der Bundeswehr
Universität München

Systemarchitektur zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen

Dipl.-Inf. Robert Heinz Koch

Von der Fakultät für Informatik der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation

Tag der Einreichung:	22.09.2011
Tag der mündlichen Prüfung:	25.11.2011

Bibliographische Informationen der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet über `http://dnb.ddb.de` abrufbar.

Herstellung und Verlag: Books on Demand GmbH, Norderstedt

Hinweis:

ISBN: 9783844801866

Prüfungskommission

Vorsitzender:	Prof. Dr. Axel Lehmann Universität der Bundeswehr München
1. Berichterstatter:	Prof. Dr. Gabrijela Dreo Rodosek Universität der Bundeswehr München
2. Berichterstatter:	Dr. ir. Aiko Pras University of Twente
1. Prüfer:	Prof. Dr. Ulrike Lechner Universität der Bundeswehr München
2. Prüfer:	Prof. Dr. Michael Koch Universität der Bundeswehr München

Meiner Familie.

Meiner geliebten Mutter für stetige Unterstützung, Hilfe und Verständnis.
Meinem innig geliebten Vater, der den Abschluß meiner Arbeit nicht mehr erleben konnte.
Meiner geliebten Schwester Sandra, die uns viel zu früh verlassen hat.
Meiner lieben Schwester Gabi, in ewiger Erinnerung an fröhliche Familienfeste.

Summary

The evolution of the Internet took place with a matchless speed over the past decades. Today, the Internet is established in numerous areas of everyday life. Purchase orders or money transfers are only two examples. The financial values which are moved over the Internet are alluring criminals. Attacks with the aid of the Internet can be executed from a safe distance, different IT laws of the countries hamper the transboundary criminal prosecution. Therefore, an underground market worth billions has been established over the past years.

For the protection of systems and networks, procedures for intrusion detection are under development for more than 30 years. Numerous systems are available and are integral security components in every bigger network today. However, even with these strong efforts, the number of security incidents is steadily increasing. Todays systems are not able to cope with challenges like targeted attacks, encrypted communication and connections or the insider threat.

The contribution of this thesis is the design and development of an architecture for intrusion and extrusion detection in encrypted environments. The architecture consists of components for the detection of external attacks as well as the identification of insiders. Statistical methods and behavior-based techniques are used, therefore there is no need for a deciphering of the data traffic. In contrast to existing approaches, the proposed architecture does not need a learning phase.

Based on a scenario of the IT infrastructure of a company, the requirements for a system for intrusion and extrusion detection are defined. Because measuring points must be located for being able to carry out a detection, an in-depth analysis of the execution of an attack is done. Afterwards, reasons for the failure of the detection of sophisticated attacks by current systems are identified based on an evaluation of the State-of-the-Art of in- and extrusion detection. An examination of encrypted network connections is used to deduce parameters which are still available after an encryption and which can be used for a detection of malicious behavior. The identified starting points are used for the development of a new architecture consisting of multiple modules for intrusion as well as extrusion detection in encrypted environments. A subsequent evaluation verifies the efficiency of the proposed architecture.

Zusammenfassung

Das Internet hat sich mit einer beispiellosen Geschwindigkeit in den Lebensalltag integriert. Umfangreiche Dienste ermöglichen es, Bestellungen, finanzielle Transaktionen, etc. über das Netz durchzuführen. Auch traditionelle Dienste migrieren mehr und mehr in das Internet, wie bspw. Telefonie oder Fernsehen. Die finanziellen Werte, die hierbei umgesetzt werden, haben eine hohe Anziehungskraft auf Kriminelle: Angriffe im Internet sind aus einer sicheren Entfernung heraus möglich, unterschiedliches IT-Recht der verschiedenen Länder erschwert die grenzüberschreitende Strafverfolgung zusätzlich. Entsprechend hat sich in den letzten Jahren ein milliardenschwerer Untergrundmarkt im Internet etabliert.

Um Systeme und Netze vor Angriffen zu schützen, befinden sich seit über 30 Jahren Verfahren zur Einbruchsdetektion in der Erforschung. Zahlreiche Systeme sind auf dem Markt verfügbar und gehören heute zu den Sicherheitsmechanismen jedes größeren Netzes. Trotz dieser Anstrengungen nimmt die Zahl von Sicherheitsvorfällen nicht ab, sondern steigt weiterhin an. Heutige Systeme sind nicht in der Lage, mit Herausforderungen wie zielgerichteten Angriffen, verschlüsselten Datenleitungen oder Innentätern umzugehen.

Der Beitrag der vorliegenden Dissertation ist die Entwicklung einer Architektur zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen. Diese beinhaltet sowohl Komponeten zur Erkennung von extern durchgeführten Angriffen, als auch zur Identifikation von Innentätern. Hierbei werden statistische Methoden auf Basis einer verhaltensbasierten Detektion genutzt, so dass keine Entschlüsselung des Datenverkehrs erforderlich ist. Im Gegensatz zu bisherigen Methoden benötigt das System hierbei keine Lernphasen.

Ausgehend von einem Szenario der IT-Struktur heutiger Unternehmen werden die Anforderungen an ein System zur Ein- und Ausbruchserkennung definiert. Da eine Detektion die Kenntnis entsprechender, messbarer Ansatzpunkte benötigt, erfolgt eine detaillierte Analyse einer Angriffsdurchführung. Auf dieser Basis sowie den Ergebnissen der Untersuchung des State-of-the-Art im Bereich der Ein- und Ausbruchserkennung wird identifiziert, warum heutige Systeme nicht in der Lage sind, ausgefeilte Angriffe zur erkennen. Anhand einer Betrachtung, welche Parameter bei verschlüsselten Verbindungen für eine Evaluation noch zur Verfügung stehen, werden Möglichkeiten zur Detektion von bösartigem Verhalten entwickelt. Hierauf basierend wird eine neue Architektur mit mehreren Teilmodulen zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen vorgestellt. Eine anschließende Evaluation zeigt die Funktionsfähigkeit der vorgestellten Architektur.

Danksagung

Die vorliegende Arbeit ist während meiner Zeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Kommunikationssysteme und Internet-Dienste an der Universität der Bundeswehr München entstanden.

Meiner Doktormutter, Frau Prof. Dr. Gabi Dreo Rodosek, gilt mein tiefer Dank für die Betreuung und Leitung meiner Arbeit, den fachlichen Rat und die umfassenden Möglichkeiten, die mir die Arbeit an Ihrem Lehrstuhl gegeben hat. Durch die von Ihr gegebenen Freiheiten in der Wahl der Forschungsrichtung konnte ich mich intensiv mit den Themen der Netzsicherheit, insbesondere dem Bereich der Einbruchserkennung auseinandersetzen.

Meinem Zweitgutachter, Dr. ir. Aiko Pras gilt ebenfalls mein besonderer Dank für die fachlichen und inhaltlichen Hinweise, welche die Struktur und Umsetzung meiner Arbeit wesentlich beeinflusst und verbessert haben, sowie für die fruchtbaren Diskussionen während meines Aufenthalts an der Universität von Twente.

Meinen Kollegen danke ich für zahlreiche fruchtbare und intensive Diskussionen und das hervorragende und lockere aber stets professionelle Arbeitsklima, für unzählige 20-Stunden-Arbeitstage und lange Wochenenden im Institut. Besonders möchte ich hier Volker Eiseler, Mario Golling und Björn Stelte danken.

Die zahlreichen Studien, Kooperationen und Kontakte des Lehrstuhl für bzw. mit Institutionen wie dem Bundesamt für Sicherheit in der Informationstechnik, der European Defence Agency, der European Network and Information Security Agency, dem Leibniz Rechenzentrum, dem Fraunhofer Institut, Secunet und vielen anderen Forschungseinrichtungen und Industriepartnern haben ferner für den Aufbau eines umfassenden Verständnis vom Stand der Forschung, den Herausforderungen und insbesondere den Restriktionen der Praxis eröffnet.

Mein Dank gilt insbesondere auch meinen Vorgesetzten während meiner Fahrenszeit an Bord der Fregatte „Mecklenburg-Vorpommern“, für hilfreiche Unterredungen, anregende Diskussionen und ihre volle, jederzeitige Unterstützung meiner Ideen und Pläne, sowie meinen zuständigen Personalplanern, welche mir diesen Schritt als Baustein meiner Karriere ermöglicht haben.

Meiner Freundin danke ich für ihre Unterstützung und Motivation sowie den wiederholten Verzicht auf gemeinsame Stunden zum Wohle des Forschens und Schreibens der Arbeit.

Meiner Mutter gilt mein tiefer Dank für eine stetige Motivation, interessanten Sichtweisen und letztendlich ihrer Stimme der Vernunft, die anderen Aspekte des Lebens nicht durch die Arbeit zu vergessen.

Meinem geliebten Vater für die Ermöglichung meines Weges. Ohne Dich wäre nichts von dem realisierbar gewesen. Du wirst immer tief in meinem Herzen sein.

Inhaltsverzeichnis

Technische Schutzmaßnahmen und Sicherheitstechnologien sind inzwischen Standard, wenn auch auf unterschiedlichem Niveau. Trotzdem scheint sich die Gefahrenlage nicht zu verbessern - im Gegenteil. Die Angreifer lernen mit, sodass technische Sicherungsmaßnahmen stets durch neue Systemlücken umgangen werden können.

Alexander Geschonneck, Forensic Technology

1 Motivation

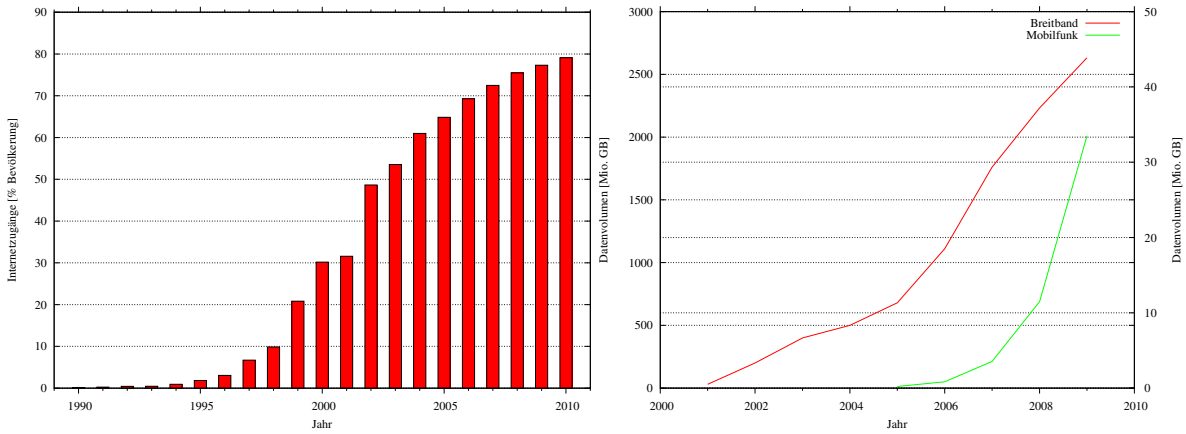
Im Jahre 2009 hat das Datenvolumen im Internet die 500 Exabyte (EB) Grenze erreicht (500 Mrd. Gigabyte (GB) bzw. 10^{18} Byte) [391]. Anders ausgedrückt, würde man diese Datenmenge in Büchern abdrucken, könnte man mit dem entstehenden Stapel zehnmal die Distanz zwischen Erde und Pluto (ca. 4,8 Mrd. Kilometer) überbrücken. Diese enormen Datenmengen müssen verarbeitet, transportiert und gespeichert werden; bereits im September 2007 hatte Google 20000 Terabyte (TB) Daten pro Tag verarbeitet [340], um erforderlichen Indizierungen, Suchanfragen, etc. für die Suchmaschine zu erzeugen. Entsprechend ist auch ein umfangreicher Schutz notwendig: Die Sicherheit von IT-Systemen und Netzen ist durch zahlreiche Faktoren gefährdet, die von höherer Gewalt bis zu illegaler Datenweitergabe durch Mitarbeiter aus dem eigenen Unternehmen reichen. Bspw. wird die Business-Webseite eines größeren Unternehmens durchschnittlich 27 mal pro Minute angegriffen [214]. Um diesen Bedrohungen entgegenzutreten, sind vielfältige organisatorische und technische Maßnahmen erforderlich. Auch mit der Nutzung entsprechender Verfahren und Systemen wie Firewalls und Intrusion Detection Systems (IDSs) bleibt ein hohes Restrisiko.

In diesem Kapitel werden die jüngere Entwicklung des Internets und dessen heutige wirtschaftliche Bedeutung kurz dargestellt sowie ein Überblick über das vorhandene Gefahrenpotential gegeben. Eine Übersicht über den Aufbau der Arbeit schließt das Kapitel ab.

1.1 Entwicklung Internet und elektronischer Handel

Nachdem 1962 die ersten Studien zur Konstruktion eines dezentralen, ausfallsicheren Netzes durch Paul Baran an der RAND Corporation [50] durchgeführt wurden und 1969 die ersten vier Knoten des ARPANET verbunden wurden, erfuhr das Netz eine beispiellose Entwicklung.

Deutschland wurde 1983 an das ARPANET angeschlossen, in den nachfolgenden Jahren hatte sich die Anzahl der Internetanschlüsse schnell vervielfacht (vgl. Abbildung 1.1a). Im Jahre 2010 hatten bereits über 79 Prozent der deutschen Bevölkerung einen Internetzugang (ca. 65 Mio. Personen), von 2000 bis 2010 entspricht dies einem Zuwachs von 171.3 Prozent [22]. Weltweit haben mittlerweile 2 Mrd. Menschen einen Zugang zum Netz. Hierbei steigt nicht nur die Anzahl der einzelnen Zugänge kontinuierlich an, auch deren Datenraten wachsen stetig: Die große Masse der Anschlüsse bietet heute Geschwindigkeiten zwischen 2 Megabit per second (Mbps) und 10 Mbps, ein Viertel ermöglicht bereits Geschwindigkeiten von mehr als 10 Mbps im Downstream [80]. Bis 2014 sollen über 75 Prozent der festen Internetanschlüsse in Deutschland Datenraten von



(a) Entwicklung der Internetzugänge gem. [126], [174].

(b) Entwicklung des Datenvolumens.

Abbildung 1.1: Entwicklung des Internet in Deutschland.

mindestens 50 Mbps aufweisen [156]. Die mobilen Zugangsmöglichkeiten zum Internet sorgen sogar dafür, dass Bevölkerungsteile Zugang zum Medium haben, obwohl sie zuhause über keine Stromversorgung verfügen: 2010 gehörten bereits 48 Mio. Menschen in Gebieten mit schlechter Infrastruktur in Asien und Afrika dieser sog. *Off-Grid, On-Net* Gruppe an, bis 2015 wird diese Zahl auf 137 Mio. steigen [96].

Die Kommunikation und der Datenaustausch über das Internet in nahe-Echtzeit sind heutzutage unverzichtbar. Im stetigen Globalisierungsprozess ist der wirtschaftliche Erfolg eines Unternehmens von der sofortigen Handlungsfähigkeit abhängig. Die zunehmende Miniaturisierung und der technologische Fortschritt sowie der schnelle Preisverfall im Hardwaresegment sorgen für eine rasante Verbreitung neuer Geräte wie bspw. Smartphones oder hochauflösender Kameras. Aber auch neue Dienste, z.B. Soziale Netzwerke wie *Facebook* oder *Twitter*, tragen zur Datenflut bei: Diese Plattformen werden immer beliebter, neben Textnachrichten werden hier auch andere Medientypen wie Fotos und Videos einfach und schnell verbreitet und bewirken einen immer schnelleren Anstieg der ohnehin schon gigantischen Datenmenge. Auch die Migration von datenintensiven Anwendungen wie Fernsehen oder Telefonie in das Internet trägt hierzu bei, z.B. durch IPTV, Video on Demand (VoD) oder Voice over IP (VoIP) [156]. *Mobile Video* wird bis 2015 voraussichtlich 66 Prozent des gesamten mobilen Datenverkehrs ausmachen, ca. 4.2 EB pro Monat [96].

Bereits 2006 lag das monatlich transferierte Datenvolumen eines Unternehmens in Deutschland im Durchschnitt bei 2777.6 GB [27]. DE-CIX, einer der weltgrößten Internet Exchanges (IXs) mit Sitz in Frankfurt am Main, hat ein derzeitiges durchschnittliches Datenaufkommen von 935.7 Gigabit per second (Gbps), Spitzenwerte können auf über 3000 Gbps steigen [164]. Cisco gibt in seinem Visual Networking Index (VNI) ein Volumen von 15 EB Internet Protocol (IP)-Datenverkehr pro Monat im Jahre 2009 an und schätzt einen Anstieg auf 64 EB pro Monat bis 2014 [94]. Mittelfristig ist weder ein Rückgang des Datenvolumens, noch eine Reduzierung der Wachstumsraten des Datenverkehrs

zu erwarten (siehe z.B. [96]).

Der enorme Erfolg und die schnelle Verbreitung des Internets und die Möglichkeiten des in den Jahren 1989 bis 1991 von Tim Berners-Lee geschaffenen Hypertext-Systems World Wide Web (WWW) [7] zog auch das Interesse des Handels auf sich. Nach der Öffnung des WWW für die kommerzielle Nutzung, entstanden 1994 die ersten Seiten für Online-Banking und bspw. die erste Online-Bestellmöglichkeit einer bekannten Pizza-Kette. In den folgenden Jahren wurden unzählige Unternehmen gegründet, die in verschiedenen technologieorientierten Feldern tätig waren und oftmals völlig überhöhte Gewinnerwartungen auf Seiten der Anleger auslösten. Die wirtschaftliche Bedeutung und Entwicklung des Internets wurde jedoch auch im Zusammenhang mit dem durch diese sog. *Dotcom-Blase* resultierenden Börsencrash des neuen Marktes (siehe [133]) im Jahr 2000 nicht beendet, sondern lediglich auf ein gesundes Niveau reduziert. Diagramm 1.2 zeigt die Entwicklung des über das Internet erzielten Umsatzes in Deutschland für die Jahre 2000 bis 2009. Auch im Krisenjahr 2009 wurde der Boom des Internet-Handels nicht gebrochen [304]: Beim Versandhandel betrug das Wachstum im Bereich Onlinehandel 15 Prozent und stieg somit auf 15.4 Mrd. € [21]. Zählt man den Umsatz weiterer digitaler Dienstleistungen wie Online-Tickets oder Übernachtungsbuchungen dazu, wurde bereits ein Gesamtumsatz von 21.8 Mrd. € erreicht [305]. Weiterhin wurde das World Wide Web Anfang 2010 mit 53.3 Prozent Anteil erstmalig der größte Umsatzträger im bundesdeutschen Versandhandel [23].

1.2 Entwicklung Internetkriminalität

Dies hat auch Auswirkungen auf die Anziehungskraft auf Kriminelle: Im Gegensatz zu den meisten anderen Gebieten, lassen sich Straftaten im Internet aus einer sicheren Entfernung heraus begehen. Befinden sich die Täter zudem im Ausland, sind Strafverfolgungen nur erschwert möglich. Seit einigen Jahren hat sich so ein erstaunlich spezialisierter Untergrund für Internetkriminalität etabliert (siehe z.B. [52], [212]). Beispielsweise sind die Verantwortlichkeiten beim Spamming u.a. zwischen Malware-Programmierern, Botnetz-Betreibern, Identitätssammlern, Spammern und Wiederverkäufern aufgeteilt. Auf Servern des Untergrund-Handels werden heutzutage Kreditkarteninformationen, Daten von Bankkonten, Schadsoftware-Toolkits und vieles mehr vertrieben. Tabelle 1.1 zeigt einen Auszug der Warenliste und der entsprechenden Preise.

Der Preis für Kreditkarteninformationen kann bspw. zwischen 0.60 € und 21 € liegen, abhängig des Kartentyps, des Herkunftslandes und des Umfangs an verfügbaren persönlichen Informationen zur Verifikation des Kartenbesitzers [138]. Eine Studie der Symantec Corporation vom November 2008 berechnete den Wert aller Waren und Dienstleistungen, die in einem Beobachtungszeitraum von einem Jahr auf Servern der Untergrund-Wirtschaft angeboten wurden, auf 276 Mio. US-Dollar. Der potentielle Maximalwert aller auf den Servern angebotenen Kreditkarten belief sich sogar auf 5.3 Mrd., der Deckungswert aller angebotenen Kontoinformationen auf 1.7 Mrd. US-Dollar [140].

Nicht nur der Handel mit Informationen, auch die Proliferation von Schadsoftware wird durch den Untergrund-Handel gefördert: Mithilfe entsprechender Tools kön-

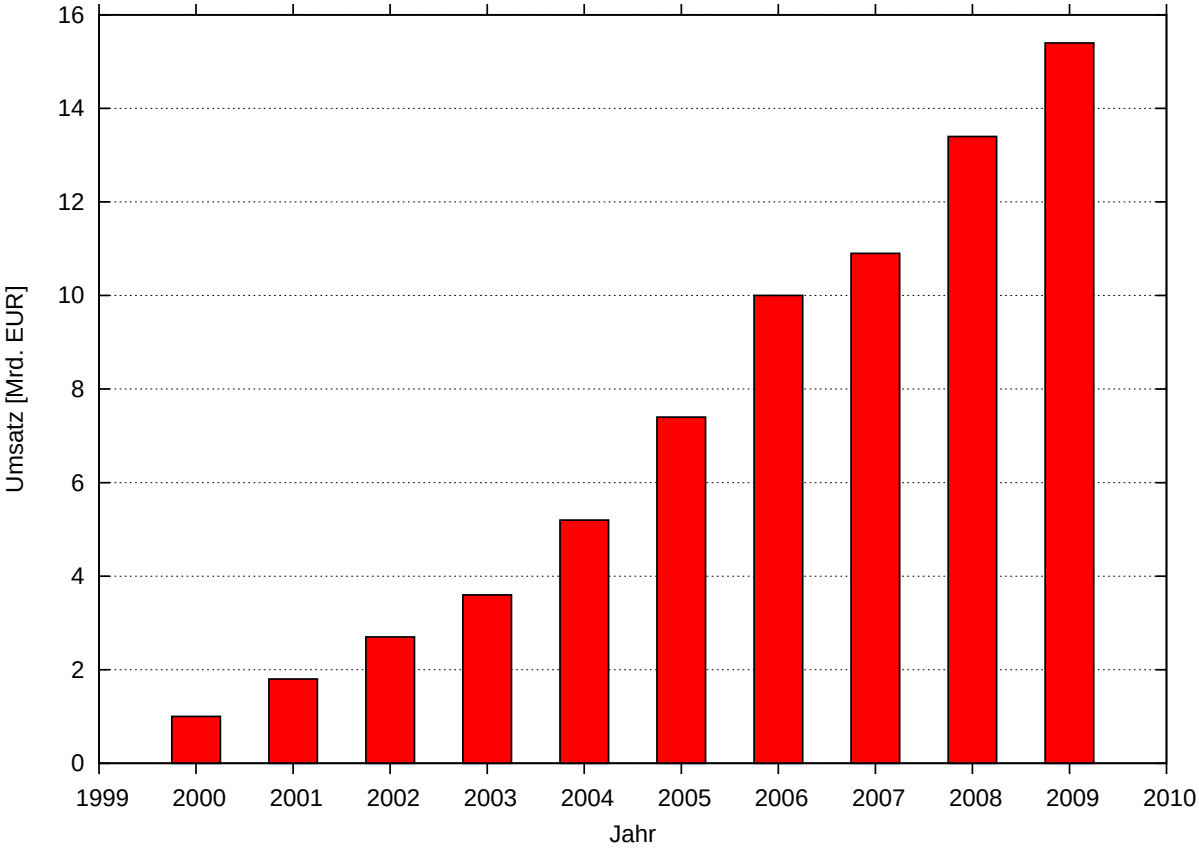


Abbildung 1.2: Entwicklung der Umsatzzahlen des Versandhandels über das Internet in Deutschland [21].

nen auch ohne Fachkenntnisse neue und gefährliche Viren und Würmer erstellt werden. Diese Programme sind mittlerweile hoch professionell und kosten bis zu mehrere tausend Euro, stellenweise auf Basis zeitlich limitierter Lizenzen. Da in diesem Bereich der Untergrund-Ökonomie ein zunehmender Wettbewerb entsteht, bieten die Autoren verschiedener Tools in jüngerer Zeit verstärkt Dienstgütevereinbarungen (Service Level Agreements (SLAs)) an: Der Support beinhaltet bspw. Updates für neue Schadroutinen oder Garantien bzgl. der (Nicht-) Detektierbarkeit [139]. Auch beinhalten manche Schadprogramme Routinen, welche andere Schadsoftware auf dem Zielrechner detektieren können und diese entfernen, um die alleinige Kontrolle über ein System zu erhalten.

Vor den immer raffinierteren Methoden der Online-Kriminellen wurde daher auch vom Bundeskriminalamt (BKA) und BITKOM gewarnt [73]. Gemäß einer Umfrage von BITKOM und Forsa hatten bereits 43 Prozent aller Nutzer Infektionen des eigenen Rechners durch Schadprogramme. Der resultierende Schaden im Bereich Online-Banking in Deutschland wurde für 2010 auf 17 Mio. € geschätzt. Bereits fünf Prozent aller Nutzer hatten demnach einen finanziellen Schaden durch Schadsoftware.

Neben diesem enormen Schadenspotential insbesondere durch Missbrauch von Kreditkarten- und Konteninformationen, gewinnt auch die Wirtschaftskriminalität wieder zu-

Tabelle 1.1: Wert verschiedener Informationen im digitalen Untergrund 2009. Quelle: Symantec [137], Wechselkurs vom 05.11.2010, gerundete Werte.

Art	Preisspanne [€]
Kreditkartendaten	0.60 – 21
Kontodaten	10 – 600
Email-Konten	0.70 – 14
Email-Adressen	1.20 je MB – 10 je MB
Identitäten	0.50 – 14

nehmend an Bedeutung (siehe z.B. [234], [104] und [82]). Gemäß dem Bundeslagebild des BKA lag der durch Wirtschaftskriminalität erzeugte Schaden für die deutsche Wirtschaft im Berichtszeitraum 2009 bei rund 3.43 Mrd. € [76]. Insbesondere der Bereich der Industriespionage hat hierbei eine erhebliche Bedeutung. Eine Studie der Corporate Trust Business Risk and Crisis Managment GmbH über die durch Industriespionage der deutschen Wirtschaft zugeführten Verluste kommt auf einen Schadenswert von 2.8 Mrd. € [104]. Der entstehende Schaden wurde in mehreren aktuellen Studien detailliert untersucht. PricewaterhouseCoopers und die Martin-Luther-Universität Halle-Wittenberg geben in ihrer Studie *Wirtschaftskriminalität 2009 - Sicherheitslage in deutschen Großunternehmen* einen durchschnittlichen Schaden von 5.85 Mio. € je Unternehmen im Bereich Wettbewerbsdelikte, Diebstahl vertraulicher Kunden- und Unternehmensdaten, wettbewerbswidrigen Absprachen sowie Wirtschafts- und Industriespionage an [82], wobei die schwersten Fälle deutlich höher liegen (vgl. Abbildung 1.3). Hiervon sind aber nicht nur Großunternehmen betroffen, im Gegenteil: Gerade mittelständische und kleine, innovative Unternehmen sind im Visier der Angreifer, da hier üblicherweise keine ausreichenden Mittel und Schutzmaßnahmen im Bereich der IT-Sicherheit vorhanden sind. Der Gesamtschaden, welcher durch Industriespionage entsteht, betrifft zu 57.6 Prozent mittelständische Unternehmen, zu 38.5 Prozent kleinere Unternehmen und lediglich zu 3.9 Prozent Konzerne [104]. Weitere Analysen, bspw. des Wirtschaftsschutzes des Niedersächsischen Ministeriums zeigen ebenfalls auf, dass kleine- und mittelständische Unternehmen im Brennpunkt der Angreifer liegen [390].

Computerkriminalität und das Tatmedium Internet spielen bei der Durchführung der entsprechenden Straftaten der Wirtschaftskriminalität eine immer wichtigere Rolle - daher wurde 2004 das Internet aus dem Lagebericht des BKA ausgekoppelt und wird seither in einem extra Bericht dargestellt.

Diese Tendenzen sind anhaltend, ein Rückgang ist auch mittelfristig nicht zu erwarten: Während die allgemeine Kriminalitätsentwicklung 2009 leicht rückläufig war, verzeichneten Straftaten mit dem *Tatmittel Internet* einen gegenläufigen Trend. 2009 wurden in diesem Bereich bundesweit 206909 Straftaten erfasst, was einem Anstieg um 23.6 Prozent im Vergleich zum Vorjahr entspricht [77]. Tabelle 1.2 zeigt die festgestellten Straftaten-Gruppen. Gut erkennbar ist, dass der größte Anstieg in der Computerkriminalität im Bereich *Computerbetrug* und *Ausspähen und Abfangen von Daten* einschließlich Vorbereitungshandlungen zu verzeichnen ist. Dahingegen ist Softwarepiraterie, sowohl im Bereich

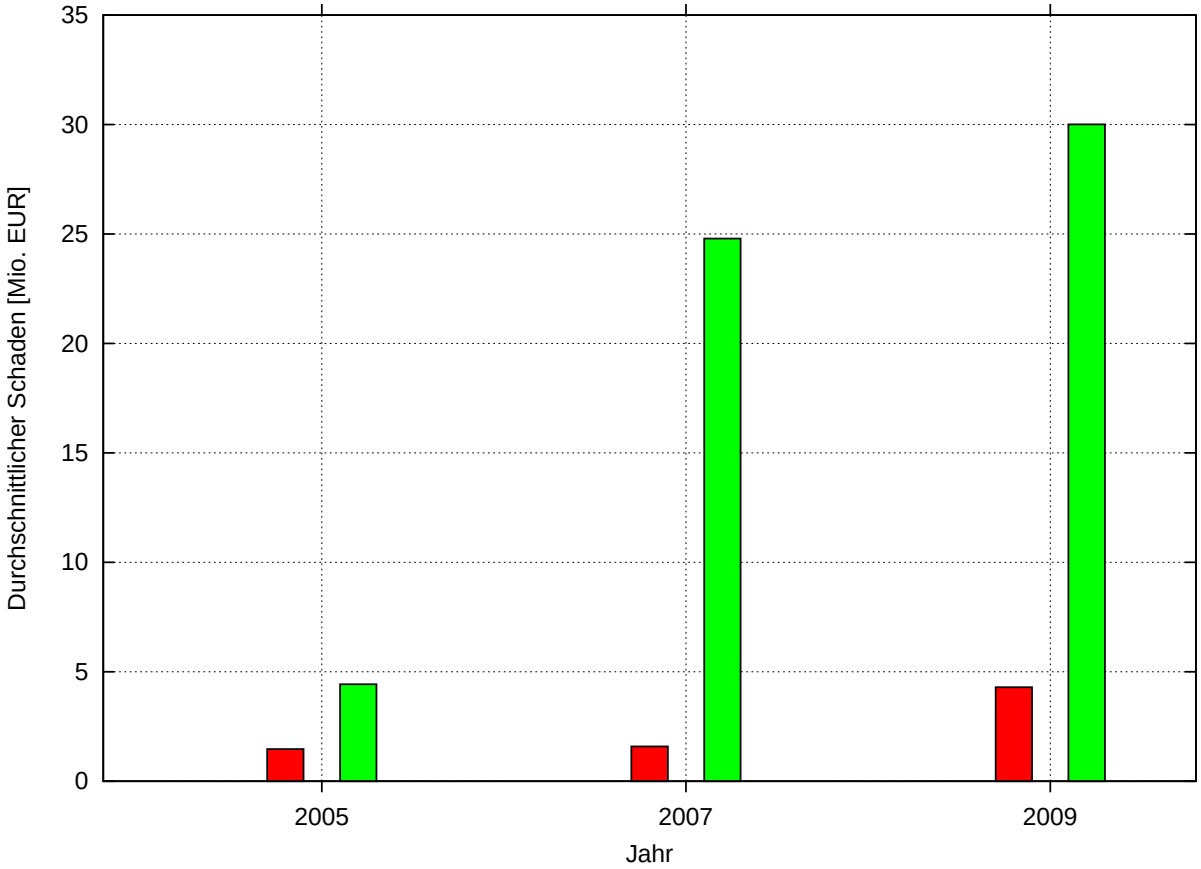


Abbildung 1.3: Durchschnittlicher finanzieller Schaden aller Wirtschaftsdelikte (rot) sowie durchschnittlicher Schaden bei den 210 schwersten Delikten [82].

privater Anwendungen als auch im Bereich gewerbsmäßigen Handels, stark rückläufig.

Hier muss angemerkt werden, dass das Gebiet Informations- und Kommunikationstechnik (IuK)-Kriminalität im engere Sinne die Bereiche *Betrug mit Debitkarten* sowie *Softwarepiraterie* **nicht** beinhaltet. Nimmt man diese für die Gesamtwertung von Tabelle 1.2 aus, ergibt sich ein Zuwachs von 32.6 Prozent bzgl. der Zahlen des Vorjahres.

Aufgrund seiner steigenden Relevanz und seinem Umfang gibt es eine eigenständige Publikation für den Bereich IuK-Kriminalität [74]. Betrachtet man das Schadensvolumen im Bereich der IuK-Kriminalität, ist dieses um rund 1 Prozent auf 36.9 € Mio. bei 50254 Fällen gesunken, jedoch beinhalten diese Zahlen keine Phänomene wie Phishing oder Bot-Netze, da diese aufgrund der unterschiedlichen Erfassung ihrer Tathandlungen nicht in der Polizeiliche Kriminalstatistik (PKS) erscheinen. Weiterhin ist in den Bereichen *Computersabotage* und *Datenveränderung* von einer erheblichen Dunkelziffer auszugehen: Oftmals wird eine erfolgreich durchgeführte Straftat nicht erkannt, da der Einbruch auf einem Rechner unentdeckt bleibt, andererseits werden viele erkannte Taten aufgrund der Angst vor einem Reputationsverlust nicht zur Anzeige gebracht (siehe z.B. [104], [390] oder [75]).

Tabelle 1.2: Entwicklung der Straftaten im Bereich Computerkriminalität [77].

Straftaten	Erfasste Fälle 2009	2008	Veränderung (%)
Betrug mit Debitkarten mit PIN	23163	23689	-2.2
Computerbetrug §263a StGB	22963	17006	+35.0
Betrug mit Zugangsberechtigungen	7205	5244	+37.4
Fälschung beweiserheblicher Daten	6319	5716	+10.5
Datenveränderung, Computersabotage §§303a, 303b StGB	2276	2207	+3.1
Ausspähen, Abfangen von Daten und Vorbereitung	11491	7727	+48.7
Softwarepiraterie (private Anwendungen)	1351	1854	-27.1
Softwarepiraterie (gewerbmäßiger Handel)	143	199	-28.1
Gesamt	74911	63642	+17.7

1.3 Bedrohungspotential

Wie wichtig ein umfassender Schutz von IT-Systemen insbesondere in vernetzten Umgebungen ist, unterstreichen neben den aktuellen Schadenzahlen die Statistiken des SysAdmin, Audit, Network, Security (SANS)-Institutes zur Überlebenszeit von Rechnern. Unter dem Begriff der sog. *Survival Time* führt das Institut die durchschnittlichen Werte, wieviele Minuten es dauert, bis ein mit einem Netz verbundenes, ungepatchtes System von Schadsoftware infiziert wurde [335]. Hierfür werden Messungen von verteilten Sensoren auf täglicher Basis ausgewertet: Detektiert bspw. ein Sensor eine Wurmaktivität auf einem bestimmten Port, wäre ein für den Fehler anfälliges System infiziert worden. Die Daten werden anhand registierter Aktivitäten von Schadsoftware für verschiedene Zielbereiche erstellt, namentlich Betriebsysteme der Windows-Familie, der Unix-Familie und Applikationen. Details können zu Peer-to-Peer (P2P)-Netzen und Backdoors abgerufen werden, weiterhin ist ein kumulativer Wert verfügbar. Da im Allgemeinen ein Betriebssystem nicht ohne Applikationen läuft, summiert sich das Gefahrenpotential für ein ungepatchtes System entsprechend auf, die Überlebenswahrscheinlichkeit sinkt. Die hierfür ermittelten Werte sind ebenfalls abrufbar. Abbildung 1.4 zeigt den Verlauf für Windows- und Unix-Systeme sowie für Applikationen und die jeweiligen kumulierten Werte für das Jahr 2010.

Abhängig der jeweiligen Einsatzumgebung des Rechners können die tatsächlichen Zei-

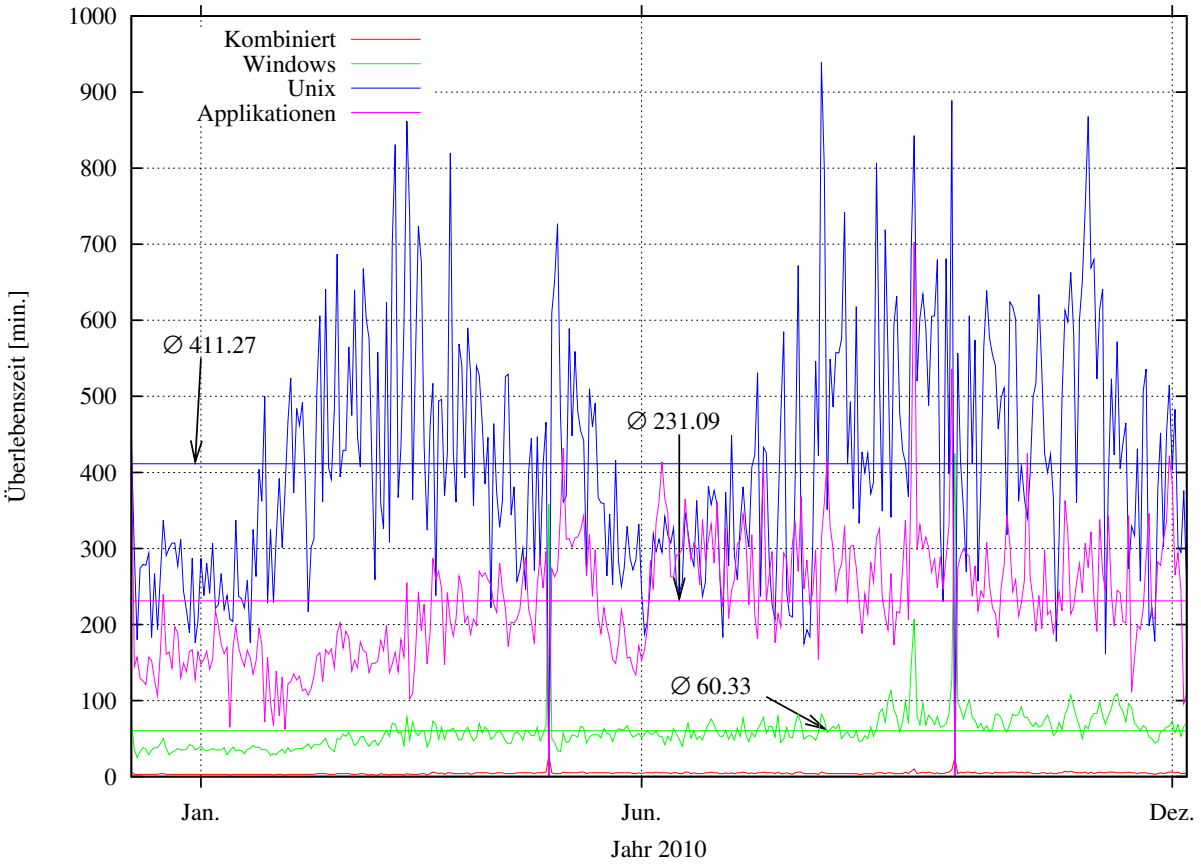


Abbildung 1.4: Überlebenszeit für ungepatchte Rechner in einer vernetzten Umgebung. Während Unix-Systeme mit durchschnittlich 411.27 Minuten die längsten Überlebenszeiten haben, kommen Windows-Systeme lediglich auf 60.33 Minuten. Die durch SANS betrachteten Applikationen kommen für 2010 auf durchschnittlich 231.09 Minuten, betrachtet man die kombinierte Überlebenszeit, also von einem System *und* entsprechend darauf laufenden Applikationen, ergeben sich lediglich 4.45 Minuten als durchschnittliche Überlebenszeit.

Tabelle 1.3: Auf den Top-500 Supercomputern eingesetzte Betriebssysteme nach Anteil der Familien im November 2010 [16].

Betriebssystem-Familie	Anzahl	Anteil %
Linux	459	91.80
Windows	5	1.00
Unix	19	3.80
BSD Based	1	0.20
Mixed	16	3.20
Gesamt	500	100

ten natürlich erheblich variieren; so wird ein durch eine entsprechende Firewall und Regelsätze geschützter Rechner in einem Firmennetz eine längere Überlebenszeit haben, als ein System an einem DSL-Anschluss ohne angepasste Schutzmaßnahmen.

Auffällig ist die deutlich geringere Überlebenszeit von Windows- im Vergleich zu Unix-Systemen. Während Unix-Systeme im Schnitt 6.9 Stunden ohne Infektion blieben, war dies bei Systemen der Windows-Plattform lediglich *eine* Stunde. Insbesondere bedeutet dies, dass die für ein Update und Patchen eines neu installierten Systems benötigte Zeit regelmäßig höher ist, als die durchschnittliche Zeit, bis ein Angriff auf eine Schwachstelle des Systems durchgeführt wurde.

Die schlechten Werte der Windows-Familie lassen sich u.a. durch die Verbreitung der jeweiligen Plattformen und somit die Attraktivität für einen Angreifer erklären: Da die Familie der Windows-Betriebssysteme mit einem Anteil von 87.7 Prozent das mit Abstand am weitest verbreitete System ist (vgl. Abbildung 1.5), ergibt sich daraus auch das lukrativste Ziel für Angriffe. Weiterhin sind die Rechner der hierbei hauptsächlichen Gruppe der Desktop-PCs typischerweise weniger gut abgesichert, als Rechner in Firmen- und anderen Netzen. Angemerkt sei, dass diese Zahlen spezifisch für den Desktop-Bereich sind. Andere Bereiche, wie bspw. Embedded Systems oder Höchstleistungsrechner, weisen andere Verteilungen auf. So entspricht der Anteil von Linux auf den Top-500 Supercomputern fast 92 Prozent (vgl. Tabelle 1.3).

Basis für das Eindringen in fremde Rechnersysteme und die illegale Datengewinnung ist eine entsprechende Schadsoftware, die eine nicht behobene bzw. gepachte Schwachstelle des Betriebssystems oder einer Applikation des Zielrechners ausnutzt. Aufgrund des geringen Risikos für den Angreifer, die hohen Gewinnaussichten und die immer einfacher zu bedienenden, gleichzeitig aber immer mächtigeren Tools zur Konstruktion von Schadsoftware auch ohne Fachkenntnisse, erfolgte in den letzten Jahren ein exponentieller Anstieg der festgestellten, neuen Schadprogramme (vgl. Abbildung 1.6).

Es ist möglich, dass die Anzahl der Schadcode-Signaturen nicht in dem hohen, exponentiellen Tempo weiter wächst: Die veröffentlichten Zahlen zum ersten und zweiten Quartal 2010 zählten 958585 respektive 457641 neue Signaturen (siehe [109] sowie [108]). Nach 921143 entdeckten Signaturen im letzten Quartal 2009 entspricht dies zu-

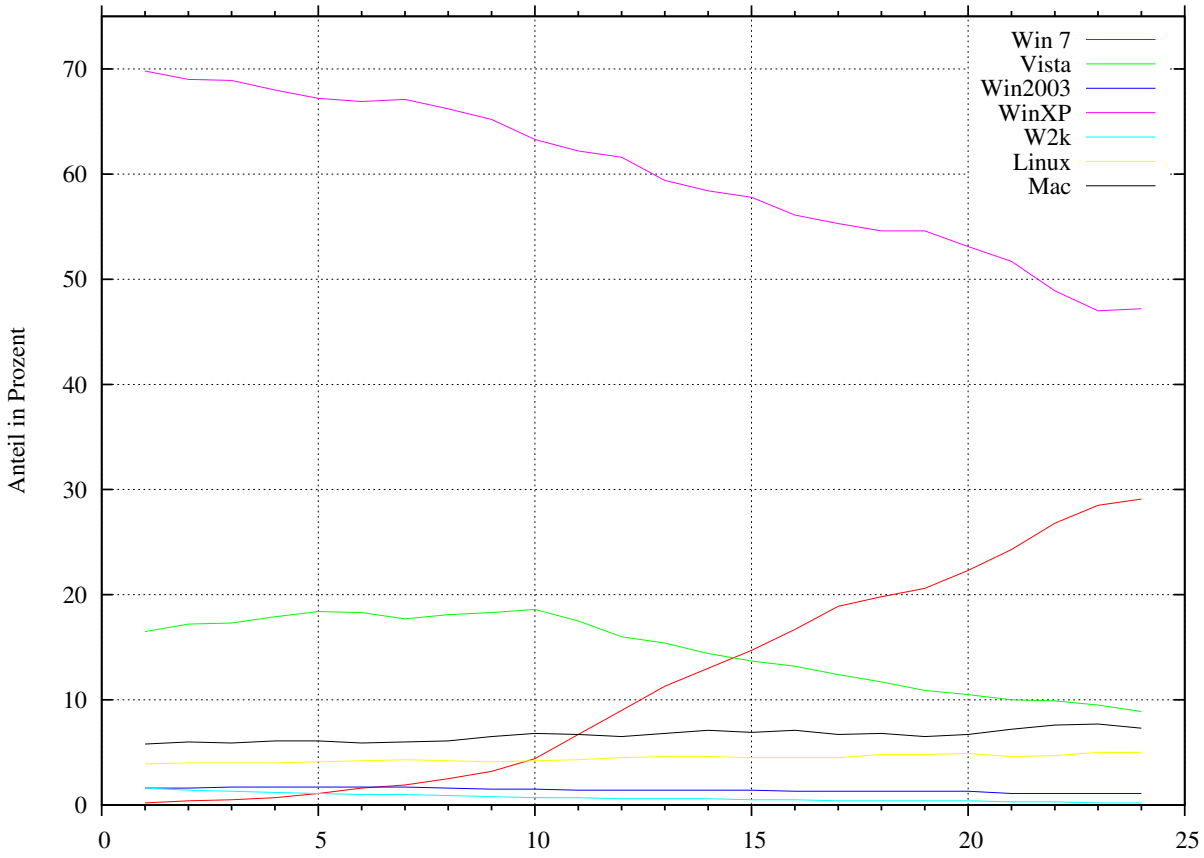


Abbildung 1.5: Anteil der Betriebssysteme im Desktop-Bereich. Die verschiedenen Versionen von Windows ergeben mit Abstand den größten Anteil, wobei auch Ende 2010 die veraltete Variante *Windows XP* die am weitest verbreitete Windowsvariante ist. Weiterhin finden sich MacOS und Linux auf dem Desktopsegment vertreten, anderer Systeme spielen nur eine untergeordnete Rolle.

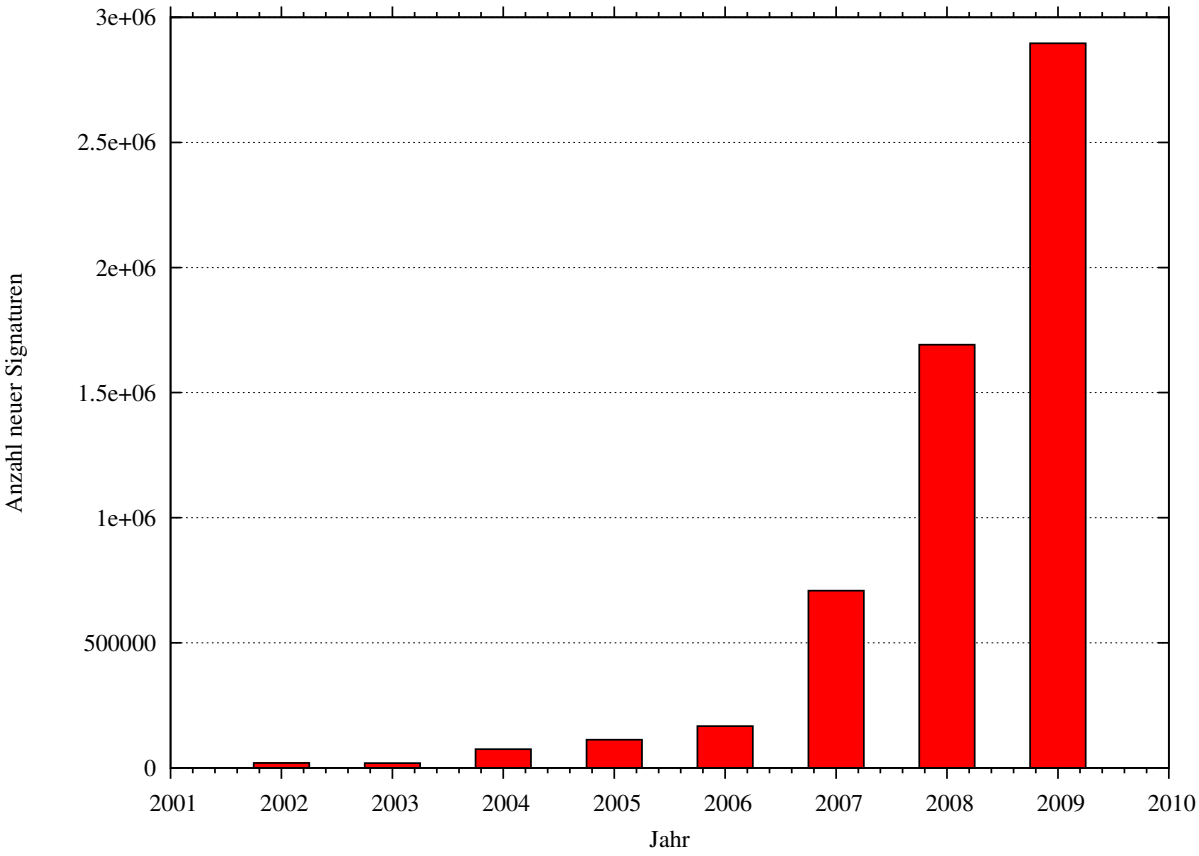


Abbildung 1.6: Entwicklung der Anzahl neuer Schadcode-Signaturen. Der exponentielle Anstieg neuer Signaturen wird durch den kommerziellen Handel mit professionellen Angriffstoolkits gefördert, was zu einer Proliferation von Schadcode führt.

nächst einem leichten Anstieg und dann einem deutlichen Rückgang der Signaturen - eine Trendwende wird hierdurch jedoch nicht eingeleitet. Der Trend geht stark hin zu mehr zielgerichteten Angriffen, auch wurden 2010 weniger Schwachstellen ausgenutzt, diese dafür technisch ausgefeilt und in aller Tiefe: Anstatt zu versuchen, eine hohe Anzahl Schwachstellen auszutesten konzentrieren sich die Angreifer auf die intensive Ausnutzung von effektiven Schwachstellen mit einer hohen zu erwartenden Rendite (vgl. [137], [108]).

Aufgrund der einfachen Möglichkeiten, hoch gefährliche Schadsoftware mit entsprechenden Toolkits zu erzeugen, steigt die Anzahl der neuen Signaturen jedoch weiterhin an: Für das Jahr 2010 wurden über 286 Millionen bösartiger Programme durch Symantec registriert [142].

Nicht nur die Anzahl von Schadcode nimmt immer schneller zu, auch die Funktionalität der Programme wird immer komplexer und trickreicher. Der Virus *NAME* durchsucht die Dateistruktur nach MPEG-1 und MPEG-2 Audio Layer 3 (mp3)-Dateien und konvertiert diese in das Format Windows Media Video (wmv). In die konvertierten Dateien werden die Schadroutinen installiert, die beim Abspielen dieser durch den Windows Mediaplayer ausgeführt werden. Die Dateiendung wird dabei nicht geändert, sondern bleibt auf mp3, damit der Nutzer keinen Verdacht schöpft.

Andere Schadprogramme, sogenannte *Ransomware* (vgl. [224]), verschlüsseln Dateien oder Festplatten und der Nutzer muss einen Betrag zahlen, damit er den Schlüssel zum Dekryptieren seiner Daten erhält. Entsprechende Programme traten das erste Mal bereits 1989 in Erscheinung (vgl. [400]), jedoch hatten diese lediglich eine einfache, symmetrische Verschlüsselung, wodurch der Schlüssel durch *Reverse Engineering* herausgefunden werden konnte. Moderne Schadsoftware nutzt Public Key-Kryptographie, wodurch der Analyst keine entsprechende Möglichkeit hat, den erforderlichen Privat Key herauszufinden. Angemerkt sei, dass es in dieser Gruppe von Schadprogrammen auch Varianten gibt, welche einen Public Key verwenden, der „gefaked“ ist, so dass also eine Entschlüsselung nicht mehr möglich ist. Dies erschwert im Zweifelsfall eine Aussage, ob ein Datenverlust aufgetreten ist oder nicht.

Während das erste Toolkit, das *Virus Creation Lab* von 1992 lediglich einfache Funktionen hatte, sind moderne Pakete wie *MPack* oder *Nukesploit* in der Lage, ausgefeilte und komplexe Schadsoftware mit einfachen Bedienschritten zu erzeugen. Dies lässt die Anzahl neuer Signaturen von Schadsoftware immer schneller ansteigen.

Neben der Tendenz, dass immer komplexere Angriffe mit immer weniger Fachwissen durchgeführt werden können, zeigt sich auch ein deutlicher Trend hin zu zielgerichteten Angriffen (*targeted attacks*). In Sozialen Netzwerken finden sich persönliche und oft auch berufliche Details, die insbesondere zur Formulierung individualisierter E-Mails herangezogen werden können. Durch eine vertraute Absenderadresse aus dem Freundeskreis und einer zu den aktuellen Gegebenheiten und dem persönlichen Umfeld passende Nachricht ist der Empfänger eher geneigt, einen Mailanhang zu öffnen und somit die Schadsoftware des Angreifers unbewusst auf dem eigenen Rechner zu installieren. Auch für fingierte Anrufe beim IT-Support oder im Sekretariat der Firma des Opfers können die Daten genutzt werden. Die Techniken des sog. *Social Engineering*, welche bereits in den 80er Jahren mit Kevin Mitnick u.a. ihre Hochblüte hatten (siehe [279]), erleben somit ein

neues Comeback (vgl. auch [309]). Eine Studie zur Computerkriminalität in der deutschen Wirtschaft zeigt, dass bereits bei über der Hälfte aller Vorfälle Social Engineering eine Rolle spielt [234].

Auf der anderen Seite nimmt die Bedeutung der IuK in immer mehr Bereichen des Lebens immer weiter zu. Das Innovationstempo ist dabei seit Jahren ungebrochen, maßgebliche Aspekte sind hierbei insbesondere [72]:

Ein *steigender Vernetzungsgrad* von IT-Systemen und Anwendern ermöglicht weltweite Kooperation und den Zugriff auf gemeinsame Datenbestände. Andererseits werden hierdurch Abhängigkeiten von den Kommunikationsinfrastrukturen und der Verfügbarkeit von Systemen erzeugt, Sicherheitsmängel einzelner Systeme können sich schnell global auswirken. Dabei weiten sich die Bereiche, in denen sich Informationstechnologie (IT) *verbreitet*, immer mehr aus. Die erforderliche Hardware wird immer kleiner und günstiger und reicht von Smartphones bis zu IP-basierter Sensorik in Automobilen oder Radio Frequency Identification (RFID)-gestützter Waren- und Besuchersteuerung. Hierbei findet eine zunehmende drahtlose Kommunikation der Geräte statt, so dass heutzutage bereits *Alltagsgegenstände* über das Internet lokalisierbar und steuerbar sind.

Motiviert durch die hohen finanziellen Anreize, das geringe Risiko sowie die zahlreichen Schwachstellen in Betriebssystemen, Applikationen und Sicherheitstools verbunden mit der immer stärkeren Digitalisierung des Alltags, neuer Dienste und der Migration traditioneller Dienste in das Internet wird dieser negative Trend bzgl. über das Internet verübter Angriffe und Straftaten weiter zunehmen. Aktuelle Sicherheitssysteme sind bereits jetzt regelmäßig nicht in der Lage, zielgerichtete Angriffe, unbekannte und neue Angriffsformen oder eine Datenausschleusung zu erkennen.

1.4 Offene Punkte

Bereits seit den 80er Jahren sind Verfahren zur Einbruchserkennung ein intensives und weit verbreitetes Forschungsgebiet, aus dem zahlreiche Arbeiten und Systeme hervorgegangen sind, auf die detailliert in Kapitel 4 eingegangen wird. Heutzutage gehören IDSs standardmäßig zur Infrastruktur jedes größeren Firmennetzes. Trotz dieser Schutzmaßnahmen steigt die Anzahl der Sicherheitsvorfälle weiter an. Dieses augenscheinliche Paradoxon lässt sich durch verschiedene technologische als auch gesellschaftliche Trends erklären, welche in Kapitel 4.6 analysiert werden. Maßgebliche Faktoren sind hier zum Beispiel die immer kürzeren Lebenszyklen von Softwareprodukten: Der freie Markt erfordert die Bereitstellung immer neuer Funktionalitäten und Extras in immer kürzeren Abständen, was sich auf die Anzahl der Programmierfehler in den Produkten niederschlägt. Auf diese Weise sind Applikationen heute der Angriffsschwerpunkt, während Betriebsysteme selber zunehmend an Attraktivität für Angreifer verlieren. Auch der Paradigmenwechsel der Angreifer von ideologischen Beweggründen hin zu rein kommerziellen Zielen hat einen umfangreichen Untergrundmarkt im Internet entstehen lassen, der Handel mit hochprofessionellen Angriffstoolkits betreibt und Daten aller Art handelt. Dies resultiert in einem exponentiellen Anstieg von immer neuen Schadprogrammen und -varianten.

Auf der anderen Seite verstärken sich Angriffe, die auf Social Engineering Techniken beruhen und die Schwachstelle Mensch ausnutzen. Wie in Kapitel 2.3 gezeigt wird, sind solche Angriffsvektoren mit den bestehenden Verfahren kaum zu detektieren. Insbesondere traditionelle, signaturbasierte Verfahren sind nicht mehr in der Lage, einen adäquaten Angriffsschutz zu bieten. Auch bestehende verhaltensbasierte Systeme können jedoch Bedrohungen wie Innentäter nicht ausreichend detektieren und sind zudem oftmals durch die notwendigen Lernphasen gefährdet. Die immer größeren Datenmengen und die Zunahme des Einsatzes von Verschlüsselung erschweren den Einsatz von IDSs noch weiter.

Ein weiterer Faktor ist die Gefahr, welche von Datenverlust und Innentätern ausgeht und die in den letzten Jahren permanent ansteigt. Verschiedene Sicherheitssysteme existieren, die jedoch nur einen rudimentären Schutz vor Datenabfluss bieten und insbesondere nicht in der Lage sind, Innentäter zu erkennen.

Die technologische Entwicklung, aber auch der gesellschaftliche Wandel in Bezug auf die Nutzung des Internets verschieben das Gleichgewicht zwischen Angreifer und Verteidiger immer weiter in Richtung der Angreifer: Der Einbezug des Internets in das alltägliche Leben eines Großteils der Bevölkerung und die damit verbundenen Verhaltensweisen, bspw. bei der Nutzung sozialer Netzwerke, eröffnen neue und durch derzeitige Systeme und Verfahren nicht auswertbare Angriffsvektoren.

1.5 Fragestellung und Vorgehensweise

Basierend auf der bereits identifizierten Problematik, dass bestehende Systeme zur Einbruchserkennung nicht in der Lage sind, den technologischen und gesellschaftlichen Entwicklungen im ausreichenden Maße Rechnung zu tragen und somit kein entsprechender Schutz vor Angriffen und Innentätern gewährleistet werden kann, ist das Ziel der Arbeit die Entwicklung eines Sicherheitssystems, welches unter den gegebenen Anforderungen eingesetzt werden kann. Der Schwerpunkt liegt hierbei auf einer verschlüsselten Umgebung, weiterhin besteht die Forderung, auf Angriffssignaturen bzw. Lernphasen zu verzichten. Diese Anforderung leitet sich direkt aus der zuvor motivierten, derzeitigen Situation im Bereich der Angriffserkennung ab.

Dieses Ziel der Arbeit wird in folgenden konkreten Fragestellungen manifestiert:

1. Ist eine verhaltensbasierte Ein- und Ausbruchserkennung ohne Lernphasen sowie ohne genaue Kenntnis des Kommunikationsverhaltens der geschützten Systeme sowie ohne Nutzung von Deep Packet Inspection (DPI) möglich?

2. Wie kann eine Architektur zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen aussehen?

3. Müssen Innentäter in einem Sicherheitssystem adressiert werden?

4. Ist eine Ein- und Ausbruchserkennung in verschlüsselten Umgebungen unter Einhaltung der rechtlichen Anforderungen möglich?

1.5.1 Vorgehensweise

Zahlreiche Arbeiten beschäftigen sich mit dem Bereich der Einbruchserkennung, wobei der Aspekt der Ausbruchserkennung bisher weniger im Fokus der Wissenschaft steht. Ziel der ersten Frage ist somit die Schaffung eines Überblicks der State-of-the-Art Verfahren sowie wichtigsten Beiträge im Bereich der Ein- und Ausbruchserkennung. Die Beantwortung dieser Frage erfolgt mittels einer umfangreichen Literaturstudie, wobei der Bereich der Detektion in verschlüsselten Umgebungen hierbei gesondert betrachtet wird und die derzeitigen Möglichkeiten und Einschränkungen abgeleitet werden. Um ein Verständnis dafür zu erzeugen, welche Ansätze in welchen Bereichen angewandt werden können, bzw. nicht zielführend sind, wird eine Klassifizierung der IDSs vorgenommen und die Problematik der Leistungsanalyse dieser Systeme besprochen.

Mittels der zweiten Fragestellung soll untersucht werden, wie die Architektur eines Systems, welches eine Ein- und Ausbruchserkennung in verschlüsselten Umgebungen durchführen kann, aufgebaut sein sollte. Zur Beantwortung dieser Frage wird zunächst eine umfangreiche Analyse der Durchführung von Angriffen vorgenommen, um die dabei entstehenden Detektionsmöglichkeiten abzuleiten. Hierbei werden insbesondere Angriffstaxonomien der Literatur herangezogen und die aktuelle Entwicklung der Angriffsfelder berücksichtigt. Weiterhin wird analysiert, welche Daten im Rahmen verschlüsselten Datenverkehrs überhaupt noch zur Verfügung stehen und mittels welcher Verfahren diese analysiert und verarbeitet werden können. Dies führt zur Entwicklung einer Architektur für sowohl die Ein- als auch die Ausbruchserkennung in verschlüsselten Umgebungen, wobei hier den zuvor gestellten Anforderungen an ein Sicherheitssystem sowohl aus Nutzer- als auch aus technischer Sicht Rechnung getragen wird. Die einzelnen Module werden anschließend umfassend validiert und hinsichtlich der Erfüllung der gestellten Anforderungen bewertet.

Da die Bedeutung des Innentäters in der Literatur sehr umstritten ist, dessen besondere Position jedoch eine maßgebliche Auswirkung auf ein IDS hat, wird diese durch die dritte Fragestellung beleuchtet. Basierend auf der Auswertung zahlreicher Studien sowie von Behörden veröffentlichter Daten und Informationen werden die stellenweise sehr unterschiedlichen Angaben und Werte gegenübergestellt und die Ursachen der Abweichungen analysiert. Auf Basis der korrelierten Aussagen und Ergebnissen wird anschließend eine Abschätzung über die Bedeutung des Innentäters geben. Ziel hierbei ist es, die Erfordernis der Berücksichtigung des Innentäters im Rahmen des Systems festzustellen.

Da der Einsatz eines Sicherheitssystems im Einklang mit den entsprechenden Datenschutzgesetzen stehen muss, um in der Praxis durchgeführt werden zu können, erfolgt mittels der vierten Fragestellung eine Untersuchung, welche rechtlichen Anforderungen im vorliegenden Kontext gegeben sind. Aufgrund der fehlenden Fachexpertise im juristischen Bereich kann im Rahmen dieser Arbeit hierfür nur eine oberflächliche Beleuchtung erfolgen. Anhand der Auswertung von Gesetzestexten und Fachliteratur werden die Einsatzmöglichkeiten und -restriktionen eines Sicherheitssystems in verschlüsselten Umgebungen ausgewertet und zusammenfassend dargestellt. Die Erkenntnisse aus dieser Fragestellung zeigen einerseits, ob der Einsatz eines entsprechenden Systems möglich ist,

andererseits welche besonderen Forderungen ggf. an das System gestellt werden müssen.

1.5.2 Beitrag der Arbeit

Der Hauptbeitrag der Arbeit ist die Entwicklung einer neuen Architektur, mit welcher eine Ein- und Ausbruchserkennung in verschlüsselten Umgebungen unter Berücksichtigung der gegebenen technischen Anforderungen und rechtlichen Rahmenbedingungen ermöglicht wird. Hierfür werden verschiedene Verfahren vorgestellt, wie der verschlüsselte Datenverkehr ausgewertet und verarbeitet werden kann, um Angriffe erkennen zu können. Die Funktionsfähigkeit der Architektur wird in umfangreichen Evaluationen validiert.

Als Voraussetzung für die Entwicklung der Architektur liefert die Arbeit eine detaillierte Analyse einer Angriffsdurchführung. Diese kann genutzt werden, um das Detektionsverfahren eines IDS auf Schwachstellen und nicht berücksichtigte Angriffsvektoren hin zu untersuchen.

Die im Rahmen der Arbeit durchgeführte rechtliche Begutachtung bzgl. den Anforderungen einer Datenauswertung zur Ein- und Ausbruchserkennung kann zur Planung entsprechender Systeme herangezogen werden.

1.6 Aufbau der Dissertation

Nachfolgend wird ein kurzer Überblick über den weiteren Aufbau der Arbeit gegeben (vgl. Abbildung 1.7).

Anhand eines grundlegenden Szenarios in Kapitel 2 wird die Situation von Unternehmen mit Hinblick auf die Erfordernisse der IT-Sicherheit im Ramen der Kommunikation beschrieben. Hierbei wird insbesondere das Konfliktfeld zwischen Absicherung mittels kryptographischer Verfahren auf der einen Seite und der Notwendigkeit der Untersuchung des Datenverkehrs mittels Systemen zur Einbruchserkennung auf der anderen Seite untersucht. Die Bedrohungen, welche für die betrachteten Unternehmen vorliegen, werden anschließend ausführlich analysiert. Hierfür werden u.a. die Grundschutzkataloge des Bundesamt für Sicherheit in der Informationstechnik (BSI) herangezogen. Aufgrund der Bedeutung im dargestellten Szenario wird im Anschluß die Rolle des Innentäters detailliert analysiert und die daraus zu ziehenden Konsequenzen für ein Sicherheitssystem abgeleitet; hierdurch wird Forschungsfrage 3 beantwortet. Um einen Angriff möglichst frühzeitig erkennen zu können, ist eine detaillierte Kenntnis des Ablaufs eines Angriffes notwendig, um die hierbei auftretenden Detektionsmöglichkeiten zu identifizieren. Kapitel 3 leitet anhand der bereits in der Motivation vorgestellten heutigen Situation, sowie insbesondere der Analyse des Szenarios und der Angriffsdurchführung die Anforderungen an ein IDS der nächsten Generation ab. Zur Feststellung, wo die maßgeblichen Schwachstellen der heutigen Systeme liegen um anschließend Lösungsmöglichkeiten zu deren Schließung zu finden, erfolgt in Kapitel 4 eine umfassende Untersuchung des State-of-the-Arts im Bereich der Einbruchserkennung. Hierbei wird zunächst ein kurzer Abriss über die Entwicklung der Einbruchserkennung gegeben, anschließend erfolgt eine Eintei-

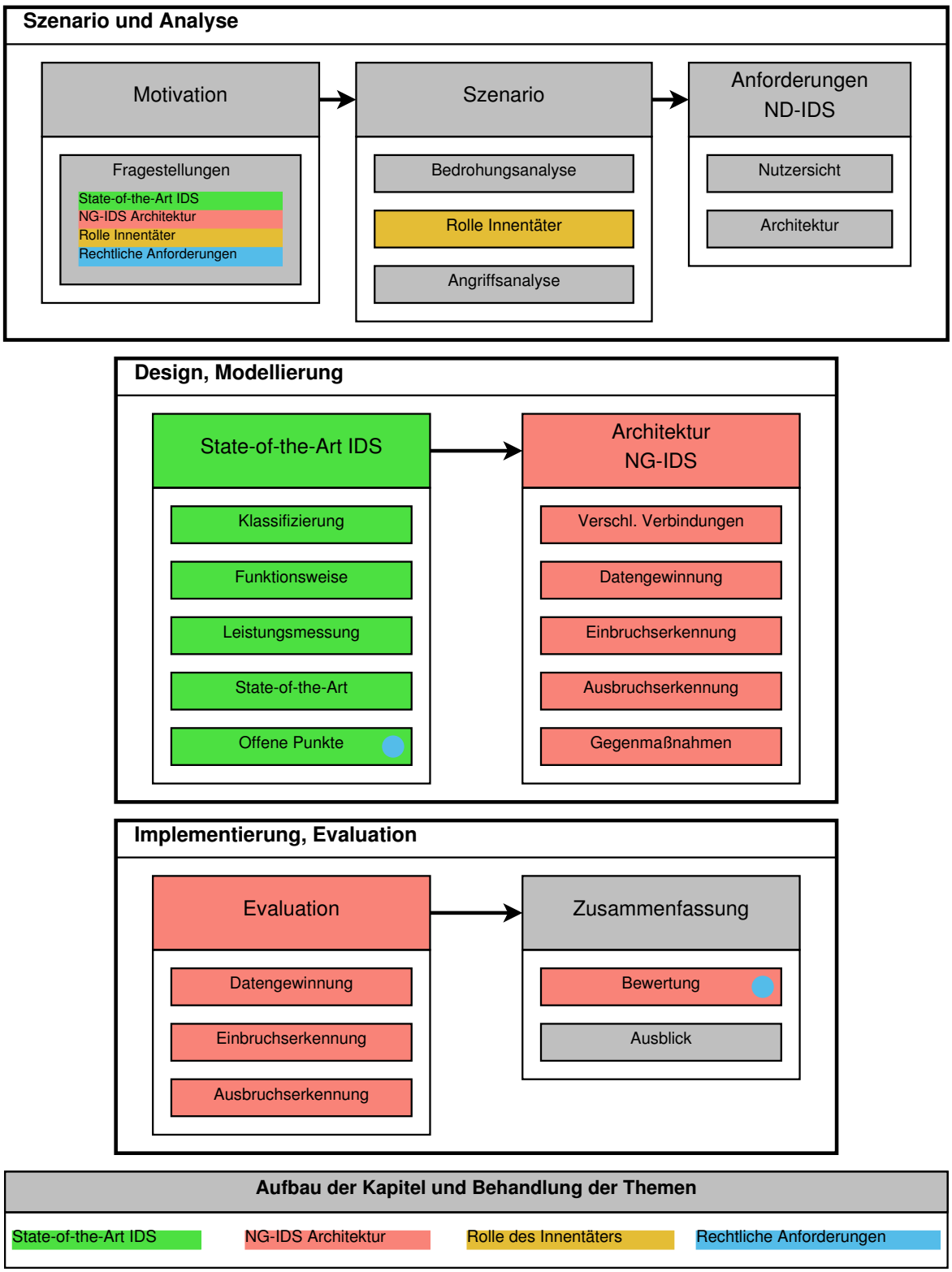


Abbildung 1.7: Aufbau der Dissertation. Die maßgeblichen Zuordnungen der Kapitel zu den jeweiligen Fragen sind hervorgehoben.

lung der Systeme anhand von Taxonomien. Die Problematik der Leistungsmessung von IDSs und die daraus resultierenden Schwierigkeiten beim Vergleich von Systemen bzw. der Einschätzung von deren Leistungsfähigkeit wird kurz diskutiert, bevor die State-of-the-Art Systeme aus Industrie und Forschung vorgestellt werden. Anhand des zuvor aufgestellten Anforderungskatalogs erfolgt eine Bewertung der State-of-the-Art, wodurch Forschungsfrage 1 beantwortet wird. Um die Schwierigkeiten und Schwachstellen der derzeitigen Systeme zu verstehen und mögliche Lösungswege zum Schließen dieser Lücken zu finden, erfolgt eine detaillierte Betrachtung der Herausforderungen für künftige IDSs. In diesem Kontext erfolgt auch eine Analyse der rechtlichen Anforderungen, welche in Bezug auf den Einsatz eines IDS im vorliegenden Szenario zu berücksichtigen sind. Diese werden im weiteren Verlauf benötigt, um nach der späteren Vorstellung der Architektur des Sicherheitssystems dessen Rechtskonformität zu prüfen, um Fragestellung 4 zu beantworten.

Die zentrale Fragestellung, wie eine Architektur zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen aussehen kann, wird in Kapitel 5 behandelt. Hierbei wird zunächst auf die Problematik der Datengewinnung bei verschlüsselten Leitungen eingegangen und untersucht, welche Daten auf welche Art analysiert werden können. Darauf basierend wird die Architektur des Systems vorgestellt, welche den gestellten Forderungen Rechnung trägt und die Schwachstellen bestehender Systeme schließt. Die einzelnen Module für die Bereiche der Ein- sowie der Ausbruchserkennung werden ausführlich besprochen. Eine Diskussion über mögliche Gegenmaßnahmen, um einer Detektion durch die neue Architektur zu entgehen sowie möglicher Reaktionen hierauf, schließt das Kapitel ab.

Die Evaluation der neuen Architektur wird in Kapitel 6 gezeigt, wobei jedes Modul einzeln ausführlich betrachtet wird. Kapitel 7 bewertet die Ergebnisse der Arbeit anhand des Eingangs aufgestellten Kriterienkatalogs. Ein Ausblick schließt die Arbeit ab.

Im Anhang der Arbeit finden sich ergänzende Informationen, detaillierte Ausgaben von Programmläufen, etc., auf die in den jeweiligen Kapiteln referenziert wird.

Unsere Sicherheiten dürfen nichts Starres werden, sonst brechen sie.

Robert Walser

2 Szenario und Angriffsanalyse

Die Auswirkungen, die technische Entwicklungen und die Internetkriminalität auf ein Unternehmen haben können, werden nun anhand eines Szenarios untersucht. Ziel des Kapitels ist die Identifizierung der Bedrohungen, die bzgl. der Kommunikation der im Szenario vorgestellten Unternehmen auftreten können. Ferner werden in einer umfassenden Angriffsanalyse die Schritte während der Durchführung eines Angriffs betrachtet (Kapitel 2.3.3), da deren Kenntnis Grundlage für die spätere Identifikation von Detektionsverfahren ist.

Zunächst werden anhand zweier repräsentativer Unternehmen die Sicherheitsanforderungen an die moderne IT-basierte Kommunikation ermittelt (Kapitel 2.1). Um diese umfassend zu analysieren, werden die Grundschutzkataloge des BSI herangezogen und bzgl. der relevanten Aussagen im Rahmen des Szenarios ausgewertet. Im Rahmen dieser Bedrohungsanalyse erfolgt ebenfalls eine detaillierte Untersuchung der Rolle des Innentäters, da dieser erhebliche Auswirkungen auf die Detektionsmöglichkeiten haben kann.

Nach der Bedrohungsanalyse erfolgt eine umfassende Betrachtung der Angriffsdurchführung. Hierzu erfolgt zunächst eine Definition von *Angriffen* (Kapitel 2.3.1), gefolgt von einer Klassifizierung von Angriffen anhand von Taxonomien (Kapitel 2.3.2) und der Betrachtung von deren Eignung im Kontext der aktuellen Entwicklung der Angriffsfelder. Anschließend werden die verschiedenen Schritte eines Angriffs detailliert untersucht (Kapitel 2.3.3), um die Detektionsmöglichkeiten der jeweiligen Stufen ableiten zu können.

Abbildung 2.1 zeigt den Aufbau des Kapitels.

2.1 Szenario

Abbildung 2.2 stellt das Szenario vor, das der weiteren Arbeit zu Grunde liegt. Betrachtet werden zwei Wirtschaftsunternehmen, zum einen ein kleines Unternehmen (vgl. EU-Definition *Kleine und mittlere Unternehmen (KMU)*, [100]) ohne eigene IT-Abteilung, weiterhin wird die IT-Struktur eines Großkonzernes betrachtet, der international agiert und die Netze mehrerer Standorte miteinander verbunden hat.

Für einen entsprechenden Zusammenschluß der verschiedenen Standorte können unterschiedliche Technologien herangezogen werden. Hierbei werden heutzutage aufgrund der geringeren Kosten (sowohl bzgl. des Betriebs als auch in Hinblick auf das notwendige Equipment) zunehmend IP-basierte Lösungen genutzt, traditionelle Standleitungen werden somit immer mehr verdrängt. Allerdings bieten Telekommunikations (TK)-Anbieter derzeit noch verschiedene Produkte für Standleitungen an: Verfügt ein Unternehmen über eine sog. *Dark Fiber*-Verbindung, kann ein hoher Grad an Sicherheit der Kommu-

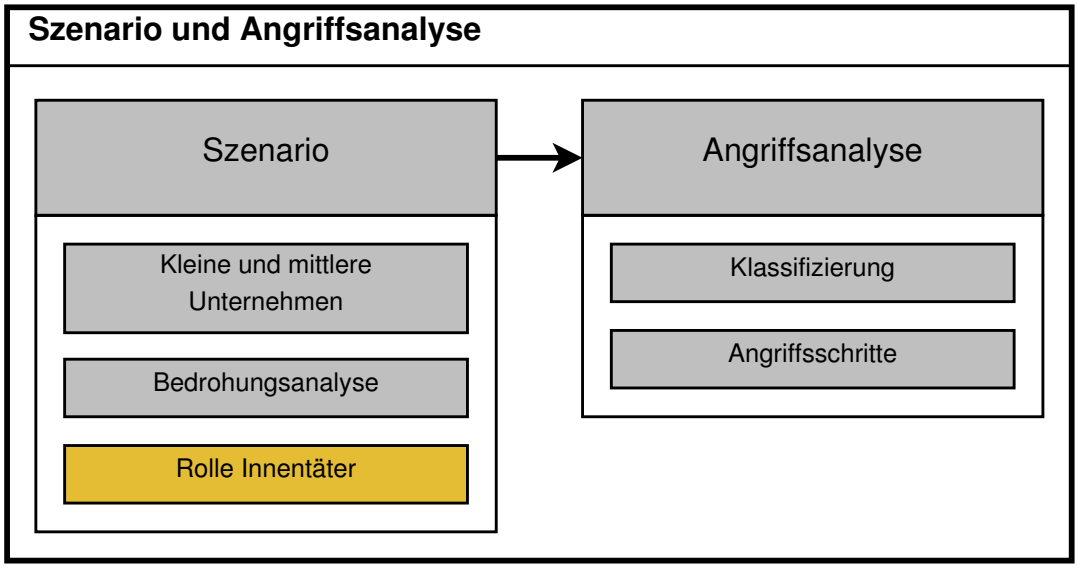


Abbildung 2.1: Aufbau des Kapitels 2. Nach der Vorstellung des Szenarios wird eine Bedrohungsanalyse durchgeführt. Die Rolle des Innentäters wird hierbei besonders untersucht. Die Angriffsanalyse betrachtet detailliert die Durchführung eines Angriffs, um im weiteren Verlauf Detektionsmöglichkeiten abzuleiten.

nikation erreicht werden. Bei dieser Art von Anbindung handelt es sich um ungenutzte Ersatz- oder Reserve-Lichtwellenleiter (LWL) eines Anbieters, die er an ein Unternehmen vermietet. Da jedoch in den wenigsten Fällen entsprechende Reserve-Leiter genau zwischen zwei gewünschten Firmenstandorten verfügbar sind, gehört Dark Fiber nicht zum Standardprodukt von TK-Anbietern [166]. Weiterhin muss hier sämtliches Equipment, welches für den Zugang und die Nutzung der Leitungen erforderlich ist, durch den Nutzer beschafft und betrieben werden. Andere Lösungen sind Satellitenübertragungen, Richtfunk oder Lösungen auf Schicht 1, sog. Standardfestverbindungen (SFVn), auch als *Leased Links* oder *Leased Lines* bezeichnet. Die funkbasierten Verfahren erfordern hierbei ebenfalls zusätzliche technische Ausstattungen und leiden unter weiteren, systemimmanenten Nachteilen, bspw. die hohen Latenzzeiten bei Satellitenstrecken von mind. 238 Millisekunden (ms) pro Richtung bzw. 476 ms Round Trip Time (RTT) (reine Übertragungszeit für die Luftschnittstelle), welche insbesondere interaktive Anwendungen wie VoIP erschweren. SFV-Lösungen werden auf Schicht 1 betrieben, hier werden dem Kunden die jeweiligen Netzabschlüsse als Zugang zur Verfügung gestellt. Die eigentlichen Datenverbindungen werden hierbei über LWL oder Kupferleitungen zu Add/Drop-Multiplexern des Providers geführt und anschließend im Backbone-Netz des Providers transportiert. Entsprechende Produkte werden von jedem größeren TK-Anbieter vertrieben, verlieren durch die zunehmende Verbreitung von IP-Plattformlösungen jedoch immer mehr an Bedeutung (vgl. [166], [68]).

Neben den kostspieligen und komplexeren dedizierten Leitungen, bieten die heutigen Weitverkehrstechnologien jedoch wirtschaftliche und einfach integrierbare Lösungen zur

Verbindung dislozierter Standorte an. Der Datentransport wird hierbei in den Providernetzen maßgeblich durch das Übertragungsprotokoll Multi-Protocol Label Switching (MPLS) bewerkstelligt. Im Gegensatz zu den dedizierten Leitungen stellt das MPLS-Transportnetz jedoch ein geteiltes Medium dar, wodurch zusätzliche Bedrohungen für die Sicherheit der Unternehmensdaten entstehen [68]. Zur Absicherung der Kommunikation über diese Providernetze werden daher Virtual Private Networks (VPNs) verwendet. Ein VPN kann verschieden definiert werden (vgl. [68]):

- Als privates Netz durch eine logische Trennung der transportierten Kundendaten, z.B. Layer-2 VPN mittels MPLS [177]. Diese VPNs sind per Definition **nicht** verschlüsselt.

- Als privates Netz durch den Einsatz von Verschlüsselung, z.B. durch Nutzung von Internet Protocol Security (IPsec), Secure Shell (SSH) oder Transport Layer Security (TLS)/Secure Sockets Layer (SSL)[1], sog. Layer-3 VPN (vgl. z.B. [176], [175]).

Layer-2 VPN werden häufig nicht über öffentliche Netze wie das Internet geleitet, sondern nutzen auch private Infrastrukturen. Da sie ohne Verschlüsselung arbeiten, gewinnen sie die Sicherheit aus dem Transport der Daten innerhalb separierter Kanäle. Allerdings finden sich in der Praxis Kombinationen privater Layer-2 Verbindungen und per Internet transportierter VPNs, basierend auf TLS, IPsec und SSH [68]. Entsprechend können durch den Zusammenschluß Angriffspunkte entstehen.

Wegen der hohen Verbreitung und steigenden Bedeutung gegenüber anderen Technologien, wird für die Konnektion der Unternehmensstandorte im Folgenden ein Layer-3 VPN mit Verschlüsselung und Nutzung des Internets als Transportmedium angenommen. Die geringeren notwendigen Sicherheitsanforderungen der Layer-2 VPNs können hierunter subsumiert werden und müssen daher nicht extra betrachtet werden.

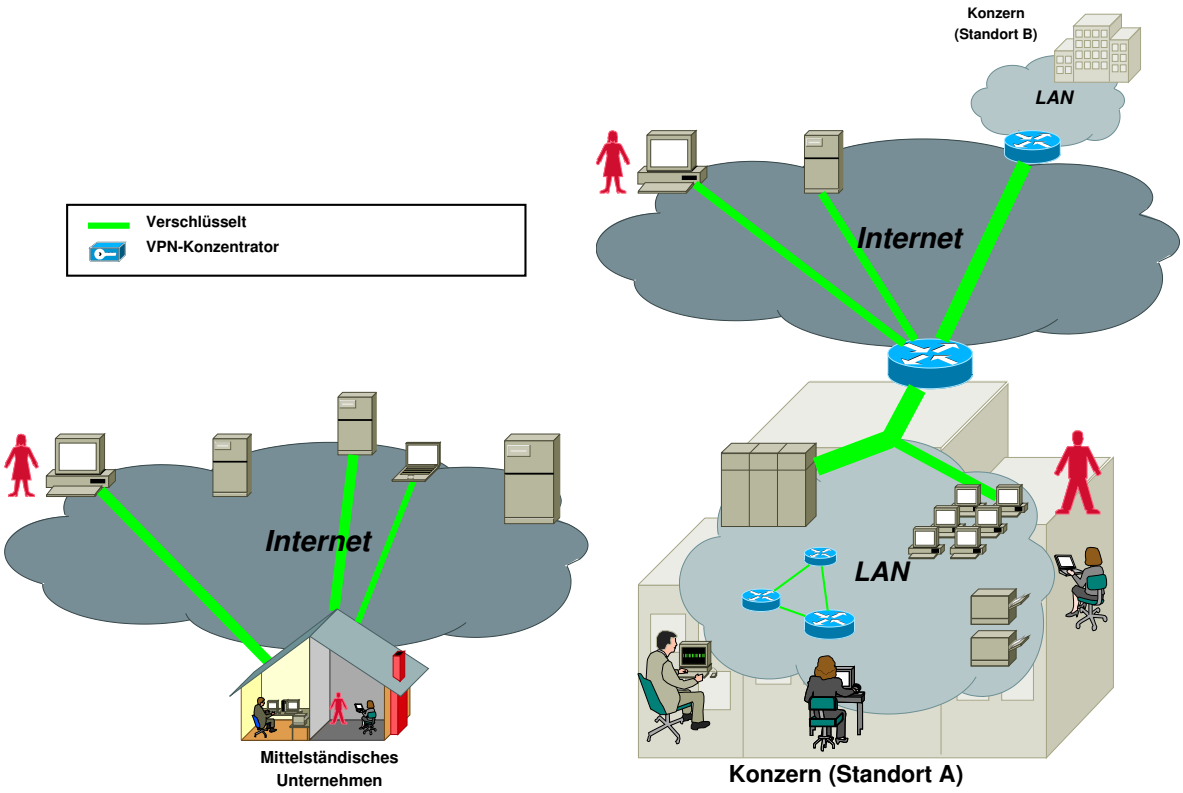
Für eine Einschätzung der anfallenden Datenvolumina müssen die Größe des jeweiligen Unternehmens (Anzahl der Mitarbeiter) und die genutzten Dienste betrachtet werden. Eine entsprechende Einordnung der Betriebsgröße anhand quantitativer Parameter gemäß den Richtlinien der EU ist aus Tabelle 2.1 ersichtlich. Darauf basierend werden für das vorliegende Szenario folgende Werte für die Betriebsgrößen genutzt:

Kleines Unternehmen: 15 Mitarbeiter, kein festangestelltes IT-Personal.

Großkonzern: 2000 Mitarbeiter, eigene IT-Abteilung.

Zur Festlegung der erforderlichen Datenraten und des zu erwartenden Datenvolumens werden weiterhin mehrere Studien und repräsentative Umfragen in Unternehmen sowie Abschätzung bzgl. der Entwicklung der Breitbandanschlüsse herangezogen: Basierend auf einer Evaluation der Fachhochschule Gelsenkirchen betrug bereits im Jahre 2006 die durchschnittliche Datenrate eines Unternehmens 95.1 Mbps, wobei Firmen ab 4000

[1] TLS entspricht der Weiterentwicklung von SSL, SSL 3.1 entspricht hierbei TLS 1.0



(a) Mittelständisches Unternehmen. **(b)** Großer, international agierender Konzern.

Abbildung 2.2: Szenario vernetzte Unternehmen. Kleine und mittelständische Unternehmen setzen Verschlüsselung ein, um bspw. Dienste und Angebote für Kunden gesichert bereitzustellen, oder um Mitarbeitern den Zugang zu Daten von außerhalb zu ermöglichen. Die verteilten Standorte eines großen Konzerns sind weiterhin über VPNs miteinander verbunden. Weitere verschlüsselte Verbindungen können durch die Nutzung entsprechender Dienste auftreten, bspw. wenn Mitarbeiter auf Webseiten im Internet zugreifen. Auch Innentäter können verschlüsselte Verbindungen missbrauchen, um Daten auszuschleusen. Schadsoftware, die sich auf Rechnern in den Firmennetzen befindet, kann Verschlüsselung nutzen um bspw. unentdeckt mit einem C&C-Server zu kommunizieren.

Tabelle 2.1: Abgrenzungskritierien für Unternehmen gem. Empfehlung der Europäischen Kommission von 1996 und der KMU-Definition der EU von 2003 (gültig seit 2005) [324], [100]

Unternehmensart	Umsatz **oder** Bilanzsumme [€]		Mitarbeiter
Empfehlung der Europäischen Kommission (03.04.1996)			
Kleinunternehmen	$<$ 7 Mio.	$<$ 5 Mio.	$<$ 50
Mittlere Unternehmen	7 - 40 Mio.	5 - 27 Mio.	50 - 249
Großunternehmen	$>$ 40 Mio.	$>$ 27 Mio.	$>$ 249
KMU-Definition der EU (01.01.2005)			
Kleinstunternehmen	$\leq$ 2 Mio.	$\leq$ 2 Mio.	$<$ 10
Kleine Unternehmen	$\leq$ 10 Mio.	$\leq$ 10 Mio.	$<$ 50
Mittlere Unternehmen	$\leq$ 50 Mio.	$\leq$ 43 Mio.	$<$ 250

Beschäftigten fast ausnahmslos mindestens $2 \cdot 155$ Mbps zur Verfügung hatten [27]. Das durchschnittliche monatliche Datenaufkommen eines Unternehmens belief sich demnach auf 2777.6 GB.

Im Jahre 2010 hatte ein Breitband-Internetanschluss in Deutschland (gemeinsame Betrachtung kommerziell und privat) im Schnitt eine Datenrate von 11.63 Mbps im Downstream und 1.28 Mbps im Upstream, weltweit betrug die durchschnittliche Download-Bandbreite 5.92 Mbps [95]. Gemäß den Plänen der Bundesregierung sollen bis 2014 bereits 75 Prozent der Haushalte Anschlüsse mit Übertragungsraten von mindestens 50 Mbps besitzen (Breitbandstrategie der Bundesregierung [156], vgl. auch [123] und [79]).

Für den Unternehmensbereich stehen weiterhin dedizierte Anbindungen zur Verfügung, d.h. Anbindungen mit fester, garantierter Bandbreite und entsprechenden SLAs. Zum Beispiel bietet Verizon Business Deutschland derzeit[2] dedizierte Zugänge von 768 Kilobits per second (Kbps) bis zu 2488 Gbps an [124].

Traditionelle Dienste wie Mail gehören zum Alltag jedes Unternehmens, aber auch die Migration traditioneller Dienste wie Telefonie in das Internet spielt eine immer wichtigere Rolle. Beispielhaft sei hier VoIP genannt, das 2006 bereits in über einem Drittel (34 Prozent) der deutschen Unternehmen eingesetzt wurde, ein weiteres Drittel (29 Prozent) plante die Einführung [386]. 2010 wurden bereits 27 Prozent aller internationalen Gespräche über VoIP geführt [375]. Dienste im World Wide Web, Austausch elektronischer Mail, virtuelle Netze zwischen verschiedenen Standorten, gesicherte Verbindungen und interaktive Dienste müssen daher betrachtet werden (vgl. [27]).

Berücksichtigt man weiterhin Studien zur Entwicklung des Breitbandes in den nächsten Jahren (siehe z.B. [94], [96], [95]) sowie Umfragen zu geplanten Bandbreitensteigerungen und dem Einsatz von VoIP, Videokonferenzen und anderen neuen Technologien, werden folgende Datenraten und -volumina für das Szenario festgesetzt:

Kleines Unternehmen: Datenrate 100 Mbps, monatl. Datenvolumen 1.2 TB

[2] Stand Februar 2011

Großkonzern: Datenrate 2 Gbps, monatl. Datenvolumen 15 TB

Nachfolgend aufgeführte Dienste bzw. Protokolle werden insbesondere intensiv genutzt:

- Webdienste, insb. Hypertext Transfer Protocol (HTTP) und zunehmend Hypertext Transfer Protocol Secure (HTTPS)
- E-Mail (Post Office Protocol (POP)/Internet Message Access Protocol (IMAP) und Simple Mail Transfer Protocol (SMTP), bzw. POP over SSL (POPS)/IMAP over SSL (IMAPS) und SMTP over SSL (SMTPS))
- Dateitransfer, insb. File Transfer Protocol (FTP) und Secure File Transfer Protocol (SFTP)
- Sicherer Fernzugriff, Terminaldienste und Remote-Desktop-Dienste, insb. SSH, Remote Desktop Protocol (RDP)
- Telefonie und Videokonferenz, z.B. VoIP

Der skizzierte Großkonzern verwendet weiterhin wie beschrieben verschlüsselte Verbindungen auf Layer-3 Basis zwischen seinen Standorten (VPN, insb. IPsec). Ebenso läuft die Kommunikation sowohl zwischen verschiedenen Firmensitzen als auch zwischen der Firma und den Geschäftspartnern verschlüsselt ab. Im Falle des Großkonzerns können Mitarbeiter auch mobil vom Internet aus auf das Firmennetz mittels kryptierter Verbindung zugreifen.

Ebenfalls ist charakteristisch, dass kleine Unternehmen keine eigene IT-Abteilung haben, erforderliche Aufgaben werden auf Auftragsbasis vergeben bzw. werden IT-Dienstleister mit verschiedenen SLAs genutzt, während Großkonzerne typischerweise über eigene IT-Kapazitäten verfügen.

Ein weiterer Unterschied ist die Auslastung der IT-Struktur; während kleine Unternehmen zu festgelegten Arbeitszeiten neigen, werden die Systeme eines weltweit agierenden Konzerns kontinuierlich genutzt: Durch die weit verteilten Standorte und die entsprechenden Zeitverschiebungen, werden das Netz und die zentralen IT-Komponenten wie Server und Datenbanken in einer 24/7-Nutzung belastet.

Die Merkmale der IT-Umgebung der beiden betrachteten Unternehmen sind in Tabelle 2.2 zusammengefasst.

2.2 Bedrohungen

Im Folgenden werden die im Szenario vorkommenden Bedrohungen analysiert.

Definition (Bedrohung)**.** *Eine Bedrohung ist ein Umstand oder Ereignis, der oder das die Verfügbarkeit, Integrität oder Vertraulichkeit von Informationen beeinträchtigen*

Tabelle 2.2: Merkmale der betrachteten IT-Umgebungen der Unternehmen.

	Kleines Unternehmen	Großunternehmen
Bandbreite	100 Mbps	2 Gbps
Datenvolumen	1.2 TB	15 TB
Dienste	Web, Mail, Telefonie, ...	
IT-Fachkräfte	bedarfsmäßig	permanent
Nutzung	Arbeitszeiten	24/7

kann, wodurch dem Besitzer bzw. Benutzer der Informationen ein Schaden entstehen kann [72].

Um eine umfassende und lückenlose Analyse der möglichen Bedrohungen zu erhalten, werden hierbei u.a. die IT-Grundschutz-Kataloge des BSI [72] genutzt. Diese beinhalten umfassende Gefährdungskataloge für alle Teilaspekte eines Informationsverbundes.

2.2.1 Bedrohungsanalyse nach IT-Grundschutz

Der IT-Grundschutz des BSI bietet eine einfache Methode, dem Stand der Technik entsprechende Sicherheitsmaßnahmen zu identifizieren und umzusetzen. Das BSI stellt hierzu zahlreiche Werkzeuge zur Verfügung, um ein angemessenes Sicherheitsniveau zu erreichen, bspw. die BSI-Standards zum Informationssicherheitsmanagement, die IT-Grundschutz-Kataloge oder die Software GSTOOL [69].

Die Kataloge geben im Baukastenprinzip verschiedene Bausteine für typische Abläufe von Geschäftsprozessen und Bereiche des IT-Einsatzes vor, z.B. für IT-Infrastruktur, bauliche Einrichtungen oder Applikationskomponenten. Jeder Baustein beschreibt die Gefährdungslage und gibt Maßnahmen für die Bereiche Infrastruktur, Personal, Organisation, Hard- und Software, Kommunikation und Notfallvorsorge an.

Die Sicherheitsaspekte des IT-Grundschutz-Modells sind in fünf Schichten gruppiert, um die Komplexität eines Informationsverbundes besser handhaben zu können (vgl. Abbildung 2.3).

Schicht 1 behandelt übergreifende Sicherheitsaspekte, die für alle oder große Teile des Verbundes gleichermaßen gelten. Hierzu gehören bspw. organisatorische Punkte, Sicherheitsmanagement und Datensicherungskonzept.

Infrastruktur-Komponenten werden in **Schicht 2** dargestellt, dies betrifft alle baulich-physischen Gegebenheiten. Hierunter zählen u.a. Gebäude, Serverraum oder häuslicher Arbeitsplatz.

Schicht 3 behandelt die einzelnen IT-Systeme wie Server, Clients oder Einzelplatzsysteme.

Vernetzungsaspekte, also die Netzverbindungen sowie die Kommunikation werden in **Schicht 4** betrachtet, während sich **Schicht 5** schließlich mit den eigentlichen Anwendungen auseinander setzt, bspw. einer Datenbank oder einem Webserver.

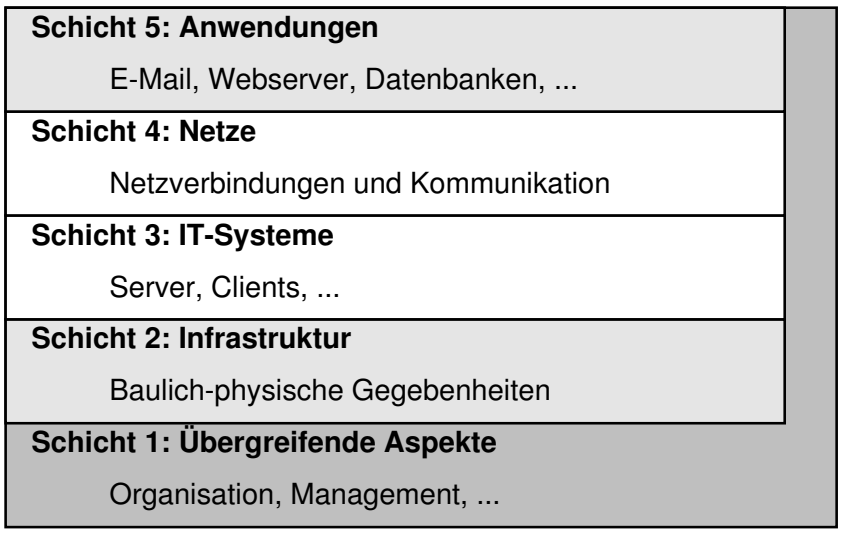


Abbildung 2.3: Die Schichten des IT-Grundschutz-Modells. Für die Sicherheitsbetrachtung des Netzes und des Datenverkehrs müssen maßgeblich die Schichten 3 und 4 analysiert werden.

Die einzelnen Schichten werden in den gleichnamigen Bausteinen detailliert untersucht, bspw. finden sich im Baustein IT-Systeme die Unterpunkte *Allgemeiner Server*, *Server unter Unix*, etc. Die jeweiligen Einträge sind weiterhin untergliedert in eine allgemeine Beschreibung, in die Analyse der Gefährdungslage gem. den Gefährdungskatalogen, sowie den Maßnahmeempfehlungen gem. den Maßnahmenkatalogen.

Für das beschriebene Szenario werden nun die Gefährdungsaspekte, welche für die Netzkommunikation relevant sind, gesammelt. Gemäß Abbildung 2.3 betrifft dies maßgeblich die Schichten *3 (IT-Systeme)* und *4 (Netze)*. Organisatorische Aspekte, Infrastruktur-Gegebenheiten, etc. fallen nicht in den untersuchten Bereich des Szenarios und können daher ausgeschlossen werden.

Folgende Bausteine[3] der Grundschutzkataloge treffen im dargestellten Szenario zu und kommen somit zur Anwendung (vgl. [71], [70]):

- Schutz vor Schadprogrammen (B 1.6)
- Kryptokonzept (B 1.7)
- Hard- und Softwaremanagement (B 1.9)
- Heterogene Netze (B 4.1)
- Netz- und Systemmanagement (B 4.2)
- VPN (B 4.4)

[3]Die Referenzierungen in Klammern geben direkt die jeweiligen Kapitel der Kataloge an, vgl. [72]

Nachfolgend werden die Punkte der Gefährdungslage der genannten Bausteine gesammelt, um sämtliche Gefährdungsaspekte zu berücksichtigen. Aspekte, die nicht im direkten Kontext eines Sicherheitssystems in einer Netzumgebung und somit dem Szenario stehen, werden hierbei ausgeschlossen (z.B. *Fahrlässige Zerstörung von Gerät oder Daten (G 3.2)*). Die konkatenierten Daten werden anschließend in einer detaillierten Betrachtung aller maßgeblichen Punkte der Gefährdungskataloge um evtl. weitere für das Szenario relevante, jedoch in den Bausteinen nicht vorhandene Aspekte ergänzt.

Anhang F.1.1 listet die identifizierten Aspekte auf und gruppiert diese anschließend zu Klassen. Nachfolgend erfolgt nur die Vorstellung der Zusammenfassung der Punkte.

Die Gefährdungslage ist in die Bereiche

- Höhere Gewalt
- Organisatorische Mängel
- Menschliche Fehlhandlungen
- Technisches Versagen
- Vorsätzliche Handlungen

gegliedert. Zu den menschlichen Fehlhandlungen zählen bspw. eine ungewollte Freigabe des Dateisystems oder eine unerlaubte Systemnutzung; technisches Versagen beinhaltet u.a. die Punkte Software-Schwachstellen und -Fehler. Der umfangreichste Teil betrifft die vorsätzlichen Handlungen, zu denen zum Beispiel der Missbrauch von Administratorrechten, Trojanische Pferde und Schadprogramme zählen.

Insbesondere können folgende sechs Kategorien abstrahiert werden:

- Angriffe bis Layer 4
- Angriffe auf höherern Layern, insbesondere Schicht 7
- Denial of Service
- Missbrauch von Privilegien
- Manipulation
- Datenabfluss

Die identifizierten Aspekte sind detailliert im Anhang F.1.1 aufgeführt und in der dort ebenfalls befindlichen Tabelle F.1 zusammengefasst.

Neben den Gefährdungen enthalten die Bausteine der Grundschutzkataloge Maßnahmen, um den jeweiligen Bedrohungen entgegentreten zu können. Dies betrifft maßgeblich die Konfiguration und Administration von Systemen und Anwendungen sowie des organisatorischen Umganges mit diesen.

Da dies insbesondere präventive Maßnahmen sind, kann die Anzahl der für das Szenario zutreffenden Punkte, welches die aktive Überwachung einer in Betrieb befindlichen Produktivumgebung wie im vorliegenden Szenario fokusiert, auf die folgenden Punkte der entsprechenden Grundschutzkataloge reduziert werden:

- Audit und Protokollierung der Aktivitäten im Netz (M 4.81)
- Sichere Konfiguration der aktiven Netzkomponenten (M 4.82)
- Kommunikation durch Paketfilter auf Minimum beschränken (M 4.98)
- Sicherung von Switch-Ports (M 4.206)
- Schutz vor unerwünschten Informationsabflüssen (M 4.345)
- Regelmäßiger Sicherheitscheck des Netzes (M 5.8)

Weiterhin können die generellen Aspekte

- Einsatz von Verschlüsselungsverfahren zur Netzkommunikation (M 5.68)
- Intrusion Detection und Intrusion Response Systeme (M 5.71)

als Notwendigkeit und Grundanforderung bzgl. der Sicherheitsanforderungen für das Szenario festgestellt werden. Tabelle F.2 im Anhang F.1.1 zeigt die Zuordnung der jeweiligen Maßnahmen zu den Gefährdungskategorien.

In die Kategorie der *Vorsätzlichen Handlungen* fällt insbesondere auch der sog. *Innentäter*. Da dessen Rolle in der jüngeren Literatur kontrovers diskutiert wurde, erfolgt nachfolgend eine Evaluation der von Innentätern ausgehenden Bedrohung.

2.2.2 Bedeutung des Innentäters

Definition (Insider)**.** *Bezeichnung für eine Person, die innerhalb eines Unternehmens eine Vertrauensstellung einnimmt und über wichtige, nicht allgemein zugängliche Informationen verfügt [379].*

Durch sein Wissen über Firmeninterna sowohl hinsichtlich geschäftlicher Daten als auch in Bezug auf Details bspw. über das Firmen-Local Area Network (LAN) sowie seine entsprechend vorhandenen, legalen Zugangsmöglichkeiten, hat ein Innentäter ein enormes Schadenspotential.

Die Rolle des Innentäters wird in der Literatur sehr kontrovers diskutiert, abhängig der jeweiligen Studie reichen Angaben von „lediglich“ 18 Prozent bis hoch zu über 80 Prozent. Hierbei hielten sich die hohen Einschätzungen von 80 Prozent, welche auf einer knapp 20 Jahre alten Studie des Federal Bureau of Investigation (FBI) basierten und über viele Jahre nicht hinterfragt wurden, bevor in jüngerer Zeit kontroverse Diskussionen aufgrund neuer Studienergebnisse aufkamen. In einer genaueren Analyse der vom FBI 2002 und 2003 veröffentlichten Statistiken kam Richard Bejtlich zum Schluß, dass es sich bei der Innentäterbedrohung um eine „Legende“ handle und diese stark überbewertet wird [54]. Auch der *Data Breach Investigation Report* von Verizon aus dem Jahre 2008 kam zu einem gleichen Schluss und stellte einen Innentäter-Anteil von 18 Prozent fest [46]. Andererseits stellen gerade neue Untersuchungen der letzten Jahre wieder einen

deutlich höheren Anteil an Innentätern fest, der sich typischerweise auf Werte zwischen 40 und 50 Prozent manifestiert.

Im Folgenden soll daher eine kurze Gegenüberstellung der Ergebnisse verschiedener aktueller Studien und Analysen erfolgen, um die reale Gefährdung abzuschätzen. Anhand der Ergebnisse wird die Gefährdung durch Innentäter im vorliegenden Szenario bewertet.

Folgende Quellen werden bzgl. der relevanten Aussagen zu Innentätern herangezogen:

- Verizon *Data Breach Investigation Report* der Jahrgänge 2008 bis 2011 (im Folgenden bezeichnet als Verizon10, etc.)
- KPMG *e-Crime-Studie 2010* (KPMG10),
- Corporated Trust *Studie: Industriespionage* (CT07),
- Studie von Pricewaterhouse Coopers und der Martin-Luther-Universität Halle-Wittenberg *Wirtschaftskriminalität 2009* (PC09)
- Berichte der Verfassungsschutzbehörden von Bund und Ländern

Verizon Data Breach Investigation Reports

Das Verizon Business Risk Team wertet seit 2004 Datenschutzverletzungen aus und hat hierfür bisher über 900 Fälle mit mehr als 900 Mio. kompromittierten Datensätzen gesammelt. Der erste Bericht dazu erschien 2008 [43] und gab einen Anteil von 18 Prozent Innentätern an, dieser blieb in der Evaluation von 2009 relativ konstant bei 20 Prozent [44]. Für das Reportjahr 2010 wurden die Datensätze, welche Verizon zur Analyse vorlagen, um Daten des United States Secret Service (USSS) ergänzt [45]. Die dadurch breitere evaluierte Datenbasis spiegelt sich in teils deutlichen Unterschieden zu den Ergebnissen der Vorjahre wieder. Abbildung 2.4 stellt die Entwicklung der Auswertungen der letzten vier Jahre dar.

Auffällig ist der starke Anstieg der Innentäter-Gruppe auf 48 Prozent im Berichtsjahr 2010, entsprechend geht hier ein starker Anstieg des Missbrauchs von Privilegien einher. Zu beachten ist weiterhin der Anstieg von *Social Engineering*-Taktiken, die typischerweise mit zielgerichteten Angriffen einhergehen (vgl. Kapitel 4.6.1). Interessant ist, dass im Berichtsjahr 2011 der Anteil an Innentätern erneut auf 17 Prozent zurück geht. Dies liegt jedoch massgeblich *nicht* an einem bedeutenden Rückgang der durch Innentäter durchgeführten Aktivitäten, sondern an einem extremen Anstieg von kleinen, externen Angriffen [42].

Corporate Trust: Studie Industriespionage

Die Studie von Corporate Trust aus dem Jahre 2007 fokusiert den Umfang der Industriespionage in Deutschland [104], demnach hatten bereits 18.9 Prozent der Unternehmen einen Fall von Spionage oder Datenabfluss zu verzeichnen, 35.1 Prozent hatten weiterhin den Verdacht, dass sie Opfer eines Informationsabflusses wurden, konnten dies jedoch

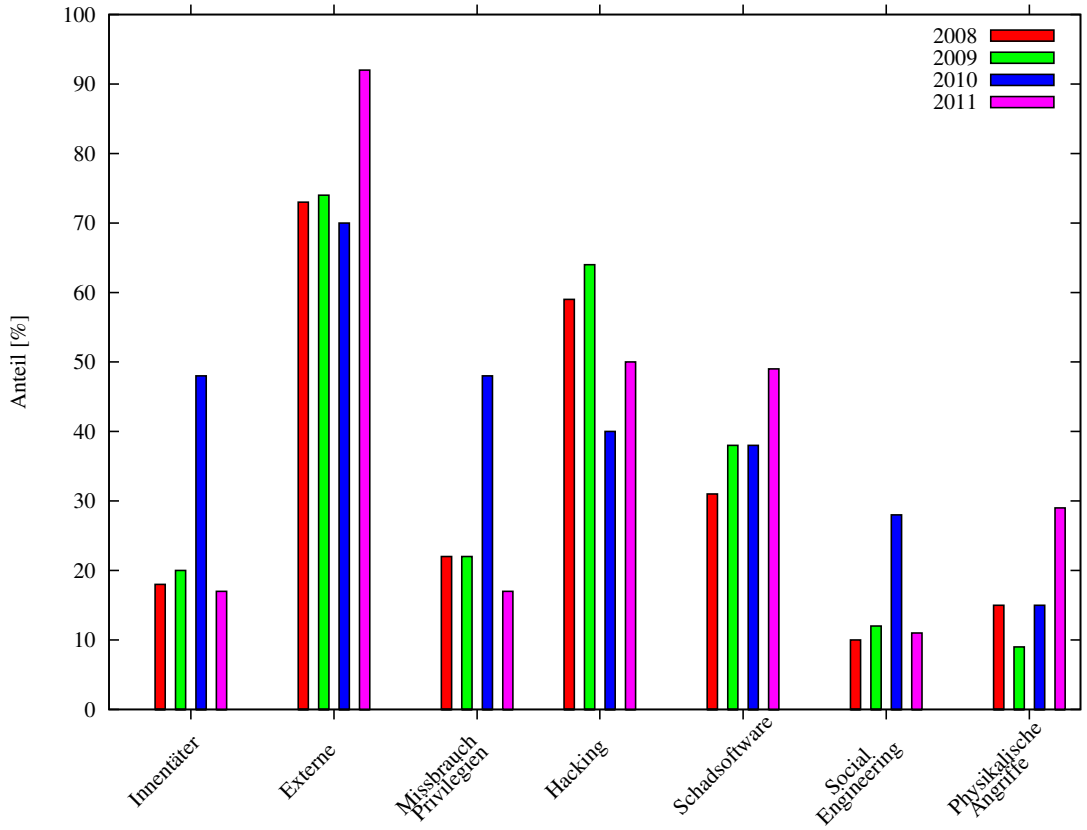


Abbildung 2.4: Tätergruppen und Datenverletzungen gem. Verizon in den Berichtsjahren 2008 bis 2011. Insbesondere ist der starke Anstieg der Gruppe der Innentäter im Jahre 2010 offensichtlich. Dieser beruht darauf, dass die analysierten Daten in diesem Jahr um die Daten des Secret Service erweitert wurden, was eine umfassendere Datenbasis wiederspiegelt. Auf der anderen Seite ist für das Berichtsjahr 2011 trotz weiterhin einbezogenen Daten des Secret Service ein erneuter Rückgang auf das Niveau vor 2010 zu verzeichnen. Dies liegt jedoch nicht an einer erneut gesunkenen Anzahl von Fällen aus dem Bereich der Innentäter, sondern an einem sehr starken Anstieg registrierter, externer Vorfälle. Entsprechend ist ein gleiches Entwicklungsverhalten in den Bereichen *Missbrauch von Privilegien* sowie *Social Engineering* erkennbar.

nicht weiter belegen. Das Risiko wird hier jedoch typischerweise unterschätzt: Zwar glauben einerseits 80 Prozent der Befragten, dass die Gefahr für Industriespionage weiterhin weltweit ansteigen wird, jedoch sehen nur 33.7 Prozent auch eine Gefahr für das eigene Unternehmen (vgl. auch Abschnitt 2.2.2).

Als Ursache für Informationsabfluss identifiziert die Studie die eigenen Mitarbeiter der betroffenen Unternehmen mit einem Anteil von 20.3 Prozent[4], gefolgt von Hackerangriffen mit 14.9 Prozent.

Betroffen sind hierbei insbesondere kleinere (38.5 Prozent) und mittelständische Unternehmen (57.6 Prozent), während Konzerne nur zu 3.9 Prozent betroffen waren.

Die Studie legt ebenfalls dar, dass bestimmte Branchen wie bspw. Pharmaunternehmen oder Banken keine Angaben zu aufgetretenen Schäden machten, obwohl alle anderen Sparten von Spionagefällen berichteten. Da es sich gerade bei diesen Branchen um sehr sensible Bereiche handelt, bei denen ein Spionagefall zu erheblichen Imageverlust führen kann, kommen die Analysten zum Schluss, dass hier von einer entsprechenden Dunkelziffer ausgegangen werden muss. Dies wird durch die Angaben der befragten Unternehmen verstärkt, in 73.9 Prozent der Fälle keine Ermittlungsbehörden eingeschaltet zu haben.

Wirtschaftskriminalität 2009

In der Studie *Wirtschaftskriminalität 2009* nehmen Pricewaterhouse Coopers und die Martin-Luther-Universität Halle-Wittenberg die Sicherheitslage in deutschen Großunternehmen zum Schwerpunkt [82]. Hierbei wurden Unternehmen ab 500 Mitarbeitern[5] stichprobenartig befragt. Die Studie stellt das hohe von Wirtschaftskriminalität ausgehende Risiko dar. Demnach berichteten 58 Prozent der Unternehmen, im Durchschnitt in acht Schadensfällen geschädigt worden zu sein, wobei der durchschnittliche finanzielle Schaden bei 5.57 Mio. € lag. Der durchschnittliche finanzielle Schaden der genannten 210 schwersten Wirtschaftsdelikte entwickelte sich dabei von 4.43 Mio. € im Jahre 2005 auf 30.01 Mio. € im Jahre 2009. Die Delikte Industrie- und Wirtschaftsspionage waren hier mit 7 Prozent, Diebstahl vertraulicher Kunden- und Unternehmensdaten mit 21 Prozent vertreten. Als von besonderer Bedeutung werden die mittelbaren Schäden durch Reputationsverlust herausgestellt. Während 2007 nur von 27 Prozent der Unternehmen von einem gravierenden Reputationsverlust berichtet wurde, waren es 2009 bereits 44 Prozent. Über den Diebstahl von vertraulichen Kunden- und Unternehmensdaten berichteten 21 Prozent aller Befragten, die damit verbundenen Schäden waren die zweithöchsten aller Delikte, im Schnitt 3.8 Mio. €. Von Delikten der Wirtschafts- und Industriespionage wurde lediglich von 7 Prozent der Unternehmen berichtet, die Studie legt jedoch nahe, dass das Dunkelfeld erheblich höher sein dürfte.

[4] Spionagehandlungen, bei denen der Täter ermittelt werden konnte, gingen in 24 Prozent der Fälle von Mitarbeitern aus.

[5] Prozentualer Anteil der Gruppen: 500-1000 (25), 1001-5000 (39), 5001-10000 (11), 10001-25000 (8), 25001-50000 (7), 50001-150000 (7), über 150000 (3).

e-Crime-Studie 2010

Die Wirtschaftsprüfungsgesellschaft KPMG AG untersucht in der *e-Crime Studie 2010* die Bedeutung der Computerkriminalität in der deutschen Wirtschaft [234]. Hierbei wurden 500 Unternehmen, die anhand ihres Umsatzes in die Gruppen „klein“, „mittel“ und „groß“[6] eingeteilt wurden, befragt. Nach den Auswertungen von KPMG waren ein Viertel der befragten Unternehmen innerhalb der letzten drei Jahre von Delikten der Computerkriminalität betroffen, wobei große Unternehmen mehr im Visier der Angreifer standen (31 Prozent), als mittlere (26 Prozent) und kleine Unternehmen (22 Prozent). Als höchstes Risiko im Bereich Computerkriminalität gaben die Unternehmen Diebstahl von Kunden- oder Arbeitnehmerdaten durch Mitarbeiter oder ehemalige Arbeitnehmer an (54 Prozent) und Diebstahl von geschäftskritischem Know-how (51 Prozent). Der Diebstahl von Daten durch Externe wurde mit einem Risiko von 41 Prozent bewertet. Bezüglich der Tätergruppe wurden ehemalige Mitarbeiter und Insider mit 70 Prozent mit Abstand am gefährlichsten bewertet.

Diese Einschätzung in Bezug auf das Risiko durch Innentäter wurde durch tatsächliche Erfahrungen der Unternehmen untermauert: In 48 Prozent aller vorgefallenen Delikte der Computerkriminalität waren Mitarbeiter die Täter. Die Schadenshöhe der analysierten Fälle von Datendiebstahl betrug demnach im Schnitt über 1 Mio. € pro Vorfall, Hauptgefahrenquelle waren hierbei Mitarbeiter und ehemalige Angestellte, sowie andere Insider wie Geschäftspartner.

Bei der Bewertung der Entwicklung der Vorfälle in den letzten Jahren gaben die Unternehmen folgende Punkte an:

- Zunehmende Komplexität der Angriffe (86 Prozent)
- Zunehmende Schwierigkeiten, Angriffe überhaupt zu detektieren (84 Prozent)
- Ausnutzung von Schwachstellen in Technologien, für die es noch unzureichende Schutzmaßnahmen gibt (82 Prozent)
- Mehr professionell organisierte Gruppen als Täter (72 Prozent)
- Mehr auf bestimmte Bereiche oder Daten zielgerichtete Angriffe (62 Prozent)
- Angreifer sitzen häufiger im Ausland (60 Prozent)
- Social Engineering spielt eine immer stärkere Rolle (53 Prozent)
- Angriffe können zunehmend durch technisch ungeschulte Personen verübt werden (33 Prozent)

[6] Als „groß“ wurden Unternehmen mit einem Jahresumsatz von mehr als 3 Mrd. € eingestuft, „mittlere“ Unternehmen von 250 Mio. € bis 3 Mrd. € und „klein“ im Bereich von 50 Mio. € bis 249 Mio. €. Hier muss angemerkt werden, dass es sich hierbei durchgängig um Großunternehmen handelt, wenn man die Definitionen der EU zu Grunde legt (vgl. Tabelle 2.1).

Weiterhin von Bedeutung ist, dass der Großteil von Delikten nicht angezeigt wurde, wenn ein Angriff durch bestehende Sicherheitsmechanismen abgewehrt werden konnte oder kein finanzieller Schaden entstand. Die Studie weisst ebenfalls darauf hin, dass zwar die Gefahr durch Insider von den Unternehmen als sehr hoch eingeschätzt wird, die von Systemadministratoren ausgehenden Gefahren jedoch meist unterschätzt werden.

Berichte der Verfassungsschutzbehörden von Bund und Ländern

Polizeiliche Kriminalstatistik 2009, IuK-Kriminalität Bundeslagebild 2009 Aus der PKS für das Jahr 2009 geht hervor, dass die Gesamtzahl von Straftaten im Jahr 2009 in Bezug auf das Vorjahr um 1 Prozent auf 6054330 Fälle zurückgegangen ist [75]. Dahingegen sind die Delikte aus dem Bereich Computerkriminalität um 17.7 Prozent angestiegen, besonders zu nennen sind hier das *Ausspähen und Abfangen von Daten* mit einem Zuwachs von 48.7 Prozent (siehe auch [74]). Extra betrachtet werden die *Straftaten mit Tatmittel Internet*, welche einen Anstieg um 23.6 Prozent im Vergleich zum Vorjahr aufzuweisen hatten.

Broschüren des Verfassungsschutzes In mehreren Broschüren weist der Verfassungsschutz von Bund und Ländern auf besondere Gefahren hin, welche insbesondere kleine und mittelständische Unternehmen betreffen und jährlich immense Schäden verursachen [154]. Insbesondere wird vor den von Innentätern ausgehenden Gefahren und dessen Bedeutung mit einem Anteil von 70 Prozent gewarnt (vgl. [153], [390]) sowie die Bedeutung elektronischer Attacken auf IuK-Strukturen hervorgehoben [151]. Die insbesondere von den Sozialen Netzwerken im Web 2.0 ausgehenden Gefahren und deren Ausnutzung mittels Social Engineering-Angriffen werden durch den Verfassungsschutz unterstrichen (vgl. [152], [121]), desgleichen die Gefährdungen für Forschung und Lehre, die durch Wissenschaftsspionage entstehen [155]. Wissenschaftsspionage wird insbesondere durch chinesische Studenten und Geschäftsleute, die außerhalb von China tätig sind, durchgeführt. Letztere sind als Gegenleistung für die Arbeit im Ausland gegenüber der Regierung verpflichtet, Spitzeldienste zu erbringen [407]. Hierdurch ist nicht nur die Wirtschaft, sondern insbesondere auch Forschung und Lehre betroffen. Der Verfassungsschutz schreibt hierzu:

> Alternativ werden Gaststudenten und Gastwissenschaftler in zukunftsträchtige Projekte eingeschleust, um für ihr Heimatland zu spionieren. Zum Teil ist dies Voraussetzung für ein Studium oder die Arbeit im Ausland. Die Pflicht oder Ehre das eigene Land zu unterstützen, lässt dabei kein Unrechtsbewusstsein entstehen [155].

Mit in den Bereich der Innentäter können auch *unfreiwillige* Preisgaben durch Beschäftigte eines Unternehmes gezählt werden, z.B. wenn Zugangsinformationen und Daten im Rahmen eines Social Engineering-Angriffs preisgegeben wurden. In diesem Rahmen kommt es somit zu „unwissenden“ Innentätern, die nicht in einer Gewinn- bzw. Schädigungsabsicht agieren, sondern im Glauben handeln, im Sinne des Arbeitgebers tätig zu werden. Da der eigentliche Angriff in diesem Falle von Extern erfolgt und daher nicht

Tabelle 2.3: Innentäter-Gefahr aus Sicht verschiedener Studien im Zeitraum 2007 bis 2010.

Studie	Verizon10	KPMG10	CT07	PC09	Verfassungsschutz
Länder	17 Länder[7]	DEU	DEU	DEU	DEU
Unternehmen	Alle Größen[8]	Konzerne	Alle Größen[9]	Alle Größen[10]	k.A.
Probengröße	141	500	741	500	k.A.
Zielgruppe	k.A.	*entfällt*	KMU	*entfällt*	KMU
Anteil Insider	48	48	20.3	k.A.	70
Social Engineering	28	53	8	k.A.	k.A.
Anzeige	k.A.	k.A.	26.1	50 / 60[11]	*„selten“*

in die Kategorie des Innentäters im engeren Sinne fällt, für die Detektion des Datenabflusses die Situation jedoch als Insider-Angriff angesehen werden kann, wird im weiteren Verlauf der Arbeit eine erweiterte Definition des Innentäters wie folgt genutzt:

Definition (Erweiterter Innentäter-Begriff)**.** *Bezeichnung für eine Person, die innerhalb eines Unternehmens eine Vertrauensstellung einnimmt und über wichtige, nicht allgemein zugängliche Informationen verfügt und durch ihr Handeln in Absicht oder unwissentlich einen Datenabfluss herbeiführt.*

Tabelle 2.3 fasst die Kernaussagen der verschiedenen Studien und Statistiken nochmals zusammen.

Die angegebenen 70 Prozent des Verfassungsschutzes liegen deutlich höher, als die anhand von tatsächlichen Vorfällen ermittelten Zahlen. Hier muss jedoch berücksichtigt werden, dass dies den Zahlen der ausgewerteten Fälle entspricht, inbesondere geben viele Unternehmen *geschätzte* Zahlen von bis zu 70 Prozent Innentäteranteil an, was sich somit mit den Werten des Verfassungsschutzes deckt. Von besonderer Bedeutung ist hier auch die Dunkelziffer, die durchgehend als sehr hoch eingeschätzt wird. Viele Unternehmen geben an, dass bereits die Feststellung, dass eine Preisgabe von Internas stattgefunden hat, immer schwieriger wird. Aufgrund der Angst vor einem Reputationsverlust kommt es hierbei auch nur selten zu Anzeigen.

Der relativ hohe Anzeigeanteil in `PC09`[12] widerspricht dem nicht, da diese Studie alle Straftaten der Wirtschaftskriminalität betrachtet, während die Werte der anderen Studien mit Schwerpunkt auf Datenverlust und Computerkriminalität ermittelt wurden. Zieht

[7] Über 50 Prozent der Fälle stammen aus den Vereinigten Staaten. Weitere Länder u.a. DEU.

[8] Prozentualer Anteil der Gruppen bzgl. der Anzahl von Mitarbeitern: 1-10 (9), 11-100 (18), 101-100 (23), 1001-10000 (26), 10001-100000 (20), über 100000 (2).

[9] Keine Aufschlüsselung.

[10] Prozentualer Anteil der Gruppen bzgl. der Anzahl von Mitarbeitern: 500-1000 (25), 1001-5000 (39), 5001-10000 (11), 10001-25000 (8), 25001-50000 (7), 50001-150000 (7), über 150000 (3).

[11] Strafanzeige gegen interne bzw. externe Täter.

[12] Im Vergleich zur Studie der Vorjahre sind diese Werte auch jeweils bereits um ca. 10 Prozent rückläufig.

man die Angaben von KPMG10 hinzu, gem. denen Anzeigen meist gemacht werden, jedoch bei erfolgreicher Abwehr oder bei fehlendem finanziellen Verlust ausbleiben, lässt sich schließen, dass Großunternehmen eher bereit sind, eine Strafverfolgung anzuberaumen als kleine Unternehmen.

Dass kleine und mittlere Unternehmen besonders von Spionage betroffen sind, lässt sich aus allen Studien schließen; die Angaben von KPMG10, wonach eher große Unternehmen betroffen sind, widerspricht dem nicht: Die als klein, mittel und groß definierten und untersuchten Unternehmensgrößen fallen nach EU-Definition sämtlich unter Großunternehmen, somit ist aus den entsprechenden Werten höchsten eine leichte Differenzierung der Gefährdung *innerhalb* der Kategorie Großunternehmen abzuleiten.

Betrachtet man die in den jeweiligen Studien berücksichtigten Unternehmensgrößen, deren Anteile am Markt, die Detektionsmöglichkeiten sowie den typischen (Nicht-) Reaktionen durch die Firmen bzgl. Meldungen und Anzeigen, ergibt sich trotz auf den ersten Augenschein hin widersprüchlichen Daten ein schlüssiges Bild.

Zusammengefasst lässt sich sagen, dass die Gefahr durch Innentäter insbesondere in kleinen und mittleren Unternehmen sehr hoch ist. Bei Großunternehmen ist die Gefährdung relativ gesehen geringer, jedoch entstehen durch die dort vorliegenden Informationswerte auch bei einer geringeren Anzahl von Fällen hohe Schadenssummen. Die Bedrohung durch Innentäter ist daher bei der Betrachtung eines Sicherheitssystems von besonderer Bedeutung.

Dieser wird daher in die Anforderungen, welche an das Sicherheitssystem gestellt werden müssen, aufgenommen.

2.3 Angriffsanalyse

Die Kenntnis über den genauen Aufbau und Ablauf von Angriffen ist von elementarer Bedeutung, um die Gefährdung für IT-Komponenten und Infrastrukturen zu analysieren; nur dann können geeignete Detektions- und Gegenmaßnahmen getroffen werden.

2.3.1 Definition

Definition (Angriff)**.** *Ein Angriff ist eine vorsätzliche Form der Gefährdung, nämlich eine unerwünschte oder unberechtigte Handlung mit dem Ziel, sich Vorteile zu verschaffen bzw. einen Dritten zu schädigen. Angreifer können auch im Auftrag von Dritten handeln, die sich Vorteile verschaffen wollen [72].*

Das Bundesministerium des Inneren (BMI) gibt im Verfassungsschutzbericht 2009 eine erweiterte Definition speziell für elektronische Angriffe an:

Definition (Elektronische Angriffe)**.** *Mit dem Begriff „Elektronische Angriffe“ werden gezielte Maßnahmen mit und gegen IT-Infrastrukturen bezeichnet. Neben der Informationsbeschaffung fallen darunter auch Aktivitäten, die zur Schädigung bzw. Sabotage dieser Systeme geeignet sind [78].*

Um Angriffe erkennen und zeitnah Gegenmaßnahmen ergreifen zu können, ist eine genaue Analyse der Gefährdungsquellen notwendig und die Kenntnis der einzelnen Angriffsschritte unerlässlich.

Im Folgenden erfolgt eine Klassifizierung möglicher Angriffe anhand von in der Literatur verbreiteten Modellen. Die für das Szenario (vgl. Kapitel 2.1) relevanten Angriffe werden herausgestellt und die verschiedenen, zugehörigen Angriffsstufen aufgeschlüsselt. Hierbei werden die genutzten Methoden sowie benötigte Informationen und Angriffspunkte analysiert.

2.3.2 Angriffs-Klassifizierung

Bei einer Klassifizierung oder Systematik handelt es sich um eine hierarchische Gliederung eines Wissensgebietes in Haupt- und Untergruppen [191]. Die Klassifizierung hat in einer Art und Weise zu erfolgen, dass mehrere Eigenschaften durch die entstehenden Gruppen erfüllt sind (vgl. z.B. [200], [220]):

- Eindeutigkeit (Wechselseitig ausschließend, *Mutually Exclusive*): Eine Zuordnung erfolgt in genau eine Kategorie, es gibt keine Überschneidungen zwischen den Kategorien.
- Vollständigkeit: Alle Kategorien zusammengefaßt decken alle Möglichkeiten ab.
- Nachvollziehbarkeit: Wiederholte Einteilung in die Kategorien führt zum gleichen Ergebnis, unabhängig von der durchführenden Person.
- Zweckdienlich: Die Klassifizierung ist geeignet, eine Hilfe für die Untersuchung und Evaluation des Gebietes zu geben.
- Verständlichkeit: Die Klassifizierung soll sowohl für Experten, als auch für in dem Gebiet interessierte Personen verständlich sein.
- Akzeptanz: Die Klassifizierung hat logisch und intuitiv zu erfolgen, um eine allgemeine Akzeptanz zu erlangen.

Das entstehende Verfahren oder Modell wird als Klassifikationsschema oder Taxonomie bezeichnet. Anhand der Einteilungen sollen das Wesen und die Charakteristika der einzelnen Klassen besser verstanden werden, um neue Systeme auf unbekannte Schwachstellen hin systematisch mittels des bestehenden Wissens zu analysieren.

In der Literatur finden sich verschiedene, vom Detaillierungsgrad teils sehr unterschiedliche Klassifizierungen für die Einteilung der in der IT vorkommenden Angriffe. Erste Veröffentlichungen entsprechender Einteilungen erfolgten bereits in den Jahren 1976 von Abbott et al. und 1978 von Bisbey und Hollingworth [189], wobei diese mehr eine Klassifizierung verschiedener Schwachstellen, als Angriffe darstellen. Das Spektrum reicht heute von generellen Modellen bis hin zu spezifischen Betrachtungen, bspw. der Taxonomie von Sybil-Angriffen durch Newsome et al. [295]. Hierbei finden sich zahlreiche eindimensionale Einteilungen, z.B. anhand der Auswirkungen eines Angriffs [118],

hierarchisch aufgebaute (z.B. Art des Angriffs (ziel- oder ungerichtet) auf der obersten Ebene, Anzahl der Angreifer als zweite Hierarchieebene, etc. [266]) und mehrdimensionale Taxonomien. Beispiel einer zweidimensionalen Gliederung ist die Angriffsmatrix nach Perry und Wallich, bei der die erste Dimension durch den Angreifer, die zweite durch die Art des Angriffs definiert wird (vgl. Tabelle F.3, Anhang F.1.3). Auch wenn die Einteilung und Belegung der Matrix von 1984 nach den heute vorhandenen Bedrohungen unvollständig und ungenügend ist, gibt sie ein Beispiel für eine entsprechende, zweidimensionale Gruppierung.

Trotz der zahlreichen existierenden Taxonomien konnte sich jedoch bisher kein allgemein akzeptiertes Modell bzw. Standard durchsetzen. Igure und Williams haben in ihren Arbeiten eine umfassende Auswertung der im Zeitraum 1974 bis 2006 veröffentlichten, sicherheitsbezogenen Taxonomien vorgelegt [206].

Grundlegend werden hier die beiden Kategorien *Angriffs-* und *Schwachstellen*-Taxonomien unterschieden. Abbildung 2.5 zeigt eine Zusammenfassung der von Igure erstellten Einteilung untersuchter Arbeiten und jeweilige Beispiele zugehöriger Veröffentlichungen.

In ihrer Analyse stellen Igure und Williams fest, dass zahlreiche Taxonomien existieren, die für spezielle Gebiete geeignet, jedoch nicht generalisierbar sind. Bspw. erfüllen einige Taxonomien nicht die grundlegenden Anforderungen wie Eindeutigkeit der Zuordnung aller Angriffe zu jeweils genau einer Klasse oder lassen nicht zuweisbare Angriffe übrig. So besteht die Einteilung von Neumann aus neun Kategorien, namentlich *Externen Angriffen*, *Maskierung*[13], *Schadprogrammen*, *Seitenkanäle* sowie den fünf Missbrauchskategorien *Aktiv*, *Passiv*, *Inaktiv*, *Indirekt* und *Hardware* [206]. Zu dieser Einteilung lässt sich jedoch keine gemeinsame Dimension angeben, so dass nicht ausgeschlossen werden kann, dass nicht weitere, unberücksichtigte Kategorien existieren. Lindquist und Johnson haben daher das Angriffsergebnis als Dimension für die Einteilung von Angriffen gewählt und sind hiermit zu den drei Klassen *Bloßstellung*, *Dienstverweigerung (Denial of Service (DoS))* und *Fehlerhafte Ausgabe* gekommen.

Eine andere Einteilung von Angriffen ist bspw. die IDS-orientierte Sichtweise von Kumar. Hier sind die Klassen anhand der Angriffssignaturen folgendermaßen eingeteilt: *Verfügbarkeit* einer Signatur, *Befehlsabfolge*, *Teilauftrag*, *Laufzeit* und *Intervall*. Diese Taxonomie ist geeignet, bekannte Angriffe während des Betriebs des Netzes zu erkennen, jedoch können die Signaturen nicht genutzt werden, um die zugehörige, ausgenutzt Schwachstelle des Systems zu identifizieren. Diese Klassifizierung ist daher nur aus Sicht einer Betriebsüberwachung, nicht aus Sicht einer Systemanalyse nutzbar. Taxonomien bzgl. der Schwachstellen in Netzen können bspw. aus Sicht der Sicherheitsbedrohungen, wie bei Jayaram und Morse, erfolgen. Hier wird zwischen *physikalischen Bedrohungen*, *Systemschwachstellen*, *Schadsoftware*[14], *Zugriffsrechte* und *Bedrohungen bzgl. der Kommunikation* unterschieden. Auch hier können Bedrohungen nicht eindeutig einer Klasse zugewiesen werden.

[13] Maskierungsangriffe werden auch als Rechteausweitung (Privilege Escalation) bezeichnet.

[14] In der Aufzählung in [206] wurde dieser Punkt fehlerhaft als *Malign problems* aufgeführt, die Quelle [219] spricht hier jedoch von *Malign Programs*.

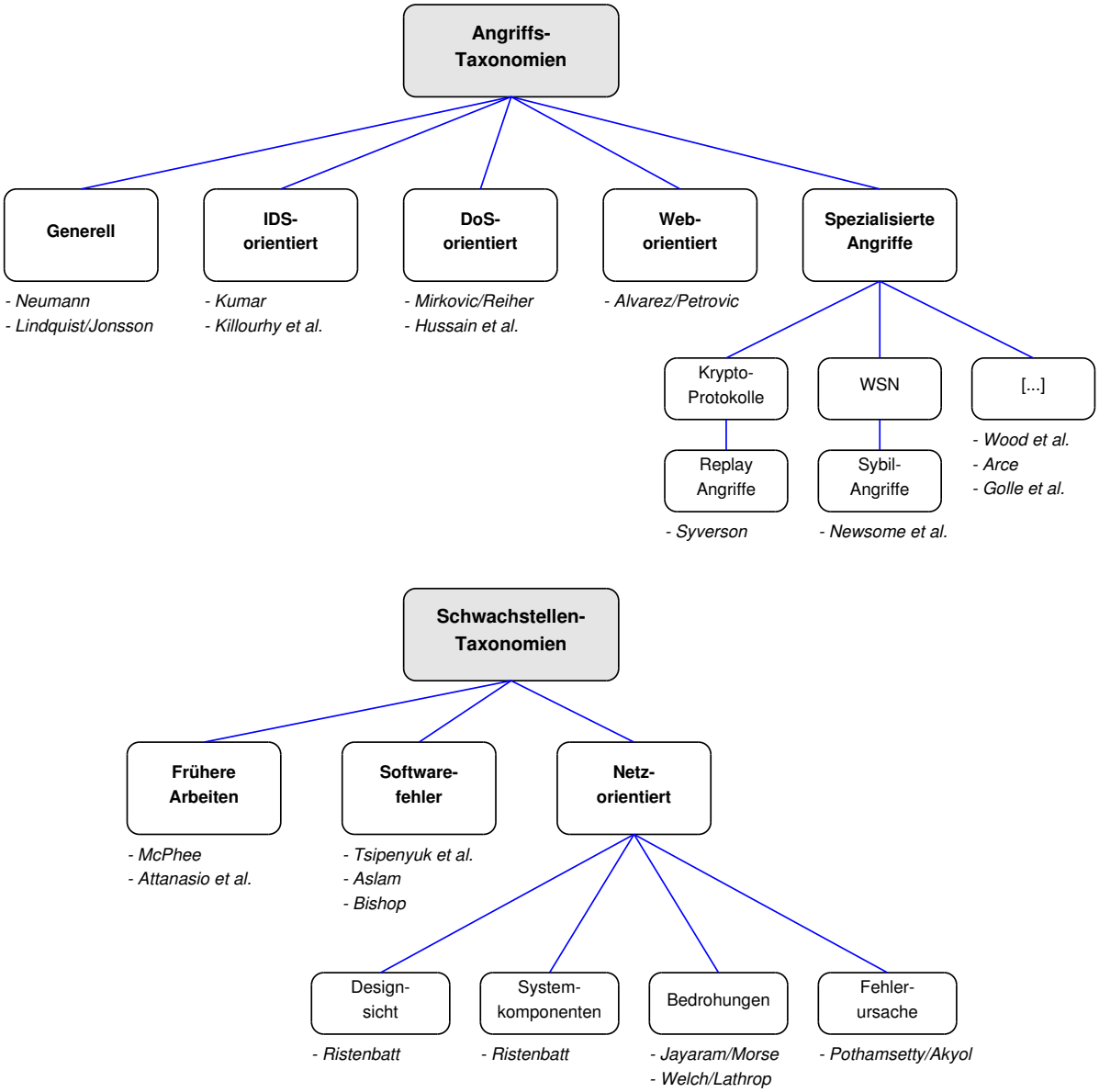


Abbildung 2.5: Aufstellung von Taxonomien nach Angriffs- und Schwachstellenbetrachtung, Zusammenstellung nach Igure [206].

Tool	Schwach-stelle	Aktion	Ziel	Nicht autorisiertes Ergebnis
Physikal. Angriff	Design	Scan	Nutzerkonto	Erhöhung Rechte
Skript / Programm	Implemen-tierung	Flood	Prozess	Informations-diebstahl
Toolkit	Konfiguration	Modify	Daten	DoS
...		...	...	...

Abbildung 2.6: Rechner und Netz-Angriffsmatrix von Howard und Longstaff [200]. Demnach bestehen Angriffe aus fünf abhängigen Teilschritten, die ein Angreifer absolvieren muss.

Eine weitere Taxonomie für die Einteilung von Sicherheitsbedrohungen in Drahtlosnetzen stammt von Welch und Lathrop. In der Arbeit werden sieben Angriffsgruppen definiert, wobei jedoch Bedrohungen hinsichtlich der Systemverfügbarkeit nicht aufgeführt werden; die einzelnen Klassen sind: *Verkehrsanalyse*, *Passives Abhören*, *Aktives Abhören*, *Nicht-autorisierter Zugang*, *Man-in-the-Middle (MITM)*, *Session Hijacking* und *Replay*-Angriffe. Die Anforderung der Eindeutigkeit wird auch in dieser Klassifizierung nicht eingehalten.

Shuyuan et al. geben einen weiteren Überblick über 25 verschiedene Klassifizierungsmöglichkeiten von Schwachstellen in Netzen [220]. Hierbei vergleichen sie die verschiedenen Klassifizierungen nach den genutzten Attributen, wie bspw. Ursprung, schadensorientierte Einteilung oder C/C++ Programmfehler. Weiterhin werden die Dimensionen der ausgewerteten Klassifizierungen angegeben. Während der größte Teil der von Shuyuan analysierten Arbeiten lediglich eine Dimension nutzt (14 der 25 untersuchten Schemata), nutzen einige wenige auch drei, vier oder mehr Dimensionen.

Als dritter Bereich wurde die Ausrichtung der jeweiligen Klassifizierungen angegeben, wobei hier die Kategorien *Allgemein* sowie *Betriebssystem-*, *WLAN-* und *angriffsorientiert* genutzt werden.

Howard und Longstaff haben in ihren Arbeiten eine Matrix möglicher Attacken entwickelt, welche zu logisch aufeinander aufbauenden Aktivitäten jeweilige Verfahren, Methoden und Schwachstellen zuordnen (vgl. Abbildung 2.6) [200].

Howard führt hier insbesondere eine Angriffs-Definition im Sinne der durchzuführenden Stufen ein:

Definition (Angriff *i.S.v. Angriffsschritten*)**.** *Eine von einem Angreifer durchgeführte Serie von Schritten, um ein nicht autorisiertes Ergebnis zu erreichen (nach [200]).*

Das eigentliche Ereignis besteht aus den Stufen *Aktion* und *Ziel*, wobei eine Aktion durch eine mittels eines Tools ausgenutzten Schwachstelle ausgelöst wird und letztendlich

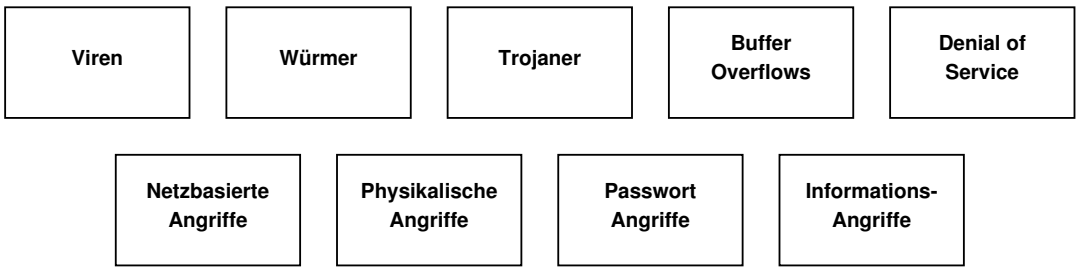


Abbildung 2.7: Angriffskategorien erster Dimension nach der Taxonomie von Hansman et al. [189].

zu einem nicht-autorisierten Ergebnis auf dem Ziel (-System) führt.

Eine detaillierte, mehrdimensionale Klassifikation erfolgt durch Hansman und Hunt, wobei nur die erste Dimension im Sinne einer Angriffs-Spezifikation relevant ist [189]. Abbildung 2.7 zeigt das oberste von drei Leveln der ersten Dimension. Die weiteren Level spezifizieren die jeweiligen Kategorien in einer feineren Detaillierung, wo dies möglich ist: Bspw. wird die Kategorie *Buffer Overflows* weiter unterteilt in Stack- und Heap-basierte Überläufe, Viren werden in reguläre, Datei-infizierende, Bootrecord-Viren und Macro-Viren unterteilt. Die tieferen Level können in der Betrachtung jedoch unberücksichtigt bleiben, da zunächst nur eine Aufstellung von Angriffskategorien erfolgen soll. Hierbei muss erwähnt werden, dass die Kategorien eins bis drei der ersten Dimension der Hansman-Klassifikation, d.h. Viren, Würmer und Trojaner, heute typischerweise unter Schadsoftware zusammengefasst werden[15].

Eine etwas abweichende Klassifizierung einer Internet Angriffstaxonomie erfolgte durch Atlantic Consulting Services im Auftrag der US Army (vgl. Abbildung 2.8) [118]. Auffällig ist hier, dass insbesondere die Ausnutzung von Schwachstellen unter die Klasse DoS subsumiert wurde, was üblicherweise extra geführt wird. Dies lässt sich bei weiterer Betrachtung der tieferen Ebenen erklären, da diese immer in der nicht-Verfügbarkeit von Systemen oder dem Absturz von Programmen enden und somit einer klassischen Definition der nicht-Verfügbarkeit von Diensten entsprechen. Da diese Subsumierung einerseits nicht verbreitet ist, andererseits aber insbesondere der heutigen Bedeutung der Ausnutzung von Schwachstellen i.S. der damit möglichen Aktionen im Erfolgsfalle nicht gerecht wird, wird diese Untergliederung im weiteren Verlauf nicht aufgegriffen.

Bei der Entwicklung des IDS *Haystack* wurde durch Smaha ebenfalls eine Klassifizierung der möglichen Angriffsarten vorgenommen [351], die auch heute noch verbreitet ist (vgl. z.B. [226]):

- Einbruchsversuche (attempted break-ins): Ein erfolgreicher Einbruchsversuch liegt vor, wenn ein nicht-autorisierter Nutzer mittels einer validen User-Id / Passwortkombination Zugang zum System erhält.

[15] Während die Tabelle 1 in [189] *Trojaner* nicht explizit aufführt, erfolgt dies im beschreibenden Text durchaus als extra Strichpunkt. Es kann daher davon ausgegangen werden, dass die Klasse *Trojaner* in der zusammenfassenden Tabelle lediglich vergessen wurde.

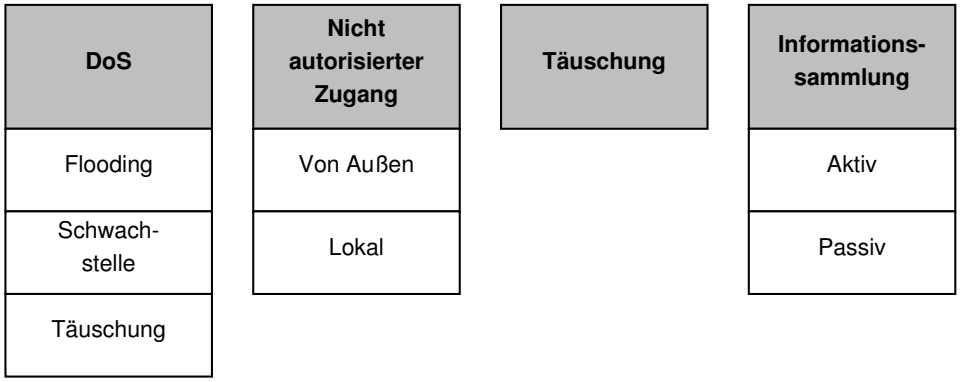


Abbildung 2.8: Internet Angriffs-Taxonomie nach Atlantic Consulting Services.

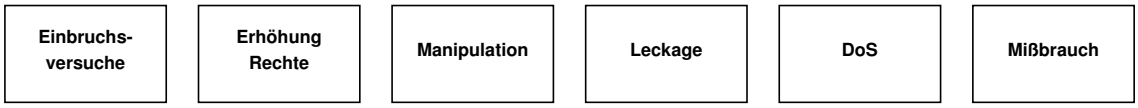


Abbildung 2.9: Einteilung der Einbruchsarten nach Smaha [351].

- Maskierungsangriff (masquerade attack): Diese auch als *privilege escalation* bezeichnete Angriffsart versucht, die zur Verfügung stehenden Privilegien zum Beispiel durch die Ausführung von Schadcode zu erhöhen (Erlangen administrativer Rechte oder höherer Nutzerrechte).
- Penetration durch Manipulation des Sicherheitssystems: Der Nutzer versucht durch Modifizierung von sicherheitskritischen Applikationen und Systemkomponenten Einfluss auf das System zu nehmen, zum Beispiel durch die Änderung von Passwortdateien.
- Leckage: Versuche, Daten aus einem System zu extrahieren, zum Beispiel durch das Ausdrucken einer Vielzahl von Dateien.
- Denial of Service: Verhinderung der regulären Nutzung von Ressourcen durch Überlastung.
- Böswillige Nutzung: Zum Beispiel Löschen von Systemdateien, etc.

Die Gemeinsamkeiten und Unterschiede der verschiedenen Modelle werden in der Tabelle 2.4 nochmals zusammengefasst. Gut zu erkennen ist die große Breite der Taxonomien; die meisten Klassifizierungen lassen sich in einen der drei Teilbereiche *Ursache*, *Durchführung / Ausgenutzte Schwachstelle* bzw. *Auswirkung* eingliedern.

Howard führt in seiner Dissertation die Taxonomien für Computer- und Netzangriffe zur Ablaufsequenz gem. Abbildung 2.10 zusammen. Bei dieser Einteilung wird die Gewichtung auf die einzelnen Teilschritte, die aufeinander aufbauen und zwingend erfüllt werden müssen, um eine Komprimittierung durchzuführen, gelegt. Es handelt sich

Tabelle 2.4: Angriffskategorien verschiedener Taxonomien. Handelt es sich um mehrdimensionale oder hierarchische Klassifizierungen, ist nur die jeweils den Angriffen entsprechende Gruppe ausgewertet.

		Perry, Wallich	Brinkley, Schell	Neumann	Lindquist, Johnson	Jayaram, Morse	Welch, Lathrop	Pothamsetty, Akyol	Hansman, Hunt	DeLooze	Atlantic Consulting
Ursache											
Extern				✓							
Missbrauch	Aktiv			✓	✓						
	Passiv			✓	✓						
	Inaktiv			✓							
	Indirekt			✓							
	Hardware			✓							
Seitenkanal				✓							
Schadsoftware				✓		✓					
	Viren								✓		
	Würmer								✓		
	Trojaner								✓		
Maskierung				✓		✓				✓	
Durchführung / Ausgenutzte Schwachstelle											
Menschliches Versagen			✓								
Missbrauch von Rechten			✓			✓	✓				✓
Zerstörung / Umgehung von Sicherheitsmechanismen			✓		✓						
Direktes Abhören			✓						✓	✓	✓
	Aktiv						✓				
	Passiv						✓				
Abhören durch Schadsoftware			✓								
Direkte Penetration / Systemschwachstellen			✓			✓					
	Buffer Overflows										
	DoS									✓	✓
	Passwort- Angriffe								✓		
Physikalisches Medium						✓			✓		
Kommunikationsbasierte Schwachstellen						✓			✓		
	Daten- Analyse						✓				
	MITM						✓				
	Session Hijacking						✓				
	Replay- Angriffe						✓	✓			
	Klartext- Kommunikation							✓			
	Protokoll- Nachrichten							✓			
	Protokoll- Stadien							✓			
	Protokoll- Authentisierung							✓			
	Entropie							✓			
Täuschung										✓	✓
Auswirkung											
Physikalische Zerstörung		✓									
Zerstörung von Informationen		✓									
Datenveränderung		✓	✓		✓						
Datendiebstahl		✓	✓		✓						
Diebstahl von Diensten / Ressourcen		✓	✓								
Nicht-Verfügbarkeit von Ressourcen			✓		✓						

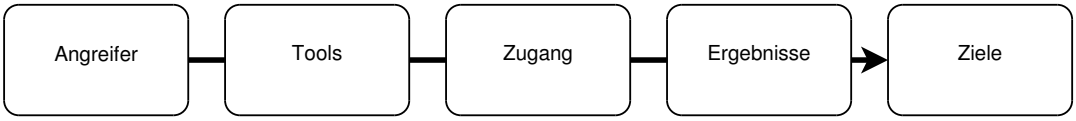


Abbildung 2.10: Ablaufsequenz von Computer- und Netzangriffen nach Howard [199].

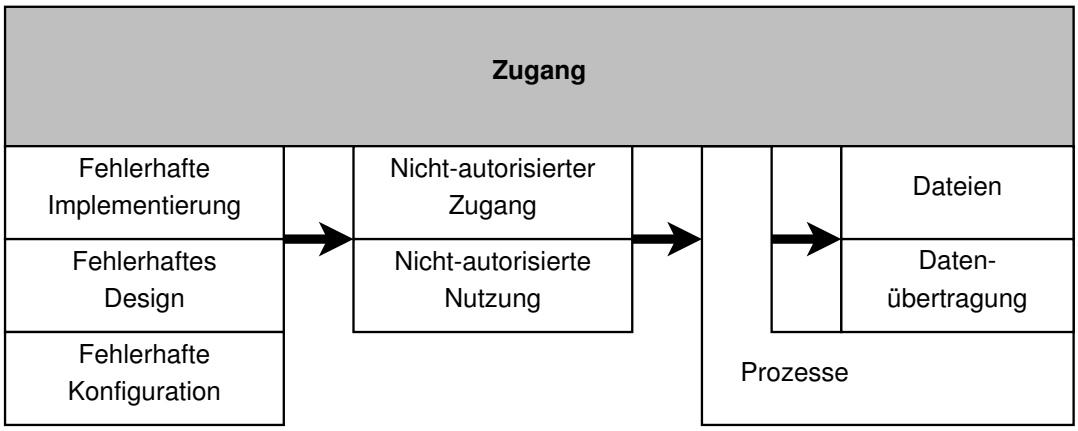


Abbildung 2.11: Erlangung des Zugriffs für einen Angriff nach Howard [199].

also um eine Gesamtbetrachtung eines Sicherheitsvorfalles; der Teilschritt *Zugang* ist widerrum im Sinne des Dreiklangs *Ursache*, *Durchführung* und *Auswirkung* weiter differenziert (vgl. Abbildung 2.11). Unter *Angreifer* zählt Howard sechs Kategorien von Tätern auf: Hacker, Spione, Terroristen, professionelle Kriminelle, Vandalen sowie die Kategorie der sog. Corporate Raiders. Unter Letzteren versteht Howard Angestellte eines Unternehmens, die aus finanziellen Absichten in Computersysteme eines Konkurrenten eindringen; im Gegensatz hierzu liegt bei professionellen Kriminellen eine private Gewinnabsicht zugrunde. Die zu den Ergebnissen gehörenden Klassen sind im einzelnen: *Verfälschung von Informationen*, *Verlust von Informationen*, *Diebstahl von Diensten* sowie *Dienstverweigerung (DoS)*. Unter den eigentlichen Zielen versteht Howard die abstrakteren Klassen Herausforderung und Status, politische Einflussnahme, finanzieller Gewinn und Beschädigung.

Für die Evaluation der zutreffenden Angriffskategorien muss der Schwerpunkt im Rahmen des vorliegenden Szenarios (vgl. Kapitel 2.1) auf Taxonomien der Gruppe *Durchführung / Ausgenutzte Schwachstelle* gelegt werden. Da ein Sicherheitssystem in eine bestehende Netz-Struktur integriert werden soll, sind Klassifizierungen welche maßgeblich auf Designfehler von Protokollen, Diensten, etc. abzielen, nicht von Bedeutung: Diese müssen in der Design- und Analysephase im Softwareentwicklungsprozess angewendet werden, um mögliche Angriffsstellen frühzeitig zu erkennen und zu vermeiden.

Nicht explizit aufgeführt sind jedoch insbesondere die Bereiche *Social Engineering* und *Targeted Attacks*, deren Bedeutung in jüngerer Zeit stetig wächst.

Entwicklung der Angriffsfelder Im Laufe der Jahre hat sich der Fokus der Angreifer und die Funktionalität der Schadsoftware stark gewandelt. Während die Theorie sich selbst reproduzierender Automaten von John von Neumann bereits in das Jahr 1949 zurückreicht, wurde Anfang der 70er Jahre das Spiel *Core Wars* von Mitarbeitern der Bell Laboratories entwickelt, bei dem es darum ging, seinen Gegnern Rechenzeit zu stehlen. Das Prinzip kam dem der späteren Würmer bereits nahe, wobei hier der Programmierer noch manuell für die Verbreitung des Codes sorgen musste.

1980 wurde von Jürgen Kraus eine Diplomarbeit über sich selbst reproduzierende Programme an der Universität Dortmund geschrieben, die allerdings unveröffentlicht in die Archive ging [327]. Der in der heutigen Form bekannte Begriff des Computervirus wurde 1981 in einem Gespräch zwischen Professor Adleman mit seinem Doktoranden Fred Cohen geprägt.

1982 wurden im Xerox Palo Alto Research Center die ersten Würmer programmiert, welche für verteilte Berechnungen genutzt werden sollten. Aufgrund eines Programmierfehlers kam es hierbei zu einer unkontrollierten Ausbreitung, welche zu einem Zusammenbruch des gesamten Systems aufgrund der hohen Systemlast führte [270].

1983 entwickelte Fred Cohen den ersten funktionsfähigen Virus im Rahmen seiner Doktorarbeit, die 1984 veröffentlicht wurde. Von diesem Zeitpunkt an begann eine rasche Entwicklung immer neuer Schadsoftware. Bspw. wurde beim *Jerusalem*-Virus (1987) versucht, Daten von der Festplatte zu löschen: Dieser ist nach seinem Entdeckungsort benannt und auch unter weiteren Namen bekannt (A-204, 1808(EXE), 1813(COM), ArabStar, BlackBox, BlackWindow, Friday13th, HebrewUniversity, Israeli, PLO, Russian). Der Virus ist konstruiert, Programme zu löschen, die an einem Freitag den 13ten ausgeführt werden (COM und EXE- Dateien) [244].

Kurz nach den ersten Viren erschienen auch die ersten Würmer. Im Gegensatz zu Viren, die sich an eine Datei anhängen und sich nicht selbstständig verbreiten können, kopieren sich Würmer selbst und bewegen sich somit hauptsächlich über das Internet weiter. Diese Automatisierung wird insbesondere zum Aufbau großer Bot-Netze genutzt, die aus vielen Tausend infizierten Rechnern bestehen können, typischerweise ohne das Wissen der Besitzer, dass ihr Rechner einem entsprechenden Netz angehört.

Häufiger Anwendungszweck dieser Bot-Netze ist das Versenden von Spam, die Durchführung von Distributed Denial of Service (DDoS)-Angriffen oder die Verschleierung der Ausgangsadresse für einen Angriff. [144] gibt zahlreiche Statistiken über den aktuellen Status von Bot-Netz-Aktivitäten und DDoS-Angriffen.

Gerade im Bereich der Spam-Mails sieht man den Wandel von destruktiven hin zu kommerziellen Interessen hinter der Nutzung von Schadprogrammen. Während zu Beginn eine wahllose Zerstörung von Daten bzw. Beeinträchtigung eines Systems im Vordergrund stand, hat sich in den letzten Jahren ein überraschend spezialisierter Untergrundmarkt im Internet aufgebaut [52].

Heutzutage können Spammer in sog. *Affiliate-Programmen* teilnehmen: Tritt ein Spammer einem entsprechenden Programm bei, bekommt er eine eindeutige ID zugewiesen, die er im Kontext der Versendung der Werbemails einsetzt. Wird aufgrund einer Spam-Mail ein Verkauf erzielt, bekommt die zugehörige ID eine entsprechende Vergütung. Von besonderer Bedeutung ist hier Spam der *Canadian Pharmacy*, der 74 Prozent des

Spamaufkommens im Jahre 2010 bestimmt hat [142].

Dabei ist die Durchführung in zahlreiche spezialisierte Aufgabengebiete eingeteilt (vgl. Anhang F.1.2). In den Prozess des Spammings sind heutzutage Botnetzbetreiber, Malwareprogrammierer, etc. involviert. Die vorliegenden Abhängigkeiten zeigen ein weiteres, wichtiges Phänomen der jüngeren Zeit: Immer mehr Spam-Mails beinhalten auch Schadsoftware, insbesondere Trojaner. Diese werden maßgeblich eingesetzt, um Identitäten, Bank- und Kreditkarteninformationen, etc. zu stehlen. Mittels dieser Informationen werden im Web Waren erworben, die widerrum weiterverkauft werden. Der kommerzielle Erfolg des Internets sowie die Möglichkeiten, aus großer Entfernung relativ risikolose Angriffe durchführen zu können, haben die Kriminalität im Internet in den letzten Jahren zu einem milliardenschweren Markt anwachsen lassen (vgl. z.B. [263], [141]). Auf den Untergrund-Servern werden Daten wie Kreditkarteninformationen, Details zu Bankkonten, Angriffstoolkits, Identitäten von Bürgern, etc. gehandelt. Dies stimuliert auch die Entwicklung von Angriffswerkzeugen: Mit Hilfe von Angriffs-Toolkits kann auch ohne eine entsprechende Fachkenntnis effektiver und gefährlicher Angriffsprogrammcode erzeugt werden. Während das erste Toolkit aus dem Jahre 1992, das *Virus Creation Lab*, lediglich Grundfunktionalitäten zur Verfügung stellte, können mittels State-of-the-Art Kits wie *MPack* oder *Nukesploit* oder Command-and-Control Toolkits wie *Spy Eye* oder *Zeus 2.0* gefährliche Schadprogramme erzeugt und Bot-Netze einfach gesteuert werden. Die Preise solcher Toolkits erreichen hierbei mehrere tausend Euro, neuerdings werden auch SLAs angeboten, bspw. mit Update-Support oder einer Garantie der Nicht-Entdeckung des Toolkits durch aktuelle Virenscanner (vgl. [139]).

Durch die einfache Nutzung der Kits und die Erzeugung gefährlicher Schadprogrammen steigt die Anzahl der entsprechenden Signaturen stark an; zwar können mit Hilfe heurisitscher Verfahren Schadcode-Familien identifiziert werden, ohne für jede Variation eine eigene Signatur vorhalten zu müssen, dies steigert aber widerrum die Anzahl der Fehlalarme und irrtümlichen Detektionen.

Auf der anderen Seite nimmt die Anzahl zielgerichteter Angriffe immer mehr zu. Ein bekanntes Beispiel der jüngeren Zeit ist der Trojaner *Hydraq* (auch bekannt als *Aurora*), von dem mehrere große Unternehmen betroffen waren (vgl. [137]). Die Angriffe wurden initiiert, indem öffentlich verfügbare Informationen der Mitarbeiter der Firmen ausgewertet wurden, die auf den Web-Präsentationen der Firmen und insbesondere auf den von den identifizierten Mitarbeitern genutzten Seiten Sozialer Netzwerke gesammelt wurden.

Soziale Netzwerke wie Facebook oder Twitter stehen zunehmend im Fokus von Angreifern, da viele Nutzer dort sehr leichtfertig mit privaten Daten umgehen ([173], vgl. auch [48]). Diese Informationen können genutzt werden, um zielgerichtet einzelne Personen anzugreifen: Anhand aktueller und für die Zielperson relevanter Themen wird unter Ausnutzung der Mail-Adresse eines persönlichen Bekannten eine E-Mail versandt, deren Anhang eine Schadsoftware beinhaltet. Durch die Relevanz und den bekannten Absender ist die Zielperson eher geneigt, den Anhang zu öffnen. Diese Vorgehensweise ist in das Gebiet des *Social Engineering* zu subsumieren, das seinen vorläufigen Höhepunkt in den 80er und 90er Jahren hatte und hier zu einer neuen Blüte kommt.

Datenverlust gehört ebenfalls zu den Problemen zunehmender Bedeutung; hierzu ist

nicht nur der durch Insider initiierte Datenabfluss zu zählen, sondern auch der versehentliche oder ungewollte Verlust von Daten (vgl. [272]); in jüngerer Zeit haben zahlreiche Datenskandale wiederholt das Interesse der Öffentlichkeit erweckt (vgl. z.B. [350], [40] oder [378]).

Neben dem ungewollten Datenverlust bekommt gerade in wirtschaftlich schweren Zeiten der Bedeutung des Innentäters ein hohes Gewicht zu, wie durch die Analyse in Kapitel 2.2.2 gezeigt. Anhand der Auswertung aktueller Studien als auch der Bewertung der kontroversen Diskussionen bzgl. der Bedeutung des Innentäters wurde gezeigt, dass der hiervon ausgehenden Gefährdung ebenfalls Rechnung getragen werden muss. Aus diesem Grunde müssen die besonderen Angriffswege, die einem Insider durch die ihm übertragenen Zugangs- und Zugriffsmöglichkeiten zur Verfügung stehen, mit in das Schema der Angriffsdurchführung aufgenommen werden.

Eine weitere starke Tendenz der letzten Jahre ist die zunehmende Verlagerung der Angriffe auf Schwachstellen von Nutzerprogrammen anstatt wie bisher, auf Fehler des Betriebssystems. Die kontinuierliche und schnelle Weiterentwicklung von Anwendersoftware, insbesondere Browsern und deren Funktionalitäten, eröffnet durch die hierdurch entstehende Komplexität der Software immer neue Angriffsmöglichkeiten. Mittlerweile richten sich bereits über 70 Prozent aller Angriffe gegen den Application-Layer [141]. Vielgenutzte Programme und deren Browser-Plugins wie der Acrobat-Reader von Adobe haben hier wiederholt zahlreiche ausnutzbare Schwachstellen angeboten (vgl. z.B. [193], [195]). Durch die Komplexität der Software hat auch die Anzahl der sog. Zero-Day Schwachstellen einen Anstieg in den letzten Jahren erfahren.

Gleichzeitig muss jedoch berücksichtigt werden, dass mehrere wichtige Softwareanbieter Patches unnötig verzögern, indem diese die entsprechenden Veröffentlichung auf der Basis von regelmäßigen *Patchdays* durchführen (vgl. z.B. [194], [98]), anstelle so früh wie möglich Updates bereitzustellen.

Die Entwicklung der Angriffsfelder lässt sich somit auf nachfolgend aufgeführte Schwerpunkte zusammenfassen:

1. Zunahme von zielgerichteten Angriffen, in diesem Kontext tritt eine intensive Nutzung von *Social Engineering*-Techniken auf.

2. Verlagerung der Angriffe weg von Betriebssystem- hin zu Applikationsschwachstellen.

3. Hohe Gefahr durch Datenverlust, sowohl ungewollt (menschliches, technisches Versagen, etc.), als auch gewollt (Insider).

2.3.3 Angriffsschritte

Um den aktuellen Entwicklungen Rechnung zu tragen, müssen die Schwerpunkte *zielgerichtete Angriffe* und *Innentäter* mit in die Betrachtung aufgenommen werden. Während Angriffe der Kategorie Innentäter unter die Kategorie *Missbrauch von Rechten* der jeweiligen Taxonomien subsumiert werden können, können auf *Social Engineering* basierende Angriffe zu der Kategorie *Täuschung* gerechnet werden (vgl. Tabelle 2.4). Da die

Nutzung solcher Komponenten jedoch erhebliche Auswirkung auf den Angriffsweg und somit die daraus resultierenden Möglichkeiten einer Detektion hat, wird das Modell der Angriffsstufen wie unter Abbildung 2.12 gezeigt um die rot eingezeichneten Elemente erweitert.

Insbesondere fehlt in der originären Klassifizierung nach Howard in der Unterteilung der Angreifer die Kategorie Innentäter. Durch seine besondere Stellung hat dieser jedoch bereits einen zumindest eingeschränkten Systemzugriff mit Benutzerrechten, so dass ein höherer Einstiegspunkt im Modell vorliegt: Einerseits kann der Innentäter den legitimen Zugang nutzen, um mittels entsprechender Angriffe höhere Rechte zu erlangen oder, falls der Zugriff auf die benötigten Daten durch seine Stellung bereits autorisiert ist, können diese auch direkt kopiert und ausgeschleust werden.

Die immer wichtigere Kategorie der zielgerichteten Angriffe setzt insbesondere auch Social Engineering-Techniken ein, um ebenfalls einen höheren Einstiegspunkt in den Angriffsstufen zu erreichen. Werden diese Techniken im klassichen Sinne betrachtet, versucht der Angreifer mittels geschickter Manipulation sein Ziel zur Herausgabe von Daten zu überzeugen; somit können die eigentlichen Stufen der Systemmanipulation ausbleiben, der Datenzugriff findet durch eine legitimierte Person statt und es verbleibt lediglich der Aspekt der Datenausschleusung (*aktives* Social Engineering gem. [92]).

Mittlerweile wird eine adaptierte Variante des Social Engineerings eingesetzt, indem das Opfer durch personalisierte und im Kontext seines aktuellen Alltags stehende Mails zum Öffnen eines Anhangs von einer ihm vermeindlich bekannten Person gebracht wird. Mittels diesem wird hierbei eine Schadsoftware installiert (*passives* Social Engineering gem. [92]). Hiermit erfolgt ein Sprung zum Einstiegspunkt *Erlangen administrativer Rechte* oder höher. In beiden Fällen wird hier das Ziel einer Social Engineering-Attacke ein *unwissender* Innentäter.

Im Folgenden werden die einzelnen Schritte sowie die damit verbundenen Maßnahmen beschrieben, um anschließend die Detektionsmöglichkeiten auf den jeweiligen Stufen zu analysieren.

Bei den identifizierten Angriffs-Stufen handelt es sich im Einzelnen um:

1. Analyse der Zielumgebung / Angriffsziele
2. Identifizieren von Schwachstellen, Auswählen von Exploits
3. Erlangen eines Fernzugriffs (*Remote-to-Local, R2L*)
4. Erlangen administrativer Rechte (*User-to-Root, U2R*)
5. Manipulation der Systemumgebung, Installation eines *Backdoors*
6. Löschen von Angriffsspuren

Diese müssen in einer logischen Reihenfolge absolviert werden, jedoch wird durch die Position des Angreifers der Einstiegspunkt maßgeblich beeinflusst. Im Sinne eines Innentäters als Angreifer resultieren daher insbesondere die Stufen

- Datenzugriff und Kopieren sowie

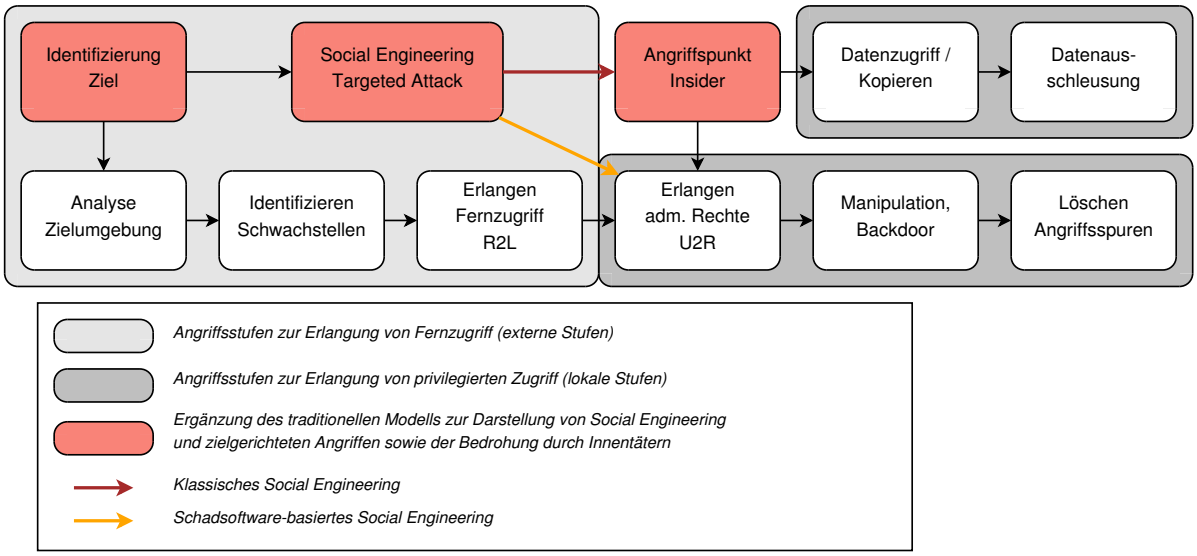


Abbildung 2.12: Ablauf der verschiedenen Angriffsstufen zur Kompromittierung eines Systems. Abzugrenzen sind die Schritte, welche extern durchgeführt werden müssen, um Fernzugriff zu einem System zu erlangen, von den Schritten, welche lokal auf dem kompromittierten System ausgeführt werden, um administrative Rechte zu erhalten. Die traditionellen Stufen müssen im Sinne der heutigen Angriffsvektoren *Social Engineering* und *Targeted Attacks* erweitert werden, da diese maßgeblich Einfluss auf die Angriffsschritte haben: Hierbei können die bisher extern durchzuführenden Schritte entfallen, da sie in einer (im Sinne eines IDS) nicht-detektierbaren Weise durchgeführt werden und somit erst in höheren Schritten oder gar nicht erkannt werden können.

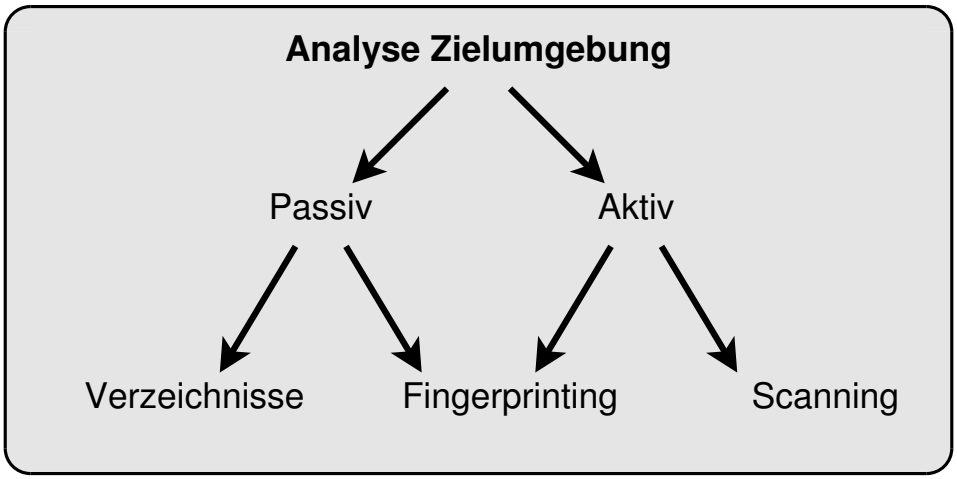


Abbildung 2.13: Angriffs-Teilschritt *Analyse der Zielumgebung* und die zugehörigen Maßnahmen.

- Datenausschleusung.

Analyse der Zielumgebung Die Erfordernisse, um die Zielumgebung insbesondere auf das Vorhandensein evtl. Schwachstellen zu analysieren, hängt naturgemäß maßgeblich von der Beobachtungsposition des Angreifers ab: Befindet sich der Angreifer außerhalb des eigentlichen Zielnetzes, sind entsprechend aufwändigere Maßnahmen zum Sammeln der erforderlichen Informationen notwendig, als wenn sich der Angreifer insbesondere in der Form des Innentäters bereits in der Zielumgebung befindet und ggf. schon mit Zugangsrechten einer bestimmten Stufe ausgestattet ist. Dies beinhaltet auch den elementaren Aspekt, ob physikalische Zugriffsmöglichkeiten auf das Zielnetz vorhanden sind, da in diesem Fall umfangreiche Einflussnahmen auf die IT-Umgebung inklusive derer Sicherheitssysteme möglich sind (vgl. auch Kapitel 4.6.1).

Befindet sich der Angreifer außerhalb des Firmennetzes, muss ein Zugriff regelmäßig über eine Firewall hinweg erfolgen, die heutzutage zu den Sicherheitsstandards fast jeden Netzes gehört (vgl. hierzu auch die gesetzlichen Anforderungen im Rahmen der Verkehrssicherungspflichten, Kapitel 4.6.5). Dies erfordert naturgemäß ein angepaßtes Vorgehen im Vergleich zu einem direkten Zugriff auf das Netz.

Um festzustellen, welche Systeme und Komponenten in einer Umgebung verfügbar sind, können verschiedene passive und aktive Maßnahmen ergriffen werden (vgl. Abbildung 2.13). Eine rein passive Informationssammlung hat den Vorteil, dass sie regelmäßig nicht zu entdecken ist; allerdings ist bei der Analyse eines Netzes von außen oder auch einer „geswitchten“ Umgebung der Zugang zu zentralen Komponenten erforderlich, um eine detaillierte Analyse der Netz-Infrastruktur zu erhalten.

Muss eine Analyse von außerhalb der Zielumgebung erfolgen, kann diese z.B. anhand der Webpräsentation des Ziels begonnen werden. Mittels einer Datenbankabfrage des Domänennamen, welche an die Domain Name System (DNS)-Rootserver gerichtet wird

und von diesen weiter zum zuständigen DNS-Server gegeben wird, erhält man die IP-Adressen des zuständigen Nameservers und des angefragten Webservers sowie dessen kanonischen Namen (CNAME Resource Record, vgl. Anhang F.1.4).

Auf diesem Ergebnis basierend können widerrum weitergehende Informationen mittels des DNS-Hilfsprogrammes `host` bezogen werden (vgl. Anhang F.1.4). Somit ist auch die Adresse des für die Domäne zuständigen Mailservers bekannt, mittels derer weitere Informationen über das Zielnetz gewonnen werden können. Der `host`-Befehl kann zusätzliche Informationen von den entsprechenden Nameservern abrufen. Somit lässt sich bspw. eine Auflistung sämtlicher registrierter Einträge der Domäne anzeigen. Da die für die Server gewählten Namen oft sprechend sind, lassen sich aus den gewonnenen Daten wertvolle Informationen wie bspw. Dienste, die auf bestimmten Rechnern laufen, ableiten.

Ebenfalls stehen die Einträge der Registrierungsinstanzen der jeweiligen Länder öffentlich zur Verfügung. Um die Stabilität des gesamten Netzes jederzeit gewährleisten zu können, sind bei der Vergabe von IP-Adressen bzw. Adressblöcken entsprechende Kontaktinformationen des Nutzers anzugeben (vgl. Anhang F.1.4). Diese sind relativ ausführlich und geben einen guten Startpunkt, um bspw. mittels Techniken des Social Engineerings weitere Informationen über die festgestellten Administratoren zu gewinnen.

Neben diesen Informationen bzgl. des identifizierten Webservers können ebenfalls weitere Details von der Registrierungsbehörde abgerufen werden, wie bspw. der vergebene Netzbereich, die verantwortliche, nationale Stelle und Kontaktinformationen der Administratoren des Zielnetzes.

Weitere Details über die Infrastruktur der Zielumgebung lassen sich gewinnen, indem bspw. ein `traceroute` zu einem Server durchgeführt wird (vgl. Anhang F.1.4). Hierbei wird der Weg der IP-Pakete verfolgt, indem vom Zielsystem die Verfügbarkeit mittels einer Internet Control Message Protocol (ICMP)-Type 8 (echo request)[16] Nachricht angefordert wird. Hierbei wird das Time-to-Live (TTL)-Feld der abgesendeten Pakete mit 1 beginnend mit jeder neuen Sendung um 1 erhöht. Jeder Router auf dem Weg zum Zielsystem dekrementiert den TTL-Wert um 1, ergibt sich hierdurch ein Wert von 0 wird eine ICMP-Antwort vom Typ 11 (time-to-live exceeded) erzeugt und das ursprüngliche Paket verworfen. Da diese Antwort auch die IP-Adresse des generierenden Routers enthält, kann auf diese Weise der Weg zum Zielsystem verfolgt werden, falls die entsprechenden Pakete unterwegs nicht durch bspw. eine Firewall verworfen werden oder die ICMP-Antwort deaktiviert wurde.

Auch die Kenntnis der Adresse des Mailservers der Domäne kann herangezogen werden, um wertvolle Informationen zu gewinnen. Sendet man an die Zieldomäne eine Mail mit dort unbekanntem Empfänger, erhält man typischerweise eine Nachricht über den erfolglosen Zustellversuch, welche ebenfalls zahlreiche Informationen über das Zielsystem beinhaltet (vgl. Anhang F.1.4). Hier lassen sich nicht nur die Arten und Versionen der eingesetzten Mailprogramme auslesen, sondern auch auf dem System laufende Dienste wie Spam-Filter und Virenscanner. Hiermit lässt sich schnell prüfen, ob ausnutzba-

[16] Unixsysteme versenden typischerweise User Datagram Protocol (UDP)-Nachrichten für die Abfrage, Windows-Tools ICMP, weiterhin existieren Implementierungen, welche Transmission Control Protocol (TCP)-Pakete verwenden.

re Schwachstellen in der eingesetzten Software vorhanden sind. Auch Hilfsprogramme wie `telnet` oder `finger` können genutzt werden, um bspw. Informationen von Server-Bannern oder laufenden Programmversionen zu gewinnen.

Um auch Systeme aufzufinden, die nicht in den DNS-Verzeichnissen eingetragen sind, kann das Zielnetz mittels eines Scanners untersucht werden. Dieser versucht, Verbindung zu den jeweiligen Ports eines Zielsystems aufzunehmen. Da ein System nicht notwendigerweise auf die Überprüfung des Status durch eine ICMP-Anfrage reagieren muss, kann der Scanner versuchen, entsprechenden Kontakt zu beliebigen IP/Port-Adresskombinationen auch ohne das Wissen, ob sich ein Rechner hinter der Adresse befindet bzw. erreichbar ist, aufzubauen [17]. Scanner können in zwei Gruppen eingeteilt werden, abhängig ob ihre Arbeitsweise aktiver und passiver Art ist.

Bei einer passiven Analyse wird der Datenverkehr lediglich abgehört und ausgewertet. Dies bedingt, dass diese Art von Analyse nur an Stellen eingesetzt werden kann, an denen der Datenverkehr des Zielnetzes zumindest partiell mitgeschnitten werden kann. Durch die rein passive Arbeitsweise werden hierbei keine zusätzlichen oder insbesondere auch ungewöhnliche Pakete, zum Beispiel mit besonderen Flag-Kombinationen, welche eine einfache Detektionsgrundlage eröffnen, erzeugt. Passive OS-Detektion beruht maßgeblich darauf, dass die Protokoll-Spezifikationen der Request for Comments (RFC) für den TCP/IP-Stack nicht alle Situationen eindeutig beschreiben, weiterhin wurden die vorgegebenen Spezifikationen nicht in allen Implementierungen der verschiedenen Betriebssysteme eingehalten, so dass hierdurch zusätzliche Unterschiede entstehen.

Beispielhaft sei nachfolgend die Festlegung für das TTL-Feld gegeben [301]:

> Time to Live: 8 bits
>
> This field indicates the maximum time the datagram is allowed to remain in the internet system. If this field contains the value zero, then the datagram must be destroyed. This field is modified in internet header processing. The time is **measured in units of seconds**, but since every module that processes a datagram must decrease the TTL **by at least one** even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime.

Wie zu sehen ist, wird keine genaue Festlegung der Werte vorgenommen. Somit kann der Initialwert vom Programmierer gesetzt werden, wodurch in der Praxis stark unterschiedliche Werte entstehen können. Insbesondere auch in Bezug auf die Handhabung von mißgeformten Paketen agieren die verschiedenen Betriebssysteme sehr unterschiedlich. *RFC 791* [301] definiert die Spezifikation für IP, *RFC 793* [302] für TCP. Diese bilden eine wesentliche Grundlage für die Kommunikation im Internet und werden in Betriebssystemen in Form des TCP/IP-Stack implementiert. Der Aufbau des IPv4-Headers ist

[17] Dieses Verfahren wird auch als Operating System (OS)-Fingerprinting oder OS-Detektion bezeichnet.

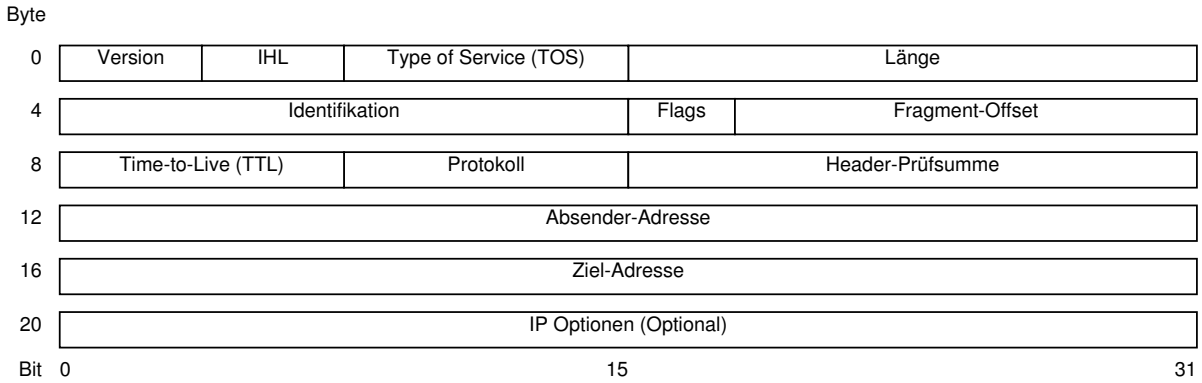


Abbildung 2.14: Aufbau des Headers bei IPv4. *Version* gibt die Version des IP-Protokolls an, *IHL* gibt die Headerlänge als Vielfaches von 32 Bit an, diese ist normalerweise 20 Byte und kann bis zu 60 Byte groß werden. Das *TOS*-Feld gilt der Regelung der Dienstgüte, *Länge* gibt die Gesamtlänge des Pakets an, die *Identifikation* gibt eine eindeutige, fortlaufende Numerierung der Pakete an, die *Flags* werden zur Handhabung von Fragmentierung genutzt, das zugehörige *Offset* gibt deren Position im ursprünglichen IP-Paket an, *TTL* beschreibt die Lebensdauer des Paketes, *Protokoll* gibt den Port des genutzten Transportprotokolls an. Weiterhin finden sich eine *Prüfsumme* und die involvierten Sender- und Empfängeradressen.

in Abbildung 2.14 gezeigt; anhand der zahlreichen Felder können genaue Schlüsse über die involvierten Betriebssysteme gezogen werden, wie nachfolgend demonstriert wird.

Tabelle 2.5 zeigt einige Signaturen im Kontext von TCP, die zur Bestimmung des Systems herangezogen werden können (vgl. [352]).

Abhängig des Betriebssystems weisen diese Werte charakteristische Eigenschaften auf, bspw. mit welchem Wert der initiale TTL-Wert gesetzt wird; Tabelle F.4 in Anhang F.1.6 zeigt eine Auswahl der Werte verschiedener Systeme, getrennt nach TCP und UDP.

Die genutzten Werte können sehr unterschiedlich sein und bereits einen Aufschluss über das betrachtete System geben. Hierbei muss jedoch beachtet werden, dass die entsprechende Information leicht geändert werden kann, um eine Identifikation zu erschweren. Bspw. reicht es in GNU/Linux aus, den Kernel-Parameter `ip_default_ttl` im laufenden Betrieb zu ändern, was mittels eines einfachen Befehls auf der Kommandozeile möglich ist (vgl. Anhang F.1.6), in Windows-Systemen muss hierfür der zuständige Registry-Key angepasst werden.

Andere Flags und systemspezifische Reaktionen können detailliertere Schlüsse zulassen, bspw. die Handhabung von *No more data from sender (FIN)*-Paketen oder die Unterdrückung von ICMP-Fehlernachrichten [157]: Wird ein FIN-Paket an einen offenen Port gesendet, ist das korrekte Verhalten nach RFC 793, *keine* Antwort zu senden. Verschiedenen Implementierungen, z.B. bei MS Windows oder Cisco, antworten jedoch mit einem Reset (RST)-Paket.

Spezifisches Verhalten wie im Falle der ICMP-Fehlernachrichten ist aufwändiger auszuwerten: In der RFC 1812 wird eine Reduzierung der Rate, mit welcher verschiedene

Tabelle 2.5: Auswahl charakteristischer Felder der TCP/IP-Protokolle zur passiven Identifizierung des Betriebssystems.

Merkmal	Wertebereich	Beschreibung
TTL	0 - 255	Alter des Paketes
Window Size	65535	Max. Größe durch 16 Bit; Erhöhung durch *Window Scale* gem. RFC 1323
DF	0 / 1	Fragmentierung des Pakets (nicht) erlaubt
TOS, DiffServ	64 Klassen	TOS (alt), Differentiated Services (neu). Bits zur Steuerung und Festlegung der Handhabung von Paketen im Rahmen von QoS. Vgl. RFC 791, RFC 1122, RFC 1349, RFC 2474, RFC 3168

Fehlernachrichten erzeugt werden, vorgeschlagen, die in den Implementierungen mancher Betriebssysteme berücksichtigt wird. Bspw. reduziert der GNU/Linux Kernel *ICMP-destination unreachable* Nachrichten (Typ 3) entsprechend den Empfehlungen der RFC (vgl. Anhang F.1.7), andere Systeme jedoch nicht.

Die Anzahl verschiedener Indizien, welche genutzt werden um Aussagen über das zugrunde liegende Betriebssystem zu treffen, ist abhängig des jeweils verwendeten Programms sehr unterschiedlich. *siphon* ist ein Proof-of-Concept und eine der ersten Implementierungen eines passiven Fingerprinting-Tools. Es nutzt lediglich drei Indizien der Paket-Header, wobei *SYN*- und *ACK*-Pakete ausgewertet werden und liefert 47 Fingerprints zur Identifikation unterschiedlicher Betriebssysteme mit. Eine Steigerung der Parameterzahl bedeutet aber nicht zwangsläufig eine Verbesserung der Genauigkeit der Detektion bzw. Reduzierung der Fehlerrate: Umso größer die Anzahl der Klassen ist, umso ähnlicher werden die einzelnen Fingerprints. Lippmann et al. haben gezeigt, dass eine Reduzierung der Klassen sowie eine spezielle, gründlich untersuchte Auswahl der Klassifizierungsmerkmale eine passive OS-Identifikation mit geringerer Fehlerrate ermöglicht [252].

Ein wichtiges und ausgereiftes passives Tool ist *p0f* von Michal Zalewski [403]. Es unterstützt vier verschiedene Detektionsmodi, die jeweils eine eigene Datenbank in Form einer Textdatei mit Fingerprints haben. Die Modi sind im Einzelnen:

- Eingehende Verbindungen (Synchronize (SYN) Modus), zur Identifizierung der verbindungsaufnehmenden Stelle (Standard).
- Ausgehende Verbindungen (SYN+Acknowledge (ACK) Modus), zur Identifizierung von Systemen, zu denen Verbindungen aufgebaut werden.
- Abgelehnte, ausgehende Verbindungen (RST+ Modus), zur Untersuchung von Systemen, die einen Verbindungsaufbau abgelehnt haben.
- Bestehende Verbindungen (Stray ACK Modus), zur Evaluation aufgebauter Verbindungen.

Tabelle 2.6: Aufbau eines Fingerprints beim passiven Fingerprintingtool p0f.

Option	Bedeutung
www	Window Size
ttt	Initial TTL
D	Don't Fragment Bit
ss	Overall SYN Packet Size
OOO	Option Value and Order Specification
QQ	Quirk Links
OS	OS genre (Linux, Solaris, Windows)
details	OS description (2.6.18, etc.)

Ein Fingerprint ist nach dem Schema

```
wwww:ttt:D:ss:OOO...:QQ:OS:Details
```

aufgebaut, die Bedeutungen der Felder sind in Tabelle 2.6 angegeben (vgl. [402]).

Grundlegend muss unterschieden werden, ob ein passiver Scanner nur dann ein Ergebnis liefert, wenn der Fingerabdruck in der Datenbank vorhanden ist, oder ob auch ähnliche ausgegeben werden können. Da es vielfach vorkommt, dass ein Fingerabdruck mehrere Systeme repräsentiert, muss auch das Verhalten des Programms in diesem Falle berücksichtigt werden: Manche Programme geben hier nur das erste gelistete System, das vollständig mit dem Fingerabdruck übereinstimmt, zurück. Die Evaluation eines einzelnen SYN-Paketes ist im Anhang dargestellt (vgl. Anhang F.1.8).

Aufgrund der Unsicherheiten und ähnlichen Parametern kann mittels passiven Methoden ein System nicht immer mit Sicherheit bestimmt werden. Insbesondere wird jedoch auch eine Aufzeichnung des Datenverkehrs benötigt, um diese entsprechend auszuwerten. Sind diese Daten nicht vorhanden, da man sich bspw. außerhalb des Zielnetzes befindet, müssen aktive Methoden genutzt werden.

Arbeitet ein Scanner aktiv, erzeugt er spezielle Pakete und sendet diese an zuvor bestimmte Adressen und Ports im Zielnetz, um die Reaktion darauf auszuwerten. Bekanntestes Beispiel eines aktiven Scanners ist das Tool *Network Mapper (nmap)* [262], aktuell in der Version 5.51[18] verfügbar. Mit dessen Hilfe können zahlreiche aktive Verfahren genutzt werden, um die in einem Netz vorhandenen Systeme sowie die durch diese zur Verfügung gestellten Dienste zu identifizieren. Hierbei wird widerrum ausgenutzt, dass verschiedene Systeme insbesondere auf fehlerhafte Anfragen, bspw. falsch gesetzte Flags im IP-Header, unterschiedlich reagieren. Für die Suche nach Hosts können z.B. Nachrichten der Typen TCP SYN/ACK, UDP, Stream Control Transmission Protocol (SCTP), ICMP Echo, ICMP Timestamp, ICMP Netmask Request oder IP Protocol Ping eingesetzt werden (vgl. Anhang F.1.4).

Sind die im Zielnetz vorhandenen Systeme gefunden, können diese in mehr Tiefe hinsichtlich der darauf laufenden Betriebssysteme sowie der Dienste und der entsprechenden Versionen untersucht werden. Ist hierbei aufgrund von zu wenig Informationsquellen nur

[18] 08. März 2011

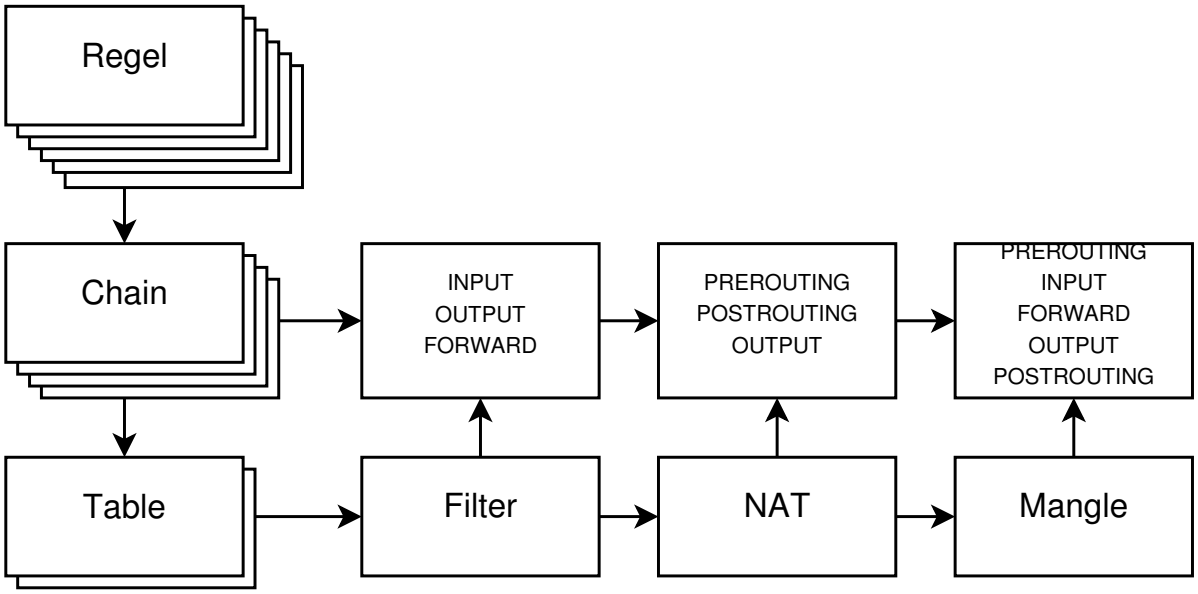


Abbildung 2.15: Gruppierung und Aufbau des netfilter-Regelwerks.

eine Schätzung möglich, gibt der Scanner entsprechende Informationen bzgl. der Qualität der gesammelten Informationen mit aus.

Der Aufbau der hier gewonnenen Fingerprints ist relativ umfangreich und unter [17] und [12] mit den entsprechend unterstützten Möglichkeiten beschrieben.

Entscheidend für die Analyse der Zielumgebung ist der entsprechend angepasste Einsatz der Scanner: Firewalls bieten heutzutage umfangreiche Regelwerke, um Netzscans zu erkennen und zu unterbinden; bspw. kann auf einfache Indizien wie speziell gesetzte Flags im Paket-Header, aber auch auf die Anzahl der (versuchten) Verbindungsaufbauten zu einem oder mehreren Systemen im hinter der Firewall liegenden Netz geachtet werden.

Der typische Weg eines Pakets durch die einzelnen Analysen einer Firewall ist anhand der *netfilter*-Firewall, die in GNU/Linux seit Version 2.4 eingesetzt wird, skizziert. Ein Regelwerk einer Firewall kann aus zahlreichen Einträgen aus verschiedenen Aufgabenbereichen bestehen, daher lassen sich die Regeln zur Übersichtlichkeit gem. Abbildung 2.15 in sog. *Chains* gruppieren, diese werden widerrum in den *Tables* zusammengefasst. Es existieren drei Tables, *Filter*, *NAT* und *Mangle*. Filter hat die Aufgabe, Pakete zuzulassen bzw. abzulehnen, NAT handhabt die entsprechenden Network Adress Translation (NAT)-Aufgaben und Mangle nimmt Paketmanipulationen vor. Jede dieser Tables hat mehrere Chains, die unterscheiden, ob das Paket an den Rechner gerichtet ist (*INPUT*), vom Rechner verschickt wird (*OUTPUT*), von diesem weitergeleitet wird (*FORWARD*) oder über ein Interface ankommt bzw. abgeht (*PREROUTING*, *POSTROUTING*).

Der Paketfluss durch die verschiedenen Instanzen der Firewall ist in Anhang F.1.10 graphisch dargestellt.

Da Firewalls und andere Sicherheitssysteme genutzt werden können, erkannte Netzscans

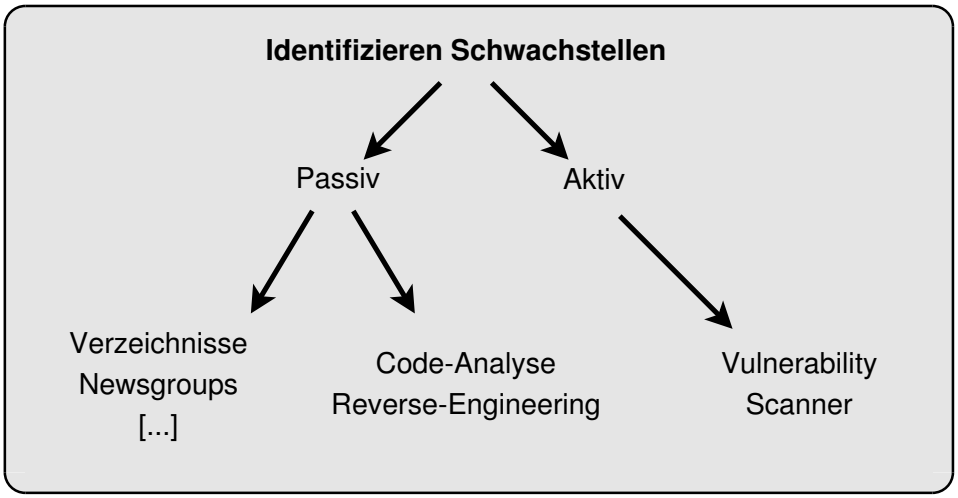


Abbildung 2.16: Angriffs-Teilschritt *Identifizieren von Schwachstellen*.

zu unterbinden, indem die zugehörigen Pakete verworfen werden, bieten Scanner umfangreiche Optionen, um der Detektion und Blockade zu entgehen. nmap bietet daher bspw. zahlreiche Timing-Optionen zur Durchführung der Netzscans an. Diese reichen von `paranoid` bis `insane` oder können darüber hinaus auch noch durch entsprechende Parameter gesteuert werden. Wird z.B. die Option `paranoid` gewählt, wird die Analyse rein seriell durchgeführt und immer nur ein Port untersucht, zwischen den einzelnen versendeten Paketen zur Untersuchung des Netzes wartet der Scanner jeweils fünf Minuten. Die Analyse eines großen Adressbereiches oder einer hohen Anzahl von Ports auf den Zielsystemen kann daher schnell viele Tage dauern, eine Reduzierung der untersuchten Ports auf die relevanten und benötigten Informationen ist hier daher unumgänglich.

Nicht nur die passive Informationssammlung kann einfach manipuliert werden, wie oben am Beispiel der TTL gezeigt wurde, auch im Rahmen der aktiven Analyse können zahlreiche Eingriffe vorgenommen werden. Beispiel hierfür ist das Programm *IP Personality* [328], das ein Kernel-Modul zur Beeinflussung charakteristischer Merkmale des Netzverkehrs zur Verfügung stellt. Es können damit u.a. Sequenznummern, Fenstergrößen und die Werte und Reihenfolge der verschiedenen TCP-Optionen manipuliert werden. Das Programm stellt hierfür ein neues Ziel in der Mangle-Table der netfilter-Firewall zur Verfügung und ermöglicht somit, das aktive Fingerprinting eines gesamten, durch eine Firewall gesicherten Netzes zu beeinflussen.

Identifizieren von Schwachstellen Ist die Zielumgebung mit den entsprechenden Rechnern, Diensten und Infrastrukturkomponenten analysiert, können angreifbare Schwachstellen ermittelt werden (vgl. Abbildung 2.16).

Grundlegendes Vorgehen ist hier, die identifizierten Dienste und Produkte auf die Anfälligkeiten bzgl. bekannter Schwachstellen hin zu überprüfen. Hierfür können bspw. die in Sicherheitstickern, auf den Seiten der Hersteller oder im Rahmen von Untergrundservern verbreiteten Informationen herangezogen werden.

Die Common Vulnerabilities and Exposures (CVE)-Datenbank ist ein wichtiges Beispiel einer Liste, in der öffentlich bekannte Informationen über Schwachstellen von Produkten aller Hersteller geführt werden. Neben dem eindeutigen CVE-Bezeichner sind zu jeder Schwachstelle ausführliche Informationen über Art, betroffene Programmversionen, Auswirkungen und externen Links, z.B. zu Hersteller-Patches gegeben. Anhang F.1.9 zeigt beispielhaft einen CVE-Eintrag.

Verschiedene Schwachstellen-Scanner sind verfügbar, die diesen Vorgang unterstützen und automatisieren. Mit die bekanntesten davon sind *Nessus* [341] und *Nikto* [367]. Nessus beinhaltet derzeit[19] 42067 verschiedene Schwachstellentests, wobei hier 14971 CVE-Einträge[20] abgedeckt werden. Das Framework bietet mittels einer einfachen Bedienoberfläche und einem modularen Konzept eine einfache und automatisierte Möglichkeit, Netze nach ausnutzbaren Schwachstellen hin zu untersuchen. Hierbei werden die anwendbaren Tests in mehrere Kategorien unterteilt, u.a. auch danach, ob es bei der Anwendung zu einem Absturz des Zielsystems kommen kann oder nicht.

Die BugTraq-Mailingliste von SecurityFocus [345] ist ein weiteres wichtiges Beispiel für die Diskussion von Schwachstellen. Entdeckte Fehler werden ebenfalls mit einer ID versehen und mit ausführlichen Informationen, betroffenen Systemen, verfügbaren PoC, Referenzen zu u.a. korrespondierenden CVE-Einträgen, etc. zur Verfügung gestellt (vgl. Anhang F.1.9).

Speziell auf die Sicherheit von Web-Applikationen sind das Framework Nikto sowie die Projekte von The Open Web Application Security Project (OWASP) zugeschnitten [15]. Nikto führt schnelle Untersuchungen von Webservern auf das Vorhandensein von gefährlichen Dateien, alten Serverversionen und versionsspezifischen Schwachstellen durch, zusätzlich werden typische Konfigurationsfehler des Servers analysiert.

Besondere Bedeutung im Rahmen der Suche nach Schwachstellen haben die sog. Zero-Days. Die Definition von Zero-Days ist nicht ganz einheitlich. Insbesondere ist die Betrachtung, inwieweit bekannte aber noch nicht gepatchte Schwachstellen hierzu zu zählen sind, umstritten (vgl. z.B. [120], [387], [369] und [62]). Die häufiger anzutreffende und im Sinne der Gefährdung auch zutreffendere Definition betrachtet Zero-Day-Schwachstellen jedoch nur im Sinne öffentlich noch nicht bekannter Schwachstellen; entsprechend werden Zero-Days nachfolgend in diesem engeren Sinne verstanden. Der Grund hierfür ist, dass oftmals die Kenntnis einer Schwachstelle auch ohne bereits vorliegenden Patch eine wenn auch mit Einschränkungen verbundene Schutzmöglichkeit vor Angriffen eröffnet, bspw. durch Workarounds bis hin zur temporären Unterbrechung eines Dienstangebotes. Da die Ausnutzung der Schwachstelle hiermit aber ebenfalls verhindert werden kann, ist eine solche Form bekannter Fehler nicht in den Bereich von Zero-Days zu zählen.

Definition (Zero-Day-Exploit)**.** *Ein Zero-Day ist eine Schadsoftware, die eine Schwachstelle eines Programms vor dem oder am Tag des Bekanntwerdens durch den Hersteller bzw. die Öffentlichkeit ausnutzt.*

Zero-Days, bzw. Schwachstellen im Allgemeinen, können durch entsprechende Analysen des Programmcodes, durch Fault-Injection oder Fuzzing-Verfahren gefunden werden.

[19] Stand März 2011.
[20] Von insgesamt 45351 CVE-Einträgen am 09.03.2011.

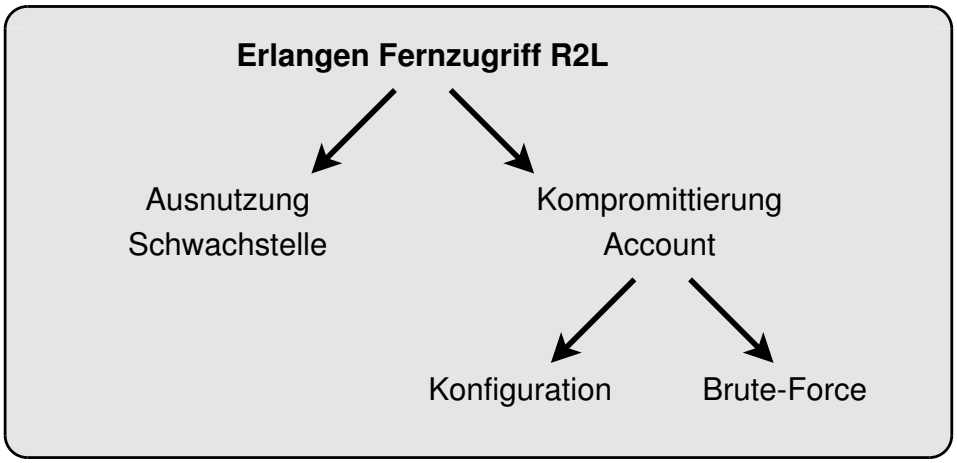


Abbildung 2.17: Angriffs-Teilschritt *Erlangen Fernzugriff R2L*.

Hierbei können zahlreiche Tools zur Automatisierung und Unterstützung, wie bspw. Debugger und Disassembler, genutzt werden. Angreifbare Schwachstellen können durch ein fehlerhaftes Design im Entwicklungsprozess, durch Fehler in der Programmierung oder Konfigurationsfehler entstehen. Ein typisches Beispiel eines Programmierfehlers sind fehlende Prüfung von Eingabedaten.

Aufgrund des mit der Suche verbundenen Aufwands sowie den umfangreichen Angriffsmöglichkeiten, die eine noch nicht behobene und öffentlich unbekannte Schwachstelle einem Angreifer bietet, sind Zero-Days im Untergrund eine wichtige Handelsressource. Hierbei können Preise im hohen, fünfstelligen Dollarbereich erzielt werden.

Erlangen von Fernzugriff Nachdem die in einem Netz verfügbaren Dienste detektiert wurden und mögliche, ausnutzbare Schwachstellen analysiert wurden, muss eine Fernzugriffsmöglichkeit auf das Zielsystem geschaffen werden. Abhängig der gefundenen Schwachstellen können maßgeblich folgende Verfahren angewandt werden (vgl. Abbildung 2.17):

- Ausnutzen einer Schwachstelle des Zielsystems
- Kompromittierung eines Accounts
  - Brute-Force / Wörterbuchangriff
  - Auslesen von Passwortdateien bei Konfigurationsfehlern oder fehlerhafter Rechtevergabe und Offline-Angriff

Wurde eine geeignete Schwachstelle gefunden, für die ein Exploit zur Verfügung steht, kann diese ausgenutzt werden, um Zugriff zum System zu erhalten. Um den Angriff durchzuführen, kann bspw. das Metasploit-Framework eingesetzt werden [256]; dieses dient dem Entwickeln, Testen und Anwenden von Exploit-Code. Mittels Konsole oder dem ebenfalls verfügbaren Webinterface können in einfacher Art die für ein Zielsystem

vorhandenen Exploits ausgewählt und konfiguriert werden, mit einem passenden Payload, bspw. zum Starten einer Shell, versehen und an das Ziel gesendet werden. Neue Exploits können auf einfache Weise entwickelt und ergänzt werden. Abhängig der ausgenutzten Schwachstelle steht in manchen Fällen bereits hier ein administrativer Zugang zur Verfügung, so dass in diesem Fall der nächste Schritt in der Angriffsfolge (*Erlangen administrativer Rechte*) übersprungen werden kann.

Eine weitere Möglichkeit, Systemzugang insbesondere dann zu erhalten, wenn keine ausnuztbaren Schwachstellen gefunden wurden, sind vorhandene Nutzerkonten. Da in einer vernetzten Umgebung oftmals ein entfernter Zugriff auf ein System ermöglicht werden muss, stehen zahlreiche Dienste zur Verfügung, die einem Nutzer einen entfernten Systemzugang bereitstellen. Früher waren hier insbesondere Programme wie `telnet`, `remote shell (RSH)`, `Remote Copy (RCP)` und `Remote login (Rlogin)` verbreitet, die allerdings alle Daten inklusive der Anmeldedaten des Nutzers unverschlüsselt über das Netz übertragen haben. Entsprechend werden heutzutage sichere Varianten der Dienste eingesetzt, bspw. `SSH` für den Fernzugriff, das auch Basis für die sicheren Varianten `Secure Copy (SCP)` bzw. `SFTP` ist, aber auch zur Absicherung von ansonsten unverschlüsselten Protokollen wie bspw. `Virtual Network Computing (VNC)` genutzt werden kann. Ist bei den entsprechenden Diensten eine Nutzer-Authentifizierung mittels Name und Passwort möglich, kann ein Brute-Force- oder Wörterbuchangriff gestartet werden. Während beim Wörterbuchverfahren eine Liste möglicher Passwörter systematisch durchprobiert wird, werden beim Brute-Force-Verfahren alle möglichen Kombinationen bis zur vorgegebenen Maximallänge versucht[21].

Ein Beispiel eines entsprechenden Tools ist `brutessh` [127], das anhand einer Passwortliste und einer Liste mit Nutzernamen die entsprechenden Kombinationen überprüft. Um die Geschwindigkeit zu erhöhen, können die Anfragen hierbei parallelisiert werden.

Im Falle fehlerhafter Konfigurationen eines Servers kann es dem Angreifer auch möglich sein, Daten auszulesen, die eigentlich nicht für den Zugriff vorgesehen sind. Ist auf diese Weise ein Zugriff bspw. auf eine Passwortdatei möglich, kann diese kopiert werden und eine lokale Passwortsuche gestartet werden, zum Beispiel mit dem Tool `John the Ripper` [312], Cain und Abel [284] oder *Rainbowtables* (s.u.).

Erlangen administrativer Rechte Hat der Angreifer durch eine erfolgreiche Penetration des Systems einen Zugriff mit bestimmten Nutzerrechten erlangt, oder hat er aufgrund seiner Position bereits einen legitimierten Systemzugang, ist der nächste Schritt abhängig des angestrebten Zieles (vgl. Tabelle 2.7): Hier sind Datenzugriff und -ausschleusung bzw. Systemzugriff und Fremdnutzung mit den jeweils benötigten Rechten zu unterscheiden. Hat der Angreifer im Falle der Durchführung eines Datendiebstahls bereits durch den Nutzeraccount Zugriff auf die erforderlichen Daten, müssen weitere Schritte nur ergriffen werden, wenn für die Extraktion der Daten aus dem System zusätzliche Rechte erforderlich sind. Sind andere oder höhere Rechte für den Zugriff erforderlich, oder soll

[21] In diesem Kontext sei darauf hingewiesen, dass auch heute noch sehr fahrlässig mit Passwörtern umgegangen wird. Verschiedenen Studien haben wiederholt gezeigt, dass Nutzer oftmals einfache Passwörter einsetzen und diese auch mehrfach nutzen.

Tabelle 2.7: Konstellationen und erforderliche Schritte nach Erlangung des Systemzugriffs.

Verfügbare Rechte		Erhöhung Privilegien	Ausführung
	Allgemein	✓	
Nutzer	*Zugriff*	✓	
	Ausschleusung		✓
Administrator			✓

das System für die weitere Fremdnutzung kompromittiert werden, ist eine entsprechende Rechteerweiterung (Privilege Escalation, Masquerading) notwendig.

Um eine erforderliche Erhöhung von Rechten vorzunehmen, können verschiedene Ansätze gewählt werden (vgl. Abbildung 2.18): Zum einen kann eine fehlerhafte Rechtesetzung bei Programmen oder Konfigurationsdateien für eine Rechteeskalation genutzt werden. Bspw. existieren in Unix und unixähnlichen Betriebssystemen das sog. Set-User-ID (SUID)-, Set-Group-ID (SGID)- und das Sticky-Bit (t-Bit), welche besondere Zugriffsrechte auf Dateien und Verzeichnisse bestimmen (vgl. z.B. [404]). Von besonderer Bedeutung ist das SUID-Bit, wenn es bei Programmen gesetzt ist: In diesem Fall werden die entsprechenden Programme mit den Rechten des Besitzers ausgeführt, z.B. administrative Tools, die dem Administrator gehören und durch Nutzer aufgerufen werden können; Beispiele hierfür sind `mount` oder `su`. Die Nutzung des SUID-Bit vereinfacht viele Aufgaben, bspw. ermöglicht es einem normalen Nutzer, sein Passwort zu ändern, obwohl dieser keine Zugriffe auf die Passwortdatei `/etc/shadow` hat, jedoch das hierbei genutzte Programm `/usr/bin/passwd`[22] mit den Rechten des Besitzers `root` ausgeführt wird.

Ein Angreifer kann in diesem Rahmen insbesondere Kommandos nutzen, um alle Dateien mit SUID oder SGID sowie *world-writable* Dateien, also Dateien, die durch jeden Nutzer änderbar sind, zu suchen:

```
$  find / -type f \( -perm -04000 -o -perm -02000 \)
  [...]
$ find / -perm -2 ! -type l -ls
  [...]
```

Hat ein fehlerhaftes Programm entsprechend gesetzte Ausführungsrechte, kann dies zum Erlangen von administrativen Rechten durch einen Angreifer führen. Auch eine falsche Rechtevergabe bei nicht-ausführbaren Dateien kann ausgenutzt werden: Lassen sich bspw. die Inhalte von Passwortdateien auslesen, können diese oftmals einfach genutzt werden, um die zugehörigen Passwörter zu ermitteln. Zwar finden sich heutzutage regelmäßig keine Klartextpasswörter mehr in den entsprechenden Dateien, sondern typischerweise nur Hashwerte, jedoch existieren effektive Möglichkeiten, diese anzugreifen. Bei Hashwerten (auch kryptographische Prüfsummen genannt) werden Zeichenketten beliebiger Länge auf Zeichenketten vorgegebener fester Länge abgebildet. Eine wichtige Forderung hierbei ist die Kollisionsfreiheit, d.h. dass es nicht effizient möglich ist, für eine

[22] Der Pfad kann system-/ distributionsspezifisch variieren, z.B. nach `/bin`.

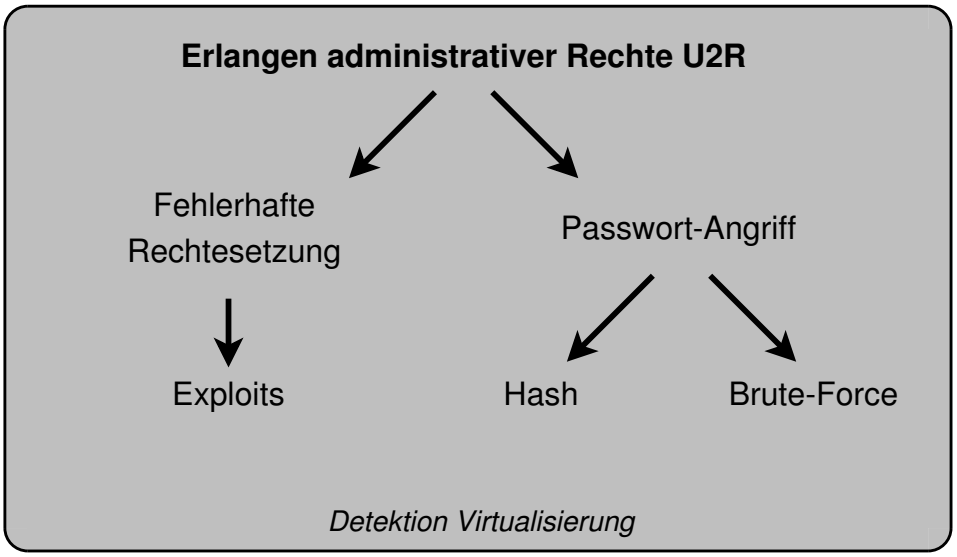


Abbildung 2.18: Angriffs-Teilschritt *Erlangen administrativer Rechte U2R*.

Hashfunktion $f\colon \mathcal{M} \to \mathcal{N}$ zwei Werte $x_1, x_2 \in \mathcal{M}$ zu finden, so dass gilt: $x_1 \neq x_2$ aber $f(x_1) = f(x_2)$. Dies bezeichnet man auch als *praktisch injektiv* [311]. Die Schwierigkeit bei solchen Einwegfunktionen liegt also darin, dass der Klartext zu einem Hashwert nicht zurückgewonnen werden kann. Eine Möglichkeit, den Klartext zu einem Hashwert zu erlangen liegt beispielsweise in einem traditionellen Wörterbuchangriff, der Berechnung der Hashes abhängig des verwendeten Algorithmus und dem anschließenden Vergleich der Werte. Dieses Verfahren dauert extrem lange; eine andere Möglichkeit ist die Vorausberechnung und Speicherung aller Klartext-Hashwert-Kombinationen, was zu riesigen Tabellen führt. Ziel ist es daher, eine hohe Zahl von Hashes kompakt zu speichern. Von besonderer Bedeutung ist hier der Rechenzeit-Speicher-Tradeoff *Rainbowtables* von Philippe Oechslin [298], [38]. Dieser ermöglicht das Erstellen effizienter Rainbow-Tabellen; zahlreiche Dienste im Netz, sowohl kostenfrei als auch kommerziell, bieten vorberechnete Tabellen unterschiedlichen Umfangs für verschiedene Hash-Algorithmen wie MD5, LM, NTLM, SHA1 oder für die Nutzung unter MySQL an (vgl. z.B. [3], [19], [6], [5], [10]).

Wurden ausnutzbare Schwachstellen in Programmen festgestellt, kann der Angreifer entsprechende Exploits auf das kompromittierte Nutzerkonto laden. Die häufigsten, generischen Angriffsmethoden sind hier *Buffer-Overflows* und *Format-Strings*. Buffer-Overflow Techniken wurden über die Jahre intensiv erforscht und zahllose Veröffentlichungen behandeln alle Aspekte dieses Verfahrens; einer der ersten davon ist der leicht verständliche Artikel „Smashing The Stack For Fun And Profit“ [404] aus dem Jahre 1996. Grundlegend nutzen diese Verfahren aus, dass in Hochsprachen wie *C* keine Überprüfungen von Speicherzugriffen während der Programmlaufzeit erfolgen und diese auch zur Übersetzungszeit nicht geprüft werden; dies resultiert in hohen Compiler- und Ausführungsgeschwindigkeiten, eröffnet jedoch Angriffsmöglichkeiten, da Daten über die eigentlichen Grenzen eines reservieren Feldes hinaus in den Speicher geschrieben wer-

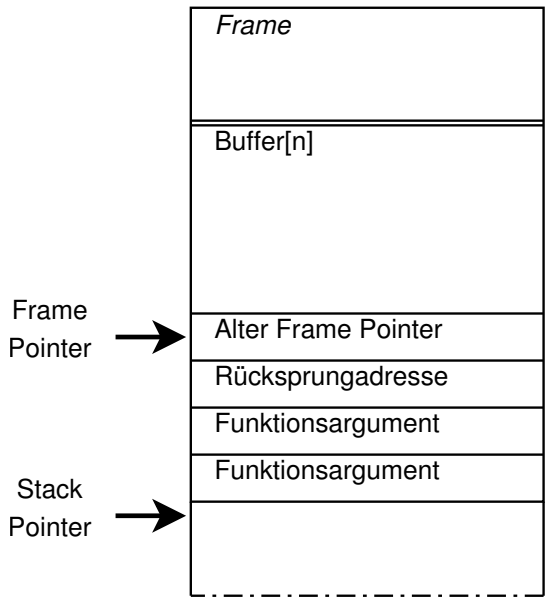


Abbildung 2.19: Aufbau des Stacks und Lage wichtiger Informationen.

den können (vgl. Abbildung 2.19). Fügt man in den Buffer an geeigneter Stelle einen passenden Bytecode ein und überschreibt die Rücksprungadresse entsprechend, wird dieser ausgeführt und kann bspw. zum Öffnen einer Konsole mit administrativen Rechten benutzt werden. Ähnliche Techniken können auch auf den Heap- und Block Started by Symbol (BSS)-Speicher angewandt werden; da hier jedoch keine Rücksprungadresse vorhanden ist, basieren entsprechende Angriffe darauf, dass dort liegende, wichtige Variablen überschrieben werden oder Funktionszeiger umgesetzt werden (vgl. [128]).

Eine Schutzmaßnahme gegen solche Angriffe bieten nicht-ausführbare Stacks, so dass in den Datenbereich eingeschleuster Schadcode nicht durch das Betriebssystem ausgeführt wird (Data Execution Prevention (DEP), vgl. z.B. [288]). Von der Seite der Angreifer kann hier mit verfeinerten Techniken wie *Return-to-libc* (vgl. z.B. [128]) oder *Return-oriented Programming* (vgl. z.B. [204], [67]) gearbeitet werden, um trotzdem zum Erfolg zu kommen.

Eine weitere Erhöhung der Sicherheit kann mittels Randomisierung der genutzten Adressen erreicht werden (Address Space Layout Randomization (ASLR)). Ohne das Wissen um die Positionen des Programms im Speicher während der Ausführung, ist es schwer für den Angreifer die notwendige Adresse für das Einbringen des Schadcodes zu bestimmen. Diese in der Kombination sehr effektiven Verfahren lassen sich jedoch bereits auch schon wieder angreifen (vgl. z.B. Techniken des *Just-in-Time (JIT)-Sprayings*, [58], [337]) und sind ihrerseits widerrum nicht immer fehlerfrei implementiert (siehe z.B. [385]).

Um die auf Schwachstellen in Programmen basierenden, vorgestellten Angriffe einfach umzusetzen, können Frameworks für Penetrationstests wie Metasploit [256], Canvas [208] oder Core Impact [373] eingesetzt werden, um die notwendigen Prozesse zu automati-

sieren und zu vereinfachen.

Ein weiterer Punkt, der im Rahmen der Rechteerweiterung und des Angriffs eines Systems beachtet werden muss, ist der zunehmenden Einsatz von Virtualisierungstechnologien. Prinzipiell sind zwei Hauptvorteile der Virtualisierung eine bessere Ressourcenausnutzung (vgl. z.B. [30]) sowie eine Erhöhung der Sicherheit: Ist das System einer virtuellen Maschine angreifbar und wird diese Schwachstelle ausgenutzt, so bleibt der mögliche Schaden auf diese Instanz begrenzt. Der Angreifer kann nur die Daten, welche zur entsprechenden Instanz gehören, kompromittieren, der Schaden wird somit eingegrenzt. Andererseits bieten sich durch den Einsatz von Virtualisierungssoftware neue Gefahren, wenn die Implementierung selbst Schwachstellen aufweist. Ist der Angreifer bspw. in der Lage, Kontrolle über den Hypervisor der Virtual Machine (VM) zu übernehmen, kann er auf alle darin laufenden Instanzen beliebig Einfluss nehmen.

Daher ist die Detektion, ob sich ein Angreifer innerhalb einer Virtualisierung befindet, ebenfalls ein wichtiger Aspekt in diesem Angriffsschritt. Hierbei gilt, dass ein komplettes Verstecken der Virtualisierung regelmäßig nicht möglich ist, bspw. können Eigenschaften der VM wie Backdoors oder bestimmte Bezeichner der darin befindlichen virtuellen Geräte entsprechende Hinweise geben (vgl. [364]). Der Befehl `dmidecode` lässt sich zum Beispiel nutzen, die Desktop Management Interface (DMI) Tabellen auszugeben, welche nach dem Bootvorgang im Speicher liegen und somit über `/dev/mem` erreichbar sind. Diese Tabellen enthalten Beschreibungen der Geräte, die laut Basic Input Output System (BIOS) im System vorhanden sind. Da ein Hypervisor Systemkomponenten wie bspw. Grafikkarte, Netzinterface und letzlich auch das BIOS der VM selbst virtualisiert, sind dort oftmals entsprechende Bezeichner des Herstellers der Virtualisierung eingetragen. Auch wird in einigen Virtualisierungsprodukten `mmap` nicht unterstützt und liefert beim Aufruf von `dmidecode` eine entsprechende Fehlermeldung. Beispielsweise kann eine VMware-Virtualisierung weiterhin anhand des vorhandenen Backdoors einfach erkannt werden (vgl. Anhang F.1.9).

Selbst wenn diese (sehr einfachen) Identifizierungsmöglichkeiten anhand von Systemausgaben oder dem Vorhandensein von Backdoors nicht greifen, kann anhand von logischen Abweichungen (bspw. in der Interrupt Description Table (IDT)) oder zeitlichen Abweichungen eine Virtualisierung detektiert werden (vgl. [365]).

Wurde durch eine der aufgezeigten Varianten eine Eskalierung der Priviliegen erreicht, kann eine entsprechende Manipulation des Zielsystems erfolgen.

Manipulation Systemumgebung Hat der Angreifer administrative Rechte in seiner Zielumgebung erworben, folgt im nächsten Schritt die Installation eines Backdoors oder Rootkits, um den Zugriff auf den Rechner dauerhaft sicherzustellen und weitere Aktionen initiieren zu können. Die Bandbreite der zur Verfügung stehenden Möglichkeiten reicht hier vom einfachen Hinzufügen eines neuen Nutzerkontos, mit dem sich der Angreifer dann äußerlich legal verbinden kann, bis hin zu komplexen und umfangreichen Rootkits, welche Systemprogramme manipulieren und austauschen, um sich zu verstecken und umfangreiche Kontrollmöglichkeiten anbieten (vgl. Abbildung 2.20).

Mittlerweile gibt es eine sehr umfangreiche Auswahl an Backdoors für alle Plattformen;

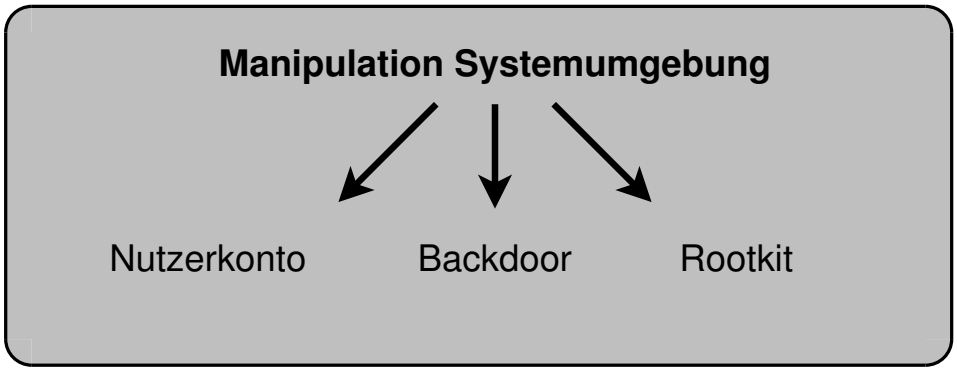


Abbildung 2.20: Angriffs-Teilschritt *Manipulation Systemumgebung.*

die Untersuchungen von Symantec zeigen, dass im Jahre 2009 der Anteil von Backdoors an der gesamten Schadsoftware bei 15 Prozent lag [141]. Für die Bereitstellung einer Backdoor kann im einfachsten Falle eine Manipulation des `inetd`-Dienstes mittels der Konfigurationsdatei `/etc/inetd.conf` erfolgen [81]:

```
discard stream tcp  nowait root /bin/bash -i
```

Nach einem Neustart des Daemons, z.B. durch Senden eines Signals `killall -HUP inetd`, wird der Server aktiv. Verbindet man sich nun mit dem Rechner, wird die entsprechende `bash`-Shell anstatt eines Dienstes gestartet. Das Tool `netcat` [162] kann ebenfalls genutzt werden, eine einfache und extrem schnelle Zugangsmöglichkeit zu schaffen [81]:

```
# Erzeugen des Backdoors
$ while true; do nc -l -p 8090 -e /bin/bash done
# Nutzung des Backdoors von Extern
$ nc 137.193.63.191 8090
```

Auch die Nutzung eines bereits installierten VNC stellt eine einfache Möglichkeit da. `Back Orifice` ist ein weit verbreitetes Backdoor, das bereits 1998 entwickelt wurde. Eine Neuimplementierung wurde unter dem Namen BO2K veröffentlicht, welche mittels eines Erweiterungskonzeptes flexibel einsetzbar ist und sowohl für Windows-, als auch für GNU/Linux-Plattformen zur Verfügung steht [1]. Professionelle Backdoors bieten auch verschlüsselte Kommunikation an, so zum Beispiel auch die für `BO2K` verfügbaren Erweiterungen.

Bei der Sicherstellung des Systemzugriffs von Außen muss jedoch berücksichtigt werden, dass der Zugang zu dem installierten Dienst von Außen, insbesondere also dem Internet, durch eine dazwischen liegende Firewall weiterhin vereitelt werden kann. Da Firewalls nicht nur im Bereich von Firmennetzen (siehe oben), sondern auch im privaten Bereich oft auf einem dritten Gerät ausgelagert sind, bspw. ein entsprechender, dedizierter Rechner in einem Firmennetz oder mittels eines Digital Subscriber Line (DSL)-Routers in einem privaten Heimnetz, kann die Erreichbarkeit von extern nicht sichergestellt werden. Dies spiegelt sich z.B. im Kommunikationsverhalten von Bot-infizierten Rechnern wider: Da der Kommunikationsaufbau vom infizierten Rechner selbst und so-

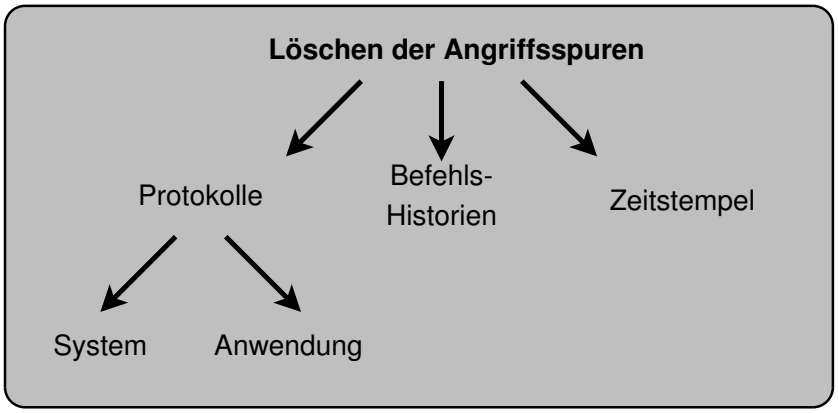


Abbildung 2.21: Angriffs-Teilschritt *Löschen der Angriffsspuren.*

mit von Innen nach Außen ausgeht, wird dieser durch die Firewall typischerweise nicht unterbunden. Das Schadprogramm kann somit Kontakt zum im Internet horchenden Server aufnehmen und erhält nach der Kontaktaufnahme seine neuen Befehle.

Um umfangreichere Einflussnahme auf das Zielsystem auszuüben, können Rootkits installiert werden, Systemprogramme ersetzt, Keylogger installiert werden u.v.m. Im Gegensatz zu Backdoors, deren Aufgabe es ist, einen Fernzugriff auf das kompromittierte System zu gewährleisten, ist der Hauptzweck von Rootkits das Verstecken der Schadsoftware selbst sowie von Applikationen auf dem Zielsystem. Rootkits gibt es ebenfalls schon seit geraumer Zeit; während anfangs lediglich installierte Hilfsprogramme gegen eigene Versionen ausgetauscht wurden, insbesondere um dadurch an höhere Zugriffsrechte zu gelangen, haben die danach entwickelten Detektionsmöglichkeiten eine Weiterentwicklung erzwungen, so dass moderne Rootkits im Kernelspace arbeiten und sich mit ausgefeilten Techniken verstecken können. Adore [366], Devil, iLLogiC [93] oder hxdef [13] sind einige Beispiele anzutreffender Rootkits.

Löschen der Angriffsspuren Sind die Manipulationen an der Systemumgebung vorgenommen, müssen als letzter Schritt die hinterlassenen Spuren gelöscht werden, um eine Entdeckung der Kompromittierung zu vermeiden. Grundsätzlich gilt es, folgende Bereiche zu bereinigen (vgl. Abbildung 2.21):

- Systemprotokolle
- Anwendungsprotokolle
- Konsolen-Historien
- Forensische Spuren an Dateien

Insbesondere die Anwendung von Exploits kann zahlreiche Einträge in den Systemprotokollen hinterlassen, die sehr auffällig sind. Nachfolgend ist beispielhaft ein (gekürzter) Eintrag nach einem versuchten Puffer-Überlauf des Unix-RPC-Dienstes abgebildet [136]:

```
Oct 21 04:18:12 server01 rpc.statd[522]: gethostbyname error for ^
    XÃ·Ã¿Ã¿^XÃ·Ã¿Ã¿^ZÃ·Ã¿Ã¿^ZÃ·Ã¿Ã¿%8x%8x%8x%8x%8x%8x%8x%8x%8x%62 716x%
    hn%51859x%hn
    \220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
  [...]
\220\220\220\220\220\220\220\220\220\220
```

Standardmäßig werden hier die Logdateien bspw. in Linux insbesondere in `/var/log` abgelegt, von besonderer Bedeutung ist hier die Datei `messages`. Aufgrund der hohen Auffälligkeit und der Gefahr einer möglichen Wiederherstellung, die mit einem kompletten Löschen einer Protokolldatei verbunden ist, werden typischerweise entsprechende Einträge aus den jeweiligen Protokolldateien einfach mittels Tools wie `vi`, `awk` oder `sed` entfernt. Neben den standardmäßigen Logdateien können jedoch auch Kopien, nichtstandardmäßige Speicherorte, etc. vorliegen. Insbesondere können die Dienste des Systems Logdateien in ihren jeweiligen, spezifischen Verzeichnissen anlegen. Der Angreifer muss daher weitere Befehle ausführen, um weitere mögliche Protokolldateien zu finden; der Befehl `grep -r 'Angreifer-IP' /` kann hier ein möglicher Ansatz sein, oder die Nutzung des Tools `find`, z.B. mittels `find . -type f -name *.txt -exec grep -l -i suchtext '' .`

Weiterhin existieren unter Unix-Systemen auch mehrere binäre Protokolldateien, bspw. für die Aufzeichnung der Nutzer-Logins. Diese können durch die Nutzung entsprechender Tools von verräterischen Einträgen bereinigt werden.

Wichtig ist weiterhin das Entfernen aller hinterlassenen Einträge aus den *history*-Dateien der Konsole, z.B. `.bash_history` oder `.history`.

Zeitstempel sind ein weiterer wichtiger Hinweis darauf, was für Aktionen durch einen Angreifer auf dem Zielsystem durchgeführt worden sind. Abhängig des genutzten Dateisystems werden hier unterschiedlich viele Informationen festgehalten. Anhang F.1.9 zeigt einen Auszug der Definition eines Inode im Dateisystem Extended 4 und der darin verfügbaren Informationen.

Entsprechend ist zu sehen, dass für jede Datei die Zeiten der letzten Bearbeitung (modification), der letzten Änderung der Eigenschaften (change), des letzten lesenden Zugriffs (access) und des Zeitpunkts des Löschens der Datei festgehalten werden. Um die Aufzeichnung entsprechender Aktualisierungen der Zeitstempel zu verhindern, können maßgeblich zwei Methoden genutzt werden (vgl. [81]). Zum einen kann das Dateisystem unter Deaktivierung des Updates der Zeitstempel neu eingebunden werden:

```
# Deaktivierung Update Zugriffszeit
$ mount -o noatime,remount /dev/sda /
# Deaktiverung Update aller Zeitstempel
$ mount -o ro,remount /dev/sda /
```

Eine andere Möglichkeit ist die Nutzung des Hilfsprogrammes `touch`, mit dem einzelne Zeitstempel gezielt beeinflusst werden können.

Sollen Spuren sicher gelöscht werden, muss der entsprechende Bereich auf der Festplatte überschrieben werden; bei Festplatten bzw. magnetischen Speicherverfahren eröffnete

sich eine Diskussion, wie oft ein Datenträger überschrieben werden muss, damit auch mittels aufwändiger Verfahren wie Magnetic Force Microscopy (MFM) keine Daten mehr zurückgewonnen werden können. Gutmann hatte in einer Veröffentlichung im Rahmen des USENIX Security Symposiums im Jahre 1996 dargelegt, dass für eine sichere Löschung von Daten die entsprechenden Festplattenbereiche bis zu 35 mal überschrieben werden müssen [184], [185]. Diese These wurde in den letzten Jahren kritisch betrachtet und widerlegt (vgl. z.B. [393], [342]), wobei die genutzten Methoden und das Vorgehen durch Gutmann in Frage gestellt wurden; auf der anderen Seite ergänzte Gutmann seine Ausführungen, dass diese für moderne Datenträger mit sehr hohen Speicherdichten nicht mehr zutreffend sind (vgl. [184]). Auch ein ausgeschriebener Wettbewerb, bei dem den professionellen Anbietern von Datenrettungsdiensten die Aufgabe gestellt wurde, die Daten einer einmalig mit Nullen überschriebenen Festplatte wiederherzustellen, blieb unerfüllt [370]. Es kann daher davon ausgegangen werden, dass ein einmaliger Löschvorgang ausreichend ist. Beachtet werden muss jedoch auch, dass weitere Spuren in den Auslagerungsdateien des Arbeitsspeichers (swap) existieren können, die ggf. gelöscht werden müssen. Angemerkt sei weiterhin, dass hierbei immer die Gefahr verbleibt, dass das Logging auf einem zentralen Server stattfindet und der Angreifer ohne entsprechenden Zugriff auf dieses System keine Möglichkeiten hat, seine Spuren zu verwischen.

Die Durchführung eines Angriffes bedarf immer mehrerer Schritte, wobei zu Beginn die Analyse der Zielumgebung, die Identifizierung von Schwachstellen sowie das Erlangen eines Fernzugriffes stehen. Insbesondere diese Schritte fallen bei zielgerichteten Angriffsverfahren und der Nutzung von Social Engineering weg, so dass deutlich weniger Detektionsmöglichkeiten zur Verfügung stehen. Hier kann direkt in die weiteren Schritte, der Erlangung von administrativen Rechten sowie der Manipulation der Systemumgebung eingestiegen werden, bzw. sogar direkt mit der Ausschleusung von Daten begonnen werden.

2.4 Zusammenfassung

Dieses Kapitel beschreibt, welche Anforderungen von Unternehmen an die moderne Kommunikation gestellt werden. Richtwerte für die Benutzung von Diensten und technischen Ausstattungen und Anbindungen der Unternehmen werden gegeben, um später als Grundlage für die Anforderungen eines Sicherheitssystems zu dienen. Die Bedrohungen, welchen die Kommunikationsstrukturen der Unternehmen ausgesetzt sind, werden anhand der Grundschutzkataloge des BSI analysiert. Hierdurch können die vorliegenden Gefährdungen zusammengefasst werden. Da die Bedeutung des Innentäters in der Literatur strittig diskutiert wird, dieser jedoch im vorliegenden Szenario erhebliche Konsequenzen für die zu treffenden Schutzmaßnahmen haben kann, wird dessen Rolle ausführlich anhand von jüngeren Studien und Statistiken diskutiert. Nachdem die Bedrohungen identifiziert sind, erfolgt eine Analyse der Durchführung eines Angriffes. Eine Auswertung von Taxonomien stellt eine umfassende Betrachtung der möglichen Angriffsarten und -vorgehensweisen sicher. Da deren genauer Ablauf von entscheidender Bedeutung für die benötigten Detektionsmöglichkeiten ist, werden die einzelnen Schritte ausführlich

betrachtet und diskutiert. Das Ergebnis kann später als Grundlage für das Design einer Architektur herangezogen werden, um entsprechende Angriffsschritte zu erkennen.

Es gibt zurzeit kein einzelnes Sicherheitsprodukt, das einen ausreichenden Schutz gegen individuell angepasste Schadprogramme bietet.

Bundesamt für Sicherheit in der Informationstechnik

3 Anforderungen an ein IDS der nächsten Generation*

Im nachfolgenden Kapitel (vgl. Abbildung 3.1) werden auf Basis des vorliegenden Szenario und der zugehörigen Bedrohungsanalyse (vgl. Kapitel 2) sowie der bereits im Rahmen der Motivation identifizierten Notwendigkeiten (vgl. Kapitel 1) die Anforderungen an ein Sicherheitssystem zur Ein- und Ausbrucherkennung der nächsten Generation vorgestellt. Ziel des Kapitels ist die Erstellung eines Kriterienkataloges, welcher zum einen zur Bewertung der verfügbaren Systeme und Ansätze genutzt werden kann und weiterhin die Designkriterien für die zu entwickelnde Architektur repräsentiert. Hierbei erfolgt eine Gruppierung der Anforderung in zwei Bereiche, zum einen Anforderungen aus Sicht des Nutzers, zum anderen Anforderungen hinsichtlich der Architektur des Sicherheitssystems. Während die Nutzerforderungen insbesondere die Einsetzbarkeit des Systems in der Praxis sicherstellen, dienen die architekturellen Forderungen der Sicherstellung der Detektionsfähigkeit der identifizierten Bedrohungen. Wo zutreffend, werden die zu den jeweiligen Anforderungen gehörigen Maßnahmen der Grundschutzkataloge referenziert (vgl. Kapitel 2.2.1 sowie Anhang F.1.1).

3.1 Forderungen aus Nutzersicht

Auf Basis der evaluierten Bedrohungen sowie der allgemeinen Anforderungen des Szenarios werden die Anforderungen an ein Sicherheitssystem zum Einsatz in der aufgezeigten

*Dieser Abschnitt enthält eine Zusammenfassung von Teilen des Artikels „Towards Next-Generation Intrusion Detection“, Proceedings of the 3rd International Conference on Cyber Conflict (ICCC), IEEE, 2011.

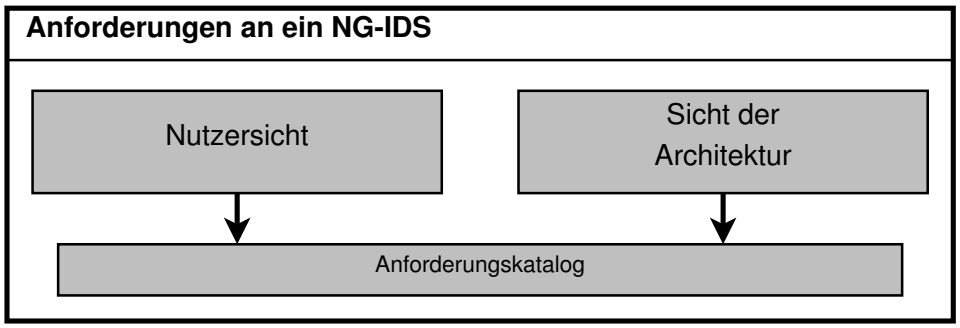


Abbildung 3.1: Aufbau von Kapitel 3.

Umgebung abgeleitet.

Für rohstoffarme Länder wie Deutschland ist der wirtschaftliche Erfolg maßgeblich abhängig vom Know-how um Verarbeitungsprozesse und technische Innovationen. Der Schaden, welcher durch Wirtschaftsspionage und Konkurrenzausspähung jährlich in Deutschland entsteht, liegt bei ca. 2.8 Mrd. € [104]. Die Sicherheit der anfallenden Daten und Informationen ist daher elementar. Maßgeblich gilt es bei der Kommunikation den Dreiklang Integrität, Authentizität und Vertraulichkeit der übertragenen Daten zu gewährleisten. Dies muss durch den geeigneten Einsatz von Verschlüsselungs- und Authentisierungsmaßnahmen sichergestellt werden (vgl. Maßnahme M 5.68).

Detektion von Angriffen und Durchführung von Gegenmaßnahmen Aufgrund immer wieder auftretender Programmierfehler in der Software und dem damit verbundenen Angriffspotential sowie der Gefahr von Datenverlust reicht die Durchführung solcher Maßnahmen jedoch nicht aus, um den Schutz der Systeme bzw. Daten zu garantieren. Vielmehr müssen insbesondere die Übertragungswege permanent auf das Vorhandensein und den Versuch von Angriffen hin überwacht werden (vgl. Maßnahmen M 5.71, M 4.81, M 5.8).

Anforderung 1 (Detektion von Angriffen)**.** *Angriffe gegen die Plattformen, Dienste, Kommunikation und Infrastruktur eines Unternehmens müssen zuverlässig erkannt werden.*

Durch die Geschwindigkeit, mit welcher sich neue Schadprogramme verbreiten können oder mit welcher Netze nach Ausnutzung einer Schwachstelle infiltriert werden können, wird eine automatische Reaktion bei Detektion eines entsprechenden Ereignis erforderlich. In kleinen Unternehmen kann eine manuelle Auswertung aufgrund fehlenden IT-Personals keine zeit- und fachgerechte Reaktion sicherstellten, in großen Konzernen wird eine ausreichende manuelle Reaktion regelmäßig durch die auftretenden Datenmengen und den damit verbundenen Alarmen verhindert, so dass auch hier eine automatisierte Reaktion ermöglicht werden muss (vgl. Maßnahmen M 4.98, M 4.345, M 5.71).

Anforderung 2 (Unterbinden erkannter Angriffe)**.** *Das System muss eine optional wählbare, automatische Reaktion bei der Erkennung von Angriffen anbieten.*

Erkennen von Innentätern Zahlreiche Firmen, insbesondere in rohstoffarmen Ländern wie Deutschland, ziehen ihr wirtschaftliches Wachstum aus ihren Erfindungen, aus dem Wissen und der Optimierung von Produktionsprozessen und ähnlichem. Der Verlust vertraulicher Daten kann schnell den wirtschaftlichen Ruin eines Unternehmens bedeuten (vgl. z.B. [82]) und muss wirksam unterbunden werden (vgl. Maßnahme M 4.345).

Anforderung 3 (Erkennen von Innentätern)**.** *Aktivitäten von Innentätern müssen erkannt werden können.*

Durch seine autorisierte Stellung geht von einem Innentäter eine erhebliche Gefahr bzgl. Datenverlust aus. Der Abfluss sensitiver Informationen ist insbesondere auch echtzeitkritisch. Eine reine Detektion ist regelmäßig nicht genug, da der Verlust von Daten insbesondere Know-how-Verlust und somit den Verlust von Wettbewerbsvorteilen,

aber auch Reputationsverlust durch z.B. den unzureichenden Umgang mit vertraulichen Kundendaten bewirken kann. Die Datenskandale der letzten Zeit und der durch den Reputationsverlust resultierende wirtschaftliche Schaden zeigen, dass eine automatisierte Reaktion erforderlich ist. Da Systeme zur Einbruchs- und Missbrauchsdetektion genauso wie Virenscanner immer unter Fehlalarmen leiden (vgl. auch Kapitel 4.4 und Abbildung 4.9), kann bei einer automatisierten Reaktion ein fehlerhaftes Unterbrechen bspw. einer korrekten Datenübertragung auf einen externen Server jedoch nicht vollständig ausgeschlossen werden. Mit Hinblick auf den höheren Schaden, der durch eine ausbleibende Reaktion bei einem echten Datenabfluss im Vergleich zu einer (seltenen) fehlerhaften Unterbrechung einer korrekten Übertragung entsteht, muss ein entsprechendes Sicherheitssystem eine *optionale*, automatisierte Reaktion anbieten (vgl. Maßnahmen M 3.345, M 5.71).

Systemintegration Hinsichtlich der Systemkomplexität des Sicherheitssystems selbst, jedoch auch aus Sicht der vorhandenen Infrastruktur ist eine möglichst transparente und autarke Integration erforderlich, um effektiv in die Infrastrukturen der betroffenen Unternehmen eingebunden werden zu können. Insbesondere muss auf Änderungen und Anpassungen der bereits bestehenden Strukturen verzichtet werden, da diese regelmäßig mit hohen Kosten verbunden sind oder die Kompatibilität zu anderen benötigten Systemen einschränken (vgl. Kapitel 4.6.3).

Anforderung 4 (Transparente Integration)**.** *Das Sicherheitssystem muss ohne Änderung der vorhandenen Infrastruktur integrierbar sein.*

Automatisierung Kleine und mittelständische Unternehmen haben typischerweise keine hohen Budgets für Ausgaben im IT-Sicherheitsbereich. Eigene IT-Abteilungen wie in großen Konzernen existieren nicht und die Absicherung der IT erfolgt meist nur rudimentär. Auf der anderen Seite wird jedoch insbesondere in kleinen und mittelständischen Unternehmen die Gefahr, Ziel eines Spionageangriffs zu werden, deutlich unterschätzt [104]. Mehrere aktuelle Studien, welche zahlreiche Vorfälle der Industriespionage und des Datenabflusses ausgewertet haben, zeigen jedoch dass gerade diese Unternehmen im Fokus der Angreifer stehen und am häufigsten betroffen sind. Der Betrieb muss daher ohne die Überwachung von IT-Personal automatisiert erfolgen, ausgelöste Ereignisse müssen weitestgehend automatisiert abgearbeitet werden. Dies unterstützt insbesondere die Nutzung des Systems durch kleine Unternehmen ohne eigene IT-Abteilung. Andererseits können Optimierungen von Betriebsparametern und manuelle Anpassung der Konfiguration dazu genutzt werden, das System für die Zielumgebung zu optimieren und somit die Fehlalarmraten zu minimieren. Dies ist insbesondere für Unternehmen mit eigener IT-Abteilung oder für sensible und besonders zu schützende Netze interessant. Eine entsprechende, manuelle Anpassung der Systemparameter muss daher optional ebenfalls möglich sein.

Anforderung 5 (System-Autarkie und Wartungsfreiheit)**.** *Das System muss so aufgebaut sein, dass es ohne umfangreichen Konfigurationsaufwand einsetzbar ist. Interaktio-*

nen mit dem Nutzer müssen auf ein Minimum beschränkt sein und durch den Endanwender entscheidbar sein. Weiterhin muss eine manuelle Anpassung der Konfiguration und Systemparameter optional möglich sein.

Echtzeitfähigkeit Die Abhängigkeit von schnellen Kommunikationswegen und weltweitem Datenaustausch ist vor dem Hintergrund der zunehmenden Globalisierung (vgl. [389], [148]) unumgänglich. Permanente Internetverbindungen gehören daher seit Jahren zu den Standardkommunikationsmitteln von Unternehmen [111] und stellen eine erhebliche sicherheitsrelevante Komponente dar (vgl. z.B. [234], [151]). Die hohen wirtschaftlichen Schäden, die mit dem Verlust von Betriebsgeheimnissen verbunden sind, erfordern daher eine zeitnahe Reaktion, wenn sicherheitsrelevante Vorgänge detektiert wurden (vgl. Maßnahme M 3.345).

Definition (Echtzeit)**.** *Echtzeit bedeutet, dass das Ergebnis einer Berechnung innerhalb eines gewissen Zeitraumes garantiert vorliegt d.h. bevor eine bestimmte Zeitschranke erreicht ist [395].*

Anforderung 6 (Nahe-Echtzeit Auswertung)**.** *Die Analyse des Datenverkehrs muss in Nahe-Echtzeit erfolgen. Insbesondere reichen Offline-Evaluationen nicht aus.*

Die Anforderungen 5 und 6 müssen mit Hinblick auf die Verhinderung von ungewolltem Datenabfluss gemeinsam realisiert werden: Ein Sicherheitssystem **muss** zum frühest möglichen Zeitpunkt reagieren, um einen entsprechenden Datenabfluss verhindern zu können. Gerade ein automatisierter Betrieb ist daher zwingend erforderlich, da die heutigen, breitbandigen Internetanschlüsse eine Übertragung großer Datenmengen innerhalb kurzer Zeit erlauben. Automatisiert bedeutet hier, dass bei Detektion eines bösartigen Ereignisses unmittelbar Gegenmaßnahmen durch das System ergriffen werden können, wenn es entsprechend konfiguriert ist. Bspw. kann hier eine IP-Adresse gesperrt werden, welche als Angreifer identifiziert wurde, oder eine Datenverbindung beendet werden, welche als Ausschleusung von Informationen erkannt wurde.

Erweiterbarkeit Der kontinuierlichen Erhöhung der Datenrate der Internetanschlüsse muss durch eine entsprechende Verarbeitungsleistung eines Sicherheitssystems Rechnung getragen werden können. Die Umsetzung des Systems muss daher insbesondere möglichst effizient erfolgen. Reicht die Leistung des Systems nicht mehr zu Analyse des gesamten Datenverkehrs aus, muss eine einfache Erweiterungsmöglichkeit gegeben sein.

Anforderung 7 (Erweiterbarkeit)**.** *Um den vorhandenen und stetig wachsenden Bandbreiten gerecht zu werden, muss das Sicherheitssystem so ausgelegt sein, dass eine nachträgliche Erweiterung möglich ist, ohne dass die bereits integrierten Komponenten wieder ausgetauscht werden müssen.*

Einsatz in verschlüsselten Umgebungen Die sich an unterschiedlichen Standorten befindlichen Netze eines Konzerns werden typischerweise mit verschlüsselten Layer-3

VPNs verbunden (vgl. auch Maßnahme M 5.68), für den mobilen Zugriff auf Firmendaten sowie administrative Arbeiten kommen weiterhin Protokolle wie SSH und TLS bzw. SSL zum Einsatz (vgl. Kapitel 2.1). Darüber hinaus nimmt jedoch auch in der privaten Internetnutzung der Anwender der Anteil abgesicherter Kommunikation stetig zu. Dem steigenden Einsatz von Verschlüsselung bei Webseiten und anderen Diensten muss entsprechend Rechnung getragen werden: Nachdem jahrelang bestehende Sicherheitslücken im Umgang mit Session-Cookies durch das One-Click Plugin *Firesheep* [83] für jeden Nutzer auch ohne Fachkenntnisse ausnutzbar wurden und Sitzungen, bspw. der weit verbreiteten Dienste *Facebook*, *Twitter* oder des Maildienstes *Hotmail*, durch einfaches Anklicken eines Eintrags im Browser-Fenster durch einen Angreifer übernommen werden konnten, geriet die Diskussion um die Sicherheit von Verbindungen zu Webdiensten in eine vielbeachtete öffentliche Diskussion und löste entsprechende Reaktionen von Seiten der Dienstanbieter aus, ihre Webangebote auch für abgesicherten Zugriff anzubieten. In diesem Kontext erfreuen sich auch Addons zur Absicherung der Zugriffe steigender Beliebtheit, zum Beispiel zum Erzwingen eines verschlüsselten Kommunikationsaufbaus, sobald eine Website entsprechendes anbietet. Künftig wird HTTP Strict Transport Security (HSTS) eine Nutzung vorhandener Verschlüsselungen einer Website erzwingen, sofern dies der genutzte Browser unterstützt (vgl. z.B. [196], [329], [313], [363] und [11]).

Dadurch, dass im Rahmen der Nutzung von firmeneigenen VPNs bzw. IPsec-Strecken die zugehörigen Schlüssel typischerweise verfügbar sind, bzw. im Falle der Nutzung von VPN-Konzentratoren die Verschlüsselung oftmals nur zwischen den verschiedenen Firmenstandorten durchgeführt wird (vgl. z.B. [160] oder [382]) und die Daten in den jeweiligen lokalen Netzen wieder unverschlüsselt vorliegen (vgl. Abbildung 3.2), muss hier nicht zwingend der *verschlüsselte* Datenverkehr analysiert werden, da auch auf die unverschlüsselten Daten zugegriffen werden kann. Hier kann entsprechend eine Untersuchung des Datenverkehrs am Übergang zwischen VPN-Appliance und dem internen Netz erfolgen, ohne Einschränkungen durch die vorherige Verschlüsselung zu haben. Andererseits müssen verschlüsselte Verbindungen Dritter, bspw. SSH- oder TLS-Verbindungen der Mitarbeiter zwingend im verschlüsselten Zustand untersucht werden, da hier dem Sicherheitssystem *kein* Schlüsselmaterial zur Verfügung gestellt werden kann bzw. wird. Da hiermit aus letzterem jedoch eine Notwendigkeit der Analyse des verschlüsselten Datenverkehrs entsteht, muss dies als entsprechende Forderung formuliert werden. Andererseits folgt hieraus auch, dass durch das System ebenfalls die VPN-Verbindungen der Firma analysiert werden können.

Aus der Forderung des Einsatzes von Verschlüsselung (M 5.68) sowie der zunehmenden Nutzung verschlüsselter Verbindungen im privaten Umfeld, der Gefahr durch Innentäter (vgl. Kapitel 2.2.2) sowie den Maßnahmen M 4.345 und M 5.71 folgt weiterhin die Notwendigkeit einer durchgehenden Untersuchung des verschlüsselten Datenverkehrs.

Anforderung 8 (Einsatz in verschlüsselten Netzen)**.** *Eine Detektion von Angriffen oder Datenabfluss muss möglich sein, auch wenn die Verbindung durch Verschlüsselung abgesichert ist.*

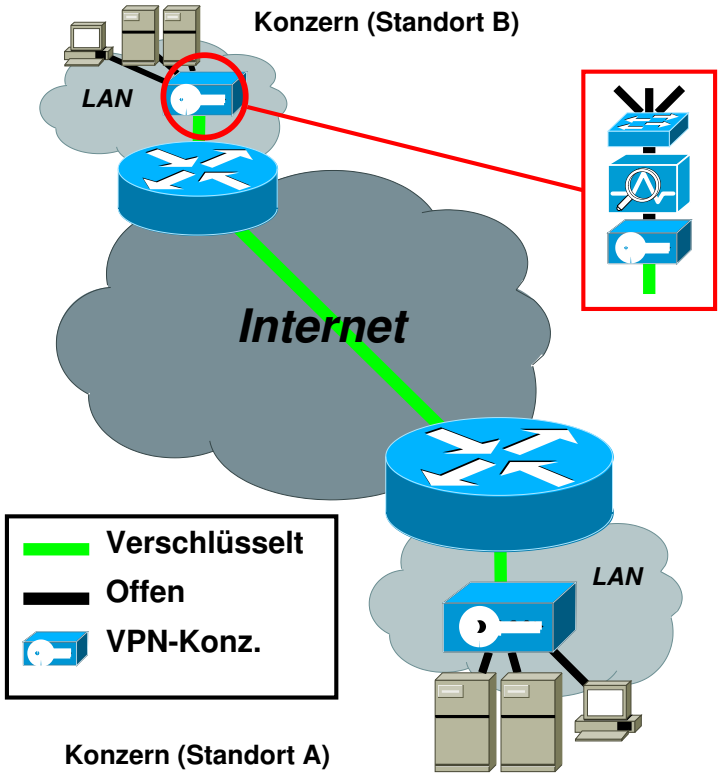


Abbildung 3.2: VPN-Verbindung zwischen verschiedenen Konzernniederlassungen. Die Nutzung von VPN-Konzentratoren ermöglicht eine Untersuchung des Datenverkehrs vor der Weiterverteilung im Firmen-LAN des jeweiligen Standortes.

Rechtskonformität Um ein Sicherheitssystem in einer Unternehmensumgebung einsetzen zu können, muss eine entsprechende Rechtskonformität sichergestellt sein. Werden Gesetze bzgl. des Datenschutzes beim Einsatz des Systems mißachtet, kann dies zu einem Verwertungsverbot und insbesondere auch zur Strafanzeige des Betreibers des Systems führen (vgl. Kapitel 4.6.5). Eine entsprechende Einhaltung des geltenden Rechts, insb. des Bundesdatenschutzgesetz (BDSG), Telekommunikationsgesetz (TKG) und der Telekommunikations-Überwachungsverordnung (TKÜV), ist daher unerlässlich.

Anforderung 9 (Rechtskonformer Systemeinsatz)**.** *Das Sicherheitssystem muss hinsichtlicht der Datenerhebung und -verabreitung rechtskonform sein.*

3.2 Forderungen an die Architektur

Auf Basis der identifizierten Bedrohungen und der Angriffsanalyse (vgl. Kapitel 2), sowie den bereits motivierten, offenen Punkten der Einbruchserkennung (vgl. Kapitel 1) werden nun die architektonischen Anforderungen an ein Sicherheitssystem zur Ein- und Ausbruchserkennung der nächsten Generation abgeleitet.

Verhaltensbasierte Evaluation Zahlreiche Faktoren erfordern den Einsatz von verhaltensbasierten Detektionsverfahren, da signaturbasierte Ansätze hier keine Analysemöglichkeiten haben:

- *Zielgerichtete Angriffe* mit speziell auf das Objekt zugeschnittenen Schadroutinen. Diese können aufgrund der nicht bekannten Signaturen nicht durch wissensbasierte Systeme erkannt werden.
- *Social Engineering-Verfahren* bringen den Nutzer dazu, Schadprogramme oder komprimittierende Aktionen selbst zu installieren bzw. auszulösen, ohne dass dem Angegriffenen dies jedoch bewusst wird. Hierdurch wird der Schadcode direkt im Zielsystem bzw. -netz installiert, ohne eine Detektionsmöglichkeit durch ggf. vorhandene, traditionelle Sicherheitssysteme.
- *Innentäter* können aufgrund ihrer Autorisierung und des vorhandenen Wissens Aktionen durchführen, die ebenfalls nicht durch signaturbasierte Verfahren erfasst werden können.
- *Unbekannte Schadprogramme und Angriffsverfahren*, für welche keine Signaturen vorhanden sind, können nicht erkannt werden.
- *Verschlüsselung des Datenverkehrs*, wodurch eine Analyse des Payloads verhindert wird. Dies verhindert ebenfalls den Einsatz signaturbasierter Verfahren.

Eine Untersuchung muss daher zwingend mittels einer verhaltensbasierten Evaluation erfolgen. Signaturbasierte Verfahren können lediglich optional als Ergänzung hinzugefügt werden, bspw. in Form von Agenten auf den jeweiligen Hosts. Hierbei muss jedoch beachtet werden, dass durch die Hinzunahme entsprechender Agenten eine Gefährdung der Evaluation des IDS entstehen kann, falls einer der Hosts kompromittiert ist. Mit steigender Anzahl von involvierten Hosts erhöht sich entsprechend die Gefährdung des Systems.

Anforderung 10 (Verhaltensbasierte Evaluation)**.** *Das System zur Ein- und Ausbruchserkennung muss verhaltensbasierte Ansätze zur Durchführung der Analyse verwenden.*

Verzicht auf eine Lernphase Die Nutzung eines verhaltensbasierten Systems bedarf typischerweise der Durchführung einer länger andauernden Lernphase in der produktiven Einsatzumgebung. Dies stellt eine massive Gefährdung für das Sicherheitssystem dar, da während dieser Phase im Netz befindliches, bösartiges Verhalten mit als Normalverhalten erlernt wird und im späteren Einsatz nicht erkannt werden kann (vgl. Anhang F.2.9).

Ein Verzicht auf eine Lernphase ist daher von Bedeutung. Ist dies nicht vollständig möglich, muss die Lernphase mit geeigneten Verfahren soweit wie möglich verkürzt bzw. vor Einflussnahme geschützt werden.

Anforderung 11 (Verzicht auf eine Lernphase)**.** *Das Sicherheitssystem soll ohne die Notwendigkeit einer Lernphase direkt in Betrieb genommen werden können. Ist dies nicht vollständig möglich, muss die Lernphase mittels geeigneter Mechanismen möglichst kurz gehalten und vor äußerer Einflussnahme geschützt werden.*

Payload-unabhängige Evaluation Eine Evaluation des Payloads muss aus verschiedenen Gründen vermieden werden:

- Nicht-Verfügbarkeit des Payloads durch Verschlüsselung.
- Hoher Aufwand durch die Datenmengen.
- Rechtliche Restriktionen der Untersuchung.

Eine besondere Herausforderung stellt die Angriffserkennung in Bezug auf die (nicht-) Nutzung der Payload-Informationen dar. Während diese einerseits aus den oben genannten Gründen nicht als verfügbar vorausgesetzt werden können und somit nicht als Forderung gesetzt werden können, sind Angriffe oberhalb der Ebene 4 wiederum primär nur mittels der Auswertung des Payloads zu erkennen. Um auf Basis dieser Konfliktsituation weiterhin eine Detektion von Angriffen auf der Applikationsebene zu ermöglichen, ohne den Payload einzubeziehen, muss eine Analyse auf Basis von sekundären Faktoren erfolgen.

Anforderung 12 (Verzicht auf DPI)**.** *Die Untersuchung des Datenverkehrs muss auf eine Analyse des Payloads (DPI) der Netzpakete verzichten.*

Netzbasierte Integration Hostbasierte Verfahren ermöglichen den Zugriff auf sämtliche am Host verfügbaren Daten sowie die Evaluation umfangreicher Systeminformationen, erfordern jedoch einen hohen administrativen Aufwand und können keine verteilte Angriffe erkennen. Weiterhin stellt eine mögliche Kompromittierung eines Hosts eine Gefährdung respektive Beeinflussungsmöglichkeit des Sicherheitssystems dar, die schwerer detektierbar sein kann, als im Falle eines netzbasierten Systems ohne Agenten. Insbesondere kann die Integration eines Network-based Intrusion Detection System (NIDS) in transparenter Art erfolgen, bspw. durch Nutzung einer Brücke auf Ebene 2 des OSI-Referenzmodells oder der Integration eines TAP-Device, so dass ein hiermit genutztes Sicherheitssystem nicht angegriffen werden kann.

Neben diesen Abwägungen spricht insbesondere noch die schnellere Reaktionsmöglichkeit für eine netzbasierte Überwachung: Erfolgt eine Analyse direkt an zentralen Netzkomponenten, kann eine unmittelbare Reaktion bei einer entsprechenden Detektion von böswilligen Vorgängen stattfinden. Wird ein hostbasiertes System für die Untersuchung genutzt besteht die Gefahr, dass aufgrund einer hohen Auslastung des Systems nicht genug Ressourcen zur Verfügung stehen und Verbindungen nicht komplett evaluiert werden. Durch die dedizierte Nutzung eines NIDS rein für Analysezwecke, kann dieser Gefahr bei entsprechender Evaluierung der Anforderungen der Netzanbindung vorgebeugt werden[1]. Ebenfalls kann eine fehlerhafte Konfiguration des Hostssystems zu Fehlfunktionen und der Nichtdetektion von Angriffen führen; insbesondere erhöht sich

[1] Beachte hierbei jedoch auch Kapitel 4.4. Oftmals decken sich die Herstellerangaben *nicht* mit den real erzielbaren Leistungswerten der Systeme. Eine entsprechende Evaluation in der Zielumgebung ist daher erforderlich.

die Gefahr einer entsprechenden Fehlkonfiguration mit der Anzahl von Systemen. Aufgrund der Diversifikation der Systene und Anwendungen kann der hierbei erforderliche, administrative Aufwand nur bedingt eingeschränkt werden. Die Nutzung eines zentralen Analysesystems reduziert den Aufwand insbesondere in großen Netzen somit erheblich.

Bei der Betrachtung bestehender Strukturen muss auch der Aufwand für die Integration eines Sicherheitssystems berücksichtigt werden. Müssen Hostkomponenten installiert werden, kann dies zu umfangreichen Arbeiten führen, ggf. müssen bestehende Software-Konfigurationen angepaßt werden oder Systeme können aufgrund von Inkompatibilitäten nicht mit integriert werden, wodurch ein Schutz für diese Komponenten entfällt. Die Nutzung eines netzbasierten Systems ermöglicht die transparente Integration in eine bestehende Infrastruktur, ohne dass Anpassungen oder Änderungen erforderlich sind. Dies lässt sich insbesondere auch sehr schnell, mit einer minimalen Unterbrechung der Konnektivität, umsetzen.

Hostbasierte Agenten können als zusätzliche Informationsquelle hinzugezogen werden, sollten jedoch aufgrund der für das IDS ausgehenden Gefährdungen durch einen kompromittierten Rechner sowie wegen des hohen administrativen Aufwands hostbasierter Verfahren vermieden werden.

Die Forderung einer netzbasierten Integration entspricht der Nutzerforderung einer transparenten Integration.

Verteilung Insbesondere mit Hinblick auf die steigenden Datenmengen und zunehmend komplexen Analysen wird eine Parallelisierbarkeit erforderlich. Dies ermöglicht sowohl eine Integration in Netze mit hohen Datenraten, als auch eine Erweiterbarkeit bei einem Ausbau der Infrastruktur und somit steigenden Anforderungen an die Evaluation.

Die Anforderung einer Verteilbarkeit entspricht der Nutzeranforderung der Erweiterbarkeit.

Automatische Reaktion Eine effiziente Gefahrenabwehr muss automatisiert und in nahe-Echtzeit erfolgen. Aufgrund der hohen Datenmengen ist eine manuelle Interaktion mit einem Sicherheitssystem höchsten für einzelne, nicht regelmäßig auftretende Meldungen möglich. Dies gilt jedoch auch nur, wenn entsprechendes IT-Personal zur Verfügung steht und zeitnah reagieren kann, bspw. in größeren Firmen mit eigener IT-Abteilung. In kleinen Unternehmen, welche typischerweise keine eigenen IT-Kräfte haben, ist dies regelmäßig nicht möglich. Für die Ausbruchsdetektion muss die Anforderung einer automatischen Reaktion noch verschärft werden, da hier bereits mit dem Abfluss von Daten ein hoher finanzieller, als auch Image-Schaden entstehen kann. Hier kommt es also auf eine unmittelbare Reaktion bei Erkennung einer Gefährdung an, bspw. das Unterbrechen einer ausgehenden Datenverbindung.

Eine automatisierte, nahe-Echtzeit-fähige Reaktion muss daher von einem Sicherheitssystem der nächsten Generation ermöglicht werden.

Diese Anforderung entspricht der Nutzeranforderung der Automatisierung.

Tabelle 3.1: Kriterienkatalog für das Sicherheitssystem, getrennt nach Nutzersicht und architektonischen Anforderungen. Die letzten drei Forderungen an die Architektur sind auf äquivalente Nutzerforderungen abbildbar und werden im weiteren Verlauf nicht extra aufgeführt.

	Nutzersicht
1	Detektion von Angriffen
2	Unterbinden erkannter Angriffe
3	Erkennen von Innentätern
4	Transparente Integration
5	System-Autarkie / geringe Wartung
6	Nahe-Echtzeitauswertung
7	Erweiterbarkeit
8	Einsatz in verschlüsselten Netzen
9	Rechtskonformer Systemeinsatz
	Architektur-Sicht
10	Verhaltensbasierte Evaluation
11	Verzicht auf eine Lernphase
12	Verzicht auf DPI
4	*Netzbasierte Integration*
7	*Verteilbarkeit*
5	*Automatische Reaktion*

Tabelle 3.1 fasst alle Punkte des Anforderungskatalogs für ein Sicherheitssystem im vorgestellten Szenario nochmals zusammen.

3.3 Zusammenfassung

Dieses Kapitel leitet den Anforderungskatalog an ein Sicherheitssystem für Ein- und Ausbruchserkennung in modernen IT-Umgebungen ab. Grundlage hierfür sind die identifizierten Bedrohungen und die durchgeführte Angriffsanalyse des Kapitels 2. Die Anforderungen werden hierbei getrennt nach Nutzersicht in Kapitel 3.1, welche die praktische Einsetzbarkeit des Systems betreffen und architektonischen Forderungen für die Sicherstellung der Detektion von bösartigen Ereignissen gem. den Anforderungen in Kapitel 3.2 aufgestellt.

Es ist ein großes Glück, wenn wir den Hindernissen, aus denen wir lernen können, möglichst früh begegnen.

Winston Churchill

4 State-of-the-Art der Intrusion Detection

Nachfolgend wird eine umfassende Betrachtung von State-of-the-Art IDSs sowie Verfahren und Techniken zur Einbruchserkennung vorgenommen. Abbildung 4.1 zeigt die Einordnung dieses Kapitels. Ziel ist es, State-of-the-Art Systeme und Forschungsansätze der Ein- und Ausbruchserkennung zu analysieren und einen Überblick der derzeitigen Möglichkeiten und Schwachstellen der Systeme zu geben. Zunächst erfolgt eine Definition der Einbruchserkennung (Kapitel 4.1). Um einen umfassenden Überblick aller Teilbereiche der Ein- und Ausbruchserkennung zu erhalten, erfolgt anschließend eine Klassifizierung anhand von Taxonomien (Kapitel 4.2) sowie eine kurze Betrachtung der Funktionsweisen (Kapitel 4.3). Die Problematik des Vergleichs von Systemen zur Ein- und Ausbruchserkennung und deren Leistungsanalyse wird in Kapitel 4.4 behandelt. Kapitel 4.5 gibt anschließend einen Überblick der aktuellen State-of-the-Art Systeme aus Industrie und Wissenschaft, wobei eine Bewertung anhand des in Kapitel 3 aufgestellten Kriterienkataloges erfolgt. Um den identifizierten Schwachstellen bestehender System und Ansätze begegnen zu können, ist ein Verständnis notwendig, warum derzeitige Systeme entsprechende Schwierigkeiten aufweisen. Hierfür werden diese Herausforderungen in Kapitel 4.6 analysiert und abschließend eine Zusammenfassung der offenen Punkte gegeben.

4.1 Definition

Um den zahlreichen Angriffsmöglichkeiten entgegenzuwirken, reicht der Einsatz von Maßnahmen wie Firewalls, etc. nicht aus. Entsprechende Komponenten können zwar einen Schutz von Angriffen von Außen gegen Dienste innerhalb eines dadurch gesicherten Netzes bieten, jedoch bleiben zahlreiche Schwachstellen offen:

- Die Schutzmaßnahmen selbst können Programmier- oder Konfigurationsfehler haben, die angreifbar sind und hierdurch eine Umgehung der Schutzmaßnahmen ermöglichen können, Analysen der Zielumgebung zulassen oder sogar eine Kompromittierung des Systems ermöglichen, so dass dieses selbst für die weitere Angriffsdurchführung missbraucht werden kann (vgl. z.B. [283], [282], [281]).
- Dienste und Rechner, die aufgrund ihres Einsatzes von Außen erreichbar sein müssen, können nicht umfassend geschützt werden, bspw. Webserver.

Um einen entsprechenden Schutz eines Netzes vorhalten zu können, müssen daher drei Gebiete adressiert werden: *Prävention*, *Erkennung* und *Reaktion*. Im Bereich der

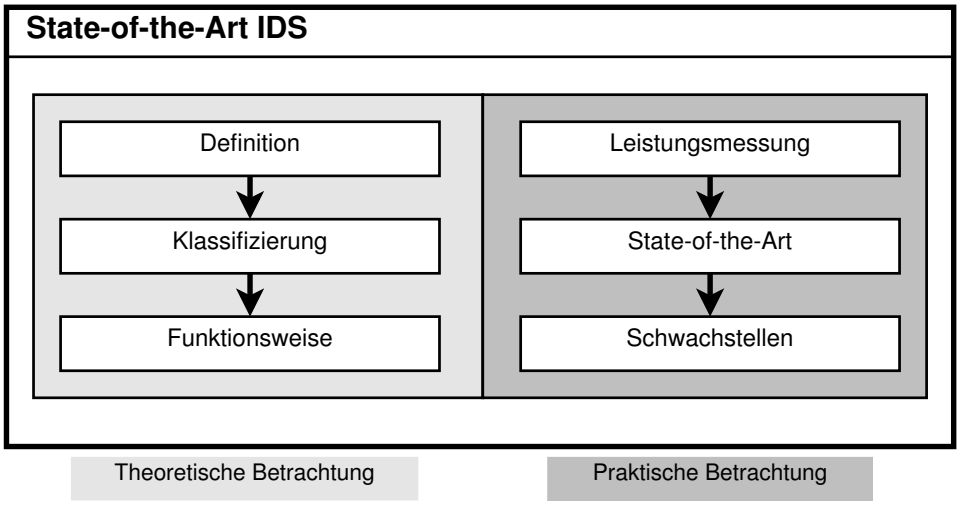


Abbildung 4.1: Aufbau von Kapitel 4. In der theoretischen Betrachtung von IDSs erfolgt zunächst eine Definition der Einbruchserkennung, gefolgt von einer Klassifizierung von Systemen. Anhand dieser erfolgt eine Betrachtung der Funktionsweisen der verschiedenen Systemarten. In der praktischen Analyse werden Möglichkeiten der Leistungsanalyse und des Vergleichs von IDSs eruiert, die Basis für eine Bewertung der folgenden State-of-the-Art Systeme sind. Die Schwachstellen der Systeme werden anschließend detailliert dargelegt.

Prävention sind insbesondere Firewallsysteme, korrekte Systemkonfiguration und sicherer Betrieb zu nennen, wobei zum sicheren Betrieb die Abschaltung nicht benötigter Dienste und Programme zählt, eine regelmäßige Systempflege und zeitnahes Einspielen verfügbarer Patches. Zur Erkennung von Sicherheitsvorfällen im Netz müssen entsprechende Detektionsverfahren eingesetzt werden, die im weiteren Verlauf vorgestellt und analysiert werden. Eine Bewertung von State-of-the-Art Produkten auf Basis des in Kapitel 2.1 aufgestellten Kriterienkataloges schließt das Kapitel ab. Reaktionen werden heutzutage oftmals mit im Rahmen der Erkennung vorgenommen, bspw. durch das Sperren von IP-Adressen nach der Detektion eines Angriffes.

Im Rahmen des IT-Grundschutzes und der Informationssicherheit definiert das BSI die Einbruchserkennung wie folgt:

Definition (Intrusion Detection)**.** *Als Intrusion-Detection wird die aktive Überwachung von Computersystemen und/oder -netzen mit dem Ziel der Erkennung von Angriffen und Missbrauch bezeichnet. Das Ziel von Intrusion-Detection besteht darin, aus allen im Überwachungsbereich stattfindenden Ereignissen diejenigen herauszufiltern, die auf Angriffe, Missbrauchsversuche oder Sicherheitsverletzungen hindeuten, um diese anschließend vertieft zu untersuchen. Ereignisse sollen dabei zeitnah erkannt und gemeldet werden.*

Als Intrusion-Detection-System wird eine Zusammenstellung von Werkzeugen bezeichnet, die den gesamten Intrusion-Detection-Prozess von der Ereigniserkennung über die

Auswertung bis hin zur Eskalation und Dokumentation von Ereignissen unterstützen [163].

Gemäß [99] wird ein System dann als IDS bezeichnet, wenn eine *automatische* Detektion erfolgt. Eine weitere, vereinfachte Definition von IDS lässt sich ebenfalls auf der Webpräsenz des BSI finden:

Definition (Intrusion Detection System)**.** *Ein Intrusion Detection System ist ein System zur Erkennung von Angriffen auf ein Rechnersystem oder Rechnernetz [150].*

Zahlreiche weitere Definitionen und Beschreibungen existieren in der Literatur, die sich jedoch nur geringfügig unterscheiden. Ein wichtiges Kriterium, in dem sich die Definitionen unterteilen lassen, ist jedoch die Berücksichtigung von *Ausbruchsversuchen.*
Das SANS-Institut gibt folgende Definition an [215]:

> Therefore, an Intrusion detection system (IDS) is a security system that monitors computer systems and network traffic and analyzes that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization.

Von besonderer Bedeutung ist hier der Systemmissbrauch und die von innen ausgehenden Angriffe; diese können zum einen durch Fehlverhalten von Nutzern oder absichtlichen Missbrauch von Systemen durch einen Innentäter erfolgen, andererseits kann eine Angriffsgefahr von Innen heraus zu externen Netzen durch sich ausbreitende Würmer, Botnetz-Aktivitäten, etc. entstehen.
Maßgeblich muss ein IDS somit Fähigkeiten in folgenden Bereichen bereitstellen:

- Einbruchs- und Angriffserkennung sowie
- Erkennung von Missbrauch.

Systeme zur Einbruchserkennung werden bereits seit den 80er Jahren erforscht. Anhang F.2.1 gibt einen Überblick über die Entwicklung und Notwendigkeit der Einbruchserkennung.
Nachfolgend werden verfügbare Klassifikationen untersucht, sowie eine Übersicht der grundlegenden Eigenschaften von Systemen zur Einbruchserkennung gegeben.

4.2 System-Klassifizierung

Ebenso wie bei der Angriffs-Klassifizierung (vgl. Kapitel 2.3), liegt im Bereich der Klassifizierung von IDSs keine allgemeingültige Taxonomie vor. Im Gegensatz zu den Einteilungen im Bereich der Angriffe liegen jedoch sehr viel weniger spezialisierte oder stark unterschiedliche Klassifizierungen vor, sondern meistens ist der Detaillierungsgrad hinsichtlich der Anzahl, etc. der Klassen unterschiedlich ausgeprägt. Im Folgenden werden einige Taxonomien vorgestellt und anschließend die wichtigsten Elemente kurz beschrieben.

Debar et al. haben mehrere Arbeiten im Bereich der Taxonomien von IDSs veröffentlicht, die hinsichtlich der genutzten Gruppen sehr weit verbreitet sind. Abbildung 4.2 zeigt die entsprechende Einteilung [115].

Die wichtigste und am häufigsten herangezogene Charakteristik von IDSs ist die **Detektionsmethode**: Hier existieren die beiden grundlegenden Konzepte der *wissensbasierten* und der *verhaltensbasierten* Detektion.

Bei *wissensbasierten Verfahren* (misuse detection, signaturbasiert, kontentbasiert) erfolgt die Erkennung eines Einbruchs durch die Festlegung des **negativen** Verhaltens. Hierbei werden Informationen über die möglichen Angriffe und Systemschwachstellen gespeichert und das IDS sucht nach Versuchen, diese mittels Schadprogrammen, Exploits, etc. auszunutzen. Somit kann eine Detektion nur erfolgen, wenn der ablaufende Angriff bzw. die eingesetzte Schadsoftware bekannt und geeignet im System repräsentiert ist. Insbesondere können keine unbekannten oder neuen Angriffe erkannt werden, ebenso wenig ist die Detektion von durch Innentäter durchgeführten Aktionen möglich, da diese meistens über legale Zugriffsmöglichkeiten verfügen und daher nicht gezwungen sind, (detektierbare) Exploits o.ä. einzusetzen. Zur Realisierung wissensbasierter Systeme lassen sich insbesondere vier Techniken einsetzen, signaturbasierte Analyse, Expertensysteme, Petrinetze sowie Zustandsübergangsdiagramme. Details zu den jeweiligen Verfahren sind in Anhang F.2.2 zu finden.

Bei *verhaltensbasierten Verfahren* (anomaly detection, kontextbasiert) wird auf Grundlage der Definition eines Verhaltensmodells, welches das **positive** Netz- bzw. Systemverhalten beschreibt, eine Angriffsdetektion auf Basis festgestellter Anomalien durchgeführt. Das Modell wird hierbei aus Referenzinformationen der Einsatzumgebung gewonnen, anschließend wird im Betrieb der gegenwärtige Zustand mit der Erwartung des Modells verglichen; wird ein festgesetzter Schwellwert überschritten, wird ein Alarm ausgelöst. Der Vorteil dieser Technologie liegt insbesondere darin, dass auch unbekannte Angriffe erkannt werden können, da keine Signaturen oder ähnliche Wissensrepräsentation notwendig ist. Aufgrund der verhaltensbasierten Analyse ist auch eine Detektion von Missbrauch von Rechten, also insbesondere Aktivitäten von Innentätern, prinzipiell möglich. Auf der anderen Seite leiden die entsprechenden Systeme bedingt durch ihre Arbeitsweise unter einer höheren Fehlalarmrate. Dies liegt u.a. daran, dass das vollständige Verhalten eines Systems nicht innerhalb einer Lernphasen erlernt werden kann, auch kann sich das Verhalten über die Zeit oder spontan ändern. Somit werden bspw. für Ereignisse, die zwar eine korrekte Nutzung des Systems darstellen, die jedoch in der Lernphase unberücksichtigt geblieben sind, entsprechende (falsche) Alarme ausgelöst. Zahlreiche verhaltensbasierte Systeme sind auf die Durchführung einer Lernphase vor dem eigentlichen Einsatz angewiesen, allerdings gibt es auch neuere Ansätze, die darauf verzichten können. Grundsätzlich muss daher beaufsichtigtes (*supervised*) und nicht-überwachtes (*unsupervised*) Lernen unterschieden werden. Weitere Verfahren sind bspw. statistischer Art oder Expertensysteme; Details zu den Lerntechniken finden sich in Anhang F.2.4.

Eine weitere, zur Klassifizierung genutzte Charakteristik ist die Art der **Reaktionen** eines IDS. Hierbei kann zwischen *passiven* und *aktiven* Reaktionen unterschieden werden.

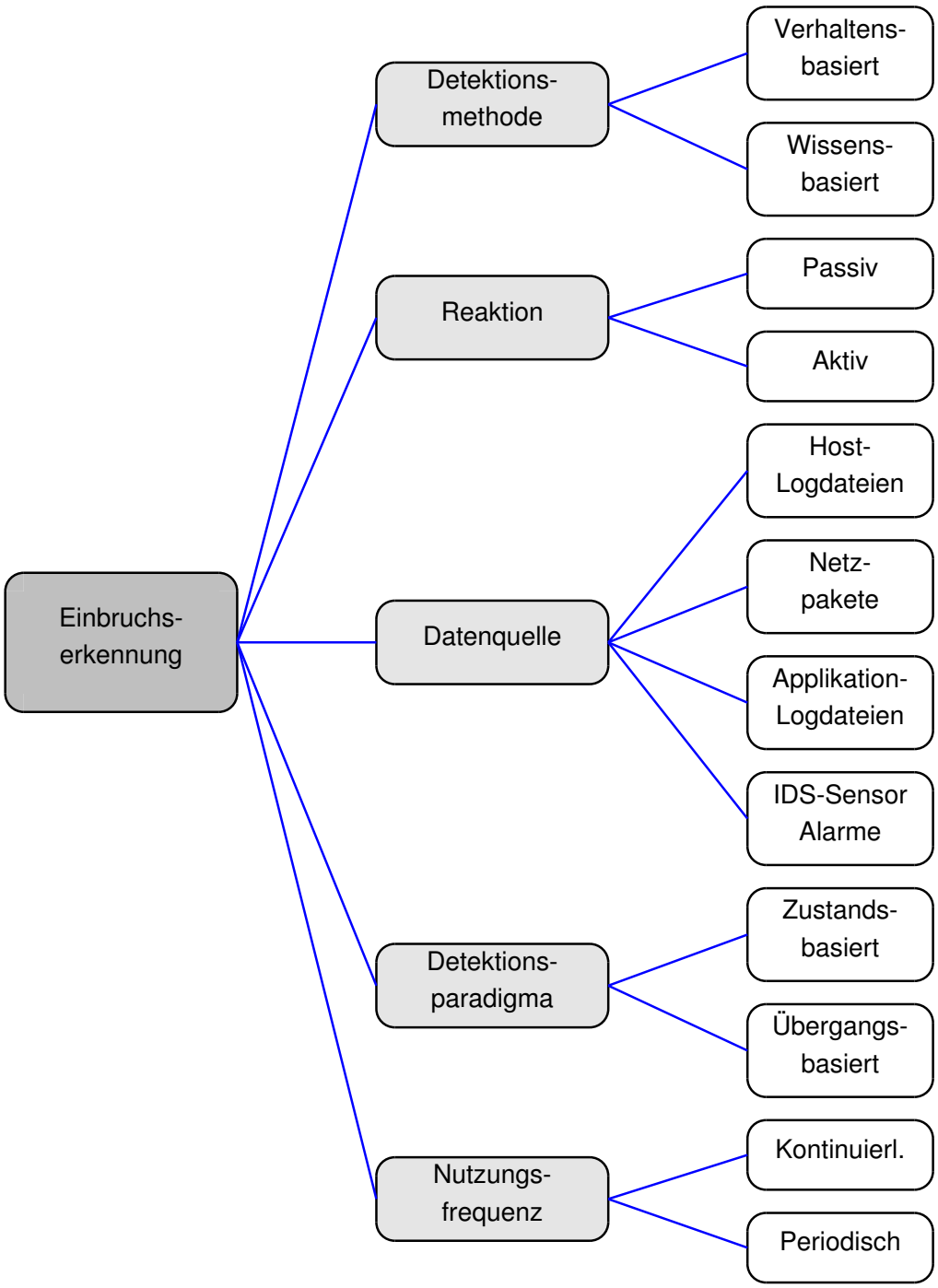


Abbildung 4.2: Taxonomie von Systemen zur Einbruchserkennung nach Debar et al. [115].

Passive Verfahren erzeugen Alarme und senden ggf. Notifikationen für den zuständigen Administrator, bspw. in Form einer E-Mail. Aktive Verfahren können zwischen *korrigierend* und *pro-aktiv* unterschieden werden; korrigierende Systeme greifen ein, indem erkannte Lücken im System geschlossen werden (z.B. Änderung von falschen Dateirechten im System), während pro-aktive vermeintliche Angreifer ausloggen, Dienste beenden, etc. Stellt ein IDS entsprechende, aktive Funktionalitäten bereit, wird es auch als Intrusion Prevention System (IPS) bezeichnet.

Die möglichen **Datenquellen** unterscheiden IDSs anhand der Informationen, welche durch die Systeme ausgewertet werden. Hier können grundlegen *Logdateien von Rechnern* oder von einzelnen *Applikationen*, Pakete des Netzverkehrs sowie Alarmmeldungen, die von anderen IDSs erzeugt wurden, genutzt werden. Grundlegend sind Host-basierte Systeme per se in der Lage, sämtliche Logdateien, aber auch weitergehende Informationen wie die Ressourcennutzung eines Nutzers oder die Systemaufrufe der von ihm genutzten Applikationen zu evaluieren, wobei sie hinsichtlich des Datenverkehrs über das Netz auf die eigene Kommunikation eingeschränkt sind; ein netzbasiertes System dahingegen hat zunächst keinen Zugriff auf detaillierte Betriebsdaten eines Hosts, kann dahingegen jedoch den gesamten Netzverkehr auswerten, wenn es an einer strategischen Position im Netz installiert ist. Verschiedene hybride Systeme existieren, die bspw. als hostbasierte Systeme installiert sind und untereinander Alarmnachrichten oder Events austauschen; dies vereinigt Vorteile beider Systeme, äußert sich jedoch in einer hohen Komplexität, einem hohen Kommunikationsaufwand und zieht weitere Nachteile wie aufwändige Administration nach sich. Von den Zugriffsmöglichkeiten ist daher zunächst zwischen host- und netzbasierter Umgebung zu unterscheiden.

Hostbasierte Systeme haben insbesondere *Systeminformationen*, *Accounting* und *Syslogs* als Quelle zur Verfügung (vgl. Anhang F.2.5).

Im Gegensatz dazu können netzbasierte Sensoren nicht über detaillierte Prozess- und einfach zuordenbare Nutzerevaluationen verfügen. Die wichtigsten Daten, die in diesem Bereich ausgewertet werden können sind die Netz-Pakete selbst, Simple Network Management Protocol (SNMP)-Daten sowie Flow-Daten. Anhang F.2.6 gibt weitere Details bzgl. der verschiedenen Informationsquellen an.

Mittels des **Detektionsparadigmas** werden die Systeme anhand des genutzten Detektionsmechanismus unterschieden. IDS können entweder Zustände des Systems analysieren, d.h. ob der jeweilige Zustand sicher oder unsicher ist, oder Zustandsübergänge, von einem sicheren zu einem unsicheren Zustand. Wird eine zustandsbasierte Detektion genutzt, kann entweder der Normalzustand festgestellt werden, dies ist jedoch sehr aufwändig, wie sich insbesondere auch durch die Fehlalarmraten in verhaltensbasierten Systemen zeigt. Eine andere Möglichkeit ist die Detektion verschiedener Fehlerstadien, bevor es letztendlich zum Ausfall der jeweiligen Funktion kommt. Dies widerrum kann auch durch eine Detektion der Zustandsübergänge, welche zu einem Fehler führen, festgestellt werden. Abbildung 4.3 veranschaulicht die drei Möglichkeiten. Weiterhin kann diese Analyse passiv, rein auf Basis der vorhandenen Daten erfolgen oder durch eine aktive Stimulation des Systems, um eine Reaktion zu erhalten.

Das letzte Klassifizierungskriterium bezieht sich auf die **Nutzungsfrequenz**, ob ein

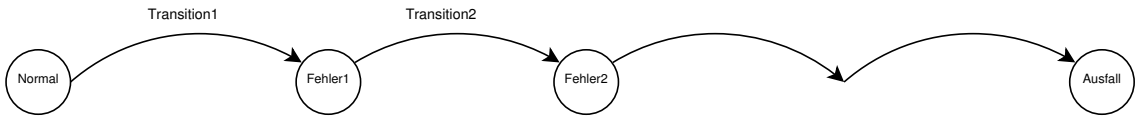


Abbildung 4.3: Detektionsmöglichkeiten anhand von Zuständen und Zustandsübergängen [115].

System eine *kontinuierliche* Auswertung ermöglicht, oder eine Evaluation zu periodischen Zeitpunkten durchgeführt wird. Insbesondere die Fähigkeit, Einbruchsversuche und Angriffe in Echtzeit zu erkennen, ist ein wichtiges Unterscheidungsmerkmal; im Bereich von breitbandigen Netzverbindungen ist eine Verarbeitung und Analyse in Echtzeit sehr aufwändig und abhängig der verwendeten Verfahren ggf. nicht vollständig möglich.

Insbesondere in den jeweiligen Unterklassen lassen sich feinere Unterteilungen vornehmen, als diese in der Arbeit von Debar et al. vorgenommen wurden. Als Beispiel seien hier die Techniken des maschinellen Lernens (vgl. z.B. [381, 203]) genannt, welche die unter *verhaltensbasierte Verfahren* aufgeführten Methoden beinhaltet, wobei die Klasse der statistischen Verfahren aufgegliedert werden kann in u.a. Bayesschen Klassifikator, Support Vector Machine (SVM), etc. Für eine allgemeine Klassifizierung sind die ursprünglich gegebenen Kategorien jedoch immernoch ausreichend, insbesondere zeigt sich dies in der nach wie vor hauptsächlich genutzten Differenzierung der Systeme nach der Detektionsmethode.

Sehr viel detaillierter ist bspw. die Taxonomie von Sabahi und Movaghar, die ältere Taxonomien insbesondere hinsichtlich neuerer Technologien erweitert (vgl. Abbildung 4.4) [333]. Hierbei kommen somit neue Klassen wie bspw. die genutzte Umgebung hinzu, die in leitungsgebundene Systeme, drahtlose und gemischte Umgebungen unterschieden wird, wobei drahtlose Systeme weiter nach den Betriebsmodi mit oder ohne Infrastruktur (AdHoc) unterschieden werden.

Im Gegensatz zu der stark erweiteren Klassifizierung von Sabahi und Movaghar, gibt Sundaram eine einfache Taxonomie basierend auf der weithin akzeptierten und genutzten Detektionsmethode (Anomalie- bzw. Missbrauchserkennung), wobei sich die einzelnen Klassen zu denen von Debar teilweise unterscheiden, die am weitest verbreiteten und genutzten jedoch in beiden Modellen vorkommen (vgl. Abbildung 4.5).

Bolzoni gibt speziell für den Bereich der anomaliebasierten Systeme eine Taxonomie an (vgl. Abbildung 4.6), wobei diese neben den oft genutzten Klassen *Algorithmus* und *Datenart* (Headerdaten, Payload oder beides) die Klasse *Granularität* definiert, die Systeme anhand der Nutzung von einzelnen Paketen oder kompletten Verbindungen unterscheidet. Dieser Aspekt trägt insbesondere neueren System Rechnung, welche mittels Flow-Daten arbeiten und somit maßgeblich statistische Daten über abgeschlossene Verbindungen nutzen.

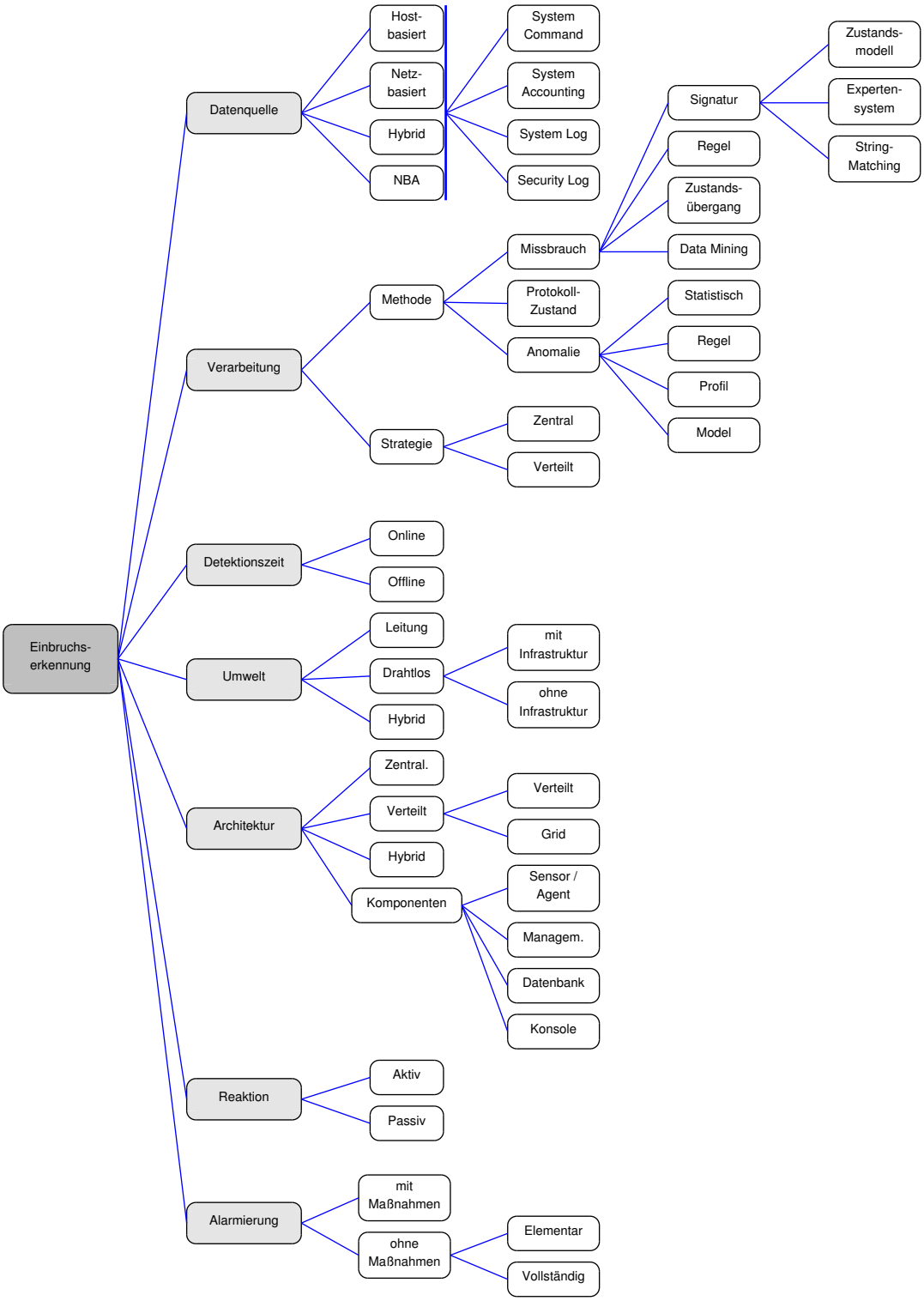


Abbildung 4.4: Taxonomie von Systemen zur Einbruchserkennung nach Sabahi und Movaghar [333].

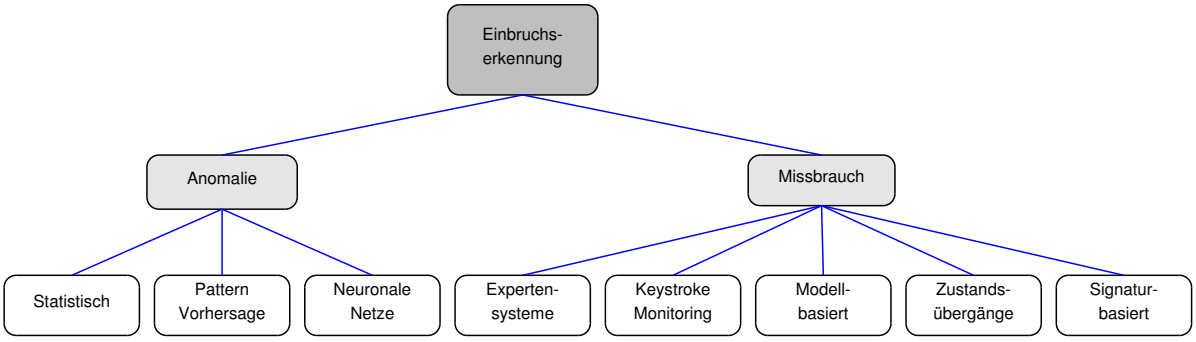


Abbildung 4.5: IDS-Taxonomie nach Sundaram [368].

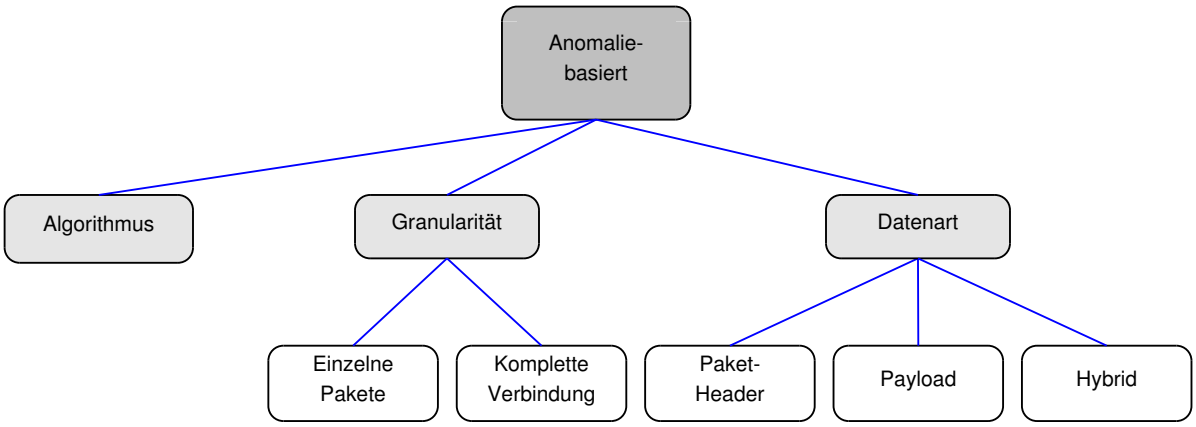


Abbildung 4.6: Taxonomie anomaliebasierter Systeme nach Bolzoni [61].

Vergleicht man die Taxonomien mit den in Kapitel 2.1 abgeleiteten Anforderungen an ein geeignetes Sicherheitssystem, lassen sich anhand dieser vorhandene Systeme nicht hinsichtlich aller Forderungen einordnen. Bspw. bieten die vorhandenen Schemata keine Möglichkeit, eine Unterscheidung von Systemen anhand ihrer Fähigkeiten, in verschlüsselten Umgebungen zu arbeiten oder anhand der Rechtskonformität ihres Einsatzes vorzunehmen. Dementsprechend wird basierend auf den vorhandenen Topologien die in Abbildung 4.7 gezeigte Systematik verwendet.

Im Gegensatz zur ursprünglichen Taxonomie wurde die Differenzierung gem. des darunterliegenden Detektionsparadigma nicht mit aufgenommen, da sich diese Klassifizierung als nicht praxisrelevant erwiesen hat und auch von keinen anderen Taxonomien aufgegriffen wurde. Dahingegen wurde die Klassifizierung um folgende drei Punkte erweitert:

- **Zulässigkeit**: Im Kontext von Datenschutzbestimmungen und den jeweils landesabhängig geltenden Gesetzen ist die Rechtskonformität eines Systems unerlässlich. Werden die geltenden Bestimmungen nicht eingehalten, führt dies nicht nur zu

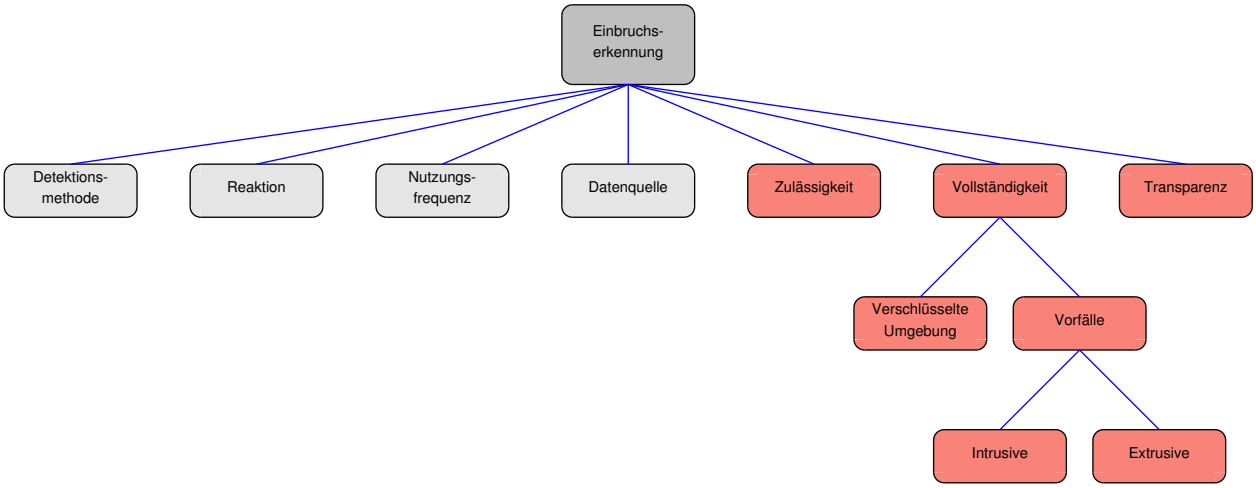


Abbildung 4.7: Klassifizierung von Intrusion Detection Systemen. Aufgrund der ungenügenden Differenzierung verfügbarer Taxonomien bzgl. den Anforderungen moderner Netzumgebungen, wurden die Kategorien *Zulässigkeit*, *Vollständigkeit* sowie *Transparenz* eingeführt.

einem Verwertungsverbot, sondern kann auch eine Strafanzeige und empfindliche Strafen nach sich ziehen (vgl. Kapitel 4.6.5). Die Klassifizierung gem. der Zulässigkeit spiegelt daher wieder, ob das jeweilige System in Einklang mit geltendem Recht eingesetzt werden darf.

- **Vollständigkeit**: Die Entwicklung sowohl der Technologie als auch der möglichen zu detektierenden Vorfälle stellt bzgl. der Detektierbarkeit hohe Anforderungen an heutige IDSs. Diesem Fakt wird durch bisherige Taxonomien nicht Rechnung getragen: Durch die schnell fortschreitende Einführung und immer stärkere Nutzung von Verschlüsselung, aber auch durch neue Angriffsverfahren wie zielgerichtete Angriffe, Innentäter und Social Engineering sind viele traditionelle Systeme nicht in der Lage, alle Gefährdungsbereiche adäquat abzudecken. Entsprechend wird diese Kategorie in die Taxonomie eingeführt und weiter in die Klassen *Verschlüsselte Umgebung* und *Vorfälle* untergliedert. Vorfälle können widerrum in die Bereiche *Intrusive* und *Extrusive* unterschieden werden; zu letzterem gehören bspw. sowohl durch Innentäter initiierte Aktionen, Datenverlust, als auch durch im Rahmen von Botnetzen ferngesteuerter Rechner durchgeführte Aktivitäten.

- **Transparenz**: Im Sinne der Integrierbarkeit eines Systems muss der Aufwand, dieses in eine bereits bestehende Struktur einzubeziehen, berücksichtigt werden. Dies wird mittels der Kategorie *Transparenz* bewertet.

4.3 Aufbau und Funktion

Generell können in jedem IDS drei Basiskomponenten identifiziert werden, Datensammlung, -analyse und Darstellung. Abbildung 4.8 zeigt die schematischen Zusammenhänge. Sensoren sammeln Daten und senden diese an eine zentrale Komponente zur weiteren Verarbeitung. Hier wird aufgrund der anfallenden Datenmengen meist ein Datenbanksystem verwendet.

Die Datensammlung hängt maßgeblich von der Art des eingesetzten Systems ab; wird ein NIDS verwendet, muss dieses insbesondere in geswitchten Umgebungen, wie sie heute typischerweise als LANs eingesetzt werden, an einer zentralen Komponente der Infrastruktur installiert werden, bspw. dem Netzübergang zum Wide Area Network (WAN). Werden hostbasierte Systeme eingesetzt, kann der gesamte für den jeweiligen Rechner bestimmte Verkehr direkt ausgewertet werden, darüber hinaus stehen detaillierte Daten über das System zur Verfügung. Da ein hostbasiertes System jedoch nur den jeweiligen Rechner selbst überwachen kann, muss hier für den Schutz eines Netzes eine entsprechende Komponente auf jedem Rechner vorhanden sein; dies kann auch in Form von Sensoren erfolgen, welche die erfassten Daten widerrum zu einer zentralen Auswerteinstanz senden. Auch im Rahmen von NIDS kann dies erforderlich werden, wenn aufgrund der Netzstruktur oder Kapazität ein einzelnes System nicht in der Lage ist, den gesamten Datenverkehr zu überwachen.

Die Analyse untersucht die eingegangenen Daten auf das Vorhandensein eines Angriffs und gibt die aufbereiteten Informationen an die Darstellungskomponente weiter. Sie ist somit die zentrale Komponente eines jeden IDS. Der Bereich Darstellung wird bei einigen Systemen weiter in die Bereiche Management und Auswertung unterteilt. Während erstere für die Konfiguration der Sensoren und des IDS verantwortlich ist, übernimmt in dieser Unterteilung die Auswertung den Analyseteil, die Darstellung der entsprechenden Daten sowie Reporting-Funktionalität. Abhängig des verwendeten Typs des IDS werden verschiedenen Sensoren zur Datensammlung eingesetzt, die zum Beispiel in der Form von Agenten Netzverkehr oder Systemereignisse in Echtzeit sammeln und zur weiteren Verarbeitung an die Analysekomponente übertragen.

Generell erfolgt zunächst eine Aufbereitung der Daten in Präprozessoren für die jeweiligen Komponenten, um anschliessend durch die Detektionsalgorithmen weiter verarbeitet zu werden. Erkannte Anomalien werden anhand von Filter- und Entscheidungskriterien an die Darstellungskomponente weitergegeben.

Anhand der durchgeführten Analyse, der dargebotenen Information und der möglichen Reaktionen lassen sich Systeme der Einbruchserkennung in vier Kategorien einteilen.

Einbruchserkennung (Intrusion Detection) IDSs entsprechen der klassischen Definition der Einbruchserkennung (vgl. Kapitel 4.1). Die Hauptaufgabe des IDS ist der Schutz des eigenen Systems oder Netzes vor Angriffen von Außen. Angemerkt sei, dass zwar bereits mehrere frühe Definitonen der Einbruchserkennung den Missbrauch durch autorisierte Personen beinhalten, klassische IDS jedoch typischerweise keine entsprechende Funktionalität bereitstellen.

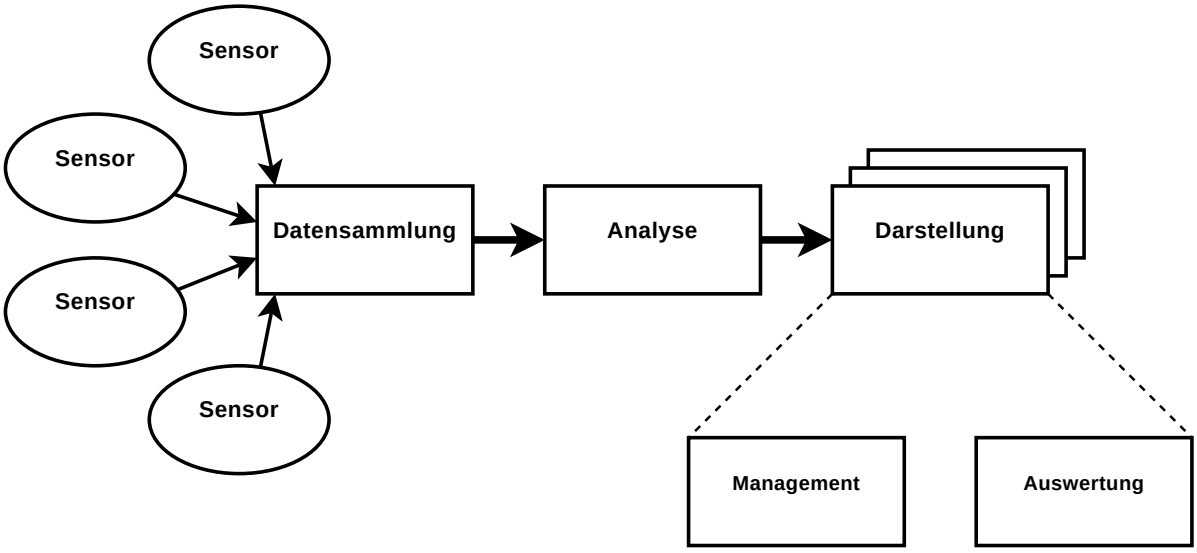


Abbildung 4.8: Komponenten von Intrusion Detection Systemen.

Intrusion Prevention Ist das System neben der Einbruchserkennung auch in der Lage, Maßnahmen im Falle der Detektion eines Angriffsversuchs zu ergreifen, bspw. das Sperren von Ports in einer Firewall oder das Blockieren von IP-Adressen, spricht man von einem Intrusion Prevention System (IPS).

Extrusion Detection In den letzten Jahren gab es immer wieder Datenskandale, die auch in der Öffentlichkeit intensiv diskutiert wurden. Oftmals handelte es sich dabei um verloren gegangene Daten, bspw. eine CD oder die eines Smartphones (vgl. z.B. [350, 40, 378]). Neben diesem unbeabsichtigten Verlust von Daten muss das gezielte Ausschleusen von Daten gegen die Intention des Inhabers unterschieden werden. Im Rahmen von Industriespionage und Innentätern, hat dieser Bereich eine besondere Bedeutung. Datendiebstahl ist somit ein Teilbereich des Datenverlustes (*Data Leakage*). Ziel der Ausbruchserkennung ist sowohl die Detektion und das Verhindern von unbeabsichtigten als auch gezielt herbeigeführten Datenabfluss.

Frühwarnung (Early Warning) Eine weitere Kategorie der Einbruchserkennung bildet der Bereich der sog. Frühwarnsysteme. Im Gegensatz zu IDSs sind Systeme der Frühwarnung weiträumiger aufgestellt. Anhand zahlreicher dislozierter Sensoren in verschiedenen Subnetzen, werden die dort auflaufenden Informationen gesammelt und ausgewertet, um so Anomalien in Teilbereichen des Gesamtnetzes (typischerweise das Internet) zu erkennen. Hierfür werden u.a. Log-Dateien von Firewalls und IDSs, Flow-Daten über den Netzverkehr sowie durch Honeynets und -pots gesammelte Daten ausgewertet. Werden Gefahren entdeckt, kann diese Information an die noch nicht betroffenen Teilnetze übermittelt werden, um eine Ausbreitung der Gefährdung schnell und effizient einzudämmen. Auf diese Weise kann z.B. die Ausbreitung eines Wurms im Frühstadium seiner Aktivi-

Tabelle 4.1: Systemarten der Einbruchserkennung.

	IDS	IPS	EPS	EWS
Erkennung von Angriffen,				
von Extern	✓	✓		
von Intern			✓	
Verhindern von Angriffen		✓		
Eigenschutz	(✓)	✓		(✓)
Kooperativer Schutz				✓
Verhindern von Datenverlust			✓	

tät durch die hiermit verbundenen Unregelmäßigkeiten in der statistischen Analyse des Netzverkehrs detektiert werden. Durch eine Evaluation des Verhaltens der Schadsoftware können daraufhin Empfehlungen weitergegeben oder automatisierte Konfigurationen durchgeführt werden, bspw. das präventive Sperren von Ports auf Firewalls.

Tabelle 4.1 fasst die Besonderheiten der verschiedenen Systemarten nochmals zusammen.

Abhängig des jeweiligen Verfahrens, lassen sich die in Kapitel 2.3.3 vorgestellten Angriffsstufen anhand verschiedener Merkmale detektieren. Tabelle 4.2 gibt die Detektionsmöglichkeiten der verschiedenen Verfahren und Quellen wider.

4.4 Leistungsmessung

Die für den Einsatz eines Sicherheitssystems erforderlichen Kriterien wurden anhand des Anforderungskataloges in Kapitel 3 definiert. Zusätzlich müssen im weiteren Verlauf meßbare Parameter betrachtet werden, mittels derer eine Bewertung und ggf. Reihung geeigneter Systeme, welche die Kriterienkataloge erfüllen, untereinander erfolgen kann.

Messbare Leistungsparameter im Rahmen entsprechender Sicherheitssysteme sind insbesondere [247]:

- Abdeckung bzgl. möglicher Angriffe
- Wahrscheinlichkeit von Fehlalarmen
- Detektionswahrscheinlichkeit

[1]Solange keine Zero-Days genutzt werden, bzw. ein Pattern vorhanden ist.
[2]Eingeschränkt, z.B. durch Logins zu untypischen Zeiten.
[3]Nur für den jeweiligen Host selbst.
[4]Bei korrekter Durchführung wird genau dies verhindert.
[5]Bei ungewöhnlichem Datenverkehr, z.B. einem Brute Force Angriff.

Tabelle 4.2: Detektion der Angriffsklassen. Den Datenpaketen kommt eine besondere Rolle zu, da diese einerseits umfangreiche Informationen zu allen Angriffsschritten beinhalten, andererseits jedoch nicht notwendigerweise verfügbar sind, bspw. durch den Einsatz von Verschlüsselungstechnologien.

Detektionsmerkmal	Analyse Zielumgebung	Identifizieren v. Schwachstellen	Erlangen Fernzugriff *R2L*	Erlangen administrativer Rechte *U2R*	Manipulation der Systemumgebung	Löschen Angriffsspuren
Detektionsverfahren						
Signaturbasiert	✓	✓	(✓)[1]	(✓)	(✓)	(✓)
Verhaltensbasiert						
Systemverhalten	✓	✓	✓	✓	✓	✓
Nutzerverhalten	✗	✗	(✓)[2]	✓	✓	✓
Detektionsparadigma						
Zustandsbasiert	✗	✗	✓	✓	✓	✗
Übergangsbasiert	✓	✓	✓	✓	✓	✓
Datenquelle						
Host-Daten (Logs, etc.)	(✓)[3]	✓	✓	✓	✓	✗[4]
Datenpakete	✓	✓	✓	✓	✓	✓
Protokoll-Daten	✓	✓	(✓)[5]	✗	✗	✗
Flow-Daten	✓	✓	(✓)	✗	✗	✗

- Resistenz vor Angriffen gegen das System selbst
- Verarbeitbare Datenrate
- Korrelationsfähigkeit[6] von Ereignissen
- Detektionsfähigkeit von neuen, unbekannten Angriffen
- Kategorisierung von Angriffen
- Beurteilung des Angriffserfolgs

Ein Vergleich von IDSs ist aufgrund der unterschiedlichen Systemdesigns, der verschiedenen Anwendungsumgebungen und der Komplexität von realem Netzverkehr sehr aufwändig und wird daher in der Literatur intensiv diskutiert (vgl. z.B. [321, 90, 317, 274, 265]). Grundsätzlich soll ein IDS Angriffe und schädliche Pakete im Netzverkehr erkennen. Neben diesen schadhaften Daten existiert der autorisierte Verkehr, der im Kontext der Einbruchserkennung als *Hintergrundverkehr (background traffic)* bezeichnet wird. Der Hintergrundverkehr beinhaltet ebenfalls Muster und Verhalten, die sich für ein IDS als böswillig darstellen können, obwohl es sich hierbei um legitimen Datenverkehr handelt; entsprechend erzeugt das IDS einen unerwünschten Alarm. Die Fehlalarmrate ist ein entscheidender Faktor beim Betrieb eines IDS. Es muss zwischen folgenden Kategorien von Alarmen unterschieden werden:

- Richtig-Positiv (*True Positive (TP)*): Ein Alarm, der als Reaktion auf einen erkannten Einbruchsversuch ausgelöst wurde.
- Falsch-Positiv (*False Positive (FP)*): Ein Alarm, der fälschlicherweise ausgelöst wurde, obwohl der analysierte Datensatz kein bösartiges Verhalten beinhaltet.
- Richtig-Negativ (*True Negative (TN)*): Es wurde korrekterweise kein Alarm ausgelöst, der analysierte Datensatz enthält kein bösartiges Verhalten.
- Falsch-Negativ (*False Negative (FN)*): Es wurde kein Alarm ausgelöst, obwohl der zugrunde liegende Datensatz einen Angriffsversuch darstellt.

Dadurch, dass die Erkennung bei wissensbasierten Systemen auf der Detektion von *bösartigem*, bei der verhaltensbasierten Detektion typischerweise[7] auf dem gutartigen Normalverhalten beruht, ergeben sich die Fehlalarme und Detektionsfähigkeiten gem. Abbildung 4.9.

[6] Hierunter sind insbesondere die Fähigkeiten eines Systems zu verstehen, Meldungen aus verschiedenen Quellen zu korrelieren. Bspw. soll der Zusammenhang von Meldungen, die an einer Firewall, einem IDS und in verschiedenen Logdateien auflaufen, erkannt werden. Ziel hierbei ist es, insbesondere stufenweise durchgeführte Angriffe zu erkennen. Die Korrelation von Ereignissen verschiedener Quellen ist ein nicht-triviales Problem und von derzeitigen Systemen nur sehr begrenzt durchführbar [247].

[7] Aber nicht notwendigerweise!

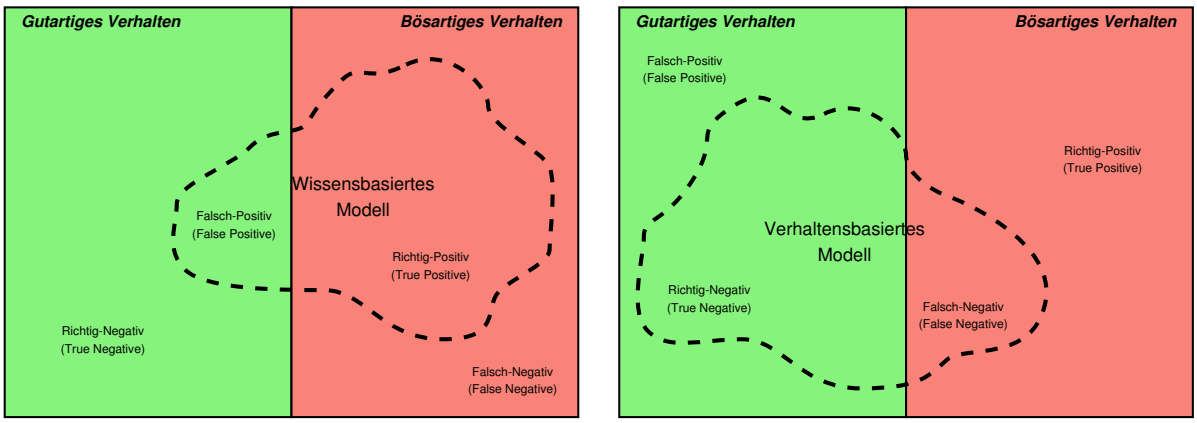


Abbildung 4.9: Fehlalarmraten und Detektionsfähigkeiten verhaltens- und wissensbasierter IDSs gem. [287].

Das wichtigste genutzte Vergleichskriterium für IDSs ergibt sich direkt anhand der Alarme (vgl. Kapitel 4.3) und lässt sich u.a. durch die Vollständigkeit eines Systems messen (vgl. [116, 117]):

Für die Klassifizierung gilt:

$$C = \frac{TP + FP}{TP + FN} \tag{4.1}$$

Die Detektionswahrscheinlichkeit bestimmt sich mittels:

$$PD = \frac{TP}{TP + FN} \tag{4.2}$$

Diese wird auch als Vollständigkeit oder Effektivität des IDS bezeichnet und beschreibt die Detektionsrate im Sinne der erkannten, echten Angriffe.

Das Fehlalarmverhältnis ergibt sich aus:

$$Ratio = \frac{FP}{FP + TP} \tag{4.3}$$

Für die Fehlalarmrate (False Alert Rate (FAR)) gilt:

$$FAR = \frac{FP}{TN + FP} \tag{4.4}$$

Insbesondere verhaltensbasierte Systeme neigen hier zu höheren Fehlalarmraten im Sinne von FPs, jedoch ist auch bei signaturbasierten Verfahren die Zahl der Fehlalarme hoch, solange diese nicht genau für die Betriebsumgebung und den dort installierten Diensten und Rechnern konfiguriert und angepasst sind. Fehlalarme lassen sich jedoch auch bei einer perfekten Konfiguration eines signaturbasierten Systems nicht verhindern, da bspw. das Pattern eines Virus als Bestandteil eines legalen Dokuments vorliegen kann. Adleman hat bewiesen, dass die Menge der Computer-Viren Π_2 – vollständig ist [26],

weiterhin hat Spinellis gezeigt, dass eine sichere Identifikation von Viren begrenzter Länge NP-vollständig ist [362]. Roşu demonstrierte, dass die Gleichheit von Datenströmen Π_2^0 – vollständig ist [326]. Dies bedeutet insbesondere, dass eine fehlerfreie Identifizierung von Viren und Schadsoftware auch mit signaturbasierten Verfahren nicht möglich ist, somit ist immer ein entsprechendes Maß an Fehlalarmen vorhanden.

Sanchez schlägt ein Zonenmodell für die Integration von IDSs vor, um mittels verschieden eingestellter Sensitivitäten der Systeme den Schutzbedürfnissen der jeweiligen Teilnetze Rechnung zu tragen und somit auch die Anzahl von Fehlalarmen zu reduzieren (vgl. Kapitel F.2.13). Hierbei nimmt die Sensitivität der Systeme von einer geringen Empfindlichkeit des vor der Firewall im Bereich des roten Netzes (WAN, hier Internet) installierten IDS, über das System zur Überwachung der Demilitarized Zone (DMZ) bis hin zum IDS für das interne LAN, welches den geringsten Datenverkehr von Außen aufweist und somit auch empfindlicher eingestellt werden kann, ohne dass dies in zu hohen Fehlalarmraten resultiert.

Für den Test eines IDS werden entsprechend sowohl Angriffsdaten, als auch Hintergrundverkehr benötigt. Folgende vier Varianten im Umgang mit Hintergrundverkehr stehen zur Verfügung [247]:

- Testdurchführung ohne Nutzung von Hintergrundverkehr
- Nutzung von realem Hintergrundverkehr
- Nutzung von geprüftem, anonymisierten Hintergrundverkehr
- Verwendung von simuliertem bzw. synthetisch erzeugten Hintergrundverkehr

Die Erstellung verifizierter Referenzdaten (die sog. *Ground Truth*, vgl. [359]) ist eine nicht-triviale Aufgabe. Grundsätzlich gibt es hier zwei mögliche Verfahren, die Erzeugung synthetischer Daten und die Nutzung und Klassifizierung von realem, aufgezeichneten Datenverkehr.

- **Aufgezeichnete Daten:** Die Aufzeichnung von realen Netzdaten ergibt einen Datensatz, der sämtliche Faktoren wie Datenmengen, genutzte Dienste, tageszeitabhängige Schwankungen, etc. wiedergibt. Diese können später bspw. mittels Hilfsprogrammen wie `tcpreplay` wieder eingespielt werden und so zum Vergleich von Systemen herangezogen werden. Nachteilig erweisen sich aber insbesondere die Aspekte, dass Ergebnisse, die auf Basis realer Daten gewonnen wurden, schwerer nachvollziehbar sind, da der zugrunde liegende Datensatz oftmals aus Gründen des Datenschutzes nicht weitergegeben werden darf. Entsprechend muss eine Anonymisierung durchgeführt werden, die neben den Adressen auch Payload-Inhalte betreffen kann, was widerrum die Nutzbarkeit der Daten für weitere Evaluationen in Frage stellen kann. Ein weiteres Problem im Umgang mit realen Daten ist die Schwierigkeit und der hohe Aufwand, eine entsprechende Markierung aller Daten nach ihrer Art (gutartig bzw. bösartig) durchzuführen. Insbesondere mit Hinblick auf neue, unbekannte Angriffe oder komplex ausgeführter, verteilter Angriffe kann dies ggf. auch in einer Analyse nicht erkannt werden und kann somit zu falschen Markierungen der Pakete führen.

- **Synthetische Daten:** Ein anderer Ansatz ist, den benötigten Hintergrundverkehr synthetisch zu erstellen. Zahlreiche Programme bieten hier Möglichkeiten an, Datenverkehr für verschiedene Protokolle zu erzeugen, wobei oftmals eine Spezialisierung auf bestimmte Bereiche wie bspw. Webserver vorliegt. Beispiele sind `httperf`, ein verbreitetes Tool zu Erzeugung von HTTP-Verkehr und zur Durchführung von Stresstests für Webserver oder `Seagull`, ein Multiprotokoll-Datengenerator. Die Problematik bei der synthetischen Erzeugung liegt darin, ein realitätsnahes Abbild des Datenverkehrs und des Nutzer- und Systemverhaltens zu erzeugen. Eine synthetische Erzeugung von Referenzdaten ist daher meist nur unbefriedigend möglich (vgl. z.B. [322, 36]). Ausnahmen können hier Bereiche sein, wo kein kompletter Netzverkehr erzeugt werden muss, sondern lediglich Aspekte oder statistische Parameter davon, bspw. bei der Flow-Analyse. [359] beschreibt die Erzeugung von generischen Referenzdaten im Bereich der Flow-Analyse, allerdings ist auch hier der Aufwand zur Erzeugung der Daten sehr hoch.

Frühe Ansätze, IDSs zu vergleichen, sind bereits mit den Arbeiten von Puketza et al. aus dem Jahre 1996 vorhanden [316]. Auf Grundlage von Prozessen aus dem Bereich von Software-Tests wird eine Methodik und Plattform zum Test von IDSs vorgeschlagen. Sowohl die Angriffe, als auch der Hintergrundverkehr werden skriptgesteuert erzeugt. Das Verfahren wurde kritisiert, da keine systematische Auswertung der Daten erfolgte und lediglich der Test eines IDS durchgeführt wurde [274].

Die bekannteste und am weitest verbreitete Evaluation für IDS ist der Datensatz der Defense Advanced Research Projects Agency (DARPA) Off-Line Intrusion Detection Evaluation von 1998 [255, 110] sowie die Erweiterung von 1999 [188, 253, 187]. Die Datensätze wurden auf Basis des Netzverkehrs einer Airforce-Basis erzeugt. Hierfür wurde der reale Datenverkehr gemessen, ausgewertet und in einem weiteren Schritt in einem isolierten Netz simuliert. Hierbei wurden mehrere tausend Unix-Systeme und mehrere hundert Nutzer nachgestellt, um den normalen Datenverkehr (Hintergrundverkehr) zu erzeugen, als Angriffe wurden weiterhin Netz-Scans durchgeführt, DoS-Angriffe, Remote-to-Local (R2L)- sowie User-to-Root (U2R)-Aktivitäten. Ziel war es, hiermit einen objektiven, wiederhol- und vergleichbaren sowie realistischen Datensatz zur IDS-Evaluation zu erstellen. Die DARPA-Datensätze haben eine weite Verbreitung und Anwendung zur Evaluation gefunden, obwohl sowohl Vorgehen zur Datengewinnung als auch Zusammensetzung kritisiert wurden (vgl. z.B. auch [265]). McHugh gibt in seiner Analyse zur DARPA 98 und 99 Evaluation insbesondere folgende Kritikpunkte an [274, 36]:

- Es erfolgte keine Auswertung der Fehlalarm-Charakteristik. Die Anzahl der Fehlalarme, welche durch den Hintergrundverkehr alleine ausgelöst werden, ist daher nicht bekannt.

- Es wurden keine Statistiken, weder zum originalen Datenverkehr, noch zur korrekten Erzeugung der synthetischen Daten, veröffentlicht.

- In den Datensätzen wurden keine Datenraten bzw. deren Variationen berücksichtigt. Insbesondere scheint die genutzte Datenrate lediglich im Bereich von Kbps

zu liegen[8].

- Die Angriffe sind über den Datensatz gleichverteilt. Darüberhinaus wurde jeder Angriffstyp (Netz-Scan, DoS, R2L, U2R) in der gleichen Häufigkeit durchgeführt, was nicht der Realität entspricht.
- In der Dokumentation der DARPA-Testsets befinden sich Inkonsistenzen.
- Es wurde keine Kontrollgruppe in die Tests involviert.
- Es sind nicht alle Angriffsklassen repräsentiert, insbesondere liegt das Gewicht auf interaktiven Angriffen auf Betriebssystem-Schwachstellen.
- Der Umfang der Trainingsdaten ist ggf. nicht groß genug. Insbesondere liegt kein Beweis vor, dass die Angriffe realistisch sind oder dass es überhaupt möglich ist, einen entsprechenden Satz zu konstruieren.
- Einige Angriffe haben TTL-Charakteristiken, die nicht dem Hintergrundverkehr entsprechen.

Die Aussagekraft von Evaluationen basierend auf Messungen der DARPA-Daten hinsichtlich der wirklichen Leistungsfähigkeit eines Sytems bzw. Algorithmus ist jedoch aus u.a. den oben genannten Gründen zweifelhaft. Mahoney und Chan konnten anhand eines einfachen IDS im Jahre 2003 u.a. feststellen, dass alle bösartigen Datenpakete des DARPA-Satzes eine TTL von 126 oder 253 aufweisen, wohingegen die gutartigen Pakete des Hintergrundverkehrs hauptsächlich eine TTL von 127 oder 254 haben. Die bösartigen TTL-Werte 126 und 253 kommen *nicht* in den (angriffsfreien) Trainingsdatensätzen vor. Eine Analyse der Pakete ergab weiterhin, dass in den 12 Millionen Paketen des Trainingsdatensatzes lediglich zehn[9] verschiedene TTL-Werte auftreten (2, 32, 60, 62, 63, 64, 127, 128, 254 und 255), eine Auswertung von einer Million Paketen anhand eines produktiven Webservers aber bereits 80 verschiedene Werte ergab [264], was daher ebenfalls ein unrealistisches Bild ergibt. Zudem sind die TTL-Werte der Datensätze im Schnitt deutlich zu hoch im Vergleich zu realem Datenverkehr. Da dies eine grundsätzliche Fehlkonzeption des Datensatzes aufzeigt, die folglich auch für den aus den DARPA-Daten gewonnenen Knowledge Discovery and Data Mining (KDD)-Cup 99-Datensatz gilt, haben mehrere Forscher von der Nutzung der DARPA-Daten abgeraten (vgl. z.B. [66]).

Weiterhin können die Daten unabhängig der Designfehler auch nicht mehr den sich seit dem Jahre 2000 stark gewandelten Angriffsspektrum Rechnung tragen (vgl. Kapitel 4.6.1). Trotz der wiederholten Kritiken an den Datensätzen, werden die DARPA-Daten auch heute noch für Vergleichszwecke und zur Darstellung von Verbesserungen im Bereich der Fehlalarmrate herangezogen (vgl. z.B. [347, 171, 248]).

Verschiedene Projekte zeichnen den Datenverkehr im Internet auf und erzeugen unterschiedliche Formen von abrufbaren Daten, bspw. komplette Datensätze, TCP-Daten,

[8] Dies ist auch unter Berücksichtigung des zeitlichen Abstands und der damals genutzten Technologie zu gering, insbesondere unter Berücksichtigung des simulierten Netzes mit einigen tausend Servern.
[9] [264] spricht hier von *8* Werten, führt jedoch die aufgezählten *10* Werte auf.

Webzugriffe, Netflow-, Routing- oder Latenzdaten. Für die Nutzung im Rahmen der Evaluation von IDSs haben hier insbesondere die Paketserien der Measurement and Analysis on the WIDE Internet (MAWI) Working Group eine Bedeutung, die als Trace-Daten eines Transpazifik-Links gewonnen werden [9]. Weitere, öffentlich verfügbare Aufzeichnungen existieren von der FH Salzburg, der Universität von Aveiro, von Traces des GÉANT-Netzes (vgl. [300]) sowie vom Konsortium Internet2 [217]. Auch ACM SIGCOMM stellt einige Datensätze zur Verfügung [349].

Einige Arbeiten haben sich mit der Erzeugung von realistischeren Synthetikdaten befasst, bspw. konnten Qian et al. eine deutlich realitätsnähere Erzeugung von HTTP-Verbindungsdaten und anderen Protokollen demonstrieren [317]. Hierfür wurden die synthetischen Daten des DARPA-Satzes mit den durch Qian erzeugten, synthetischen Daten sowie Realdaten aus einem Vergleichsnetz betrachtet. Kayacik und Zincir-Heywood haben ein Framework zur Erzeugung von HTTP-Datenverkehr auf Basis von Logdateien vorgestellt [225]. Mittels eines Self-Organizing Feature Maps (SOM)-basierten IDS wurden die Daten des KDD 99, des vorgestellten Ansatzes sowie eines realen Datenstromes ausgewertet und verglichen, wobei das Verfahren von Kayacik eine höhere Ähnlichkeit zu realem Datenverkehr aufweisen konnte, als der KDD-Datensatz. Für die Erzeugung von Angriffen zur Evaluation anomaliebasierter Systeme und deren Auswertung finden sich gesonderte Ansätze, bspw. bei Maxion und Tan [271].

Da die Auswertung von Flowdaten im Rahmen der verhaltensbasierten Angriffserkennung zunehmend an Bedeutung gewinnt, wird hier der Bedarf entsprechender Datensätze erforderlich. Entsprechende Datensätze werden durch das TOolbox for Traffic Engineering Methods (TOTEM)-Projekt zur Verfügung gestellt [20], ein markierter und anonymisierter Datensatz insbesondere zur Evaluation Flow-basierter IDSs wurde von Sperotto vorgestellt [359].

Trotz der verfügbaren Datensätze werden diese nur selten für die Evaluation und den Vergleich von Systemen herangezogen. Dies liegt insbesondere auch daran, dass die von realen Verbindungen gewonnenen Daten typischerweise nicht markiert sind, also keine Aussage getroffen werden kann, ob ein Paket einer Verbindung gut- oder bösartiger Natur ist. In der Praxis konnten sich bisher keine Datensätze neben DARPA 99 und KDD 99 hinreichend etablieren.

Werden Messungen der entsprechenden Parameter anhand eines bekannten und bewerteten Datensatzes wie bspw. den DARPA-Daten vorgenommen, können diese Werte anschließend für verschiedene Systeme verglichen werden. Hierbei muss jedoch, neben bereits zuvor geäußerten Bedenken bzgl. der zugrunde liegenden Vergleichsdaten, auch die jeweils genutzte Metrik genau betrachtet werden, um Fehlschlüsse zu vermeiden.

Zahlreiche Veröffentlichungen basieren auf der Verbesserung der Fehlalarmrate und optimieren entsprechende Algorithmen dahingehend (vgl. z.B. [318, 31, 245]). In diesem Kontext sei an das Phänomen der sog. *Base Rate Fallacy* erinnert, das im Anhang F.2.14 kurz beschrieben wird. Demnach gilt bei der Nutzung von Bayesschen Wahrscheinlichkeiten als Vergleichsmetrik von IDSs, dass auch bei hohen Detektions- und geringen Fehlalarmwahrscheinlichkeiten des Systems die Wahrscheinlichkeit, dass bei Alarmierung auch tatsächlich ein Angriff erfolgte, geringer liegen, als man dies gefühlsmäßig erwarten würde.

Das Information Technology Laboratory (ITL) des National Institute of Standards and Technology (NIST) erarbeitet Analysen und Empfehlungen für Thematiken mit besonderem Interesse aus dem Themenbereich der IT. Die Ergebnisse werden in den sog. NIST Interagency or Internal Reports (NISTIRs) veröffentlicht. NISTIR 7007 [276] gibt einen Überblick über die benötigten quantitativen Messungen im Bereich der IDSs, beschreibt die hierbei entstehenden Probleme und gibt Empfehlungen an die Forschergemeinschaft, diesen Problemen zu begegnen. Zu den Problemen, die immer noch bzgl. dem Vergleich von IDSs vorliegen, werden hier maßgeblich folgende Punkte aufgezählt [247]:

- Sammlung von Angriffsskripten und Opfersoftware
- Unterschiedliche Anforderungen beim Vergleich von anomalie- und signaturbasierten IDSs
- Nutzung von Hintergrundverkehr

Das NIST gibt daher die Empfehlung Möglichkeiten zu erforschen, Datensätze mit normalem und Angriffsverhalten geeignet zu erzeugen, zu markieren und zu verteilen, was bisher nicht befriedigend gelöst werden konnte. Die beste Aussagekraft hat somit derzeit ein Direktvergleich (Gruppentest) mehrere Systeme unter den gleichen Bedingungen.

4.5 State-of-the-Art Systeme

Nachfolgend wird ein Überblick über derzeitige State-of-the-Art Systeme geben, wobei sowohl Produktivsysteme als auch wissenschaftliche Ansätzen bzw. Prototypen herangezogen werden. Weitere in der Forschung befindliche Ansätze, die jedoch nur Verbesserungen in spezifischen Teilbereichen anstreben (z.B. Reduzierung der Fehlalarmrate, etc.) werden in den jeweiligen Abschnitten des Kapitels 4.6 aufgeführt.

Im Bereich der IPS werden regelmäßige Vergleichstest der wichtigsten, am Markt befindlichen Produkte vorgenommen. Der letzte IPS-Gruppen-Vergleichstest der NSS-Labs[10] führt 13 Systeme auf, die detailliert anhand zahlreicher Kriterien bewertet wurden[11], u.a. Erkennungsraten, Durchsatz bzgl. verschiedener Protokolle, Erkennung von Täuschungsversuchen, etc.

Zum Vergleich der Systeme werden in einer Testumgebung mit realistischem Netzverkehr eines Produktivnetzes[12] 1179 reale Exploits eingespielt.

Folgende Systeme wurden hierbei verglichen: Check Point Power-1 11065, Cisco IPS 4260, Endace Core-100 (IDS), Fortinet Fortigate 3810, IBM GX6116, Juniper IDP 8200, Juniper SRX 3600, McAffee M-8000, NSFOCUS NIPS 1200, Palo Alto Networks PA-4020, Sourcefire 3D 4500, Stonesoft IPS 1205 sowie Stonesoft IPS 3205.

[10] Vierteljährlicher Bericht, *Comparative Group Test Report*, IV. Quartal 2010, vom 10.01.2011.
[11] Testmethodik siehe [238].
[12] Zusammenstellung der Protokolle siehe *Test Methodology V6, 6.6 „Real-World“ Traffic* [241].

Die Gesamtstudie ist nicht kostenfrei erhältlich, allerdings sind die detaillierten Testberichte einzelner Systeme auf den Seiten der Hersteller abrufbar[13]. Insgesamt wurde im Durchschnitt eine Effektivität von 62 Prozent erreicht, d.h. 62 Prozent der durchgeführten Angriffe konnten durch die Systeme abgewehrt werden. Der Bericht stellt ebenfalls heraus, dass die Standardkonfigurationen vieler Systeme keine optimalen Ergebnisse erzielen und erst nach einer durch von Fachpersonal der jeweiligen Herstellerfirmen durchgeführten Optimierung verbessert werden konnten. Ohne diese Optimierungen lag die durchschnittliche Effektivität lediglich bei 31 Prozent. Auch der von den Herstellern angegebene, maximale Datendurchsatz konnte von vielen Systemen nicht erfüllt werden. Bei einem System wurde unter Last sogar **weniger als 3 Prozent** des angegebenen Wertes erreicht. Das Testlabor stellte ebenfalls fest, dass die Leistungsfähigkeit bzgl. des Durchsatzes der Systeme im Jahre 2010 generell rückläufig war.

Für die nachfolgende Bewertung von State-of-the-Art wurden drei der getesteten Systeme, die von den Test-Laboren im Vergleich ausgezeichnet wurden, aufgenommen: Check Point Power-1 11065, McAffee M-8000 sowie NSFOCUS NIPS 1200. Das IPS McAffee M-8000 erwies sich als das leistungsfähigste System mit dem höchsten Durchsatz (11533 Mbps) und den höchsten Detektionsraten im Vergleich der NSS-Labs. Mit den Voreinstellungen konnten 91.9 Prozent der eingespielten Angriffe geblockt werden. Hierfür wurde das System anhand von 1179 Exploits getestet. Dies entspricht nach Aussagen der NSS-Labs dem umfangreichsten Test in diesem Segment, verglichen mit der Anzahl aktuell vorhandener Schadsoftware stellt dies jedoch nur einen geringen Prozentsatz dar (vgl. Kapitel 4.6.1). Weiterhin muss berücksichtigt werden, dass insbesondere zielgerichtete Angriffe, Zero-Day Angriffe und neue, unbekannte Angriffe (welche nur anomaliebasiert erkannt werden können) hier nicht eingehen. Daher kann der durch die Labore ermittelte Leistungswert der jeweiligen Systeme nur für einen Vergleich der getesteten Systeme *untereinander* anhand der vorliegenden Bedingungen genutzt werden und muss für weitere Vergleiche kritisch betrachtet werden. Kommt das System an die Lastgrenzen seiner Ressourcen, werden neu eingehende Verbindungen geblockt. Check Point Power-1 11065 sowie NSFOCUS NIPS 1200 sind weitere durch die NSS-Labs ausgezeichnete Produkte mit Durchsätzen von 2433 Mbps bzw. 874 Mbps.

Von besonderer Bedeutung sind die Systeme Snort [355] und Bro [239], welche als Open Source zur Verfügung stehen. Mit 400000 registrierten Nutzern gilt Snort als defacto Standard im Bereich von NIDSs und wird durch die Firma Sourcefire weiterhin in mehreren Produkten kommerziell vermarktet. Verschiedene Erweiterungen sind verfügbar, bspw. um E-Mails nach Viren zu scannen oder für eine nahe-Echtzeit Analyse von mehrfach gepackten Dateien, um versteckte Schadprogramme zu finden. Bro untersucht zunächst die Semantik der Applikationsebene des Datenverkehrs und führt dann eine ereignisorientierte Analyse zur Suche nach bekannten Angriffspattern (sowohl Signaturen als auch Ereignisse) durch. Weiterhin werden Verhaltensanomalien ausgewertet.

Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) verei-

[13] Angemerkt sei, dass die Evaluationen der Systeme, welche Auszeichnungen erhalten haben, auf den jeweiligen Herstellerseiten zum Download verfügbar waren. Leider (aber verständlicherweise) konnten keine Berichte von *schlecht* beurteilten Systemen zum freien Download gefunden werden.

nigt Ansätze der signatur- und verhaltensbasierten Bereiche [216]. Es ist skalierbar für hohe Datenvolumen ausgelegt, in einem modularen Systemdesign umgesetzt und leicht an neue Regeln und Ziele anzupassen. PAYL ist ein Payload-basierter Anomaliendetektor [383]. Der normale zu beobachtende Applikations-Payload wird automatisch und ohne Überwachung modelliert, hierzu wird eine Verteilung der Paketfrequenzen erstellt. In der Detektionsphase erfolgt ein Vergleich der Mahalanobis-Distanz zwischen den neuen Daten und den Erwartungen des modellierten Profils. Packet Header Anomaly Detector (PHAD) lernt den normalen Wertebereich von 33 Feldern der Ethernet-, IP-, TCP-, UDP- und ICMP-Protokolle [264]. Im Gegensatz zu vielen anderen Systemen werden die meisten Angriffe unabhängig von einem anormalen IP-Adressbereich oder Portnummern detektiert. Bei einer Evaluation mittels des DARPA 99-Datensatzes konnten 29 von 59 Angriffstypen erkannt werden. Mittels der Erkennung von Verhaltensmustern in den Angriffen werden bei Learning Intrusion Detection System (LIDS) Attacken erkannt [112]. Das System ist agentenbasiert und kommuniziert über ein Blackboard, durch eine Online-Lernfähigkeit kann sich das System Änderungen in der Netzumgebung anpassen. Hu et al. nutzen einen AdaBoost-basierten Algorithmus zur Einbruchserkennung [201]. Das System setzt schwache Klassifizierer für sowohl kategorische als auch kontinuierliche Merkmale ein und kombiniert diese widerum zu einem starken Klassifizierer. Mit diesem Vorgehen erreicht das System eine Erkennungsrate von ca. 90 Prozent und eine Fehlalarmrate unter 2 Prozent. SSAD ist ein Algorithmus zur anomaliebasierten Auswertung von Datenströmen und basiert auf einem halb-unterstützten Lernverfahren, um den erforderlichen Umfang an markierten Daten für die Lernphase zu reduzieren [401]. Mittels einer Dämpfungsregel wird der Einfluß älterer Daten auf die Detektion reduziert. SilentDefense ist ein anomaliebasiertes IDS, das aus mehreren Komponenten kombiniert und konfiguriert werden kann [61]. Das Kernmodul bildet Poseidon, welches eine zweistufige Evaluation auf Basis einer SOM sowie eines modifizierten PAYL-Algorithmus [383] durchführt. Zur Reduzierung der Fehlalarmrate dient die Komponente Atlantides, die sowohl im Bereich signatur- als auch anomaliebasierter Verfahren eingesetzt werden kann. Sphinx ist ein webbasiertes IDS, das den Nutzer mittels regulärer Ausdrücke eine Kontrolle über den Detektionsvorgang ermöglicht. Zur automatisierten Klassifizierung von Anomalien dient schließlich das Modul Panacea. Boggs et al. tauschen Alarme zwischen verschiedenen Domänen aus, welche von Anomaliedetektoren in den jeweiligen Domänen erzeugt wurden [59]. Auf Basis der Korrelation dieser Alarme ist es möglich, Zero-Day-Angriffe zu erkennen.

Zahlreiche Ansätze nutzen spezielle Hardware, um insbesondere signaturbasierte Verfahren zu beschleunigen. Field Programmable Gate Arrays (FPGAs) werden bspw. von Kořenek und Kobierský genutzt, um die Detektionsleistung der Snort-Regeln zu erhöhen [232]. Das System erreicht einen Durchsatz von 6.4 Gbps auf Basis einer COMBO6X-Karte[14].

Otey et al. haben elementare verhaltens- und signaturbasierte Algorithmen auf einem Netzinterface implementiert [306]. Das Vorgehen soll insbesondere die Manipulationssi-

[14]Die Autoren geben in ihrem Paper *nicht* an, welches Snort-Regelwerk verwendet wurde, die Anzahl der umgesetzten Regeln ist somit unbekannt.

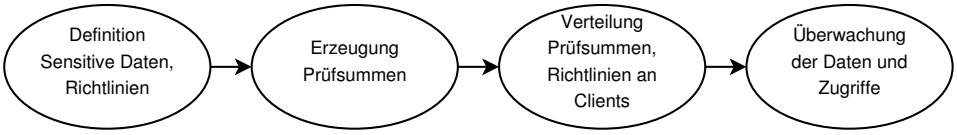


Abbildung 4.10: Grundlegende Funktionsweise eines DLP- Systems.

cherheit des IDS erhöhen. SafeCard ist ein IPS, das ebenfalls auf Basis eines programmierbaren Netzinterface (Intel IXP2400) basiert [114]. Das System vereint verschiedene Detektionstechniken wie Flow-Informationen, Payload-Evaluation und Auswertung von Feldern der Applikationsebene. Es ist in der Lage, auf einem Gigabit-Link eingesetzt zu werden. Ebenfalls auf spezieller Hardware setzt der Ansatz von Gao et al. auf [158]. Auf Basis von kaskadierten Ternary Content-Addressable Memories (TCAM) wird ein Signaturvergleich für lange Pattern unter hohen Bandbreiten möglich. Der Ansatz wurde bis 2 Gbps evaluiert, nach Angabe der Autoren lassen sich theoretisch Geschwindigkeiten bis zu 16 Gbps erreichen.

Die erste Umsetzung in Richtung Data Leakage Prevention (DLP) waren die Sicherheitserweiterungen des National Information Assurance Research Laboratory der NSA und der Secure Computing Corporation für den GNU/Linux-Kernel, welche unter dem Namen Security-Enhanced Linux (SELinux) 2000 als Open Source veröffentlicht wurden [28]. SELinux implementiert Zugriffskontrollen auf Ressourcen auf Kernel-Ebene. Seit 2001 haben mehrere Firmen Produkte im Bereich DLP herausgebracht, bspw. Code Green Networks [211] oder Trend Micro [278]. Für die Einführung eines DLP-Systems müssen zunächst die Daten bestimmt und definiert werden, die im jeweiligen Netz als sensitiv anzusehen sind. Hieraus erzeugt das DLP ein Datenmodell, das zum Vergleich der ein- und ausgehenden Daten genutzt wird. Einfache Systeme greifen hier widerrum auf Pattern-Matching zurück, somit ist die Funktionalität eines DLP in diesem Rahmen wie die eines entsprechend konfigurierten, den ausgehenden Netzverkehr überwachenden IPS. Weitere Verfahren, wie wörterbuchbasierte Suchen, die Nutzung von Prüfsummen oder Bayesschen Wahrscheinlichkeiten zur Berechnung, mit welcher Wahrscheinlichkeit ein Datensatz sensitiv ist, werden ebenfall genutzt [246]. Abbildung 4.10 zeigt schematisch die notwendigen Schritte eines DLP- Systems.

Im Bereich der Frühwarnsysteme befinden sich mehrere State-of-the-Art Systeme in Betrieb und Aufbau. Eines der ältesten Monitoring-Systeme im Internet ist das Internet Storm Center (ISC), das 2001 von SANS gegründet wurde und als Analyse- und Warnungsdienst für das Internet fungiert. Das System verfügt über Sensoren, die von über 500000 IP-Adressen rund um den Globus Firewall- und IDS-Logdaten sammeln, welche an die DShield-Datenbank des ISC übermittelt werden. Anormale Trends und Aktivitäten werden durch maschinelle Auswertung und aufgrund der Bewertung und Beurteilung eines Operateurs vorgenommen. Aufgrund der Auswertung wird die jeweilige Warnstufe im sog. Infocon-Monitor veröffentlicht [89], die möglichen Stufen reichen hier von *Grün* zur Anzeige von störungsfreiem Betrieb bis hin zu *Rot*, wobei massive Störungen und Konnektivitätsverlust in weiten Teilen des Internets auftreten. Mehrere

andere Frühwarnsysteme überwachen ebenfalls das Internet oder analysieren insbesondere länderspezifische Teilbereiche, bspw. ARAKIS des polnischen Computer Emergency Response Team (CERT) [289] oder werden zur umfassenden Auswertung der Netzsicherheit von Kunden betrieben, bspw. Active Threat Level Analysis System (ATLAS) von Arbor Networks [293] oder DeepSight Early Warning System (EWS) von Symantec [107]. Das Projekt Worldwide Observatory of Malicious Behaviors and Attack Threats (WOMBAT) untersucht neue Ansätze zur Gewinnung von Echtzeitdaten, Analysetechniken sowie Verfahren zur Ursachen-Erkennung [122]. In Kapitel F.2.15 ist der typische Aufbau eines EWS beschrieben.

Tabelle 4.3 zeigt eine Zusammenfassung der State-of-the-Art Systeme. Die Systeme der Gruppen DLP und EWS wurden aufgrund der Ähnlichkeit der Systeme und der nur wenig verfügbaren Detailinformationen zusammengefasst.

Ein zusätzlicher Entscheidungsfaktor kann sich aus den Kosten des Systems ergeben. Während einige Systeme als Open Source kostenlos zur Verfügung stehen, bspw. Snort oder Bro, können die Anschaffungskosten proprietärer Systeme auch in einen sechsstelligen € Bereich gehen, hierzu können noch weitere Kosten wie Lizenzen und Wartung kommen. Bspw. kostet die *McAffee Network Security Platform M-8000* 224995 US-Dollar im Anschaffungspreis[15], unter Berücksichtigung der durch Installation und Wartung entstehenden Kosten beläuft sich die Total Cost of Ownership (TCO) für ein Jahr auf 270740 US-Dollar, für drei Jahre auf 361030 US-Dollar [242]. Diese Ausgaben sind für ein kleines Unternehmen, wie es ebenfalls im Szenario in Kapitel 2.1 betrachtet wird, nicht zu tragen. Daher müssen die Kosten für ein System ggf. mit in den finalen Auswahlprozess einbezogen werden.

4.6 Schwachstellenanalyse

Wie die Evaluation von State-of-the-Art Systemen in Kapitel 4.5 gezeigt hat, können aktuelle Systeme keinen umfassenden Schutz, wie er für das dargestellte Szenario benötigt wird, gewährleisten. Maßgeblich wurden folgende Bereiche festgestellt, welche durch die vorhandenen State-of-the-Art Systeme nicht befriedigend oder gar nicht erfüllt werden können:

- Genauigkeit Detektionsverfahren (Anomalie- und signaturbasierte Verfahren, Verfahren und Evolution von Schadsoftware)
- Risiko von Datenverlust und Innentätern (Erkennen von Datenabfluß, Verhindern von Datenabfluß)
- Wartungsfreiheit
- Entwicklung und Wachstum Datenverkehr (Skalierbarkeit)
- Einsatz in verschlüsselten Netzen

[15] Stand: September 2010.

Tabelle 4.3: Vergleichsmatrix des Anforderungskataloges für ein Sicherheitssystem.

	Industrie				Wissenschaft								
	McAffee M-8000	NIPS 1200	Snort	*DLP*	EMERALD [SRI00]	PAYL [Wang04]	PHAD [Mahoney01]	LIDS [Cannady03]	SSAD [Yu09]	SilentDefense [Bolzoni09]	[Korenek07, Kong09, ...]	[Otey03, DeBruijn06]	TCAM [Gao06]
Angriffs-detektion[1]	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reaktion	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	(✓)	✗
Erkennen Innentäter	✗	✗	(✗)	(✓)	✗	✗	✗	✗	✗	✗	✗	✗	✗
Transparente Integration	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗
Wartungsarm	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗
Echtzeit-auswertung	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Skalierbarkeit	✓	✓	✗	✗	✗	✓	✓	(✓)	✗	✓	✗	(✓)	(✓)
Verschlüsselte Netze	✗	✗	✗	✗	✗	✗	(✗)	✗	✗	✗	✗	✗	✗

✓ Erfüllt ✗ Nicht erfüllt () Mit Einschränkungen

- Rechtskonformität

Der Bereich *Reaktion* wird ebenfalls durch zahlreiche Systeme nicht hinreichend erfüllt, in dem Sinne, dass es sich dabei um reine Detektionssysteme (IDSs) handelt, die keine entsprechenden Reaktionen beim Erkennen eines Angriffs initiieren (im Gegensatz zu IPSs). Da entsprechende reaktive Maßnahmen, typischerweise eine Sperrung von Ports, einzelner IPs oder ganzer IP-Blöcke auf einer Firewall jedoch relativ einfach anhand eines Signales wie bspw. einem Alarm integriert werden können, wird dieser Bereich nicht tiefer betrachtet. Insbesondere bleibt aber die Forderung bestehen, dass ein Sicherheitssystem in der Lage sein muss, entsprechende Reaktionen bei Detekton zu ergreifen. Gao leitet aus der Notwendigkeit einer *sofortigen* Reaktion zur Verhinderung von Schäden die Erfordernis ab, dass hierfür ein netzbasiertes System eingesetzt werden muss, da dieses frühzeitiger auf ein erkanntes Event reagieren kann als ein hostbasiertes System [158].

Nachfolgend werden die Ursachen der aufgeführten Herausforderungen im Detail analysiert.

4.6.1 Detektionsverfahren

Wie bereits in Kapitel 4.2 gezeigt, existieren im Bereich der Einbruchserkennung maßgeblich zwei Detektionsverfahren, wissensbasierte sowie verhaltensbasierte Detektion. Während die Kombination beider Verfahren theoretisch einen umfassenden Schutz eines Systems ermöglichen sollte, liegen in der Praxis für beide Verfahren zahlreiche Herausforderungen und Einschränkungen vor.

Wissensbasierte Detektion

Die Haupteinschränkung der signaturbasierten Analyse ist die systemimmanente, reaktive Arbeitsweise [161]. Da für jeden Angriff bzw. jede Schadsoftware eine entsprechende Signatur in der Datenbank vorhanden sein muss, können insbesondere neue, unbekannte Angriffe und für ein Ziel speziell angepasste Methoden nicht erkannt werden.

Das Durchsuchen des Datenverkehrs ist weiterhin ein relativ aufwändiger Vorgang, der bis zu 80 Prozent der gesamten durch ein IDS erzeugten Systemlast ausmachen kann [269, 356]. Bei den heute verfügbaren hohen Datenraten kann es daher durch die Analyse des Payloads zu Engpässen und Performance-Einbrüchen kommen [231]. Aufgrund dessen sind softwarebasierte NIDS kaum in der Lage, Datendurchsätze über 200 Mbps zu analysieren (vgl. [361, 232]).

Um diesen Einschränkungen zu begegnen, wurden in den letzten Jahren zahlreiche hardwarebasierte Ansätze vorgestellt, die den Einsatz signaturbasierter Verfahren unter höheren Datenraten ermöglichen. Der erste hardwarebasierte Ansatz für Textsuche wurde von Pryor et al. bereits im Jahre 1993 vorgestellt [314]. Mittels der Nutzung von Splash2-Systemen[16] konnten hierbei 50 Millionen Zeichen pro Sekunde für eine textba-

[16] Am Supercomputing Research Center, Bowie, MD, USA (SRC) entwickelte Architektur bestehend aus bis zu 16 sog. Splash-Boards, jedes bestückt mit 16 Xilinx 4010 FPGAs.

sierte Schlüsselwortsuche erreicht werden.

Heutzutage können zwei Ansätze im Bereich der hardwarebeschleunigten Verfahren unterschieden werden, Logik- und Speicherarchitekturen [250]. Logikarchitekturen nutzen hauptsächlich Logikbausteine von FPGAs, um die regulären Ausdrücke der Signaturensuche in Zustandsmaschinen oder kombinatorische Verfahren umzusetzen. Hierdurch werden deutliche Geschwindigkeitszuwächse erreicht und entsprechend höhere Bandbreiten möglich (vgl. z.B. [280, 47, 171]). Andererseits muss berücksichtigt werden, dass bei diesen Verfahren erhebliche Ressourcen des Chips für die Umsetzung der Regelsätze benötigt werden. Kong et al. argumentieren daher, dass diese *On-Chip* Verfahren auf Dauer nicht skalieren werden [231]. Zahlreiche Arbeiten zielen daher darauf ab, die erforderliche Anzahl von Logikeinheiten pro Suchcharakter zu reduzieren (vgl. z.B. [250, 357]). Die hohe Resourcennutzung von FPGAs bereits durch leichtgewichtige IDSs wie Snort werden von Gao kritisiert [158]. Insbesondere führen die wachsenden Regelsätze und die Nutzung der erforderlichen FPGAs mit einer entsprechenden Anzahl von Logikelementen zu höheren Latenzzeiten der Verbindungsstrukturen. Diese widerrum äußert sich in einer geringeren, maximal nutzbaren Taktfrequenz und senkt somit den maximal erreichbaren Datendurchsatz. Eine weitere, bedeutende Einschränkung bei der Nutzung von FPGAs ist die Notwendigkeit der Synthese- und Designschritte sowie die Rekonfiguration des Chips. Während dieser Zeit ist ein entsprechendes System nicht in der Lage, eine Detektion durchzuführen, es muss also die Updaterate bzw. das entsprechende zeitliche Intervall, sowie die für ein Aufspielen und Rekonfigurieren des Chips notwendige Zeit berücksichtigt werden. Gao nutzt daher TCAMs für die Speicherung und das effiziente Durchsuchen von Signaturen. Bei TCAM handelt es sich um ein hardwarebasiertes Suchverfahren, das deutlich effizienter ist als algorithmische Ansätze [308]. Insbesondere weist diese Art Speicher die Eigenschaft auf, zu einer Eingabezeichenfolge *genau ein* passendes Ergebnis innerhalb von $O(1)$, also innerhalb eines Taktes, zurückzugegeben [158]. Hierbei werden bei einem ternären CAM drei Bit-Zustände ermöglicht: 0, 1 sowie der Zustand „don't care“. Einschränkungen bestehen insbesondere durch die feste Datenbreite von typischerweise 64 Bytes, was eine entscheidente Restriktion für die Suche nach Signaturen darstellt; der Ansatz von Gao ermöglicht die Kaskadierung von TCAMs und ermöglicht somit eine Suche nach langen Signaturen.

Speicherarchitekturen haben gegenüber den Logikarchitekturen entsprechend den Vorteil, dass umfangreichere Signaturbanken untergebracht werden können. Die jeweiligen Pattern werden bei diesem Verfahren in einen endlichen Automaten übersetzt, die zugehörigen Zustandsübergangstabellen werden im Speicher abgelegt [250]. Techniken wie das Teilen von gleichen Präfix- oder Suffix-Zeichenfolgen [231] oder Vorfilterung [232] ermöglichen Geschwindigkeiten von 4 Gbps bzw. 6.4 Gbps. Hierdurch werden fliegende Aktualisierungen der Signaturen möglich, ohne dass eine Synthese oder Rekonfiguration erforderlich ist. Speicherarchitekturen sollten daher besser in der Lage sein, den Anforderungen bzgl. steigender Signaturzahlen und wachsender Datenraten stand zu halten.

Andere hardwarebasierte Lösungen sind bspw. die Integration eines IDS direkt auf der Network Interface Card (NIC). Otey et al. führen hierbei eine reine headerbasierte Eva-

luation durch [306], während der Ansatz von Bruijn et al. eine umfangreiche Analyse auf mehreren Abstraktionsebenen umsetzt [114]. Neben der Auswertung von Paketinformationen fliessen hierbei bspw. auch Datenströme, etc. mit ein. Das System ist auf einem Gigabit-NIC implementiert. Eine kommerziell umgesetzte Variante hierbei ist bspw. bei einer Serie von 3Com-Netzkarten[17] erhältlich. Diese integrieren allerdings nur grundlegende Firewall-Funktionalitäten, Paket-Analyse, etc. wird nicht durchgeführt.

Entsprechende Verfahren bzw. direkt auf der NIC installierte Sensorik kann künftig noch eine höhere Bedeutung erhalten, da mittlerweile das Netzinterface auch zum Einschleusen von Schadsoftware mißbraucht werden kann (vgl. z.B. [119]).

Trotz ihrer aufgrund des Verfahrens grundsätzlich besseren Eigenschaften bezüglich Fehlalarmen, erfordern gerade signaturbasierte Verfahren einen hohen administrativen Aufwand, um die Fehlalarmraten in einem akzeptablen Rahmen zu halten. Betrachtet man bspw. die Standardinstallation eines Snort-Systems in einer produktiven Netzumgebung mit ca. 150 Rechnern und Servern, ergeben sich täglich mehrere tausend Alarme und Warnungen (vgl. Anhang F.2.3).

Um die Anzahl der Alarme auf ein akzeptables Niveau zu senken, muss die IDS-Installation detailliert an die Einsatzumgebung angepasst werden. Hierfür ist es u.a. notwendig, sämtliche Rechner, Server und Netzperipherie wie bspw. Drucker inkl. der jeweiligen Betriebssysteme, Dienste und Diensteversionen anzugeben. Weiterhin muss das Regelwerk auf nicht benötigte Regeln hin untersucht werden, die entsprechend zu deaktivieren sind. Nach der Anpassung an das Einsatznetz muss die Installation stetig gepflegt werden, bspw. bei Updates von Systemen oder Programmversionen, etc.

Somit ergeben sich für signaturbasierte Verfahren insbesondere folgende Probleme:

- Detektion nur bekannter Angriffe
- Datenbankgrößen und Bandbreiten erfordern den Einsatz dedizierter Hardware
- Aktualität der Datenbanken
- Hoher Konfigurations- und Pflegeaufwand

Verhaltensbasierte Detektion

Im Gegensatz zur wissenbasierten Detektion erfordert ein verhaltensbasierter Ansatz keine Datenbank o.ä, in der Pattern vorgehalten werden müssen, stattdessen wird anhand der Erwartung eines Modells des Netzes und dem Vergleich zum aktuellen, gemessenen Zustand eine Bewertung vorgenommen. Die Modellierung des normalen Verhaltens eines Systems oder Netzes ist komplex und ein aktives Forschungsgebiet (vgl. z.B. [377]). Eine Schwierigkeit bei verhaltensbasierten Ansätzen liegt darin, dass legitime, jedoch zuvor noch nicht gesehene Aktionen von Nutzern als Angriffe missinterpretiert werden, wodurch die Fehlalarmrate entsprechender Systeme im Vergleich zu signaturbasierten

[17] 3CR990-Serie, veröffentlicht 2001, nach der Übernahme von 3Com durch HP im Jahre 2010 Umbenennung der Modelle in *HP Secure*.

Verfahren steigt. Um das notwendige Wissen über die Systemumgebung zu erhalten, verwenden viele Systeme eine Lernphase, in der die Eigenschaften des Netzes wie bspw. genutzte Dienste, Datenmengen und tageszeitliche Schwankungen analysiert werden. Hierbei müssen Online- und Offline-Lernverfahren unterschieden werden: Beim Online-Lernen werden die Daten in inkrementeller bzw. sequentieller Weise zur Analyse zur Verfügung gestellt; der Lernvorgang erfolgt dann in serieller Art und Weise, jeweils auf einem ausgewählten oder zufälligen Sample. Wird ein Offline-Ansatz gewählt, wird der gesamte Datensatz der Problemstellung in einer Lerniteration analysiert [57].

Um die richtigen Reaktionen anhand der vorgestellten Daten zu erlernen, benötigen zahlreiche Algorithmen eine entsprechende Markierung, ob die präsentierten Daten gut- oder bösartiges Verhalten darstellen (vgl. z.B. [372, 248]). Abhängig davon, ob die Daten für die Lernphase synthetisch erzeugt werden oder realer Netzverkehr genutzt wird, entstehen verschiedene Probleme (vgl. auch Kapitel 4.4). Während die Nutzung synthetischer Daten typischerweise nicht in der Lage ist, das Verhalten eines realen Netzes vollständig naturgetreu wiederzuspiegeln, bringt die Nutzung der realen Netzdaten die Problematik der Markierung und die Gefährdung durch nicht erkannte, in den Daten vorhandener Angriffe mit sich.

Almgren und Jonsson nutzen daher ein aktives Lernverfahren, um den hohen Bedarf markierter Daten signifikant zu reduzieren [32].

Zahlreiche State-of-the-Art Systeme benötigen jedoch eine Lernphase in der produktiven Netzumgebung, um das erforderliche Modell für die Evaluation aufzubauen (vgl. z.B. [213, 354, 240]). Dies birgt die Gefahr, dass Schadsoftware, die sich bereits im Netz befindet, oder Angriffe, welche während der Lernphase durchgeführt werden, als normales Netzverhalten erlernt werden und somit ein fehlerhaftes Netzmodell erzeugt wird [372]. Später kann das entsprechende, böswillige Verhalten nicht vom System erkannt werden (vgl. z.B. [115, 228, 60]). Aufgrund der Ähnlichkeiten von Schadsoftwarefamilien können somit *zahlreiche* Angriffe vor dem IDS unerkannt, bzw. falsch als gutartig interpretiert werden. Diese Gefährdung wird insbesondere verstärkt, da die Lernphase typischerweise eine bis mehrere Wochen dauert, um die erforderliche Datenbasis zu gewinnen und die zeitlichen Schwankungen zu analysieren (vgl. z.B. [240, 213]).

Lernphase verhaltensbasierter Systeme Neben der oben vorgestellten Gefährdung der Lernphase durch bereits in der Netzumgebung vorhandenen Schadsoftware kann diese auch gezielt manipuliert werden, wenn ein Angreifer oder Insider Kenntnis über deren Durchführung hat. Die typische, mehrwöchige Dauer der Lernphase eröffnet hierbei zahlreiche Angriffspunkte.

Für die Erstellung der notwendigen Modelle werden von den State-of-the-Art Systemen typischerweise Flow-Informationen verwendet (vgl. Kapitel F.2.6, die durch aktive Netzkomponenten wie Router oder Switches erzeugt werden. Diese analysieren den weitergeleiteten Datenverkehr und erzeugen daraus Flow-Pakete mit statistischen Informationen, die zur weiteren Verarbeitung an entsprechende Analysesysteme gesendet werden können.

Da hierfür typischerweise keine getrennten Netze zur Datenübermittlung verwendet

werden, jedoch selbst beim Vorhandensein dieser durch die Gefahr von Innentätern oder bereits kompromittierter Systemen ebenfalls mit einer Manipulationsmöglichkeit gerechnet werden muss, ergibt sich hier ein angreifbarer Kanal. Um die daraus resultierende Verwundbarkeit verhaltensbasierter Analyseverfahren zu untersuchen, wurden umfangreiche Evaluationen mit verschiedenen Auswertesystemen sowohl mittels sFlow, als auch mittels NetFlow durchgeführt (vgl. Kapitel F.2.9). Die Auswertungen zeigen, dass die Lernphasen verhaltensbasierter Systeme einen gefährlichen Angriffspunkt darstellen und Verhalten, welches auf Basis der Evaluation von Flowdaten generiert wird, leicht beeinflußt werden kann.

Reduzierung der Lernphase Um die Gefährdungen für das Sicherheitssystem zu minimieren, welche durch die Notwendigkeit einer Lernphase entstehen, können verschiedene Ansätze herangezogen werden. Winter et al. schlagen vor, dass die Lernphase anhand von bösartigem Verhalten durchgeführt wird, anstatt der Nutzung des gutartigen Datenverkehrs des Netzes. Sie nutzen eine Ein-Klassen SVM zur Analyse von Flow-Daten, nachdem das System mit negativem Verhalten angelernt wurde [388]. Das Verfahren zeigt in manchen Bereichen deutlich bessere Ergebnisse bzgl. der Fehlalarmraten als andere verhaltensbasierte Verfahren, ist jedoch stark abhängig von der Nutzung von Source- und Destination-Ports, was eine reale Anwendung erheblich einschränkt. Weiterhin muss berücksichtigt werden, dass auch bei der Nutzung von bösartigem Verhalten die Vollständigkeit der Lernphase regelmäßig nicht zu gewährleisten ist. Bspw. wurde daher das System von Winter nur für die Untersuchung von HTTP- und SSH-Verbindungen evaluiert.

Aus den oben aufgeführten Gründen heraus ist der Verzicht auf eine Lernphase anzustreben. Hierzu können nicht-überwachte Lernverfahren wie bspw. k-means oder SOM (vgl. Kapitel 4.2) als Basis herangezogen werden. Stellenweise finden sich auch Ansätze, welche überwachte und nicht-überwachte Verfahren kombinieren. Bspw. nutzen Carrascal et al. eine Hierarchie aus SOMs zur Modellierung des Datenverkehrs und lernende Vektorquantisierung zur Klassifizierung der Netzpakete [86]. Ziel der Arbeit ist die Senkung der Fehlalarmrate auf ein für eine Produktivumgebung akzeptables Niveau. Den Autoren ist es mittels ihres Verfahrens möglich, die Fehlerrate im Vergleich zu anderen Techniken deutlich zu reduzieren, jedoch nur auf Kosten der Detektionsrate (vgl. Tabelle 4.4).

Angemerkt werden muss, dass die präsentierten Werte auf einem Vergleich mittels der DARPA-Datensätze basieren. Somit ist lediglich ein Vergleich der Verfahren untereinander möglich, eine Aussage über die tatsächliche Detektionsfähigkeit in einer produktiven Umgebung kann nur ungenügend abgeleitet werden (vgl. Kapitel 4.4).

Casas et al. schlagen ein robustes Clustering-Verfahren vor, um abnorme Daten-Flows mittels eines temporären, sequentiellen Schiebefensters zu erkennen [87]. Mittels der durch das Clustering gefundenen Anomalien werden automatisch Filter-Regeln erzeugt, die zu Anomalie-Signaturen kombiniert und wiederum an andere Sicherheitssysteme wie Firewalls oder IDSs exportiert werden können. Der Ansatz ist vollständig

Tabelle 4.4: Detektions- und Fehlalarmraten verschiedener Lernverfahren nach [86].

Verfahren	Detektionsrate	Fehlalarmrate
Clustering	93 %	10 %
k-NN	91 %	8 %
SVM	98 %	10 %
Hierarchische SOM	89 %	7.6 %
Kombiniert (Carrascal)	72 %	2 %

nicht-überwacht und kann Angriffe ohne die Notwendigkeit von Signaturen, markierten Trainings-Datensätzen oder einer Lernphase erkennen. Das System wird direkt in der Netzumgebung angeschlossen und kann dann unmittelbar mit der Analyse beginnen. Da die Evaluation jedoch maßgeblich auf Parametern wie IPs, Ports und Flags beruht, können zielgerichtete Angriffe, etc., nicht detektiert werden.

Folgende Probleme treten somit für den Einsatz verhaltensbasierter Verfahren auf:

- Unvollständigkeit des Modells und somit Fehlalarme
- Notwendigkeit angreifbarer Lernphasen vieler Systeme in der produktiven Netzumgebung
- Keine Detektion von zielgerichteten Angriffen
- Verlust der Echtzeitfähigkeit durch die Nutzung von Flows
- Keine Detektion von Innentätern

Evolution von Schadsoftware

Wie bereits in Kapitel 2.3.2 angerissen, haben sich die auf Rechnersysteme durchgeführten Angriffe in den letzten Jahrzehnten grundlegend gewandelt. Während anfangs die reine Zerstörung von Daten im Vordergrund stand, hat sich mit dem kommerziellen Erfolg des Internets ein umfassender Untergrundhandel aufgebaut, der professionalisiert und mit ausgeprägten Geschäftsmodellen Angriffe auf Nutzer und Rechner durchführt.

Abbildung F.17 in Kapitel F.2.10 zeigt einen Überblick der Entwicklung der Angriffstechnologien im Vergleich zum für den Einsatz dieser erforderlichen Wissen. Es liegt hierbei eine stetige Professionalisierung der Angriffswerkzeuge vor, welche mittlerweile sämtliche Bereiche beginnend mit der Suche nach Schwachstellen, der automatisierten Anwendung von Exploits, dem Abhören von Datenverkehr und der Nutzung von Verschleierungstechniken abdeckt und selbst die einfache Steuerung kompletter Botnetze ermöglicht.

Proliferation von Schadsoftware Während sich selbst verbreitender Code zu Beginn als eine Forschungsarbeit im Bereich der Parallelisierung darstellte, wurden die Verfahren schon bald für Angriffe auf Rechner und Daten genutzt (vgl. Kapitel 2.3.2). Durch den kommerziellen Erfolg und die Möglichkeiten, Angriffe aus sicherer Entfernung durchzuführen, haben sich Schadprogramme mit exponentieller Geschwindigkeit vermehrt. Im Jahre 2010 zählte die Symantec Corporation bereits über 286 Millionen verschiedener Varianten von Schadsoftware [142]. Durch die immer professionelleren Angriffstoolkits, welche im Baukastenprinzip die Erstellung neuer Schadprogramme ermöglichen, entstehen immer mehr Schadcodevarianten in kürzeren Zeiten. Sobald ein Schadcode durch einen Hersteller analysiert und eine entsprechende Signatur für seine Sicherheitssysteme veröffentlicht wurde, erzeugen die Angreifer leicht modifizierte Varianten ihres Angriffs, um wiederum unentdeckt zu bleiben. Die effiziente Signatursuche durch die einerseits zahlreichen Pattern, die großen Datenmengen heutiger breitbandiger Verbindungen andererseits, ist damit eines der Hauptprobleme der Missbrauchserkennung geworden. Die immense Anzahl von Signaturen sorgt dafür, dass die zugehörigen Datenbanken bereits heute nur noch unter Schwierigkeiten für effiziente Suchen nach Schadcode genutzt werden können (vgl. Kapitel 4.6.1).

Der Handel mit Angriffscode für noch nicht öffentlich bekannt gewordene Schwachstellen (Zero-Day-Exploits) ist durch das enorme, erreichbare Schadenspotential zu einem lukrativen Sektor im Untergrundmarkt geworden. Verschiedene Studien und Berichte zeigen, dass für entsprechende Exploits Summen bis zu 80000 US-Dollar und mehr geboten werden.

Zero-Days und Pattern-Aktualität Die Anzahl der Schwachstellen in Programmen als auch der Zero-Days hat in den letzten Jahren zugenommen, was sich durch die zunehmende Komplexität insbesondere der Anwendersoftware erklären lässt. Nachdem im Jahre 2009 ein leichter Rückgang der neu entdeckten und öffentlich bekannten gewordenen Schwachstellen von 5491 im Jahre 2008 auf 4501 zu verzeichnen war, hat die Zahl 2010 ein neues Rekordhoch mit 6253 neu entdeckten Schwachstellen erreicht. Eine besondere Gefährdung entsteht hierbei durch die Gruppe der Zero-Days (vgl. Kapitel 2.3.3), da in diesem Falle eine ausnutzbare Schwachstelle noch nicht bekannt wurde. Während in den Jahren 2008 und 2009 von 9 bzw. 12 Zero-Days berichtet wurde [141], waren es im Jahre 2010 bereits 14 entsprechende Schwachstellen [142]. Der Zeitstrahl in Abbildung 4.11 gibt eine Darstellung der wichtigsten Stufen zwischen der Entdeckung einer Schwachstelle und der Bereitstellung eines entsprechenden Patches wieder. Von besonderer Bedeutung sind die Dauer des Wirkungsbereiches eines Zero-Day-Exploits sowie die Zeitspanne zwischen dem Bekanntwerden beim Hersteller und der Verfügbarkeit des Patches, die Verwundbarkeitszeit (Window of Exposure).

Jedoch ist auch die Gefährdung durch Schwachstellen, die bereits länger öffentlich bekannt sind, nicht zu unterschätzen. Einige Hersteller schließen entsprechende Lücken erst nach geraumer Zeit (vgl. z.B. auch Kapitel 4.6.1), weiterhin ist das Update-Verhalten der Nutzer oftmals unzureichend: Die am häufigsten durchgeführten Angriffe zielen oftmals gegen Schwachstellen ab, für die es bereits für längere Zeit entsprechende Patches

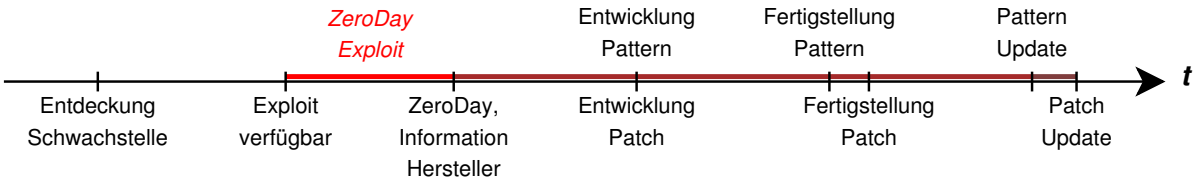


Abbildung 4.11: Zeitstrahl der Gefährdung durch einen Zero-Day-Exploit.

verfügbar gibt.

Um Angriffe auf mögliche Schwachstellen erkennen zu können, müssen bei signaturbasierten Verfahren entsprechende Pattern in der jeweiligen Datenbank des Sicherheitssystems vorhanden sein. Dementsprechend ist die Zeitspanne zwischen dem Bekanntwerden einer Lücke und der Veröffentlichung einer zugehörigen Signatur für das Sicherheitssystem von entscheidender Bedeutung. Abbildung 4.12 zeigt die Erkennungsraten bekannter Antiviren-Produkte verschiedener Hersteller im Oktober 2010 an. Während die höchste Detektionsrate bei 92.08 Prozent lag, waren einige Produkte nur in der Lage, einen Bruchteil der eingespielten Schadsoftware zu erkennen. Dies zeigt nachhaltig die Problematik der signaturbasierten Verfahren, welche durch die Notwendigkeit von aktuellen Pattern entstehen. Die hierfür erforderlichen Updateraten können erheblich zwischen den verschiedenen Herstellern schwanken, Abbildung 4.13 stellt die Anzahl der Updates in der 43. und 44. Kalenderwoche 2010 für verschiedene Produkte dar.

Ein ähnliches Bild zeigt sich, wenn man die Update-Intervalle für die Signaturen der IDSs verfolgt: Hier liegen typischerweise größere Intervalle bzgl. den Updates der Signaturdatenbanken vor, als dies bei Virenscannern der Fall ist. Oft erscheinen entsprechende Aktualisierungen nur im Rhythmus mehrere Tage oder sogar Wochen. Bspw. waren die aktuellen Signaturen von wichtigen IDSs am 20. Januar 2011 wie folgt datiert:

- *Juniper IDP DI*, 18. Januar 2011
- *IntruShield*, 11. Januar 2011
- *Sourcefire IPS*, 18. Januar 2011
- *Proventia*, 11. Januar 2011

Die entsprechend aktuellen Versionen und Aktualisierungen werden durch das Talisker-Radar [258] geführt und stehen öffentlich zur Verfügung. Vergleicht man diese Zeitspannen mit der durchschnittlichen Anzahl an täglich neu erscheinenden Signaturen (ca. 8000, vgl. [141]), entsteht hier ein erhebliches Schutzdefizit. Lippmann analysierte die Auswirkung der Identifizierung von Schwachstellen und Erstellung von Software-Patches mit Hinblick auf IDSs und den Aktualitäten von deren Signaturdatenbanken [254]. Die Evaluation zeigt, dass Signaturen oftmals nicht schneller verfügbar sind, als die jeweiligen Patches der Hersteller zur Behebung der Schwachstellen. Dies deckt sich auch weiterhin mit den angegebenen Beobachtungen der Signatur-Updates und der Veröffentlichung

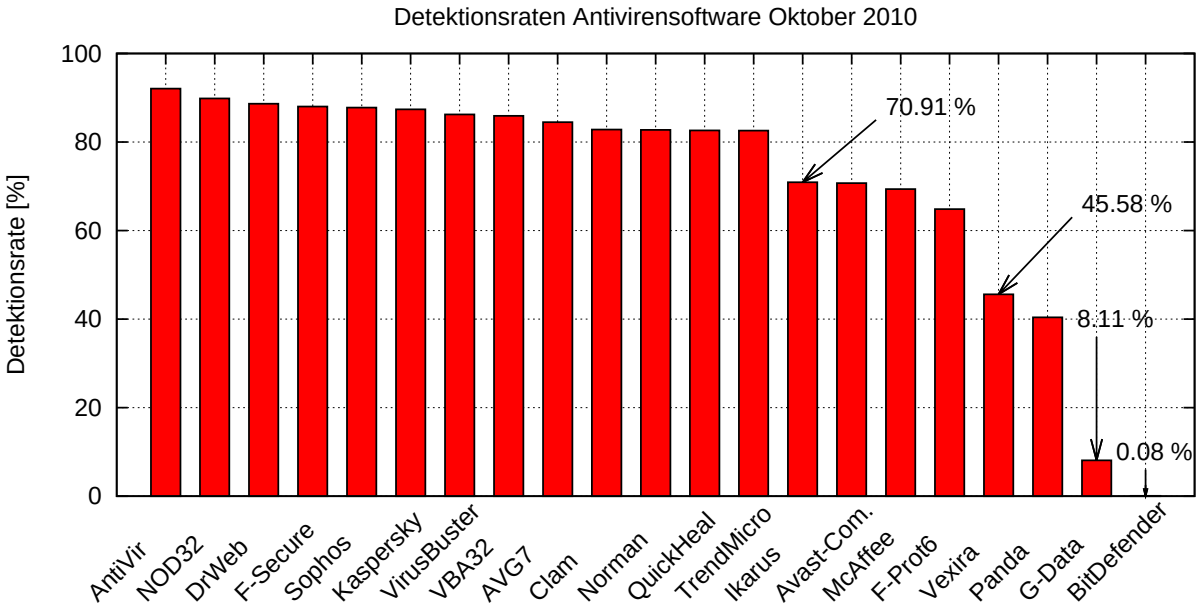


Abbildung 4.12: Erkennungsraten von Virenscannern verschiedener Hersteller im Oktober 2010 (Auswertung gem. [144]). Gute Produkte haben Detektionsraten von 85 bis 95 Prozent der bekannten Schadsoftware, manche Produkte erkennen in der Praxis jedoch nur Bruchteile der eingespielten Schadprogramme. Oftmals zeigen Vergleichstests von Antivirensoftware durchgehend sehr hohe Detektionsraten von über 98 oder 99 Prozent. Hier muss die angewandte Methodik betrachtet werden, oftmals entstehen solche Ergebnisse mit kleinen Virendatenbanken, die bereits Wochen vor dem Vergleich eingefroren wurden, Zero-Days, welche die eigentliche Gefahr darstellen, werden hierbei nicht betrachtet.

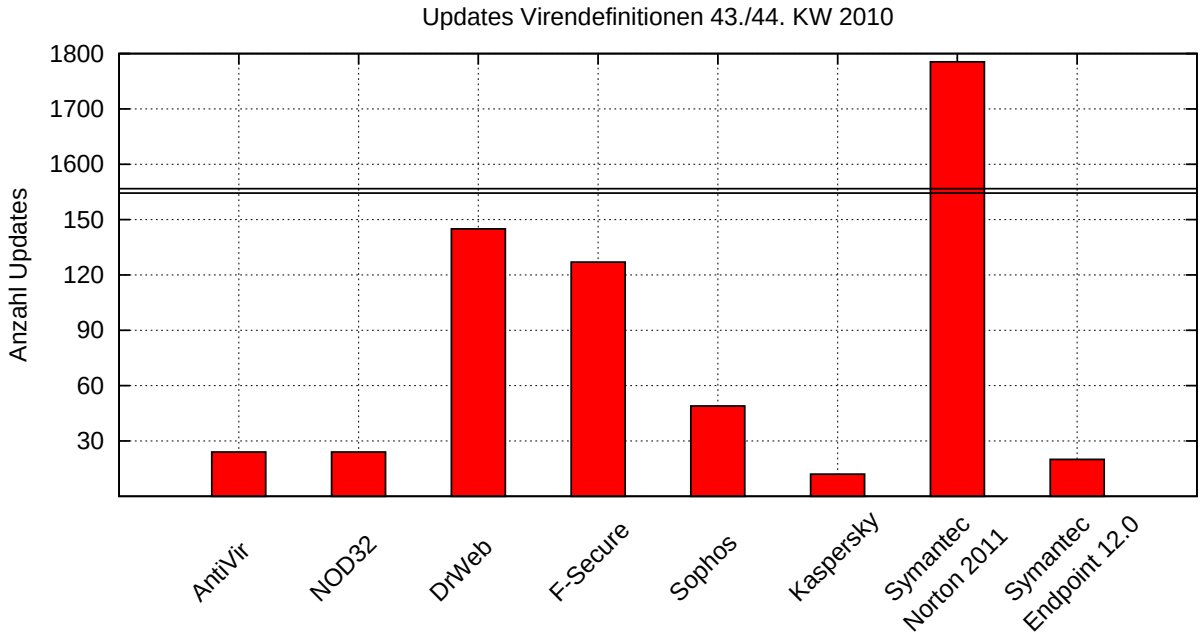


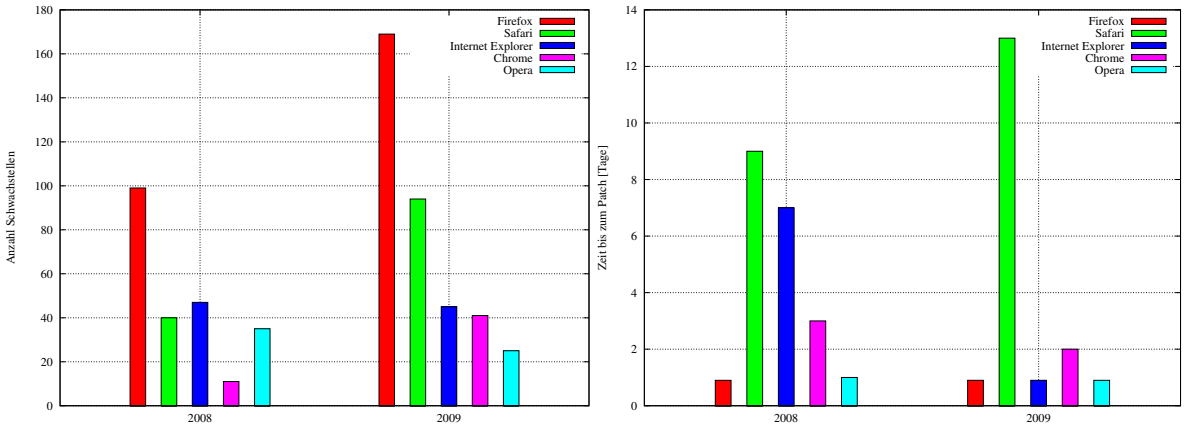
Abbildung 4.13: Updateraten der Signaturen für Produkte verschiedener Hersteller in der 43. und 44. KW 2010 (Auswertung gem. [37]). Aufgrund der steigenden Anzahl von neuen Signaturen, die in einer Größenordnung von mehreren hundert neuen Schadcodevarianten pro Tag liegt, sind kurzfristige und regelmäßige Updates unerlässlich.

von Patches. Andererseits verzögern jedoch Anbieter wie Microsoft, Adobe oder Oracle die Veröffentlichung von wichtigen Updates zur Korrektur von Schwachstellen bewusst durch die Nutzung sog. *Patch-Days*. Hierbei werden zu festgesetzten Termin, bei Microsoft bspw. an jedem zweiten Dienstag eines Monats, zahlreiche Patches auf einmal zur Verfügung gestellt. Allerdings sind auch hier nicht immer alle aktuellen Softwarekorrekturen beinhaltet: Bspw. wurden wichtige Patches für eine Schwachstellen im Internet Explorer [343] und weitere in den Miniaturbildern [344] nicht beim nachfolgenden Patch-Day einbezogen, obwohl bereits Exploits veröffentlicht wurden und ebenfalls ein einfach anzuwendender Code in das Metasploit-Framework aufgenommen wurde [147].

Angriffstechnologie und -verfahren

Mit den gewinnorientierten Absichten heutiger Angriffe gehen auch entsprechende Wandlungen der hierfür genutzten Technolgien und Verfahren einher.

Angriffe auf Applikationsebene Während zu Beginn der Programmierung von Schadsoftware maßgeblich Schwachstellen in Betriebssystemen und Protokollen angegriffen wurden, haben sich die Angriffsziele in den letzten Jahren verschoben. Zum einen sind die Schwachstellen der Betriebssysteme heutzutage besser durch entsprechende Schutzmaßnahmen abgesichert, zum anderen haben sich leichter angreifbare Ziele etabliert: Ein starker Trend geht hin zu Angriffen auf Applikationsebene. Insbesondere Webbrowser



(a) Entwicklung der Anzahl von Schwachstellen in verbreiteten Browsern.

(b) Durchschnittliche Dauer bis zur Verfügbarkeit eines Patches.

Abbildung 4.14: Schwachstellen in Webbrowsern und die durchschnittliche Zeit bis zur Verfügbarkeit eines Patches [141].

sind ein beliebtes Angriffsziel, da sie durch ihre Komplexität und den stetig wachsenden Funktionsumfang anfällig für Sicherheitslücken sind. Zudem ist hierbei durch die inherente Funktion eines Browsers in Bezug auf die Notwendigkeit von Netzverbindungen eine Kontaktierung eines bösartigen Servers im Internet, bspw. zum Versand abgehörter Daten oder zur Entgegennahme von Befehlen, vereinfacht möglich.

Die Anzahl der Fehler für weit verbreitete Browser und die jeweilige, durchschnittliche Dauer bis zur Verfügbarkeit eines Patches ist in Abbildung 4.14 dargestellt. Erkennbar ist, dass die Anzahl der Schwachstellen in den meisten Fällen innerhalb des Beobachtungszeitraumes angestiegen oder konstant geblieben ist, wobei sich die durchschnittliche Patchdauer bei den meisten Herstellern auf unter einen Tag gesenkt hat. Lediglich der Safari-Browser zeigt hier eine negative Entwicklung und hatte im Jahre 2009 eine durchschnittliche Verwundbarkeitszeit von 13 Tagen. Neben den Browsern selbst bieten auch die immer vielfältigeren und komplexeren Browser-Plugins, welche zusätzliche Funktionen in die Browser integrieren, zahlreiche angreifbare Schwachstellen an. Abbildung 4.15 zeigt eine Übersicht der Bereiche mit den meisten Schwachstellen.

Die zahlreichen, vorhandenen Schwachstellen und die oftmals schwierige Detektion haben zu einer intensiven Ausnutzung dieser Lücken geführt. Entsprechend stieg der Anteil von Angriffen gegen Webapplikationen im Jahre 2010 stark an, und erhöhte sich um 93 Prozent gegenüber den Werten des Vorjahres [142]. Dies wird insbesondere durch die zunehmende Verbreitung von Webattack-Toolkits vorangetrieben.

Targeted Attacks Eine Analyse der Angriffe der letzten Jahre zeigt, dass der Anteil von zielgerichteten Angriffen stetig zunimmt. Anstelle des Ausschaltens von Infrastrukturen werden gezielt Organisationen und einzelne Personen angegriffen [41]. Hierbei wird nicht versucht eine möglichst große Anzahl von Rechnern zu infizieren, wie dies bei tra-

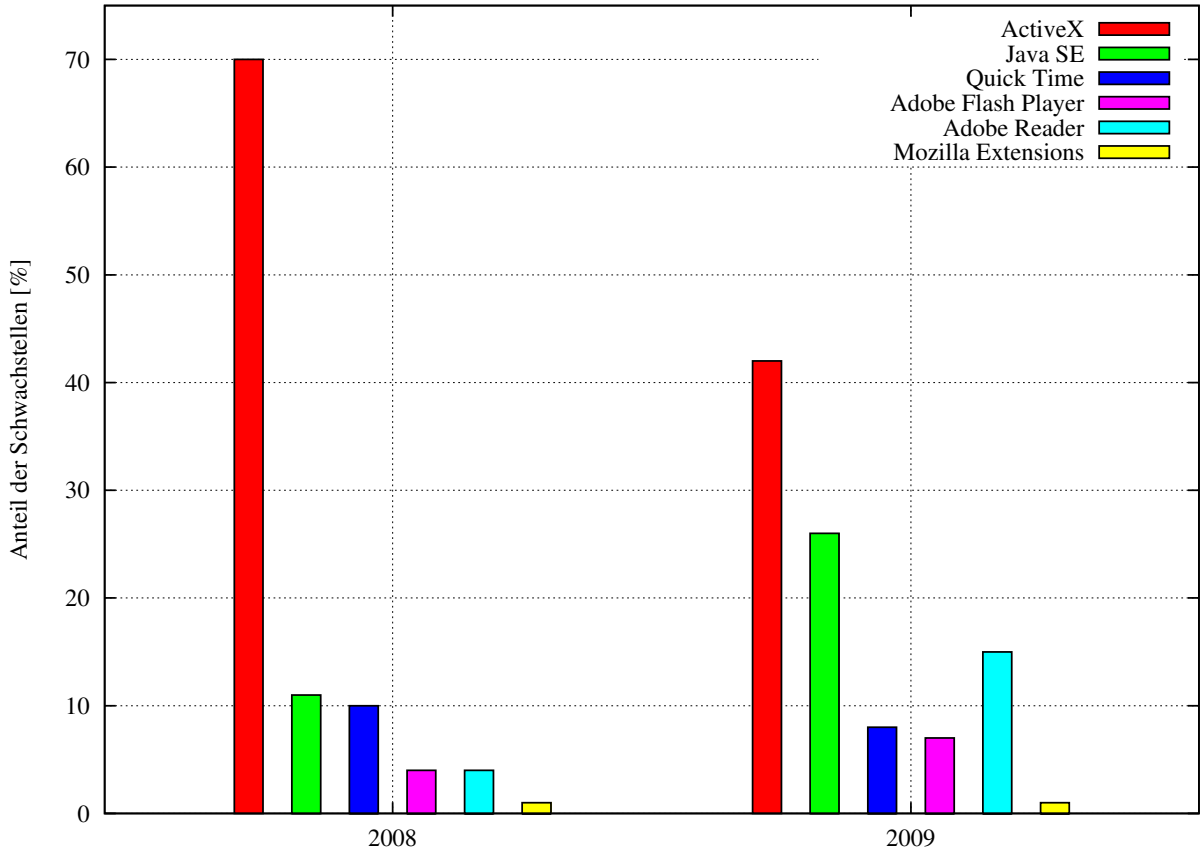


Abbildung 4.15: Schwachstellen in Browser-Plugins, prozentualer Anteil von insg. 424 (2008) bzw. 321 (2009) Schwachstellen [141]. Während sich einige Bereiche etwas verbessern konnten, stieg die Anzahl entdeckter Schwachstellen in anderen deutlich an. Schwachstellen in Anwendersoftware stellen mittlerweile den Hauptangriffspunkt für Hacker dar, da zunehmende Softwarekomplexität und immer kürzere Release-Zyklen stetig neue Angriffspunkte eröffnen.

ditionellen Verfahren erfolgt, sondern lediglich ein oder eine geringe Zahl ausgewählter Ziele angegriffen. Hintergrund ist, dass zum einen nur besonders profitable Ziele betrachtet werden, zum anderen die genutzten Angriffsverfahren und involvierten Exploits durch den gezielten Einsatz für längere Zeit unerkannt bleiben und Signaturen entsprechend spät entwickelt werden können. Dadurch, dass der jeweilige Angriff genau an ein Ziel angepasst ist, steigt die Erfolgswahrscheinlichkeit entsprechend: Ein bekanntes Beispiel ist die Zero-Day-Lücke Hydraq (bekannt als Trojan.Hydraq bzw. Aurora), mittels derer einige große Unternehmen kompromittiert wurden (vgl. z.B. [141]). Die Durchführung ist beispielhaft für die Vorgehensweise zielgerichteter Angriffe: Zunächst werden öffentlich verfügbare Informationen über Angestellte der anvisierten Firma gesammelt, bspw. von der Firmenhomepage oder aus den Daten Sozialer Netzwerke. Die bereits in den 80er Jahren verbreiteten Techniken des Social Engineerings erleben somit eine neue Blüte. Soziale Netzwerke wie Facebook und Twitter sind aufgrund der zahlreich verfügbaren Daten beliebte Informationsquellen für Angreifer. Viele Nutzer gehen sehr leichtfertig mit persönlichen Informationen im Kontext Sozialer Netzwerke um. Diese können bspw. genutzt werden, um personifizierte Mails an Nutzer zu senden. Durch die Einbettung in eine aktuelle, für das Ziel relevante Thematik und die Nutzung einer Mailadresse eines seiner Bekannten ist die Wahrscheinlichkeit, dass das Ziel die Mail und einen entsprechenden verseuchten Anhang öffnet besonders groß. Auf diese Weise lässt sich die Schadsoftware des Angreifers einfach in das Zielsystem oder -netz einschleußen. Der genutzte Schadcode wird dabei typischerweise für das jeweilige Ziel speziell konstruiert; dies hat eine besondere Relevanz, da diese durch signaturbasierte Verfahren nicht detektierbar sind. Zahlreiche andere Verfahren werden intensiv genutzt, um Soziale Netzwerke zu mißbrauchen und deren Nutzer anzugreifen. Beispiele hierfür sind bösartige Links, die Ausnutzung von Kurz-Uniform Resource Locator (URL)-Diensten, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF) und Clickjacking [394].

Die Verbreitungswege sind dabei nicht auf Netze wie das Internet und Dienste wie Mail beschränkt: Ein häufiges Vorgehen ist bspw. die Nutzung von Universal Serial Bus (USB)-Sticks, um den Schadcode in das Zielsystem einzuschleusen. Dieser kann auf verschiedene Art und Weise motiviert werden, bspw. indem er an einer ausgewählten Position als zufällig verloren hinterlegt wird. Hierbei ist die Wahrscheinlichkeit groß, dass ein Finder den USB-Stick aus Neugierde an ein System anschließt und somit die Infizierung initiiert. Eine andere, beliebte Möglichkeit ist das Verteilen von USB-Sticks als Werbegeschenke, bspw. auf Messen [53]. Die Schadsoftware ist hierbei mit auf dem Stick installiert, nutzt der Empfänger das Speichermedium an einem seiner Systeme, wird die bösartige Software eingespielt. Mittels dieser Art der Offline-Verbreitung ist es auch möglich, zuvor als sicher geltende Systeme und Netze anzugreifen. Ein bekannte Beispiel ist der Wurm Stuxnet, der im Juni 2010 im Iran durch Wissenschaftler des Anti-Viren Herstellers VirusBlokAda gefunden wurde [260]. Sein Ziel ist es, Supervisory Control and Data Aquisition (SCADA)-Systeme anzugreifen. Große Industrieanlagen werden heutzutage vollständig durch Rechner gesteuert [323]. Die Fertigungsprozesse werden durch die Systeme auf Basis der aufgezeichneten Sensordaten, bspw. Temperatur- und Druckinformationen, innerhalb der Betriebsparameter der Anlage gehalten, Manipulationen dieser Regelung können somit zu ernsten Konsequenzen führen. Die entsprechenden Steue-

rungen basieren oft auf dem von Siemens entwickelten S-7-System, das aus zahlreichen Programmable Logic Controllers (PLCs) besteht. Stuxnet nutzt zur Verbreitung sowohl das LAN als auch USB-Sticks, um Rechner zu erreichen, welche nicht mit einem Netz verbunden sind. Infiziert der Wurm einen Rechner, werden zunächst eigene Treiber installiert, die mit Hilfe zweier gestohlener Zertifikate signiert sind und installiert weiterhin ein Rootkit, um sich zu verstecken. Stuxnet sucht nach WinCC/Step 7- Systemen, die zur Programmierung und Überwachung der PLCs eingesetzt werden. Wird der Schadcode fündig, wird der PLC mit einem Zero-Day-Exploit kompromittiert und anschließend reprogrammiert; wird er nicht fündig, werden keine Aktionen ausgeführt. Da für die Durchführung des Angriffes hochgradiges Insider-Wissen über die Funktionsweise und insbesondere die Programmierung der Anlage notwendig ist, sind entsprechende Angriffe auf ein spezielles Ziel fokussiert. Weiterhin ist die Konzeption des Wurmes so ausgerichtet, dass die Entdeckungswahrscheinlichkeit möglichst minimiert wird. Dies zeigt sich nicht nur in der Nutzung eines Rootkits zum Verstecken des Schadprogrammes, sondern auch in der begrenzten Anzahl der Ausführungen, bevor der Code inaktiv wird. Die große Verbreitung von Stuxnet ist daher eher als ungewollt anzunehmen und durch das Einspielen auf an das Internet angeschlossene Systeme im eigentlichen Zielobjekt, das iranische Atomkraftwerk Bushehr, zurückzuführen. Auch die Funktionsweise des Wurmes, der nur nach der erfolgreichen Erkennung der Zielumgebung aktiv wird, unterstreicht dies. Weiterhin muss berücksichtigt werden, dass der Schadcode spezifisch für die genauen Gegebenheiten der Fabriksteuerung programmiert werden muss und daher generell nicht allgemein einsetzbar ist. Dennoch hat der Wurm eine erhebliche Verbreitung gefunden. Gem. einer Studie des Antivirenherstellers McAffee haben 59 Prozent der befragten Strom-, Gas- und Wasseranbieter in Deutschland den Schädling in ihren Systemen entdeckt, der internationale Durchschnitt liegt bei 41 Prozent [197]. Dies stellt eine neue Stufe der Gefährdung für solche Systeme dar, deren Schwachstellen durch eine steigende Anzahl von motivierten und qualifizierten Hackern angegriffen werden [85].

Professionelle Werkzeuge Die mittlerweile in der Untergrundszene verfügbaren Angriffstools sind hochentwickelt und ermöglichen auch fachlich nicht versierten Personen, effiziente Angriffe auf IT-Systeme durchzuführen. Florierende Geschäftsmodelle (vgl. Kapitel 2.3.2) lassen sowohl den Umfang an Funktionalität immer größer, als auch die Reaktionszeiten immer kürzer werden. Entsprechende Servicemodelle bieten einem Käufer bspw. die Garantie, dass die genutzten Schadroutinen bei Entdeckung durch Antivirensoftware durch den Entwickler modifiziert werden, um wiederum der Entdeckung zu entgehen. Zusätzlich zum Handel mit Angriffstools und Informationen sind zahlreiche Dienste verfügbar, bspw. das manuelle Lösen von Captchas oder die Installation von Schadsoftware. Bei letzterem können sich Nutzer auf entsprechenden Seiten einschreiben und werden mit Schadsoftware versorgt. Die Bezahlung erfolgt nach der Anzahl von Installationen auf Fremdrechnern und kann zu Einkünften von mehreren hundert US-Dollar pro Tag führen [125].

Neue Angriffstechniken durch neue Dienste Durch die schnelle Weiterentwicklung des Internets migrieren zahlreiche Dienste und es entstehen immer neue Protokolle und Verfahren. Online-Banking, VoIP oder VoD sind hier nur einige Beispiele; die IP-Fähigkeit eines Gerätes ist heutzutage schon eine Designforderung in vielen Bereichen. Die Entwicklung der Car2Car- und Car2Infrastructure- Kommunikation ist hierfür ein Beispiel (vgl. z.B. [102]). Gerade unter den marktwirtschaftlichen Anforderungen der Wettbewerbsfähigkeit und aufgrund von Kostenfragen, wird hierbei der Sicherheit der Systeme oftmals nicht genügend Rechnung getragen. Zusätzlich steigt die Wahrscheinlichkeit für Fehler durch die zunehmende Anzahl von Lines of Code (LoC) und die immer komplexeren Systeme und Anwendungen. Dies eröffnet insbesondere auch neue Angriffsmöglichkeiten, die zuvor nicht bekannt waren.

Bedeutung von Botnetzen Von Botnetzen geht heute eine große Gefahr aus: Bots sind kleine Programme, die ohne Wissen des Besitzers eines Rechners auf diesem installiert werden. Ein Botnetz ist ein Netz von infizierten Rechnern, die durch einen Command and Control, auch C2 (C&C)-Server gesteuert werden, typischerweise von einem Kriminellen oder organisierter Untergrundkriminalität [51]. Die größten Botnetze umfassen mittlerweile viele hunderttausend Rechner, *Rustock* kontrollierte 2010 bereits über eine Million Zombie-Rechner [142].

Botnetzen kommt eine besondere Bedeutung zu, da diese für die Durchführung zahlreicher Angriffe und dem Ausspähen der Nutzer verwendet werden. Maßgebliche Gefahren sind [209]:

- Durchführen von DDoS-Angriffen
- Ausspähen von sensitiven Nutzerdaten, u.a. mittels Keyloggern, Identitätsdiebstahl
- Proxy-Funktionalität zur Verschleierung der Identität eines Angreifers (Missbrauch des Bots als Sprungbrett)
- Versendung von Spam
- Verbreitung von Schadsoftware
- Click Fraud (automatisches Anklicken von Werbung, um Profit bei sog. pay-per-click Werbungen zu erzeugen) und Adware (automatisches Einblenden von Werbung auf dem Bot-Rechner)
- Phishing

Auch entstehen neue Geschäftsmodelle, wie das Vermieten ganzer Botnetze [51].

Zu dem umfangreichen Gefahrenpotential kommt eine erschwerte Detektion hinzu, da die Aktionen vom infizierten System heraus gestartet werden können und somit die typischerweise nach Innen absichernden Firewalls passieren können und einer geringen Detektionswahrscheinlichkeit durch IDSs, die lediglich den eingehenden Datenverkehr

untersuchen, unterliegen. Das Erkennen und Ausschalten von Botnetzen wird auch dadurch erschwert, dass die früheren, zentralisierten Ansätze mit einem C&C-Server durch neue Verfahren wie der Nutzung von P2P-Kommunikation und der Einführung von proprietären und verschlüsselten Kommunikationsprotokollen ersetzt wurden. Zu beobachten ist, dass bei der Weiterentwicklung der Bots ebenfalls ein immer höherer Stellenwert auf die Unsichtbarkeit dieser gelegt wird.

Einen Überblick über die aktuell aktiven C&C-Server ist in [143] ersichtlich. Derzeit[18] sind durchschnittlich 5600 entsprechende Server aktiv. Jeder dieser Server verwaltet im Schnitt 20000 infizierte Rechner [51], so dass täglich mehrere Millionen Bots online sind. Aufgrund der hohen Anzahl von Teilnehmer eines Botnetzes ist es einfach, effektive DDoS-Angriffe durch diese zu fahren: Im Durchschnitt kann jeder Bot-Client hierfür 40 KB/sec Datenrate zur Verfügung stellen, so dass bereits ein kleines Botnetz das Netz einer durchschnittlichen Firma lahm legen kann, ein großes Botnetz kann sogar einen Internet Service Provider (ISP) gefährden.

Da einzelne, durch einen Bot durchgeführte Aktionen durchaus legal im Sinne der auf dem Rechner initiierten Aktion sein können (z.B. das Versenden einer Mail im Rahmen einer Spam-Verteilung des Botnetzes), müssen statistische Verfahren zu deren Detektion herangezogen werden. Gleiches gilt für die Kommunikation zwischen Bot und dem C&C-Server, insbesondere wenn diese verschlüsselt ist.

Die aus der Evolution der Schadsoftware resultierenden Probleme sind somit maßgeblich:

- Zielgerichtete, Social-Engineering Angriffe
- Angriffe auf Applikationsebene
- Einfacher Einsatz und Erzeugung von Schadcode ohne Fachwissen
- Sehr große Signaturdatenbanken, ressourcenfordernde Suchverfahren
- Zu große Verzögerungen bei der Bereitstellung von Signaturen

Abbildung 4.16 zeigt die maßgeblichen Faktoren und deren Zusammenhänge in Bezug auf signatur- bzw. verhaltensbasierte Systeme. Aufgrund der derzeitigen Technologietrends wie Verschlüsselung und der Bedrohungsentwicklung wie eine steigende Anzahl zielgerichteter Angriffe oder der Bedrohung durch Innentäter, ist der Einsatz signaturbasierter Verfahren diesbezüglich beschränkt und fordert andererseits die Nutzung von verhaltensbasierten Ansätzen. Weitere Faktoren, wie bspw. die Notwendigkeit ohne spezielle Hardware auszukommen sowie ressourcenschonend zu arbeiten, um auch der steigenden Mobilität von Systemen und deren Anforderungen Rechnung zu tragen, verstärken diese Forderung. Andererseits werden die typischen Funktionsweisen verhaltensbasierter Verfahren ihrerseits durch Gefährdungspotentiale gehemmt, bspw. die Durchführung von Lernphasen oder die Notwendigkeit echtzeitfähiger Reaktionen.

[18] April 2011.

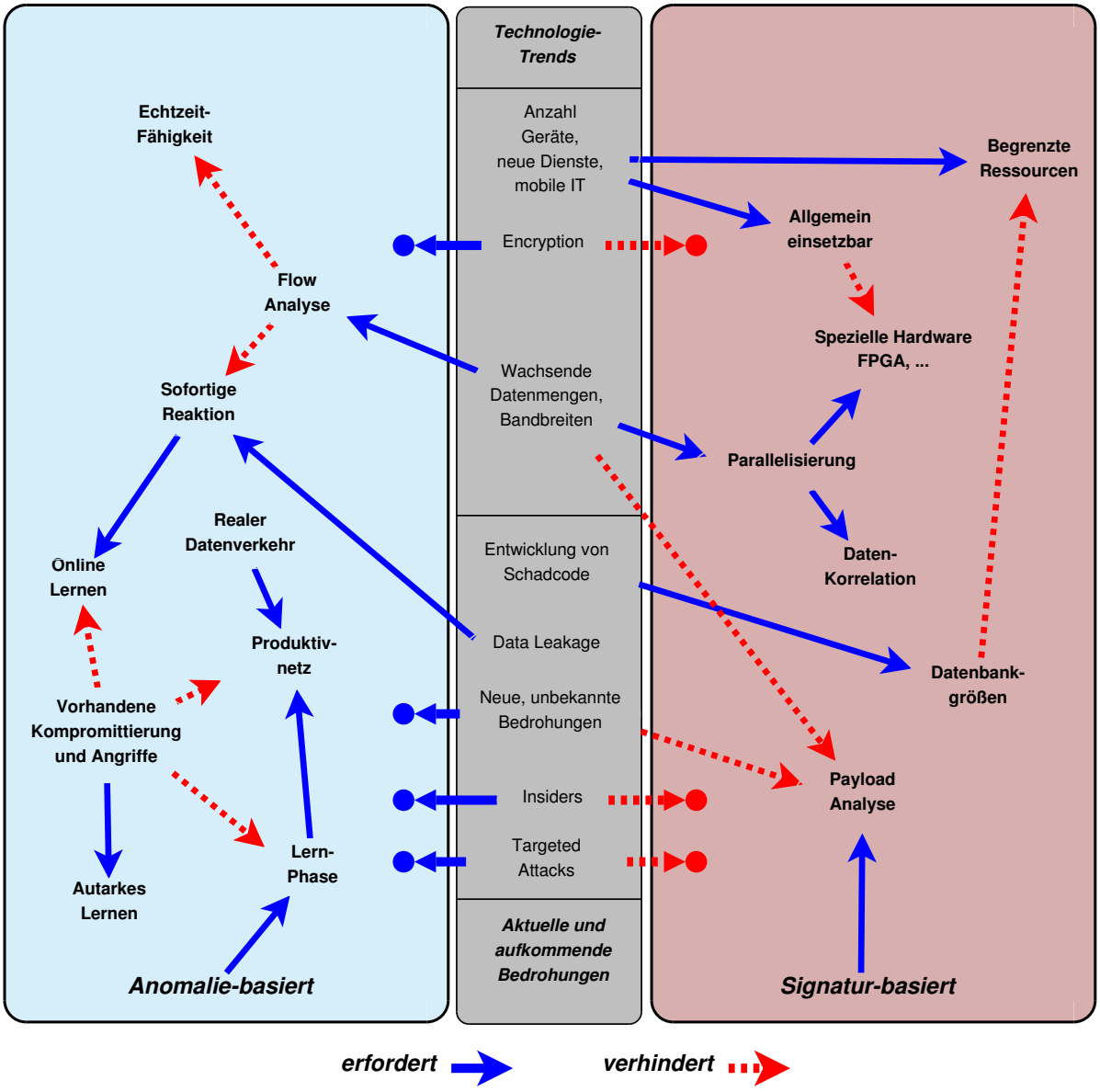


Abbildung 4.16: Auswirkungen verschiedener Einflußfaktoren auf signatur- und verhaltensbasierte Analyse. Von besonderer Bedeutung sind Faktoren, die sowohl durch Anforderungen (blau), als auch durch Restriktionen (rot) erreicht werden: Dies stellt maßgebliche Einschränkungen in der Nutzung bzw. Forderung der jeweiligen Punkte dar.

4.6.2 Entwicklung und Wachstum des Datenverkehrs

Wie bereits in Kapitel 1.1 erwähnt, wachsen die täglich über das Internet übertragenen Datenmengen in einem rasanten Tempo an, so dass bis 2014 mit einem monatlichen Datenaufkommen von 64 EB zu rechnen ist. Der globale IP-Datenverkehr wird sich damit im Zeitraum 2009 bis 2014 vervierfachen.

Infrastruktur und Bandbreite

Neben der steigenden Zahl an Nutzern und Geräten lässt sich der starke Zuwachs an Datenvolumen mit dem Nutzerverhalten erklären. Visuelle Netzapplikationen werden immer häufiger gleichzeitig während der Nutzung anderer Applikationen - möglicherweise wiederum visueller Art - eingesetzt. Auch wenn der Nutzer mit anderen Aufgaben beschäftigt ist, laufen diese oftmals im Hintergrund weiter. Dieser Trend wird als „widgetization“[19] von Internet und Fernsehen bezeichnet [210]. Die beiden hier vorliegenden Eigenschaften, Multitasking und passives, vernetztes Arbeiten, sind maßgeblich für die sog. Hyperkonnektivität. Diese wird durch vier Aspekte ermöglicht: Einer immer stärkeren Verbreitung von Breitbandanschlüssen, der Zunahme der Darstellungsfläche und Auflösung von digitalen Anzeigen, der Verbreitung von netzfähigen Geräten und der Zunahme der Leistungsfähigkeit von TK-Geräten. Im Zeitraum von 2009 bis 2014 wird sich das durch Videokommunikation erzeugte Datenvolumen versiebenfachen, das mobile Datenaufkommen wird bis 2014 sogar auf das 39-fache ansteigen [210].

Der Ausbau der Datennetze in Europa und der damit verbundene Anstieg des Datenvolumens ist in Kapitel F.2.11 kurz skizziert.

Die immensen und stetig steigenden Datenmengen stellen eine besondere Herausforderung dar. Insbesondere werden auch Mobilgeräte in naher Zukunft für eine Vervielfachung der Datenvolumina sorgen (vgl. auch Tabelle F.9).

Durch die Vielzahl von Signaturen sind die zugehörigen Datenbanken von Virenscannern und IDSn heute bereits auf viele Megabyte (MB) angewachsen. Nicht nur die Datenbankgröße, sondern auch eine performante Untersuchung auf das Vorkommen von Signaturen im überwachten Datenverkehr wird immer anspruchsvoller und ist bei hohen Bandbreiten nur noch bedingt in (nahe-) Echtzeit möglich. Insbesondere eine DPI ist nur begrenzt möglich. Spezielle Hardware und hochgradige Parallelisierung, z.B. mittels des Einsatzes von FPGAs, ermöglicht zwar ein Anheben dieser Grenze, schränkt jedoch die Anwendbarkeit ein und ist daher in vielen Bereichen nicht einsetzbar: Insbesondere bei Mobilgeräten und Smartphones sind die verfügbaren Ressourcen stark begrenzt. Eine möglichst lange autarke Energieversorgung per Batterie ist hier entscheidend, weshalb keine zusätzlichen Hardwareelemente, bspw. für eine effiziente, parallele Pattern-Analyse, erwünscht sind. Die eingeschränkte Rechenleistung sowie begrenzter Speicher erschweren zusätzlich die Integration signaturbasierter Detektoren.

[19] Widgets sind Grafikelemente eines Desktops, die Möglichkeiten zur Interaktion mit dem Nutzer anbieten und beliebig in den Desktop integriert werden können, bspw. zur Darstellung von Nachrichtentickern, Videos oder Wettervorhersagen.

Gerätezahl und neue Dienste

Insbesondere die steigenden Fähigkeiten mobiler Geräte sorgen für eine starke Zunahme der Verbreitung und zur Entstehung immer neuer Dienste. Zu den zahlreichen, multifunktionellen Geräten gehören u.a. Handys, Smartphones, mobile Spielekonsolen, Pikoprojektoren, GPS-Geräte, e-Book-Reader, etc. Der Anstieg der damit verbundenen Datenmengen ist im Anhang F.2.16 exemplarisch in Tabelle F.9 dargestellt.

Besonderer Bedeutung kommen bei den Mobilgeräten einer Vielzahl neuer Dienste zu, bspw. ermöglicht durch Adhoc-Vernetzung. Andererseits spielt die Energieaufnahme der Gerät eine wesentliche Rolle; für eine hohe Mobilität und lange Akkulaufzeiten muss eine minimale Energieaufnahme erreicht werden, was sich insbesondere auch in der Architektur und Leistungsfähigkeit der Prozessoren wiederspiegelt.

Das enorme Wachstum des Netzes und die schnell steigende Anzahl von Geräten haben den verfügbaren Internet Protocol Version 4 (IPv4)-Adressraum schnell knapp werden lassen. Insbesondere aufgrund der Nutzung von NAT und Verfahren wie zeitlich begrenzter Vergabe dynamischer IP-Adressen konnte die Notwendigkeit, hier Änderungen durchzuführen, bisher herausgezögert werden, so dass IPv4 weiterhin das dominante Protokoll ist (vgl. Kapitel 4.2). Nach der Vergabe der restlichen IPv4-Blöcke und dem steigenden Bedarf an Adressen, insbesondere auch in Entwicklungs- und Schwellenländern wird das Nachfolgeprotokoll Internet Protocol Version 6 (IPv6) künftig eine wichtigere Position einnehmen. Abbildung F.22 im Kapitel F.2.18 zeigt die Auslastung des IPv4-Adressraumes. Diese sehr inhomogene Aufteilung führt insbesondere in den Ländern mit hohen Wachstumszahlen und geringen verfügbaren Adress-Ressourcen zu erheblichen Herausforderungen. Die Aufteilung des Adressraumes von IPv6 ist aus der Tabelle F.10 im Anhang F.2.17 ersichtlich.

Die aus der Entwicklung und dem Wachstum der Netze resultierenden Probleme sind somit insbesondere:

- Weiterhin kontinuierlicher, starker Anstieg des Datenvolumens
- Neue, bisher unbekannte Dienste und Kommunikationsmöglichkeiten
- Migration auf das Protokoll IPv6

4.6.3 Einsatz von Verschlüsselung*

Sowohl signaturbasierte- als auch verhaltensbasierte IDS beruhen auf der Verfügbarkeit und Auswertung der Paketinformationen des Netzverkehrs. Während verhaltensbasierte Systeme hierbei anhand charakteristischer Eigenschaften wie gesetzter Flags, Verbindungen zu bestimmten Ports oder Datenmengen versuchen, Anomalien zum normalen

*Dieser Abschnitt enthält eine Zusammenfassung von Teilen des Artikels „Command Evaluation in Encrypted Remote Sessions“, Proceedings of the 2010 Fourth International Conference on Network and System Security (NSS 2010), Seiten 299–305, IEEE Computer Society, 2010.

Verhalten des Systems zu entdecken und darauf basierend einen Alarm auslösen, untersuchen signaturbasierte Verfahren die Pakete auf das Vorhandensein bestimmter Muster. Bei der Untersuchung der Datenpakete kann zwischen zwei Verfahren unterschieden werden:

- Shallow Packet Inspection [294]: Dieses Verfahren wird hauptsächlich in Firewalls eingesetzt, um die Art des Datenverkehrs anhand des genutzten Ports zu erkennen. Eine Analyse erfolgt nur bis zum Layer 4 des Open System Interconnection (OSI)-Referenzmodells, ausgewertet werden insbesondere die Absender- und Zieladressen sowie die Informationen zum Verbindungsstatus (Stateful Packet Inspection). Hierdurch ist nur eine rudimentäre Analyse einer Verbindung möglich. Nutzt ein IDS eine Shallow Inspection, können lediglich Scanning-Aktivitäten oder ggf. der Missbrauch von Protokollen detektiert werden.

- Deep Packet Inspection [218]: Im Gegensatz zur Shallow Packet Inspection wird hierbei auch der Payload eines Paketes untersucht. Anhand dieser Informationen sind Auswertungen überhalb der Schicht 4 möglich, bspw. kann erkannt werden, um welche Datenart es sich handelt. Da sich im Payload auch die Signaturen von Viren oder Exploits verbergen, kann eine entsprechende Evaluation durch ein IDS durchgeführt werden, um solche Angriffsversuche bzw. Schädlinge zu erkennen und zu stoppen.

Es ist ersichtlich, dass eine reine Shallow Packet Inspection im Falle eines IDS nur geringen Erfolg verzeichnen kann:

- Die Mehrzahl der Angriffe wird heutzutage auf der Applikationsebene durchgeführt, sie sind somit nicht auf den Layern bis 4 erkennbar.

- Eine Vielzahl von Applikationen nutzt keine Standard-Ports mehr (vgl. z.B. P2P), oftmals werden auch von traditionellen Diensten nicht Standard-Ports verwendet: Ein verbreitetes Beispiel ist hier SSH, das von vielen Administratoren nicht mehr auf Port 22 betrieben wird, da hier eine hohe Zahl automatisierter Scan- und Angriffsversuche durchgeführt werden.

Daher ist prinzipiell eine DPI anzustreben. Wenn dies nicht möglich ist, muss eine Shallow Inspection mit geeigneten Mitteln bereichert werden, um eine ähnliche Fähigkeit zu erhalten.

Einschränkungen in verschlüsselten Umgebungen

Aufgrund zahlreicher Datenskandale der jüngeren Zeit ist der Datenschutz mehr in das öffentliche Licht geraten (vgl. Kapitel 2.3.2). Das Tool Firesheep demonstrierte bspw. beeindruckend, wie einfach die Übernahme einer fremden Mailsitzung, einer Verbindung zu einem Sozialen Netzwerk wie Facebook, etc. sein kann (vgl. Kapitel 3.1). Dies führt zu einer Forderung nach verschlüsselter Datenübertragung, auf welche die Diensteanbieter zunehmend reagieren. Eine intensivere Nutzung von IPv6, deren integraler Bestandteil

IPsec zur Sicherstellung der Integrität, Authentizität und Vertrauchlichkeit ist, wird die Nutzung von Kryptographie bei der Datenübertragung im Internet zusätzlich vorantreiben.

Mit Hinblick auf den Einsatz von Sicherheitssystemen bringt dies erhebliche Einschränkungen mit sich, da hierdurch eine Analyse des Payloads ausgeschlossen wird und somit keine DPI mehr möglich ist.

Analyseverfahren in verschlüsselten Umgebungen

Aufgrund der beschriebenen Einschränkungen der Verfügbarkeit von Informationen in verschlüsselten Umgebungen verbleiben drei grundlegende Verfahren, Einbruchserkennung in verschlüsselten Netzen zu betreiben:

1. Erkennen des Missbrauchs von Protokollen
2. Modifikation von Protokollen oder der Netz-Infrastruktur
3. Statistische Analyse des verschlüsselten Datenverkehrs

Im Folgenden werden die einzelnen Verfahren kurz vorgestellt.

Erkennen des Missbrauchs von Protokollen Anhand der Protokolle ist der ordnungsgemäße Kommunikationsablauf der jeweiligen Dienste definiert, die in Standards vorgegeben und bspw. durch RFCs publiziert werden. Zahlreiche Angriffe beruhen darauf, dass die Vorgaben der Protokolle nicht immer vollständig sind, oder Abweichungen in der realen Implementierung der verschiedenen Betriebssysteme auftreten (vgl. Kapitel 2.3.3). Ein IDS kann dies ausnutzen, indem es die jeweiligen Protokollzustände bzw. -transitionen überwacht. Werden Anomalien bzgl. der normalen, spezifizierten Nutzung des Protokolls festgestellt, kann eine Alarmierung erfolgen. Da hierfür keine Entschlüsselung des Datenverkehrs notwendig ist, kann das System generell eingesetzt werden. Der Verzicht auf eine Entschlüsselung ist gleichzeitig positiv im Sinne der Ressourcen-Nutzung: Für das Mitverfolgen der Zustände des Verschlüsselungsprotokolls werden nur geringe Systemmittel benötigt.

Yasinsac nutzt sowohl signatur- als auch verhaltensbasierte Ansätze, um Angriffe auf Sicherheitsprotokolle zu erkennen [398]. Hierbei werden Angriffstaxonomien und Funktionsprinzipien der Protokolle herangezogen, um eine verhaltensbasierte Detektion zu ermöglichen; bspw. kann eine bestimmte Protokollsequenz Anzeichen für einen Angriff darstellen. Entsprechende Sequenzen werden als Signaturen gespeichert und aktive Verbindungen in Echtzeit mittels der Nutzung von Zustandsübergangsmaschinen auf deren Vorkommen hin untersucht. Die von Yasinsac vorgeschlagene Architektur umfasst drei Bereiche, erstens zur Erzeugung der Wissensdatenbank für bekannte Angriffe und Verhaltensprofile, eine Sonde zur Aufzeichnung des Datenverkehrs in der Zielumgebung sowie die Untersuchung der aufgezeichneten Daten auf das Vorkommen von Angriffen gemäß der Wissensdatenbank. Hierbei werden sowohl die Signaturen als auch die Informationen der verhaltensbasierten Detektion in der Datenbank abgelegt.

Joglekar et al. haben ein entsprechendes System zur Detektion von Protokollmissbrauch auf Basis gemeinsam genutzter Programmbibliotheken (Shared Libraries) der Kryptographie und der Applikationsprotokolle vorgeschlagen [221]. Die Überwachung der Zustände wird in Systemteilen, welche für die Handhabung der Protokolle verantwortlich sind, integriert. Dies ermöglicht eine effiziente Erkennung von Angriffen auf das Verschlüsselungsprotokoll, Angriffe auf Appliktionsebene können jedoch nicht erkannt werden. Somit sind durch diese Verfahren lediglich entsprechende Missbrauchsversuche des Protokolls detektierbar, bspw. durch einen Brute Force-Angriff oder den Versuch, Ressourcen zu blockieren. Angriffe, die im Payload verschlüsselt sind, z.B. Versuche einer Rechteerhöhung oder Exploits, können nicht erkannt werden.

Fadlullah et al. schlagen eine Architektur aus verteilten Beobachtungspunkten (monitoring stubs) an den Routern eines Netzes vor, um mittels verhaltensbasierter Analyse Angriffe auf Kryptoprotokolle zu erkennen [131, 371]. Zur Angriffserkennung wird eine nichtparametrische kumulative Summe über die TCP-Header der Verbindungen erzeugt, Abweichungen vom Normalprofil erzeugen einen Alarm. Die einzelnen Beobachtungspunkte extrahieren hierbei die benötigten Protokolldaten mittels `tcpdump` und können auch genutzt werden, einen Angriff zurückzuverfolgen. Dies wird erreicht, indem Ergebnisse, welche einen Alarm ausgelöst haben, zwischen den verschiedenen Beobachtungspunkten korreliert werden. Hohe Werte zeigen, dass der Angriff über die jeweiligen Router weitergeleitet wurde. Angemerkt werden muss, dass dies nur funktionieren kann, wenn entsprechend alle[20] Router mit entsprechender Monitoring-Funktionalität ausgestattet wurden. Weiterhin lässt sich das Verfahren nicht für alle Kryptoprotokolle anwenden, da es auf den Zugriff auf die Flags des TCP-Frames angewiesen ist; diese stehen bspw. bei SSH oder TLS zur Verfügung, jedoch z.B. *nicht* bei der Nutzung von IPsec.

Während für diese Vorgehensweise der Protokollanalyse zwar einerseits keine Entschlüsselung des Datenverkehrs notwendig ist, können jedoch andererseits lediglich Angriffe auf der Netzebene erkannt werden. Insbesondere können die vorgestellten Verfahren keine Angriffe auf der Applikationsebene detektieren, welche jedoch heutzutage die größte Gefährdung darstellen und weiterhin ansteigen (vgl. Kapitel 4.6.1).

Modifikation von Protokollen oder der Netz-Infrastruktur Ein anderer Ansatz ist das Design der Netze mit Hinblick auf die verschlüsselte Kommunikation oder das Anpassen von bereits vorhandenen Strukturen. Dies kann bspw. durch eine Modifikation der genutzten Verschlüsselungsprotokolle, durch MITM-Verfahren oder spezielle Hardwarekomponenten erfolgen. Typischerweise sind diese Verfahren nicht allgemein einsetzbar, da durch Änderungen von Protokollen Inkompatibilitäten entstehen können oder spezielle Komponenten für die Kommunikation erforderlich werden. Entsprechende Ansätze wären somit bspw. innerhalb eines Firmennetzes bzw. zwischen den Netzen verschiedener Firmenniederlassungen möglich, jedoch nicht für eine generelle Analyse des verschlüsselten Datenverkehrs anwendbar.

Goh et al. haben ein auf diesem Konzept basierendes IDS vorgeschlagen, welches in

[20] Bzw. mindestens die Edge-Router aller Teilnetze.

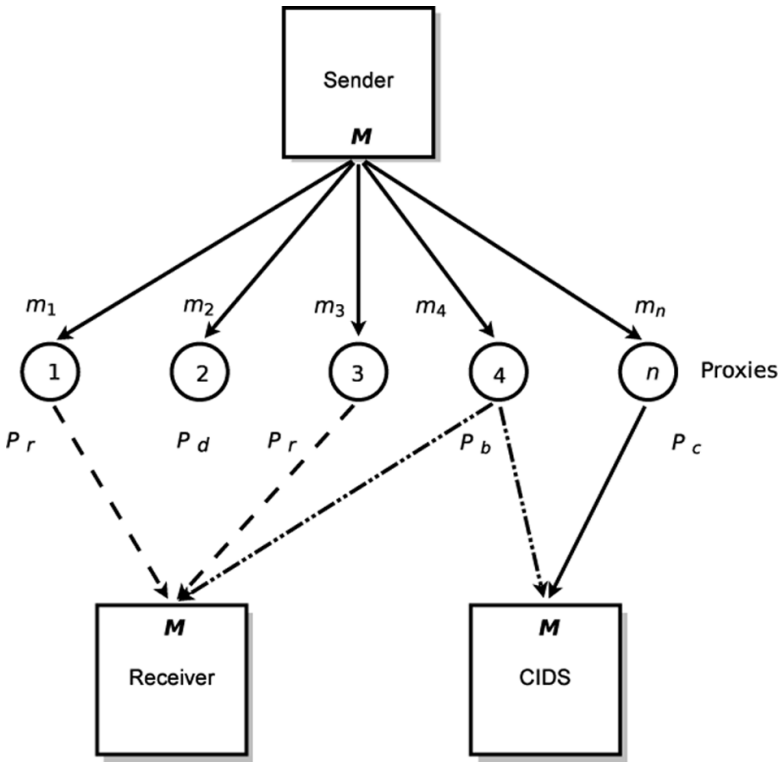


Abbildung 4.17: Central IDS nach Goh für den Einsatz in verschlüsselten Umgebungen [169].

der Lage ist sowohl eine Payload-Analyse der Datenpakete der verschlüsselten Verbindung durchzuführen, als auch gleichzeitig die Vertraulichkeit und Integrität der Daten zu gewährleisten [169, 168, 170, 167]. Ihr System beruht auf einem hierfür entwickelten Protkoll, das den Netzverkehr eines Sender repliziert und sowohl an den eigentlichen Empfänger, als auch zusätzlich an ein zentrales IDS (Central IDS (CIDS), ein separierter und dedizierter Server), weiterleitet. Das Protokoll selbst setzt auf ein darunter liegendes VPN auf und fügt einen zusätzlichen Layer zur Netz-Ebene hinzu. Während das VPN die Vertraulichkeit der Daten im Netz garantiert, ermöglicht die Datenkopie dem CIDS eine entsprechende Analyse mittels einer DPI. Zusätzlich zu diesen Komponenten werden mehrere Proxy-Systeme benötigt, an welche die Netzpakete mit verschiedenen Wahrscheinlichkeiten geschickt werden (vgl. Abbildung 4.17). Jeder Sender muss für seine Übertragung die Proxies nutzen, die wiederum die Verteilung an das CIDS und den Empfänger vornehmen. Dieses Vorgehen verhindert, dass eine Sendung ohne die Einbeziehung des CIDS direkt an den Empfänger erfolgen kann. Um die Vertraulichkeit der Pakete auch bei den Proxies zu garantieren, wird ein geteiltes Geheimnis nach Shamir verwendet, da mittels dieses Verfahrens auf die Nutzung einer komplexeren Public Key Infrastructure (PKI) verzichtet werden kann. Jedes Paket M wird in die Teile $\{m_1, m_2, ..., m_n\}$ zerlegt, anschließend wählt der Sender n Proxies aus der Liste der verfügbaren Instanzen aus und sendet jeweils einen Teil an einen der Proxies. Der Proxy

Tabelle 4.5: Matrix der Erkennungsraten unterschiedlicher Protokolle innerhalb eines verschlüsselten Tunnels gem. Wright [392] bei der Nutzung des Viterbi-Klassifikators.

	AIM	SMTP-out	SMTP-in	HTTP	HTTPS	FTP	SSH	Telnet	none
AIM	80.8	2.9	1.4	1.6	3.1	0.9	5.4	3.2	0.7
SMTP-out	7.1	73.2	6.9	1.2	1.9	2.3	1.9	5.2	0.3
SMTP-in	2.5	10.6	77.2	0.1	0.2	4.6	0.8	3.9	0.1
HTTP	0.7	0.3	0.1	90.3	6.4	0.3	1.3	0.4	0.1
HTTPS	0.9	0.8	0.1	5.9	88.5	0.6	1.9	0.8	0.5
FTP	7.1	4.1	11.1	0.9	2.1	57.7	6.0	11.0	0.0
SSH	3.4	1.8	9.3	1.5	6.8	2.8	69.1	1.9	3.2
Telnet	2.2	1.0	1.8	3.5	2.2	2.6	3.2	82.9	0.4

leitet wiederum mit verschiedenen Wahrscheinlichkeiten das Paketteil entweder an das CIDS und den Empfänger, nur das CIDS, nur den Empfänger oder gar nicht weiter. Dieses Vorgehen ermöglicht eine Erkennung von Umgehungsversuchen des CIDS.

Aufgrund der komplexen Systemstruktur und insbesondere der Nutzung zusätzlicher Protokolle, ist das Verfahren nicht allgemein anwendbar. Da das Protokoll per Design verlustbehaftet ist, wird ebenfalls die Netz-Performance reduziert, insbesondere bei schlechter Wahl für die Wahrscheinlichkeiten der jeweiligen Paketweiterleitung.

Statistische Analyse Eine weitere Möglichkeit, Einbruchserkennung bei verschlüsselten Verbindungen zu realisieren, ist die Nutzung statistischer Analysen. Wie bereits beim ersten vorgestellten Ansatz, erfolgt auch hier keine Entschlüsselung der Daten. Da hierbei nur beobachtbare Parameter des Datenverkehrs analysiert werden, bspw. die Verteilung von Paket-Größen, Muster von transferierten Datenvolumina oder das Timing der Übertragung, ist dieser Ansatz generell anwendbar.

Die statistischen Ansätze zur Analyse von verschlüsseltem Datenverkehr lassen sich maßgeblich in drei weitere Gruppen unterteilen, der *Protokollanalyse*, der *Webseitenerkennung* und der *Einbruchserkennung*.

Im Bereich der Protokollanalyse wird versucht, das innerhalb einer verschlüsselten Verbindung genutzte Protokoll zu identifizieren, bspw. HTTP für den Zugriff auf Webseiten oder Jabber für die Sofortnachrichtendienste. Wright et al. evaluieren die beobachtbaren Merkmale Paketgröße, Zeitintervall sowie die Transportrichtung, um auf die Protokolle innerhalb eines verschlüsselten Tunnels zu schließen [392]. Hierbei untersuchen sie zum einen Tunnel, die zahlreiche TCP-Verbindungen gleichzeitig transportieren, als auch solche, die nur einzelne Verbindungen absichern. Wright et al. nutzen verschiedene Methoden wie k-Nearest Neighbor (k-NN) Hidden Markov Models (HMMs) und Viterbi zur Untersuchung und Klassifizierung der Datenströme. Mittels des vorgeschlagenen Verfahrens ist es möglich, verschiedene Protokolle mit Wahrscheinlichkeiten bis über 90 Prozent bei aggregierten, mit bis über 80 Prozent bei einzelnen Verbindungen zu erkennen (vgl. Tabelle 4.5).

Alshammari et al. nutzen Signaturen und Flow-Attribute zur Erkennung von SSH-

Strömen [33]. Für die Identifizerung des SSH-Datenverkehrs werden drei Lernalgorithmen genutzt, SVM, Naïve Bayesian und C4.5, welche die benötigten Signaturen erzeugen. Diese werden von Alshammari als *allgemeine* Signaturen bezeichnet, da sie nicht nur in der Netzumgebung, in welcher sie erlernt wurden, gültig sind, sondern auch in völlig anderen Netzen angewandt werden können. Das Verfahren ist in der Lage, SSH-Verbindungen mit Wahrscheinlichkeiten bis zu 97 Prozent Detektionsrate zu erkennen, bei einer Fehlerrate von 0.8 Prozent.

Eine Klassifizierung von verschlüsselten Datenströmen in Echtzeit wird von Bar-Yanai et al. vorgeschlagen [49]. Um eine Detektion in Echtzeit zu erreichen, kombinieren die Autoren k-means und k-NN Klassifikatoren und können somit die Komplexität der Berechnung reduzieren unter gleichzeitiger Beibehaltung guter Klassifikationsergebnisse: Das System ist in der Lage, 97.3 Prozent verschiedener Protokolle wie bspw. HTTP, SMTP oder ICQ zu erkennen und liegt damit nur geringfügig unter den Klassifikationsresultaten eines reinen k-NN Algorithmus (99.1 Prozent), jedoch unter einem deutlich verbesserten Laufzeitverhalten (bspw. 42-mal schneller bei einer Datensatzgröße von 9000).

Ein Verfahren zur Erkennung von Webseiten wurde von Bissias et al. vorgeschlagen [56]. Mittels einer Analyse des verschlüsselten Datenverkehrs wird bei der Nutzung von realem Datenverkehr nach einer Trainingsphase von 24 Stunden eine Erkennungsrate von 23 Prozent erreicht, wird nur die Gruppe mit einfacher identifizierbaren Seiten betrachtet, steigt die Genauigkeit auf 40 Prozent an; werden drei Schätzungen für jede Seite vorgenommen, liegt die Erkennungsrate bei 100 Prozent. Das System arbeitet auf Datenströmen von Webseiten, welche mittels einem Datenstrom verschlüsselt werden, wie es bspw. bei Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), IPsec und SSH-Tunneln der Fall ist.

Die dritte Gruppe statistischer Ansätze wird für die Einbruchserkennung eingesetzt.

Yamada et al. haben ein System zur Angriffserkennung in verschlüsselten HTTP--Datenströmen präsentiert [396]. Für die Evaluation werden die Paketgrößen und deren Zeitmessungen für jeden einzelnen Webclient herangezogen, Zugriffshäufigkeiten bestimmt und bösartige Aktivitäten anhand von Regeln, die auf den Zugriffshäufigkeiten und den Charakteristika des Datenverkehrs beruhen, erkannt. Die Eigenschaften des HTTP-Datenverkehrs werden wiederum durch die maximalen Anfrage- und Antwortgrößen der jeweiligen Verbindung gekennzeichnet. Der daraus erzeugte Merkmalsvektor besteht aus 20 Werten, jeweils den 10 größten Paketen der jeweiligen Transportrichtung nach absteigenden Paketgrößen sortiert. Die Vektoren werden anhand des euklidischen Abstands gruppiert und deren Auftretenswahrscheinlichkeiten festgestellt, um mit den typischen Zugriffscharakteristika verglichen zu werden. Typische Anfragen an einen Webserver haben bspw. eine geringe Datengröße in Richtung des Webservers und eine hohe Datengröße in Richtung des Clients. Betrachtet man dahingegen die Durchführung eines Buffer Overflow-Angriffs, wird bspw. eine größere Anfrage zum Überlaufen des anfälligen Puffers gesendet. Schwellwerte für maximale Abfrage- bzw. Antwortgrößen sowie Zugriffsfrequenzen müssen für den jeweiligen Server festgelegt werden. Wird eine Frequenzanalyse zur Filterung von falsch-positiven Ergebnissen verwendet, werden unter bestimmten Bedingungen der beteiligten Paketgrößen der Anfragen und Antwor-

ten gute Detektionsresultate erreicht. Angemerkt sei, dass gemäß der Evaluation von Yamada das System bei maximalen Paketgrößen von 3000 für die Anfragen respektive 400 für die Antworten Fehlalarmraten von bis zu 9 Prozent nach Bereinigung durch die Frequenzanalyse erreicht, jedoch bei den realistischeren Werten für einen Webserver von maximalen Paketgrößen von 800 in der Anfrage und 2000 in der Antwort lediglich kombinierte Fehlalarmraten von 47 Prozent nach Bereinigung erlangt.

Ein anderes Sicherheitssystemen basierend auf der Evaluation von statistischen Daten haben Fouroshani et al. vorgestellt [134]. Für die Bewertung der Verbindungen werden lediglich die Paket-Größen sowie die zeitlichen Abstände zwischen den Paketen herangezogen. Anhand der Zugriffshäufigkeit auf einen Webserver und den Spezifikationen des TCP-Verkehrs werden Angriffssignaturen erstellt. Das System wurde mittels der Ein- und Ausgabedaten eines HTTP-Servers evaluiert. Durch die statistische Auswertung der Anfragen an den Server und dessen Antworten, konnte Fouroshani Angriffe im verschlüsselten Datenstrom entdecken. Bspw. sind bei der Durchführung eines Scan-Angriffes die Antworten des Servers typischerweise kleiner als reguläre Pakete, auch wenn die Anfragen des Scans selbst äquivalente Paketgrößen zu legitimen Zugriffen haben. Ein auf einer Skriptsprache basierender Angriff sendet kleine Anfragen in Bezug auf die Paketgrößen aus und erhält im Falle des Erfolges eine große Antwort zurück, was dem typischen Verhalten einer HTTP-Verbindung entspricht. Dies trifft jedoch nur im Erfolgsfall zu; kann der Angriff nicht greifen, ist die Antwort des Servers typischerweise ebenfalls klein.

Nachteilig an diesem Verfahren ist die Notwendigkeit, dass für jeden betrachteten Server zunächst ein Profil über die Zugriffsfrequenzen und -eigenschaften erzeugt werden muss. Ebenfalls zeigt das System von Fouroshani eine hohe Fehlalarmrate, die stark von der genauen Festlegung einiger Schwellwerte abhängt und nur selten unter 20 Prozent fällt.

Weitere Arbeiten im Bereich der Einbruchserkennung in verschlüsselten Umgebungen existieren, die sich ebenfalls in die drei genannten Kategorien einordnen lassen und gleiche Ausrichtungen und Schwachstellen haben, wie die vorgestellten Arbeiten (vgl. z.B. [399, 336]).

Nutzung von Host-Sensorik Eine weitere Möglichkeit, eine Analyse auf Payload-Ebene durchzuführen ohne jedoch aufwändig in den Netzverkehr eingreifen zu müssen, ist eine hostbasierte Evaluation. Der Empfänger der Pakete ist natürlicherweise in der Lage, diese zu dekryptieren, so dass eine anschließende Untersuchung erfolgen kann. Um nicht die Vorteile einer netzbasierten Einbruchserkennung zu verlieren, können die Evaluationen in Form von Sensoren in den jeweiligen Hosts durchgeführt werden und das Ergebnis an ein entsprechendes NIDS übermittelt werden. Ein diesbezüglicher Ansatz wird von Abimbola et al. gewählt [25]. Das vorgeschlagene Systems setzt sog. NetHost-Sensoren ein, welche im Netz-Stack der jeweiligen Rechner installiert sind. Diese befinden sich zwischen der Vermittlungs- und der Transportschicht, da hier die Pakete wieder unverschlüsselt vorliegen und fragmentierte Pakete wieder zusammengefügt sind. Der Datenverkehr wird durch die Sensoren an die zentrale, netzbasierte Auswerteinstanz

Tabelle 4.6: Übersicht der Verfahren zur Einbruchserkennung bei verschlüsseltem Datenverkehr.

	Protokoll	Modifizierend	Statistisch	Host
Erkennung auf Layer 7	✗	✓	✓	✓
Transparenter Einsatz	✓	✗	✓	✗
Konfigurationsfrei	✓	(✓/✗)[21]	(✓)[22]	✗
Fehlalarmraten	✓	(✓/✗)[21]	(✗)[23]	(✓/✗)[21]

gesendet. Das Verfahren bringt mehrere Nachteile mit sich, u.a. den hohen, administrativen Aufwand für die Pflege der Host-Sensoren, der Gefährdung des IDS durch einen kompromittierten Host oder der Gefahr von Performance-Einbrüchen bei hoher Last auf dem Netzlink. Weiterhin können Ende-zu-Ende-Verschlüsselungen auf Applikationsebene nicht analysiert werden.

Tabelle 4.6 zeigt eine Übersicht der genannten Verfahren sowie ihrer jeweiligen Vor- und Nachteile. Wie erkennbar ist, folgt aus den Anforderungen gem. Kapitel 3 unmittelbar, dass lediglich statistische Verfahren für die Entwicklung des geforderten Sicherheitssystems herangezogen werden können.

Die Einführung von Verschlüsselung bringt somit insbesondere folgende Probleme mit sich:

- Detektion nur von Angriffen spezifischer Gruppen
- Keine Nutzung des Payloads *oder*
- Komplexe und meist inkompatible Modifikationen von Protokollen und/oder Infrastrukturen
- Hohe Fehlalarmraten

4.6.4 Risiko von Datenverlust und Innentäter

Datenverlust hat sich zu einem der Hauptprobleme für Unternehmen entwickelt. Dieser kann sowohl durch fahrlässiges Verhalten von Mitarbeitern, falsche Konfigurationen oder Policies oder auch durch Innentäter ausgelöst werden (vgl. Kapitel 2.2.2). Nachfolgend aufgeführte, charakteristische Eigenschaften erschweren die Verhinderung von Datenverlust insbesondere:

[21] Abhängig des im System verwendeten Detektionsverfahrens.
[22] Möglich, jedoch bedürfen die bisherigen Verfahren typischerweise einer Konfiguration oder Erstellung von Profilen.
[23] Derzeitige Verfahren haben typischerweise hohe Fehlalarmraten von oftmals deutlich über 20 Prozent, vgl. Kapitel 4.5.

- Zahlreiche Sicherheitssysteme sind nicht darauf ausgelegt, Datenverluste zu erkennen, sondern untersuchen lediglich eingehende Verbindungen auf das Vorhandensein von Angriffen.

- Der Datenabfluß erfolgt typischerweise in einer systemkonformen Art: Durch die vorhandenen Rechte eines Innentäters oder eines unbewusst handelnden Mitarbeiters ist die jeweilige Aktion legitimiert und löst typischerweise keinen Alarm aus.

- Die Reaktion auf einen Datenabfluß kann nur erfolgreich sein, wenn sie insbesondere in Echtzeit stattfindet. Ein Vorgehen muss daher automatisiert sein, da bspw. ein abgeschlossener, unerwünschter Datentransfer erhebliche Schäden für eine Firma nach sich ziehen kann. Eine nachträgliche Analyse des Vorfalls kann den Schaden regelmäßig nicht reduzieren; daher muss eine entsprechende Übertragung automatisiert abgebrochen werden, sobald sie detektiert wird.

- Auf dem Markt befindliche Systeme zur Verhinderung von Datenabfluss (DLP) suchen typischerweise nach bestimmten Schlüsselwörtern oder zuvor erzeugten Dateimarkierungen. Insbesondere die Suche nach Signaturen kann bei deren Kenntnis jedoch durch geeignete Maßnahmen, bspw. Konvertierungen, etc. umgangen werden.

4.6.5 Rechtliche Betrachtung*

Der Einsatz eines IDS in einem produktiven Umfeld erfordert auch die Einhaltung der jeweiligen für den Datenschutz relevanten Gesetze. Wird gegen Datenschutzbestimmungen oder, wenn anwendbar, das Fernmeldegeheimnis verstoßen, führt die unzulässige Datenerhebung zu einem Beweisverwertungsverbot und rechtliche Schritte können nicht ergriffen werden. Im Gegenteil, wurden die Daten unrechtmäßig erhoben, droht eine Strafanzeige nach §206 StGB und §44 BDSG sowie ein Bußgeld nach §43 BDSG oder Schadensersatzanspruch nach §7 BDSG.

Zunächst ist daher entscheidend, welche durch ein IDS auszuwertenden Daten überhaupt unter den Datenschutz fallen. Hier gilt, dass alle Datenschutzgesetze lediglich den Schutz *personenbezogener Daten* bezwecken. Daten, die bspw. für statistische Zwecke erhoben werden und keinen Rückschluss auf natürliche Personen ermöglichen, unterliegen nicht dem Datenschutzgesetzen. Gemäß §3 Abs. 1 BDSG sind personenbezogene Daten „Einzelangaben über persönliche oder sachliche Verhältnisse einer natürlichen Person“ [149]. Hierzu zählen insbesondere Name und Adresse, Beruf, Telefonnummer, Mail- und IP-Adresse, Mailinhalte, Verbindungsdaten, etc. Gerade die IP-Adresse hat im Kontext eines IDS natürlich eine entscheidende Stellung und muss daher im weiteren Verlauf besonders betrachtet werden.

*Dieser Abschnitt fasst maßgeblich die für den vorliegenden Kontext relevanten Punkte des Buches „Praxis des IT-Rechts – Praktische Rechtsfragen der IT-Sicherheit und Internetnutzung“, 2. Auflage, Vieweg Verlag, 2007, zusammen.

Grundsätzlich ist jede Datenerhebung zunächst generell verboten (präventives Verbot mit Erlaubnisvorbehalt), es existieren jedoch zahlreiche Ausnahmen in Form von Erlaubnisvorbehalten. Ein zum Datenschutz rechtskonformer Einsatz eines IDS ist daher unerlässlich.

Bzgl. des Datenschutzes müssen maßgeblich folgende Gesetzestexte betrachtet werden (vgl. auch [101]):

- BDSG, insb. §§3a, 4, 4a, 5, 11, 14, 19a, 28 und 31
- TKG
- Teledienstedatenschutzgesetz (TDDSG), insb. §§ 4-6
- Richtlinie 95 / 46 / EG
- Im Rahmen der *Einführung*: Betriebsverfassungsgesetz (BetrVG) sowie Personalvertretungsgesetz des Bundes (BPersVG)

Welche Erhebungen und Untersuchungen des Datenverkehrs zulässig sind, hängt entscheidend vom vorliegenden Szenario ab. Im Rahmen eines großen Unternehmensnetzes können folgende Punkte zum Tragen kommen (vgl. Kapitel 2.1):

Fernmeldegeheimnis Zur Wahrung des Fernmeldegeheimnisses ist jeder Dienstanbieter gem. §88 Abs. 2 TKG verpflichtet. Hierunter wird eine nachhaltige Bereitstellung von Telekommunikation für Dritte mit oder ohne Gewinnerzielungsabsicht verstanden (vgl. §3 Nr. TKG), wobei Telekommunikation den technischen Vorgang des Versendens und Empfangens von Zeichen, Text, Sprache, Bildern oder Tönen mittels technischer Anlagen meint (§3 Nr. 22 TKG). Diese Anforderung ist bereits erfüllt, wenn ein Arbeitgeber einem Arbeitnehmer den Internetzugang gewährt.

Grundlegend muss unterschieden werden, ob eine Privatnutzung des Internets am Arbeitsplatz erlaubt ist oder nicht. Ist die Privatnutzung gestattet, gilt das Fernmeldegeheimnis und eine Überwachung des Datenverkehrs ist weitgehend beschränkt. Ist dahingegen eine private Nutzung nicht gestattet, gelten lediglich die Bestimmungen des BDSG, wodurch Kontrollen leichter möglich sind.

Anwendbar ist das Fernmeldegeheimnis nur auf private Daten, nicht jedoch auf beispielsweise Äußerungen in Newsgroups, die öffentlich zugänglich sind.

Im Falle des Geltungsbereiches des Fernmeldegeheimnisses dürfen nach §88 Abs. 3 TKG lediglich Daten für die technische und organisatorische Bereitstellung des Telekommunikationsdienstes erhoben werden, zudem herrscht eine Löschungspflicht. Ausnahmen vom Kontrollverbot werden jedoch durch §100 TKG definiert, u.a. zur Störungsbeseitigung und zur Missbrauchsbekämpfung. Im Rahmen der IT-Sicherheit kann daher die Gewährleistung eines sicheren und störungsfreien Ablaufs als Erlaubnistatbestand betrachtet werden.

Für die Aufzeichnung von Verbindungsdaten gilt, dass diese zunächst erhoben und für etwaige Notfälle vorgehalten werden dürfen, bspw. einem Missbrauchsfall. Inwieweit

Daten dann jedoch eingesehen und ausgewertet werden dürfen, ist eine Abwägungsentscheidung im Einzelfall.

Weitere datenschutzrechtliche Anforderungen sind im TDDSG definiert, welches jedoch nur zur Anwendung kommt, wenn der Arbeitgeber über die Zugangsvermittlung hinaus eine inhaltliche Nutzung zur Verfügung stellt. Bei einer reinen Bereitstellung des Internetzuganges findet es keine Anwendung.

Bundesdatenschutzgesetz Handelt es sich bei der Internet-Nutzung nur um eine dienstliche und ist eine private Nutzung nicht erlaubt, kommt lediglich das BDSG zur Geltung. Dieses schützt lediglich personenbezogene Daten und kommt bei nicht-öffentlichen Stellen nur bei einer automatisierten Verarbeitung zur Anwendung. Diese kann bei Erhebung, Verarbeitung und Nutzung der Daten erfolgen und liegt z.B. bereits vor, wenn eine dateigebundene Speicherung erfolgt. Zu beachten ist, dass auch bei einem Privatnutzungsverbot das Mitlesen aufgrund des Eingriffs in das Persönlichkeitsrecht i.A. nicht gestattet ist. Inhaltskontrollen kommen nur in Frage, wenn ein dringender Verdacht auf strafbare Handlungen besteht. Im Gegensatz dazu dürfen die äußeren Verbindungsdaten (Datum, Absender, Dauer der Verbindung [235] bzw. auch Datum, Uhrzeit, Datenumfang, Anzahl der E-Mails und Teile der Mail-Header [358]) festgehalten werden.

Tarifverträge und Unternehmesvereinbarungen können genutzt werden, um die Speicherung und Verarbeitung von Arbeitnehmerdaten zu regeln und verdrängen oder modifizieren das BDSG in Form einer zulässigen Ermächtigungsgrundlage.

Weiterhin schreibt das BDSG eine Zweckbindung bei der Datenerhebung vor, §3a BDSG legt außerdem das Gebot der Datenvermeidung und Datensparsamkeit bei der Erhebung persönlicher Daten vor. Bei der Gestaltungsauswahl von Datenverarbeitungssystemen sollen daher personenbezogene Daten so wenig wie möglich erhoben werden. Insbesondere soll die Möglichkeit der Anonymisierung von Daten genutzt werden. Weiter gilt gem. §4, dass die Zulässigkeit der Erhebung, Verarbeitung und Nutzung personenbezogener Daten durch andere Rechtsvorschriften oder durch eine explizite Einwilligung der Betroffenen erteilt werden kann.

Gemäß §14 (2) gilt weiterhin, dass das Speichern, Verändern oder Nutzen von Daten zulässig ist, wenn es zur Verfolgung von Straftaten oder Ordnungswidrigkeiten erforderlich ist.

Verkehrssicherungspflichten Der Bundesgerichtshof (BGH) spricht im Rahmen der Haftungssystematik von den sog. Verkehrssicherungspflichten (vgl. z.B. [299], [223]):

> Wer eine Gefahrenquelle eröffnet oder sich an ihr beteiligt, muss Dritte schützen und hierfür geeignete Schutzmaßnahmen ergreifen.

Die Kommunikationsvorgänge in Intranet und Internet eröffnen vielfältige Gefahren, sind also Gefahrenquellen im Sinne der Verkehrssicherungspflichten. Diese Verkehrssicherungspflichten bestehen im Wesentlichen aus:

- Organisationspflichten bezüglich betrieblicher (technischer) Abläufe und

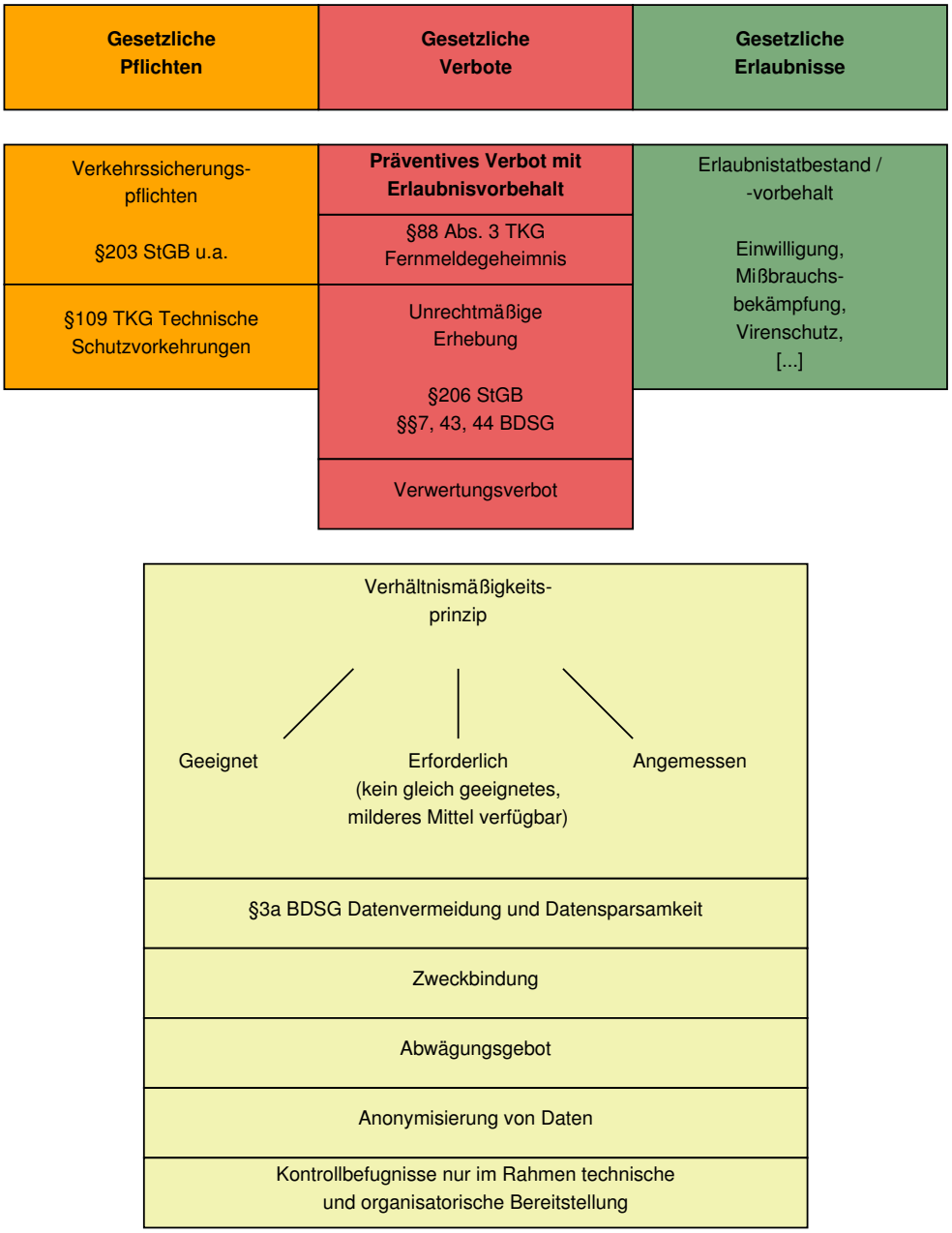


Abbildung 4.18: Übersicht der gesetzlichen Pflichten und Rechte. Pflichten wie das Treffen technischer Schutzvorkehrungen auf der einen Seite stehen umfangreichen Einschränkungen in Bezug der Datenerhebung gegenüber. Um dennoch effektive Maßnahmen ergreifen zu können, sind gesetzliche Erlaubnisse bspw. durch einen Vorbehalt in Form einer Einwilligung vorgesehen. Dennoch müssen die geltenden Restriktionen wie bspw. das Verhältnismäßigkeitsprinzip beachtet werden.

- Aufsichtspflichten des Arbeitgebers gegenüber seinen Mitarbeitern.

Verkehrssicherungspflichten ergeben sich aus verschiedenen gesetzlichen Bestimmungen. Beispielsweise begründet §203 StGB eine Garantenstellung bestimmter Berufsgruppen wie Kredit- und Finanzinstituten für sensible Daten. Diese müssen über angemessene Sicherheitsvorkehrungen für den Einsatz ihrer Elektronische Datenverarbeitung (EDV) verfügen.

Das BDSG §9 mit Anlage enthält die Grundsätze ordnungsgemäßer Datenverarbeitung [299]. Hierunter zählen u.a. auch Zugangskontrolle durch Passwörter sowie der Einsatz von Firewalls.

Sicherungslücken Auch wenn ein Unternehmen aufgrund seiner Tätigkeit nicht mit umfangreichen und rechtlich bindenden Verkehrssicherungspflichten belegt ist, erfordern die in der Rechtsprechung als Sicherungslücken bezeichneten Schwachstellen eine Absicherung des eigenen Netzes, um im Schadensfall überhaupt über eine Handhabe zu verfügen: Wird eine Sicherungslücke ausgenutzt, ohne dass eine entsprechende Zugangssicherung vorliegt, liegt *keine* Strafbarkeit vor, da der virtuelle Hausfriedensbruch in Netzen *nicht* strafbar ist. Dies zeigt die Bedeutung, welche der Ergreifung adäquater Schutzmaßnahmen zukommt.

https-Scanning Eine besondere Stellung nimmt das sog. https-Scanning ein. Aus Sicherheitsgründen werden Webseiten zunehmend verschlüsselt, die relevanten Standards sind hier SSL bzw. TLS. Nicht nur Webseiten für Online-Banking oder Zugriffe auf E-Mail-Provider, auch reguläre Seiten im Internet mit weniger sensiblen Daten werden hierbei zunehmend verschlüsselt. Aufgrund immer einfacher anzuwendender Programme ist die Gefährdung unverschlüsselter Kommunikation in den letzten Jahren stark angestiegen. Eindrucksvoll wurde dies durch das Firefox-Addon Firesheep bewiesen (vgl. Kapitel 3.1).

Mit der zunehmenden Verschlüsselung, deren Erfordernis aus Datenschutzgründen abgeleitet werden kann, der gleichzeitigen Forderung eines Schutzes vor Viren und Schadsoftware andererseits, entsteht jedoch ein Spannungsfeld zwischen Daten- und Systemschutz, bspw. in der Anlage zu §9 BDSG. Beide Elemente, Verschlüsselung und Virenschutz, sind wesentliche Elemente zur Gewährleistung der datenschutzrechtlichen Anforderungen. Ohne ein Aufbrechen der Verschlüsselung ist ein Virenscan nicht möglich. Für ein gesetzeskonformes Untersuchen der Verbindung müssen daher die Straftatbestände

- §202a Strafgesetzbuch (StGB) Ausspähen von Daten
- §206 StGB Bruch des Fernmelde-/Telekommunikationsgeheimnisses
- Ordnungswidrigkeit nach §43 BDSG

verhindert werden. Insbesondere können daher nachfolgende Zulässigkeitsvoraussetzungen abgeleitet werden:

- Vorhandensein eines konkreten Gefährdungspotentials

- Erfordernis der Maßnahme zur Gefahrenabwehr
- Möglichkeit zu optionalen Ausnahmen
- Geschlossenheit des Systems
- Aufbau des Systems als Blackbox, keine Einsichtnahmemöglichkeit durch Administratoren oder sonstige Dritte

Abbildung 4.19 veranschaulicht die zutreffenden Gesetze und daraus resultierende Möglichkeiten für verschiedene Szenarien. Gut zu erkennen ist, dass ohne Einholen eines Erlaubnisvorbehaltes im Falle einer erlaubten Privatnutzung keine ausreichenden Maßnahmen zur Ein- und Ausbruchserkennung ergriffen werden können. Insbesondere verhindert das Auswertungsverbot eine angemessene Nutzung von DLP-Systemen, da diese nur in Echtzeit aufgabengemäß arbeiten können. Eine Auswertung, die erst bei erkannter Gefahr im Verzug vorgenommen werden kann, ist hier zu spät. Daher ist bei erlaubter Privatnutzung ein Erlaubnisvorbehalt in Form einer Nutzervereinbarung einzuholen, durch welchen eine Analyse des Datenverkehrs unter den gezeigten Auflagen ermöglicht wird. Generell ist jedoch auch bei nicht erlaubter Privatnutzung eine entsprechende Nutzervereinbarung anzuraten. Bezogen auf das hier zugrunde gelegte Einsatzszenario in einer großen Netzumgebung einer Firma bedeutet dies, dass folgende Optionen genutzt werden können:

- Aufzeichnung der äußeren Verbindungsdaten
- Aufzeichnung statistischer Daten
- Ein entsprechender Erlaubnisvorbehalt erlaubt unter Beachtung des Verhältnismäßigkeitsprinzips eine entsprechende Auswertung

4.7 Übersicht der offenen Punkte

Obwohl IDSs bereits seit über zwanzig Jahren untersucht und weiterentwickelt werden, sind sie nicht in der Lage, einen umfassenden Schutz zur Verfügung zu stellen. Im Gegenteil, durch die rasch fortschreitende technologische Weiterentwicklung und den (oftmals nicht sichtbaren) Einzug des Internet in immer mehr Bereiche und Gegenstände des Alltags, wird es zunehmend schwerer, den immer ausgefeilteren Angriffsmethoden entgegenzutreten.

Tabelle 4.7 zeigt einen Überblick der festgestellten Problemfelder für den Einsatz aktueller IDSs.

Wie gut erkennbar ist, sind insbesondere wissenbasierte Verfahren nicht mehr in der Lage, den aktuellen Forderungen Rechnung zu tragen. Nicht vorhandene Signaturen, eine wachsende Anzahl zielgerichteter Angriffe, steigende Datenraten und immer größere Signaturdatenbanken sind hier Kernprobleme. Durch die zunehmende Verschlüsselung des Datenverkehrs geht sogar die prinzipielle Einsetzbarkeit dieser Systeme immer weiter

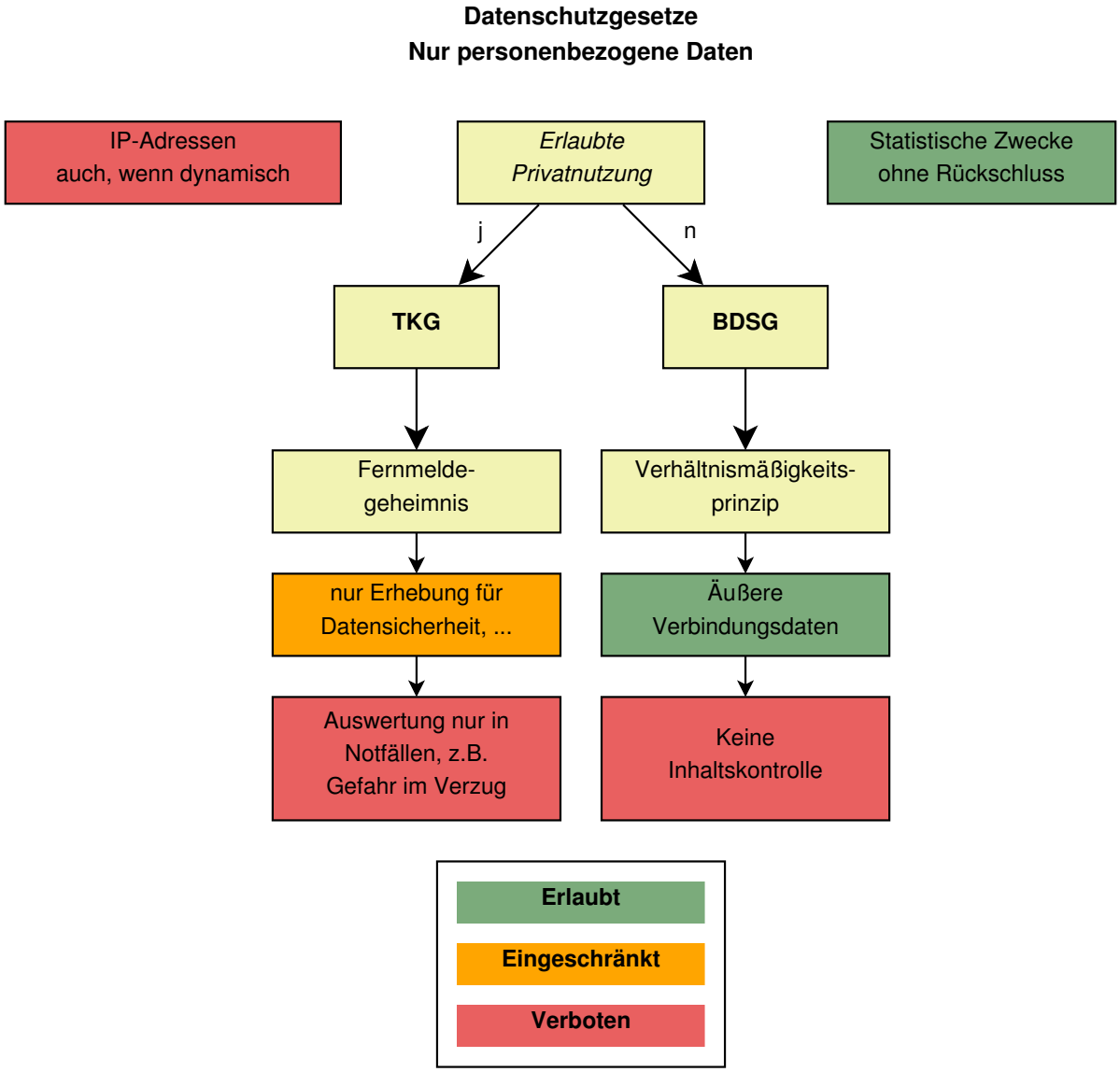


Abbildung 4.19: Auswahl anzuwendender Gesetze bzgl. der Datenauswertung. Abhängig einer erlaubten oder untersagten Privatnutzung gelten die Gesetze des TKG bzw. des BDSG.

Tabelle 4.7: Problemfelder aktueller IDS / IPS und DLP Systeme. ✓ bedeutet erfüllt, ✗ zeigt Schwächen der jeweiligen Systeme an, () bedeutet eingeschränkte Anwendung, ¬ steht für nicht zutreffend. EWSs sind inherent netzbasiert, entsprechend entfällt hier die Spalte hostbasierter Systeme.

	Intrusion Detection						
	Wissensbasiert		Verhaltensbasiert		Data Leakage		Early Warning
	Host	Netz	*Host*	Netz	Host	Netz	*Netz*
Konfiguration	✗	✗	✓	✓	✗	✗	(✓)
Zero Days	✗	✗	✓	✓	¬	¬	✓
Erfordernis Signaturen	✗	✗	✓	✓	¬	¬	✓
Datenraten	✗	✗	✓	(✓)	✓	(✓)	(✓)
Datenbankgrößen	✗	✗	✓	✓	✓	✓	(✓)
Angriffe auf Schicht 7	(✓)	(✓)	(✓)	✗	✓	(✓)	✓
Verschlüsselte Verbindungen	✓	✗	✓	✗	✓	✗	✗
Zielgerichtete Angriffe	✗	✗	(✓)	(✓)	¬	¬	✗
Verteilte Angriffe	✗	(✓)	✗	✓	¬	¬	✓

zurück. Verhaltensbasierte Systeme sind eher geeignet, den Anforderungen Rechnung zu tragen. Aktuelle Systeme und Forschungsarbeiten sind jedoch nicht in der Lage, alle Anforderungen zu erfüllen. Gerade im Bereich der Analyse verschlüsselter Verbindungen und der Detektion zielgerichteter Angriffe liegen hier offene Problempunkte. Rein von der grundlegenden Verfahrensweise sind diese Systeme jedoch eher in der Lage, diesen Punkten Rechnung zu tragen, als wissensbasierte Ansätze dies können.

4.8 Zusammenfassung

Das Kapitel führt eine umfassende Betrachtung der Systeme zur Ein- und Ausbruchserkennung durch. Hierfür wird ein Überblick über die geschichtliche Entwicklung gegeben und anschließend eine Klassifizierung anhand von Taxonomien gegeben. Die Problematik der Leistungsanalyse und dem Vergleich von IDSs wird dargelegt. Anhand des in Kapitel 3 aufgestellten Kriterienkataloges erfolgt eine Analyse und Bewertung der State-of-the-Art Systeme und Arbeiten aus der Forschung. Die offenen Punkte werden hervorgehoben und umfangreich analysiert, da diese Defizite als Basis für das Design einer Architektur für Ein- und Ausbruchserkennungssysteme der nächsten Generation herangezogen werden müssen.

Konzentriere Dich auf die Dinge, die vor Dir liegen, nicht auf die, die Du hinter Dir lässt.

Alan Cohen

5 Architektur eines IDS für verschlüsselte Umgebungen

Um den Defiziten der aktuellen IDSs Rechnung zu tragen, wird im vorliegenden Kapitel eine Architektur für ein Sicherheitssystem für verschlüsselte Umgebungen vorgestellt. Abbildung 5.1 zeigt den Aufbau des Kapitels.

Ziel des Kapitels ist es, neue Verfahrensweisen zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen vorzustellen. Hierfür wird zunächst untersucht, welche nutzbaren Informationen bei verschlüsselten Verbindungen noch zur Verfügung stehen und wie diese ausgewertet werden können, anschließend wird die Architektur des neuen Sicherheitssystems für Ein- und Ausbruchserkennung in verschlüsselten Umgebungen vorgestellt. Die notwendigen Module werden für die jeweiligen Teilbereiche der Datengewinnung (Kapitel 5.1.3), der Einbruchs- (Kapitel 5.1.4) sowie der Ausbruchserkennung und Identifizierung von Innentätern (Kapitel 5.1.5) gegliedert vorgestellt. Abschließend werden Betrachtungen über möglichen Gegen- bzw. Täuschmaßnahmen, die darauf abzielen, das Sicherheitssystem zu umgehen, angestellt und diskutiert.

5.1 Ein- und Ausbruchserkennung in verschlüsselten Umgebungen

Die auf Basis der Designanforderungen an ein IDS der nächsten Generation gestellten Forderungen werden nun als Grundlage für die Entwicklung einer entsprechenden Architektur herangezogen. Hierfür müssen insbesondere die eingeschränkt verfügbaren Informationen einer verschlüsselten Umgebung berücksichtigt werden.

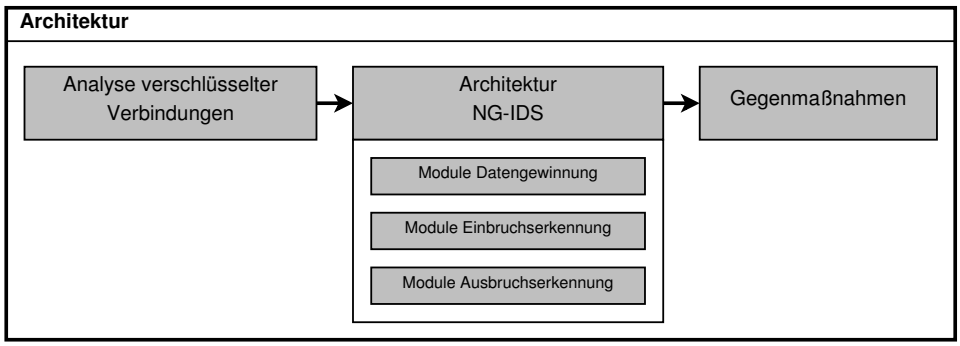


Abbildung 5.1: Aufbau von Kapitel 5.

5.1.1 Analyse verschlüsselter Verbindungen

Wie bereits in Kapitel 4.6.3 gezeigt, stellt die Einbruchserkennung in verschlüsselten Umgebungen besondere Herausforderungen dar, da entweder eine Änderung von Protokollen bzw. der Infrastruktur erforderlich ist und somit keine Kompatibilität bzw. generelle Einsetzbarkeit mehr gewährleistet werden kann, enorme Kosten entstehen, oder nur bestimmte Angriffe erkannt werden können bzw. die Fehlalarmraten zu hoch für einen Einsatz in einer Produktivumgebung sind.

Aus den grundlegenden Herausforderungen, verschlüsselte Verbindungen untersuchen zu können und dabei einen generell einsetzbaren Ansatz zu entwickeln, schließen sich Eingriffe in Protokolle bzw. Infrastruktur aus, womit lediglich eine statistische Evaluation der beobachtbaren Parameter möglich ist.

Nachfolgend werden daher die nutzbaren Informationen, welche bei einer Betrachtung der verschlüsselten Datenübertragung von Außen beobachtet werden können, beschrieben und analysiert.

Auswertbare Daten

Für den Informationsaustausch in lokalen Netzen, welche im vorliegenden Kontext betrachtet werden, werden maßgeblich Protokolle der IEEE 802.3 Ethernet- Familie sowie TCP/IP eingesetzt. Abbildung 5.2 zeigt den Aufbau eines tagged Ethernet-II Frames, wie er hauptsächlich in der Kommunikation verwendet wird. Die Präambel wird zur Synchronisation des Empfängers eingesetzt und besteht aus einer Schwingung von 0 und 1 für eine Dauer von $6.4\mu s$; Ziel- und Sender- Media Access Control (MAC) geben die Adressen des Empfängers bzw. Absenders an. Das Tag-Feld kann für die Nutzung von Virtual LANs (VLANs) gemäß 802.1q genutzt werden, das Typfeld spezifiziert das verwendete Protokoll der Schicht 3. Das Feld Frame Check Sequence (FCS) stellt einen Cyclic Redundancy Check (CRC)-Wert des Frames zur Verfügung, zum Abschluss folgt der Interframe-Gap mit einer Länge von $9.6\mu s$. Für die Nutzung des weit verbreiteten Internet Protocol wird das Typfeld für IPv4 auf $0x0800$, für IPv6 auf $0x86DD$[1] gesetzt. Auf der Transportschicht wird im Rahmen verschlüsselter Verbindungen weiterhin TCP eingesetzt, wodurch zuverlässige Verbindungen garantiert werden.

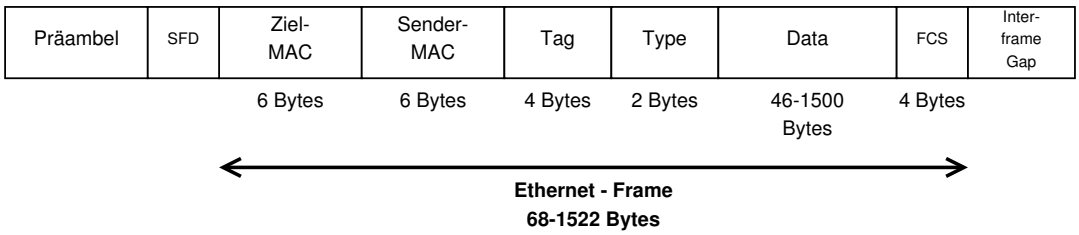


Abbildung 5.2: Aufbau eines tagged Ethernet-II Frames.

[1] Ether Types, Internet Assigned Numbers Authority (IANA).

Tabelle 5.1: Bei TCP vorhandene Flags, sowie Erweiterung der klassischen Flags um ECE und CWR zur Staukontrolle gem. RFC 2481.

Flag	Bedeutung
ACK	Acknowledge
FIN	Finish
PSH	Push
RST	Reset
SYN	Synchronisation
URG	Urgent
ECE	ECN Echo Bit
CWR	Congestion Window Reduced

Welche Daten nach einer Enkryption noch zur Verfügung stehen, zeigen Abbildung 5.3 und 5.4 beispielhaft anhand von IPsec bzw. SSH.

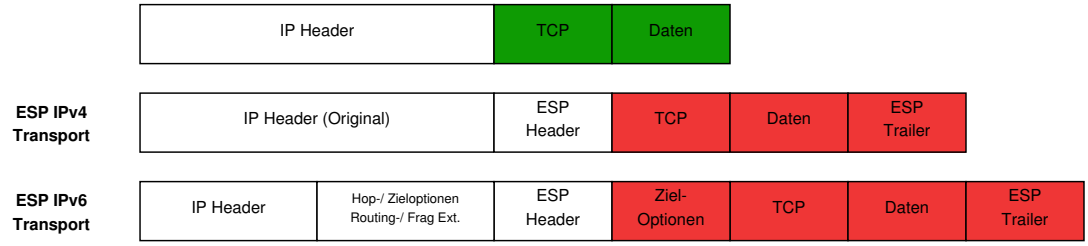


Abbildung 5.3: Verschlüsselung mittels IPsec im Transportmodus bei IPv4 und IPv6.

Wird IPsec zur Verschlüsselung von mittels IPv4 übertragenen Daten verwendet, wird der Header im Original übernommen und um den Encapsulating Security Payload (ESP)-Header ergänzt; der TCP-Frame und die Nutzdaten werden komplett verschlüsselt. Der Aufbau von IPv6 unterscheidet sich von dem von IPv4, natürlich gilt aber auch hier, dass sämtliche Daten des TCP-Frames sowie der Payload verschlüsselt sind. Tabelle 5.1 fasst die Flag-Informationen zusammen, welche durch die Verschlüsselung des TCP-Frames nicht mehr zur Verfügung stehen. Dies bedeutet insbesondere, dass Informationen bzgl. des Status des Verbindungsaufbaus nicht mehr zur Verfügung stehen.

Betrachtet man eine per SSH verschlüsselte Verbindung, ergibt sich die in Abbildung 5.4 dargestellt Situation; in diesem Falle sind die Flags des TCP-Frames *nicht* verschlüsselt, jedoch die weiteren Daten des Pakets: Die Länge des Pakets (ohne das Längenfeld selbst und die Message Authentication Code-Felder), die Nutzdaten sowie das Padding. Letzteres erweitert die Daten um 1 bis 8 Byte zufälliger Daten, um Angriffe, die auf

bekannten Klartext basieren, zu verhindern[2].

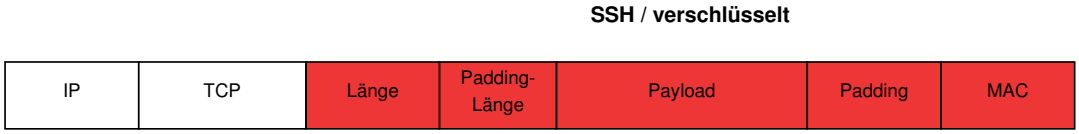


Abbildung 5.4: Aufbau eines SSH-verschlüsselten Datenpakets.

Somit sind folgende Parameter für eine statistische Auswertung verschlüsselter Datenpakete verfügbar:

- Die Paketgrößen der verschlüsselten Daten
- Eingeschränkt Flag-Informationen des genutzten Transportprotokolls

Um eine allgemeine Anwendbarkeit zu gewährleisten, werden nachfolgend keine Flags für die Angriffsdetektion genutzt. Ein weiteres Merkmal, das jederzeit bei einem Paketstrom beobachtet werden kann, ist der Auftretenszeitpunkt eines jeden Pakets an einem Beobachtungsort auf der Route der Pakete. Dieser wird neben den Paketgrößen als zweites, allgemein verfügbares Merkmal herangezogen.

Insbesondere IPsec kommt künftig eine besondere Bedeutung zu, da dieses inhärenter Bestandteil von IPv6 ist. Mit einer durch das Ende der freien IPv4-Adressblöcke[3] zukünftig zu erwartenden, stärkeren Nutzung von IPv6 wird somit der Anteil verschlüsselten Datenverkehrs zusätzlich ansteigen.

Die beobachtbaren Parameter Beobachtungszeitpunkt und Paketgröße sind ausschnittsweise für eine beispielhafte Verbindung in Tabelle 5.2 dargestellt.

Neben dem Zeitpunkt der Detektion[4] ist die laufende Paketnummer der Übertragung, die Zeit zwischen den jeweils aufeinanderfolgenden Paketen (Zwischenankunftszeit Δ_t), die Größe des Payloads sowie Quell- und Zieladresse im IPv4-Format aufgezeichnet. Pakete mit einer Payload-Größe von 0 Bytes und gesetztem ACK-Flag sind nicht aufgeführt; diese Pakete dienen der Bestätigung des Empfangs an den Absender und werden im weiteren Verlauf nicht benötigt. Grund hierfür ist, dass der leere Payload keine relevanten Informationen enthält und für die nachfolgend vorgestellten Verfahren nicht herangezogen wird.

Kreuzkorrelation

Um auf Basis der beobachtbaren Daten verwertbare Aussagen treffen zu können, muss eine entsprechende Analyse betrieben werden. Da die eigentlichen Inhalte der Verbindungen durch die Verschlüsselung nicht mehr verfügbar sind, müssen erkennbare Eigenschaften ausgewertet werden. Der Gedanke ist hierbei, dass charakteristische Eigenschaften

[2] Vgl. z.B. [227].
[3] Vergabe der letzten Adressblöcke im Februar 2011.
[4] Grundlage UNIX-Timestamp, Sekunden seit dem 01.01.1970.

Tabelle 5.2: Statistische Daten der aufgezeichneten Pakete einer Datenverbindung (`ACK`-Pakete sind nicht aufgeführt).

Beobachtungszeit	Seq.-Nr.	Δ_t	Paket-Größe	Quelladresse	Zieladresse
1271361720.982397	186	0.011087	48	192.168.0.20	192.168.0.20
1271361721.827251	188	0.833673	48	192.168.0.10	192.168.0.10
1271361721.838364	189	0.011113	848	192.168.0.20	192.168.0.20
1271361722.394998	191	0.545447	48	192.168.0.10	192.168.0.10
1271361722.406128	192	0.011130	48	192.168.0.20	192.168.0.20
1271361722.643165	194	0.225903	48	192.168.0.10	192.168.0.10
1271361722.654261	195	0.011096	48	192.168.0.20	192.168.0.20
1271361723.907355	197	1.241911	48	192.168.0.10	192.168.0.10
1271361723.918444	198	0.011089	560	192.168.0.20	192.168.0.20
1271361724.715200	200	0.785568	48	192.168.0.10	192.168.0.10
1271361724.726323	201	0.011123	48	192.168.0.20	192.168.0.20

von typischen Verbindungen wie bspw. einem gutartigen Zugriff auf eine Webseite oder einem bösartigen Passwortangriff auch nach der Verschlüsselung erhalten bleiben und für die Beurteilung und Klassifizierung der Verbindung herangezogen werden können.

Hierfür können Ähnlichkeitsmaße (Similaritäten) und Distanzmaße eingesetzt werden. Während sich bei Ähnlichkeitsmaßen eine starke Ähnlichkeit zwischen den Variablen in hohen Ergebniswerten ausdrückt, ergeben sich bei Distanzmaßen hier geringe Werte; letztere werden daher auch als Unähnlichkeitsmaße bezeichnet [65].

In der Literatur sind zahlreiche Verfahren definiert (vgl. z.B. [186, 29]). Abhängig des jeweiligen Anwendungsbereiches können bspw. Verfahren eingesetzt werden, welche alle Meßwerte gleich behandeln, gewichten, Häufigkeiten der Werte berücksichtigen oder bei denen die Differenzen der Abstände der Werte unterschiedlich stark eingehen. Im Rahmen einer Echtzeitauswertung eines IDS muss hierbei auch die Komplexität der genutzten Methode berücksichtigt werden, um die entsprechend für einen Datenlink benötigte Analyseleistung erbringen zu können.

Eine der einfachsten Ähnlichkeitsmessungen ist die Summe absoluter Differenzen:

$$SAD = \sum_{i}^{n} |X_i - Y_i| \tag{5.1}$$

Die Berechnung des Euklidischen Abstands ist ein weiteres, einfaches Maß und erfolgt gem. nachfolgender Formel:

$$ED = \sqrt{\sum_{i}^{n} (X_i - Y_i)^2} \tag{5.2}$$

Der wichtigste Unterschied zwischen den Ergebnissen der Formeln 5.1 und 5.2 ist, dass beim Euklidischen Abstand eine Quadrierung der Differenzen erfolgt und somit

große Differenzen stärker in die Berechnung eingehen, als dies bei der Summe absoluter Differenzen der Fall ist.

Betrachtet man eine Datenverbindung, lassen sich deren Wertereihen auch als ein Signal über die Zeit t interpretieren.

Um hier eine entsprechende Ähnlichkeitsbetrachtung anzustellen, eignen sich insbesondere die Verfahren der Kreuzkorrelation, mit welchen die Ähnlichkeit zweier Zeitfunktionen bestimmt werden kann. Die Kreuzkorrelation zweier Funktionen $x(t)$ und $y(t)$ ist wie folgt definiert:

$$s(\tau) = \int_{-\infty}^{\infty} x(t)\, y(t+\tau)\, \mathrm{d}t \tag{5.3}$$

Wird diese Funktion auf $x(t)$ und $y(t)$ angewandt, wird die Funktion $y(t)$ zeitlich um den Wert τ verschoben, mit der Funktion $x(t)$ multipliziert und anschließend integriert. Somit ergibt sich ein Maximum für die Verschiebung τ, bei der sich die betrachteten Funktionen am ähnlichsten sind.

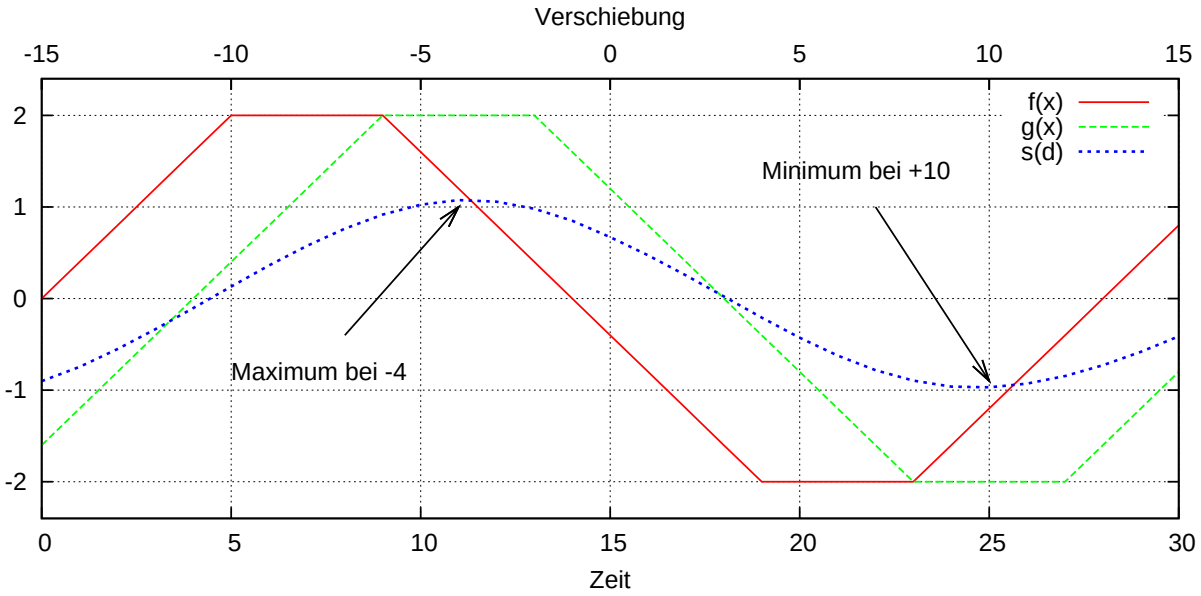
Abbildung 5.5 zeigt ein einfaches Beispiel einer Kreuzkorrelation zweier trapezförmiger Signale $f(x)$ und $g(x)$.

Da die gemessenen Informationen regelmäßig und insbesondere im hier betrachteten Fall als Datenverbindungen nicht als Funktionen, sondern als zeitdiskrete Wertefolge vorliegen, kann das vorangegangene Integral nicht direkt verwendet werden, sondern muss in eine entsprechende Summenfunktion überführt werden. Hierfür werden die Werte f_i bzw. g_i zu diskreten Zeitpunkten $t_0 + \Delta t, ..., t_0 + i \cdot \Delta t, ...t_0 + N \cdot \Delta t$, mit $i = 1..N$ genutzt, anschließend kann das Ergebnis durch den Effektivwert der Wertefolgen geteilt werden, um es zu normieren:

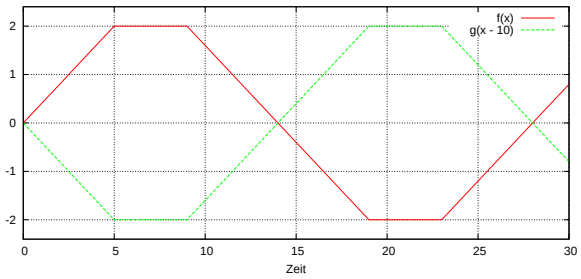
$$s_d = \frac{\sum_{i=1}^{n} \left[(f[i] - m_f) \cdot (g[i-d] - m_g) \right]}{\sqrt{\sum_{i=1}^{n} (f[i] - m_f)^2} \sqrt{\sum_{i=1}^{n} (g[i-d] - m_g)^2}} \tag{5.4}$$

Hierbei entsprechen m_f und m_g den Mittelwerten der jeweiligen Wertereihen von $f[x]$ bzw. $g[x]$, d ist die jeweils betrachtete Verschiebung von $g[x]$. Sind die Werte der Wertereihen für ein bestimmtes d gleich, ergibt die normierte Kreuzkorrelation den Wert 1 (vgl. Abbildungen 5.5a und 5.5c), sind sie gegenphasig, ergibt sich der Wert -1 (vgl. Abbildungen 5.5a und 5.5b). Ergibt die normierte Korrelation für alle d für die komplette Wertereihe Null, sind die Reihen unkorreliert.

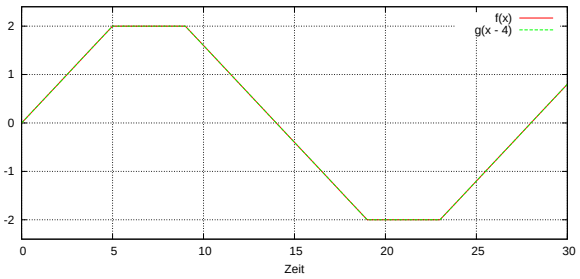
Abbildung 5.6 zeigt beispielhaft Kreuzkorrelationen der in Tabelle 5.2 dargestellten Paketserie mit drei weiteren Paketserien, einer konstanten Serie von Datenpaketen einheitlicher Größe sowie zwei Ausschnitte ebenfalls aufgezeichneter Serien. Abhängig der jeweiligen zeitlichen Verschiebung im Bereich $[-10, .., 10]$ ergeben sich die in der Grafik dargestellten Korrelationswerte. Gut zu erkennen ist die hohe Korrelation mit der Paketserie 1, in welcher die gleichen Paketsequenzen enthalten sind, wie in der Paketserie gem. Tabelle. Betrachtet man die Korrelationsverläufe mit den anderen Paketserien,



(a) Kreuzkorrelation zweier Signale $f(x)$ und $g(x)$. $s(d)$ zeigt die Ergebnisse der Korrelation für Verschiebungen im Bereich $[-15; 15]$.



(b) Verschiebung der Funktion $g(x)$ zum Minimum der Korrelation. Die Funktionen liegen achsensymmetrisch.



(c) Verschiebung der Funktion $g(x)$ zum Maximum der Korrelation. Die beiden Funktionen decken sich und haben in dieser Lage die höchste Ähnlichkeit.

Abbildung 5.5: Veranschaulichung der Kreuzkorrelation.

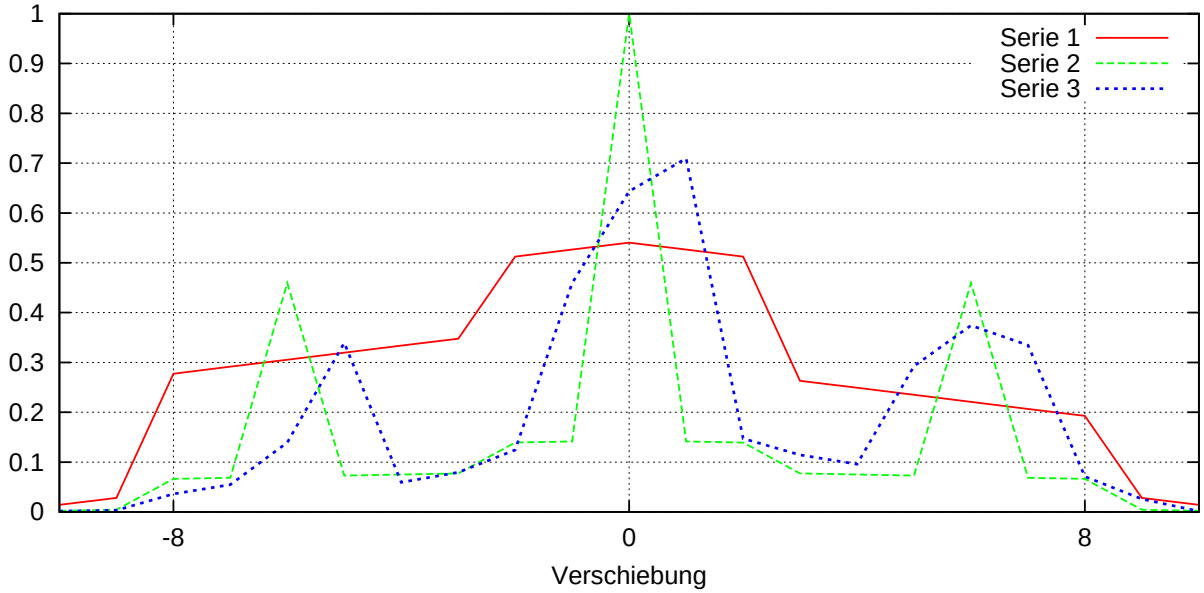


Abbildung 5.6: Beispielhafte Kreuzkorrelation einer Serie verschiedener Paketgrößen mit drei unterschiedlichen Referenzserien. Bei der optimalen Verschiebung und der passenden Referenzserie wird eine hohe Korrelation erreicht.

ergeben sich geringere, maximale Werte, da in diesen nur kürzere Fragmente gleicher Paketfolgen enthalten sind. Auffällig ist weiterhin der regelmäßige Verlauf im Falle der Korrelation mit ähnlichen Paketserien, während bei mehr unterschiedlichen Serien unregelmäßigere Strukturen entstehen.

5.1.2 Sicherheitssystem für verschlüsselte Umgebungen*

Nachfolgend wird die Architektur für ein Sicherheitssystem, welches den gestellten Anforderungen genügt, vorgestellt. Abbildung 5.7 zeigt den schematischen Überblick des ursprünglichen Systems, welches auf die Erkennung von Innentäter-Angriffen und den Missbrauch kompromittierter Nutzerkonten spezialisiert ist.

Abbildung 5.8 zeigt die Erweiterung der Systems um die Komponenten *SSH-Brute Force-Erkennung* sowie *TLS-Angriffs-Erkennung*. Diese beiden Module ergänzen die Architektur um Detektionskomponenten, welche eine Einbruchserkennung im klassischen Sinne, durchgeführt durch einen externen Angreifer, bereitstellen. Die Komponenten *Befehlsevaluierung* und *Nutzeridentifizierung* arbeiten auf dem zuvor in Cluster aufgeteilten Datenstrom und dienen der Erkennung von Innentätern sowie der Nutzung kompromittierter Nutzerkonten durch externe Angreifer.

Die unterschiedliche Nutzung der Paket- bzw. Clusterströme und die verschiedenen

*Dieser Abschnitt enthält eine Zusammenfassung des Posters und Kurzartikels „Security System for Encrypted Environments (S2E2)“, Proceedings of the 13th International Conference on Recent Advances in Intrusion Detection (RAID), Springer-Verlag, 2010.

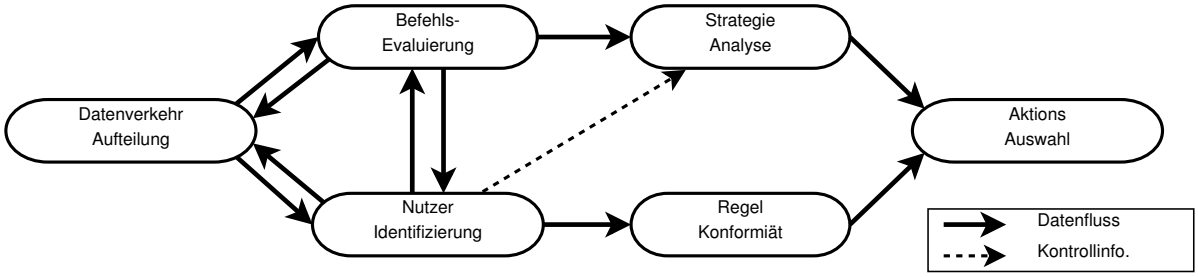


Abbildung 5.7: Schematische Sicht des Sicherheitssystems für verschlüsselte Umgebungen (*Security System for Encrypted Environments, S2E2*).

Korrelationen, welche in den jeweiligen Modulen zur Angriffsdetektion eingesetzt werden, sind im weiteren Verlauf ausführlich dargelegt.

Abbildung 5.9 zeigt die Gesamtübersicht der Architektur. Die Architektur des Sicherheitssystems lässt sich in die drei Bereiche

- Datengewinnung- und Aufbereitung
- Einbruchserkennung
- Ausbruchs- und Innentätererkennung

gliedern. Die *Datengewinnung* dient der Generierung der benötigten statistischen Daten anhand der beobachteten Datenpakete des Netzes. Um eine effiziente Verarbeitung im System zu ermöglichen, werden diese anschließend mittels der Nutzung von Hash-Tabellen *aufbereitet*. Der Bereich der *Einbruchserkennung* dient der Detektion von Angriffen, bei denen noch kein Zugang zum Zielsystem vorhanden ist, während die *Ausbruchs- und Innentätererkennung* von einem kompromittierten Zugang (Vorhandensein von Nutzerrechten) oder autorisierten Handlungen (Innentäteraktivität) ausgeht und eine entsprechende Erkennung ermöglicht.

Nachfolgend werden die einzelnen, zugehörigen Module des Systems detailliert vorgestellt.

5.1.3 Module zur Datengewinnung

Zur Datengewinnung gehören die Datensonde, welche die Netzpakete abfängt und die erforderlichen Daten extrahiert, ein Modul mit Hashing-Funktionalitäten sowie die Clustererzeugung. Da letztere *nur* für die Ausbruchs- bzw. Innentäterdetektion benötigt wird, ist sie logisch im entsprechenden Bereich der Ausbruchserkennung aufgehängt (vgl. Abbildung 5.9).

Datensonde (Probe)

Die Analyse des Netzdatenverkehrs erfolgt mittels einer Sonde, die beliebig in das zu überwachende Netz integriert werden kann. Die Positionierung auf einer transparenten

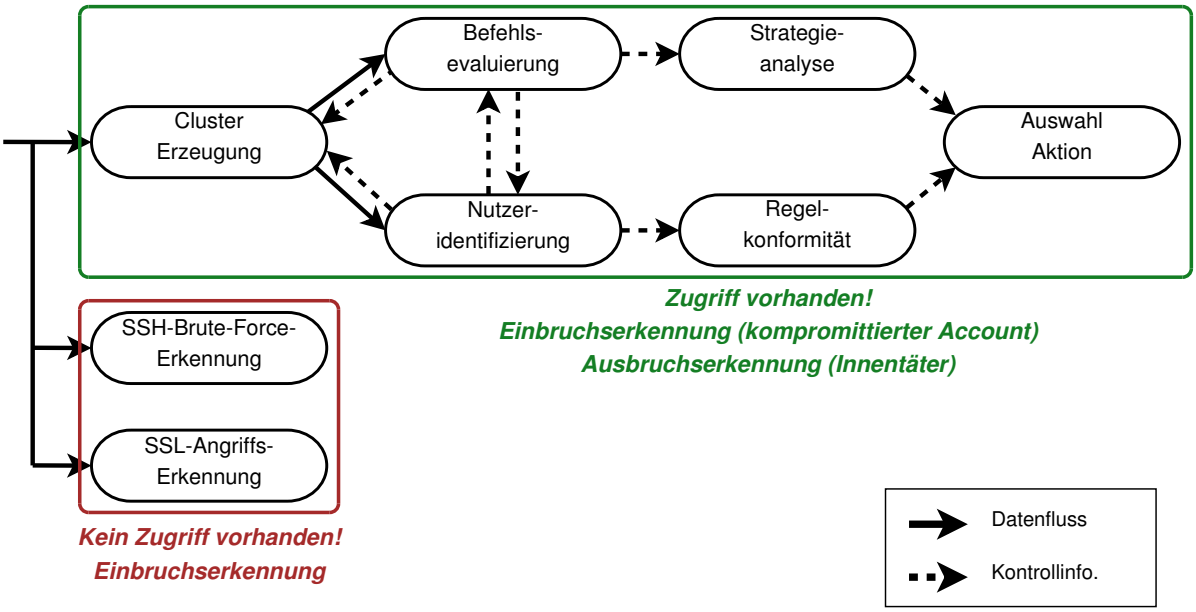


Abbildung 5.8: Erweiterung des Sicherheitssystems für verschlüsselte Umgebungen (*S2E2*). Während die Komponenten der Befehlsevaluation und Nutzeridentifikation die Erkennung von Innentätern und kompromittierten Accounts ermöglichen, dienen die Module der Brute Force- und Angriffserkennung der klassischen Einbruchserkennung von externen Tätern.

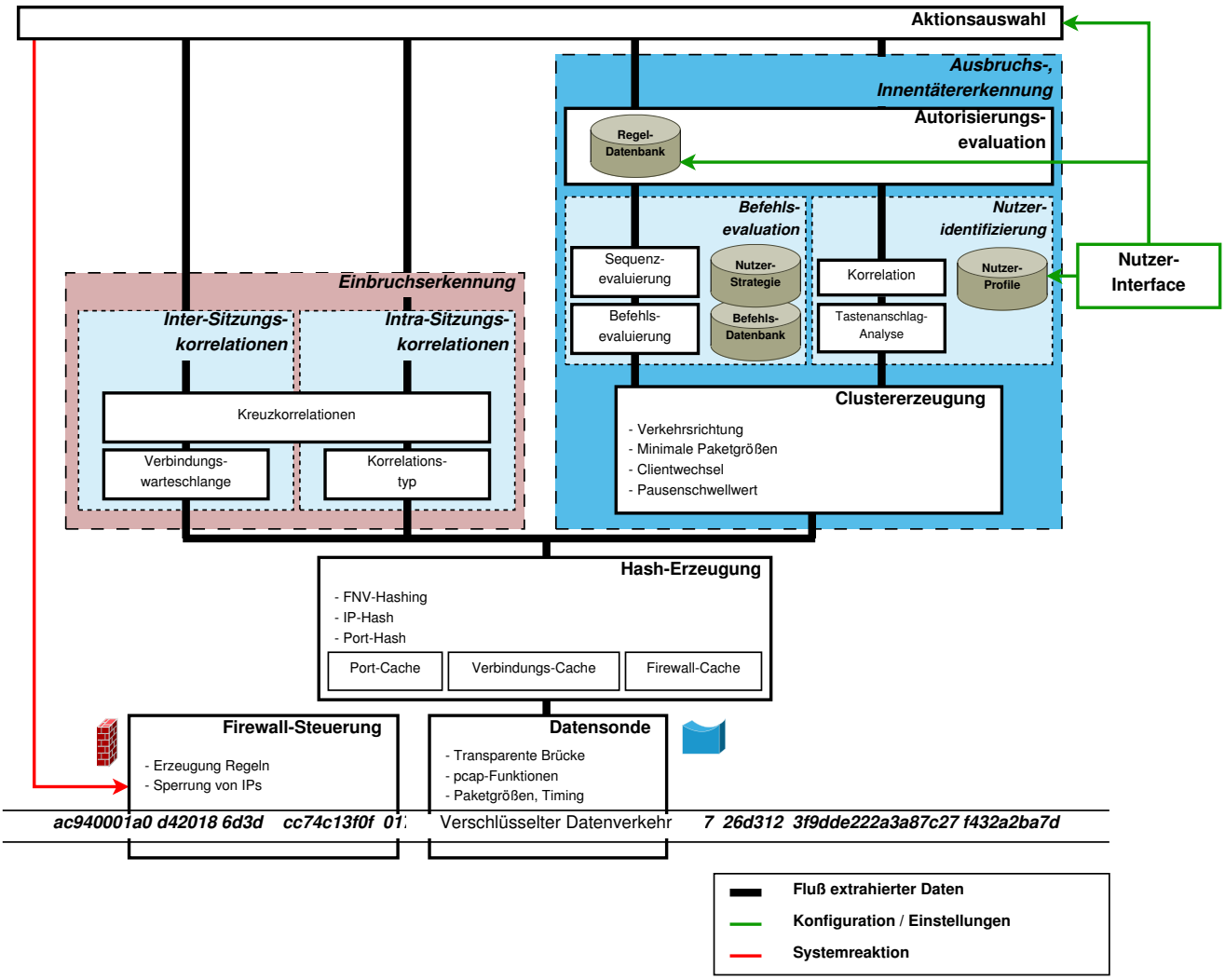


Abbildung 5.9: Architektur des Sicherheitssystems für verschlüsselte Umgebungen. Die Datensonde fängt die Netzpakete ab und bereitet die statistischen Daten auf, die in Hash-Tabellen mit den zugehörigen Verbindungen verwaltet werden. Anschließend werden die Daten an die verschiedenen Module der Ein- und Ausbruchserkennung weitergegeben.

Brücke oder einem Gateway bzw. dem Netzübergang zum Internet bietet sich an, um den gesamten Datenverkehr umfassend und mit einer minimalen Anzahl von Sonden zu untersuchen.

Um einen Betrieb in Umgebungen mit hohen Datenraten zu ermöglichen, ist eine performante und parallelisierbare Implementierung der Sensorik notwendig. Gem. der Analyse in Kapitel 5.1.1 können folgende Daten aufgezeichnet werden:

- Zeitpunkt, zu welchem jedes Netzpaket die Sonde passiert
- Größe des Payloads
- Gesetzte Flags

Wie unter Kapitel 5.1.1 gezeigt, sind Flags abhängig der genutzten Verschlüsselung nicht notwendigerweise verfügbar und werden daher *nicht* aufgezeichnet bzw. berücksichtigt.

Die Integration der Datensonde in einer transparenten Netzbrücke ermöglicht eine Untersuchung des Datenverkehrs, ohne dass ein Angreifer davon Erkenntnis erlangen kann. Für den Zugriff auf den Netzverkehr werden Funktionen der Packet Capture Library (pcap) [8] genutzt. Diese stellen eine Schnittstelle zum Zugriff auf alle Pakete des Netzes bereit, auch solche, die nicht an den Rechner gerichtet sind. Hierzu wird zunächst ein Handler auf der entsprechenden Netzschnittstelle installiert, sowie ein Filterausdruck für die zu kopierenden Daten angegeben. Da im Rahmen der Netzüberwachung jegliche Daten berücksichtigt werden sollen, muss hier keine Einschränkung vorgenommen werden. Jedoch kann hier bereits eine Möglichkeit der Parallelisierung des Systems angesetzt werden, indem mehrere Sonden jeweils verschiedene Teilbereiche des Datenverkehrs analysieren, bspw. getrennt durch die Zielserver im lokalen Netz. Mittels des kompilierten Filterausdrucks und der bei Erhalt eines neuen Pakets aufzurufenden Funktion wird der Abhörvorgang gestartet (vgl. Kapitel F.3.3).

Trifft ein neues Paket am Netzinterface ein, wird dieses durch den pcap-Handler abgefangen und durch das Modul weiter verarbeitet. Hierfür steht die Funktion `got_packet()` zur Verfügung. Diese untersucht das Netzpaket zunächst dahingehend, ob es im Rahmen des Sicherheitssystems betrachtet werden muss. Erfüllt es nicht die geforderten Eigenschaften, bspw. ein ACK-Paket mit einer Payload-Größe von 0 Byte, wird es verworfen und nicht durch das System betrachtet. Wurde ein Paket als zu verarbeitendes erkannt, wird dieses an die Hash-Erzeugung weitergegeben.

Hash-Erzeugung

Um eine effiziente und schnelle Verarbeitung der Daten zu ermöglichen, werden für jede neu erkannte Verbindung Hashwerte zur eindeutigen Identifikation und weiteren Verarbeitung erzeugt. Hierbei werden von den unterschiedlichen Modulen mehrere Arten von Hashwerten genutzt: Einerseits Werte, die lediglich die IP-Adressen der beteiligten Kommunikationspartner berücksichtigen und zum anderen Datensätze, welche auch die genutzten Ports mit beinhalten. Dies ist erforderlich, da für einige Module eine besondere Handhabung verschiedener Ports einer IP-Adresse erfolgen muss, für andere aber

nicht. Neben einer effizienten Verarbeitung innerhalb der Module ermöglicht das Hashing der IP-Adressen gleichzeitig eine Anonymisierung der beteiligten IP-Adressen, um den Anforderungen gem. Kapitel 4.6.5 Rechnung zu tragen. Da jedoch auch andererseits eine Reaktion bei Erkennung eines Angriffes bzw. Sicherheitsverstoßes gem. dem aufgestellten Kriterienkatalog erfolgen können muss, kann in diesem Fall die IP-Adresse des Angreifers *nicht* vollständig anonymisiert werden, da sie im Rahmen der zu ergreifenden Maßnahmen, insbesondere der Sperrung der Adresse mittels einer Firewall, weiterhin vorliegen muss. Soll lediglich eine Detektion von Angriffen erfolgen, ist wiederum eine komplette Anonymisierung der Daten möglich; entsprechend wird diese Einstellung als Konfigurationsoption verfügbar gemacht. Für die Erzeugung der Hashwerte aus den IP-Port-Kombinationen wird das Hashing-Verfahren Fowler, Noll und Vo (FNV) genutzt [296]. Dieses ist mit Hinblick auf die Geschwindigkeit unter gleichzeitig geringer Kollissionswahrscheinlichkeit entwickelt. Die hohe Streuung der erzeugten Hashwerte ermöglicht insbesondere einen Einsatz für das Hashing fast identischer Strings, wie sie bspw. bei IP-Adressen vorliegen (vgl. Kapitel F.3.4). Der FNV-Algorithmus wird hierbei unter Nutzung der Hashtable-Implementierung von Christopher Clark in das Sicherheitssystem integriert (vgl. Kapitel F.3.5). Als Schlüssel für die Hashtabellen werden die IP- und ggf. Port-Informationen von den Kommunikationspartnern des abgehörten Datenpakets verwendet, hierfür werden zwei Werte für den Schlüssel und den inversen Schlüssel gebildet (Pseudocode, vgl. Kapitel F.3.6):

```
   test_ip_str = ip_src + ip_dst + src_port;
rev_test_ip_str = ip_dst + ip_src + dst_port;
```

Da das erste Paket, welches eine Kommunikation zwischen Client und Server initiiert, immer vom Client gesendet wird, kann dies zur Erkennung der Transportrichtung genutzt werden. Die Mitnutzung des Ports ist hier entscheidend, um mehrfache Verbindungen eines Clients für die nachfolgende, weitere Bearbeitung unterscheiden zu können. Ist bereits ein Paket der Verbindung vorhanden, kann die vorliegende Transportrichtung durch die beiden aufgestellten Schlüssel bestimmt werden. Dieses Vorgehen ist notwendig, da beim Zugriff auf die Hashtabellen nur der Schlüssel effizient überprüft werden kann und die Zusatzinformationen über die jeweilige Rolle einer Adresse (Client oder Server) hier nicht direkt enthalten ist[5]. Abbildung 5.10 zeigt den Ablauf der Erkennung der Transportrichtung mittels der Hashtabellen-Schlüssel.

Neben diesen Hashwerten werden auch entsprechende ohne Port-Informationen erzeugt, die jedoch erst für Bewertungen innerhalb der jeweiligen Module eingesetzt werden und nicht zur Identifizierung der Verbindungen dienen. Mittels des erkannten bzw. neu erzeugten Hashwertes einer Verbindung wird eine zugehörige Datenstruktur verwaltet, welche insbesondere die benötigten, statistischen Parameter der jeweiligen Verbindungen

[5] Eine andere Möglichkeit ist die Aufteilung der Datenstruktur oder das Durchsuchen der Hashtabelle mittels eines Iterators, die jedoch aus Effizienzgründen im Rahmen des vorliegenden Systems verworfen wurden.

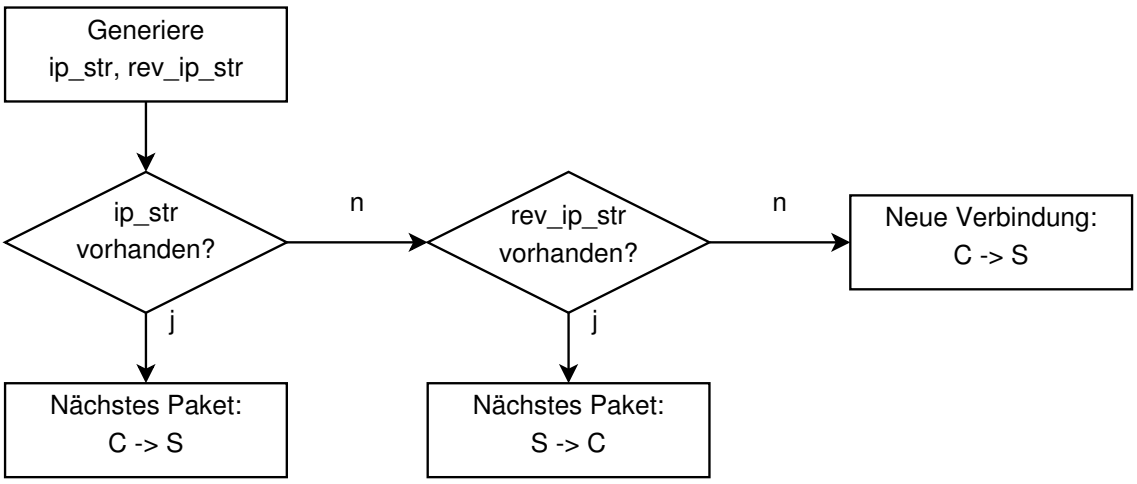


Abbildung 5.10: Erkennung der Übertragungsrichtung anhand der vorhandenen Hashtabelleneinträge.

vorhält (vgl. Kapitel F.3.7). Verbindungen werden während der weiteren Ausführung des Sicherheitssystems aus den Tabellen gelöscht, wenn sie als nicht bösartig klassifiziert wurden und abgeschlossen werden, oder wenn bei einer nicht korrekt abgebauten Verbindung für eine konfigurierbare Zeitspanne kein Datenpaket mehr gefunden wurde (Timeout-Mechanismus).

Wurden die statistischen Werte des Paketes in die zugehörige Datenstruktur eingetragen, erfolgt die weitere Verarbeitung wie nachfolgend aufgeführt:

- Die Clustererzeugung bereitet den Datenstrom zur weiteren Verarbeitung in Befehls-Antwort-Sequenzen auf und übergibt diese an die Module für die Ausbruchs- bzw. Innentätererkennung, also der Befehlsevaluation und der Nutzeridentifikation.
- Die Brute Force-Detektion findet direkt auf dem eingehenden Datenstrom statt, hierfür werden lediglich die Größen des Payloads der jeweiligen, *einzelnen* Verbindungen ausgewertet und analysiert.
- Die TLS-Angriffserkennung wird ebenfalls direkt auf dem eingehenden Datenstrom durchgeführt, hierfür werden die Paketgrößen des Payloads *verschiedener* Verbindungen miteinander korreliert.

Clustererzeugung

Wurde ein Netzpaket mittels der Datensonde aufgezeichnet und dessen statistische Information extrahiert und in das Datenfeld der jeweiligen Verbindung aufgenommen, ist die weitere Verarbeitung der Daten abhängig des anschließenden Detektionsbereiches, *Einbruchserkennung* oder *Ausbruchs-* bzw. *Innentätererkennung*. Der Prozess der Clustererzeugung gehört logisch zur Ausbruchserkennung, da die entsprechende Aufbereitung der Daten nicht für die Einbruchserkennung benötigt wird (vgl. Abbildung 5.9).

Aufgrund seiner architektonischen Zugehörigkeit zur Datenaufbereitung wird er jedoch hier vorgestellt.

Bei der Ausbruchs- und Innentätererkennung werden insbesondere die Befehle, die der jeweilige Nutzer eingibt, betrachtet und bewertet. Hierbei dient der Clustering-Prozess der Aufteilung des Datenstroms in Befehlssequenzen, um diese später gegen bekannte und bewertete Inhalte zu prüfen. Ein Cluster in einer verschlüsselten Sitzung ist wie folgt definiert:

Definition (Cluster)**.** *Ein Cluster beinhaltet sämtliche verschlüsselten Pakete einer Sitzung, die zu einem bestimmten Befehl gehören. Hierzu zählen die Pakete der Eingabe sowie die zur Antwort gehörigen Pakete des Servers, jedoch ohne ACK-, Echo- oder anderer administrativen Pakete.*

Abbildung 5.11 zeigt den Ablauf einer Sitzung und der prinzipiellen Erkennung der Clustergrenzen, die zur Trennung der Befehle rekonstruiert werden müssen. Da Befehle durch den Nutzer nur in sequentieller Abfolge eingegeben werden können, gehört eine Paketserie immer zu einem Befehl, während die Grenzen zwischen den Paketen verschiedener Befehle durch charakteristische Paketgrößen einerseits, insbesondere jedoch auch durch das Timing der übertragenen Pakete gefunden werden können. Zu Beginn des Sitzungsaufbaus einer verschlüsselten Verbindung durch den Client erfolgt die Authentifizierung des Nutzers, welche durch den Server initiiert wird. Im Gegensatz zur späteren Übertragung werden die Daten hierbei nicht durch einzelne Pakete versendet, sondern die gesamte Anmeldeinformation aus Nutzername und Passwort wird komplett übermittelt. Nach der Legitimierung des Nutzers steht die Sitzung zur Verfügung. Von diesem Zeitpunkt an werden alle Eingaben unmittelbar an den Server gesendet, um den Benutzer ein flüssiges und angenehmes Arbeiten zu ermöglichen. Die Eingaben in Form von Tastaturanschlägen sind anhand der charakteristischen Paketgröße, die ein einzelnes Zeichen mit der jeweiligen Verschlüsselung darstellt, zu erkennen[6]. Jedes vom Nutzer eingegebene und übermittelte Zeichen wird durch den Server mittels eines ACK-Paketes mit einem null Byte großem Payload bestätigt, weiterhin sendet dieser ein Paket der gleichen Größe des ursprünglichen Paketes an den Client, welches das Echo-Zeichen, also das beim Nutzer lokal angezeigte Zeichen, enthält. Bspw. ist bei der Nutzung von SSH und dem Verschlüsselungsalgorithmus Advanced Encryption Standard 128 bit Cipher Block Chaining (AES128-CBC) die charakteristische Paketgröße 48 Byte. Ist die Eingabe eines Kommandos abgeschlossen und wird dieses nach Betätigung der Eingabe-Taste auf dem Server ausgeführt, antwortet dieser mit einem oder mehreren Paketen, die typischerweise deutlich größer sind, als diejenigen welche im Rahmen der Befehlseingabe mittels Tastaturanschlägen versendet werden. Nach dem Empfang der Antwort des Servers ist eine Nutzereingabe abgeschlossen und es werden keine weiteren Pakete über das Netz übertragen, bis die nächste Taste betätigt wird. Da der Nutzer typischerweise eine kurze Pause nach der Eingabe und der zugehörigen Antwort des Servers einlegt,

[6] Ein entsprechendes Verhalten ist auch für eine Interaktion mit anderen Eingabegeräten wie bspw. der Maus beobachtbar; für die weitere Arbeit wird hier jedoch nur die Interaktion mit der Tastatur berücksichtigt.

um diese auszuwerten und den nächsten Schritt einzuleiten, kann hierdurch die Grenze zwischen zwei aufeinanderfolgenden Befehlen erkannt werden[7]. Dies kann zusätzlich durch die Paketgrößen verifiziert werden: Nach einer Abfolge konstanter Paketgrößen durch die Zeichen der Eingabe des Nutzers und den entsprechenden Echo-Paketen des Servers erfolgt eine deutlich größere Antwort des Servers und hierauf wiederum eine Serie konstanter Eingabezeichen durch den Nutzer und den zugehörigen Echo-Paketen.

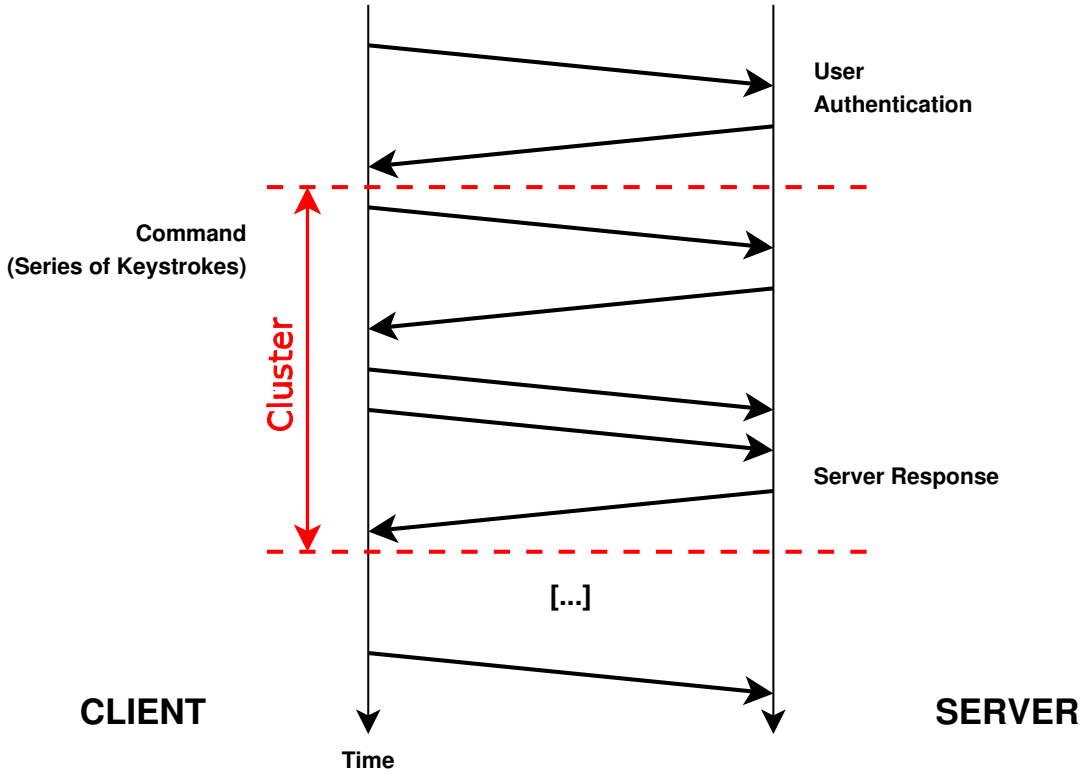


Abbildung 5.11: Erzeugung von Clustern aus den Paketen des Datenstroms.

Zusammengefasst werden folgende Bedingungen zur Detektion der Clustergrenzen geprüft:

- Es muss eine Serie von Paketen charakteristischer Payloadgröße, gesendet vom Client an den Server, vorliegen.
- Die zugehörige Serie von Echo- und administrativen Paketen ist abgeschlossen.
- Eine durch den Server gesendete Antwort überschreitet eine minimale Paketgröße (insbesondere größer als die charakteristische Paketgröße).

[7] Hier sei angemerkt, dass bereits die Pausen des Nutzers ein charakteristisches Bild ergeben können, welches für eine Identifizierung ausreichend sein kann: Bspw. demonstrierten Chen et al., dass eine Identifikation anhand der von einem Spieler eingelegten Pausen und Phasen ohne Aktivität bei Computerspielen möglich ist. Die Auswertung der Verteilung zwischen aktiven Phasen und Ruhezeiten zeigt individuelle Profile für verschiedene Spieler und erreicht eine Detektionsgenauigkeit von über 90 Prozent [91].

Tabelle 5.3: Auszug einer Serie aufgezeichneter Netzpakete einer verschlüsselten Verbindung.

Beobachtungszeit	n	Größe	Quelle	Beobachtungszeit	n	Größe	Quelle
1271362395.540057	2	48	C	1271362396.329887	13	0	C
1271362395.554030	3	48	S	1271362396.587828	14	48	C
1271362395.567991	4	0	C	1271362396.598803	15	48	S
1271362395.635884	5	48	C	1271362396.609901	16	0	C
1271362395.649194	6	48	S	1271362396.715838	17	48	C
1271362395.660328	7	0	C	1271362396.726849	18	48	S
1271362395.731922	8	48	C	1271362396.737989	19	0	C
1271362395.742829	9	48	S	1271362396.747659	20	816	S
1271362395.753772	10	0	C	1271362396.758546	21	0	C
1271362396.307897	11	48	C	1271362398.660010	22	48	C
1271362396.318838	12	48	S				

- Der Beginn eines neues Befehls wurde durch das erste Paket charakteristischer Größe nach einer minimalen Wartezeit zwischen Serverantwort und erneuter Eingabe detektiert (sog. *Cluster Break Time*).

Berücksichtigt werden muss allerdings, dass die verschlüsselte Verbindung so konfiguriert sein kann, dass sog. *keep-alive-* bzw. *heartbeat*-Pakete in regelmäßigen Intervallen übertragen werden. Diese Pakete dienen bspw. dem Aufrechterhalten der Verbindung und verhindern ein Timeout, können jedoch anhand ihrer charakteristischen Größen und insbesondere dem Timing leicht erkannt werden.

Der in Kapitel F.3.11 aufgeführte Algorithmus 2 setzt die Suche der Clustergrenzen um.

Tabelle 5.3 zeigt beispielhaft einen Ausschnitt eines aufgezeichneten Datenstromes; neben dem Unix-Zeitstempel in einer Genauigkeit von μ-Sekunden, der logischen Paketnummer der Verbindung und der Größe des Payloads ist die Quelle der jeweiligen Transaktionsrichtung angegeben.

Basierend auf den beschriebenen Eigenschaften leitet sich der zugehörige Cluster entsprechend der in Tabelle 5.4 dargestellten Werte ab. Hierbei entfallen entsprechend die 0 Byte großen ACK-Pakete sowie die jeweils 48 Bytes großen Echo-Pakete, das Ende des Cluster wird durch die Payloadgröße von Paket 20 und der deutlich höheren Zwischenankunftszeit des Client-Pakets 22 erkannt. Wichtig ist die Betrachtung der Kombination beider Faktoren insbesondere deshalb, da die Serverantwort auch aus einer Reihe von Paketen, auch kleinerer Größe, bestehen kann. Führt diese Kombination nicht zu einer eindeutigen Grenze, kann diese im weiteren Verlauf der Evaluation unter Zuhilfenahme weiterer Ergebnisse nochmals überprüft und ggf. angepasst werden.

Anhand der Ergebnisse aus Tabelle 5.4 lassen sich die Eigenschaften zur Detektion der Clustergrenzen nochmals nachvollziehen. Angemerkt sei, dass durch die Einbeziehung der Nutzerpause zwischen der Ausgabe eines Kommandos und der nachfolgenden Eingabe ein Cluster immer erst dann erkannt werden kann, wenn die Eingabe des näch-

Tabelle 5.4: Aus der in Tabelle 5.3 dargestellten Paketserie extrahierter Cluster.

Beobachtungszeit	n	Δ_t	Größe	Quelle
1271362395.540057	2	0.000000	48	C
1271362395.635884	5	0.095827	48	C
1271362395.731922	8	0.096038	48	C
1271362396.307897	11	0.575975	48	C
1271362396.587828	14	0.279931	48	C
1271362396.715838	17	0.128010	48	C
1271362396.747659	20	0.031821	816	S
1271362398.660010	*22*	*1,912351*	*48*	*C*

sten Clusters begonnen hat. Dies führt einerseits zwar zu einer kurzen Verzögerung der Evaluation eines Clusters, ist jedoch für eine korrekte Bestimmung der Clustergrenzen notwendig.

5.1.4 Module zur Einbruchserkennung

Die Einbruchserkennung im vorgestellten Sicherheitssystem entspricht Angriffen der klassischen Art, also solche, bei denen noch keine Zugriffsrechte zum Zielsystem vorhanden sind. Das System implementiert zwei entsprechende Detektionsmodule, welche auf verschiedene Arten die Daten verschlüsselter Verbindungen korrelieren: Zum einen die Daten *einer* Verbindung bzw. *eines* Nutzers mit sich selbst (*Intra*-Sitzungskorrelationen), zum anderen die Verbindungen *mehrerer* Nutzer untereinander (*Inter*-Sitzungskorrelationen).

Intra-Sitzungskorrelationen

Intra-Sitzungskorrelationen führen Berechnungen auf dem beobachteten Datenstrom eines jeden Nutzers getrennt durch. Angewendet kann das Verfahren insbesondere auf neue, sich im Aufbau befindliche verschlüsselte Sitzungen werden: Nach dem Verbindungsaufbau vom Client zum Server wird die Authentifizierung durchgeführt, die bspw. durch die Eingabe der Kombinationen eines Nutzernamens und Passwortes erfolgen kann. Ist die Anmeldung erfolgreich, kann mit der eigentlichen Sitzung begonnen werden, im Falle eines Fehlers durch eine ungültige Eingabe wird diese typischerweise noch mehrere Male abgefragt, bevor die Verbindung durch den Server zurückgesetzt wird. Dies eröffnet natürlich die Angriffsoption, dass ein Unbefugter durch das systematische Ausprobieren von Nutzer / Passwortkombinationen versucht, Zugang zu einem System zu erhalten.

Brute Force-Angriffe stellen ein häufig eingesetztes Element bei der Angriffsdurchführung dar, um Einstiegspunkte in Systeme zur Vorbereitung der nächsten Schritte

Tabelle 5.5: Top-5 Angriffe im Internet am 01.02.2011 [293]. *Angriffe* gibt die Anzahl der Angriffe pro Subnetz an.

Beschreibung	Angriffe	Anteil
POLICY Reserved IP Space Traffic - Bogon Nets 1	274.31	18.7 %
Microsoft SQL Server version buffer overflow attempt	257.58	17.6 %
POLICY Reserved IP Space Traffic - Bogon Nets 2	140.10	9.6 %
SSH brute-force login attempts	**133.27**	**9.1 %**
SCAN Sipvicious Scan	95.87	6.5 %

zu erhalten (vgl. Kapitel 2.3). Wie weit diese spezielle Angriffsform verbreitet ist (vgl. auch [346, 360]), zeigt bspw. ein Blick auf die Statistiken von ATLAS zu den häufigsten Angriffen im Internet: Am 01.02.2011 waren SSH Brute Force Login-Versuche mit 9.1 Prozent aller Angriffe auf Position vier der am häufigst detektierten Attacken (vgl. Tabelle 5.5).

Die hohe Beliebtheit dieser Angriffsart liegt auch darin begründet, dass zahlreiche Nutzer keine gute Passwortpolitik betreiben und zu einfache Passwörter wählen, Passwörter mehrfach einsetzen oder insbesondere in Systemen, die regelmäßige Passwortwechsel erzwingen, Passwörter auf Basis einfacher Bildungsgesetze verwenden. Zahlreiche Studien der letzten Jahre belegen (vgl. z.B. [339, 88, 146, 198]), dass sich diese Situation trotz der Informationen und Warnungen an die Nutzer kaum gebessert hat. Da auf Basis eines kompromittierten Nutzerkontos eine erhebliche Gefährdung für das System bzw. Netz ausgeht, ist eine frühzeitige und effiziente Detektion von Brute Force Angriffen erforderlich.

Für die Durchführung entsprechender Angriffe können Tools wie zum Beispiel `brutessh` oder `SSHater` eingesetzt werden, die auf Basis von Wörterbüchern oder von Bildungsregeln automatisch Nutzer-Passwort-Kombinationen durchprobieren. Hierbei können bspw. auch die Anzahl der parallel zu öffnenden Verbindungen, die zeitlichen Abstände der Versuche oder die zu nutzenden Kryptoalgorithmen bestimmt werden.

Eine Detektion eines solchen Angriffes lässt sich rein auf Basis der statistisch beobachtbaren Daten der Netzpakete realisieren, da die Verbindungen abhängig der erfolgreichen oder gescheiterten Authentifizierung verschiedene Charakteristika aufweisen.

Tabelle 5.6 gibt den Verlauf der Payloadgrößen sowohl bei mehrmaligen, fehlerhaften Zugangsversuchen, als auch bei einem erfolgreichen Zugang an (SSH-Sitzungen unter der Nutzung der Verschlüsselung AES128-CBC).

Auffällig ist insbesondere die Wiederholung der Paketserien gleicher Größe für die mehrfachen, fehlerhaften Loginversuche. Weiterhin ist die Paketserie des erfolgreichen Logins deutlich unterschiedlich zu den Paketgrößen der abgewiesenen Zugriffe.

Tabelle 5.6: Serien des Paketpayloads bei zurückgewiesenen Logins (links) und erfolgreichem Login (rechts).

Beobachtungszeit	n	Größe	Quelle	Beobachtungszeit	n	Größe	Quelle
1270060353.574124	5	39	S	1270060249.377474	5	39	S
1270060353.598878	7	39	C	1270060249.399566	7	39	C
1270060353.618841	9	792	C	1270060249.419443	9	792	C
1270060353.629592	10	784	S	1270060249.430136	10	784	S
1270060353.659618	13	24	C	1270060249.459943	13	24	C
1270060353.670451	14	152	S	1270060249.480212	15	152	S
1270060353.690932	16	144	C	1270060249.500324	17	144	C
1270060353.701767	17	720	S	1270060249.511116	18	720	S
1270060353.712832	18	16	C	1270060249.521987	19	16	C
1270060353.733700	20	48	C	1270060249.542695	21	48	C
1270060353.754584	22	48	S	1270060249.563423	23	48	S
1270060353.765548	23	64	C	1270060249.574313	24	64	C
1270060358.650097	25	64	S	1270060254.474934	26	64	S
1270060361.241796	27	144	C	1270060257.881910	28	144	C
1270060363.388655	29	64	S	1270060257.944587	30	32	S
1270060365.873666	31	144	C	1270060257.965065	32	128	C
1270060367.569295	33	64	S	1270060259.208001	34	48	S
1270060370.225513	35	144	C	1270060259.218886	35	448	C
1270060371.942316	37	64	S	1270060259.239482	37	112	S
				1270060259.250433	38	368	S
				1270060259.445127	40	80	S
				1270060275.393190	42	48	C
				1270060275.404175	43	176	S
				1270060275.415097	44	64	S
				1270060275.445443	47	32	C

Da mehrer Möglichkeiten bestehen, wie die Verbindung aufgebaut bzw. die Authentifizierung initiiert wird und von einer Adresse auch eine bestehende Verbindung vorliegen kann, während weitere Zugangsversuche stattfinden[8], müssen folgende Fälle unterschieden werden (vgl. Abbildung 5.12):

- Aufbau einer Verbindung und anschließende, erfolgreiche Authentifizierung.
- Aufbau einer Verbindung und anschließende, wiederholt fehlerhafte Authentifizierungsversuche; Abbau der Verbindung nach der maximalen Anzahl fehlerhafter Versuche und erneuter Verbindungsaufbau.
- Aufbau paralleler Verbindungen und jeweils Durchführung eines (oder von n) Authentifizierungsversuche(n), danach Beendigung der Verbindung durch den Client.
- Aufbau einer Verbindung und anschließende, erfolgreiche Authentifizierung sowohl parallel zur bestehenden Sitzung fehlerhafte Authentifizierungsversuche.
- Aufbau und Ausführung mehrerer, authentifizierter Sitzungen.

Die weiteren möglichen Kombinationen (bspw. erfolgreiche Verbindung, parallel Zugangsversuche mit maximaler Anzahl fehlerhafter Versuche und anschließendem, erneuten Verbindungsaufbau) subsumieren sich in der Evaluation durch das Sicherheitssystem in die vorgestellten Fälle.

Betrachtet man diese Fälle und berücksichtigt die auch mittels Tabelle 5.6 demonstrierte Diversifikation zwischen erfolgreichen und abgewiesenen Authentifizierungsversuchen, zeigt sich dass mittels einer Korrelation der jeweiligen Abschnitte innerhalb einer Verbindung bzw. zwischen verschiedenen Verbindungen einer Adresse (vgl. Abbildung 5.13), fehlerhafte Logins identifiziert werden können.

Dies kann zunutze gemacht werden, um einen effiziente Brute Force-Schutz für verschlüsselte Dienste eines ganzen Netzes wirksam und einfach mittels einer transparenten Brücke zu integrieren. Wird ein Brute Force-Angriff erkannt, kann dies je nach Konfiguration des Systems zu einer automatischen Sperrung der Angreifer-Adresse führen. Diese Funktionalität zum Schutz vor Brute Force-Angriffen wird als Modul in das Sicherheitssystem integriert, Abbildung 5.14 skizziert die Einbindung in das Gesamtsystem.

Die Funktion `got_packet()` stellt die Funktionalitäten der Datensonde (vgl. Kapitel 5.1.3) bereit; wie bereits oben beschrieben, muss eine Clusterung des Datenstromes nur für die Module der Ausbruchserkennung erfolgen. Für die hier vorliegenden Intra-Sitzungskorrelationen reicht es, da sich insbesondere die fehlerhaften Logins durch kurze Paketsequenzen ausdrücken, entsprechend kurze Abschnitte auszuwählen und mit den jeweiligen Verschiebungen über die Paketserie zu korrelieren. Werden hierbei hohe und regelmäßige Korrelationen gefunden, zeigt dies das Vorhandensein eines entsprechenden Angriffs. Die Funktion `correlation_type()` identifiziert hierbei, welche der vorgestellten Verbindungsarten vorliegt und somit, wie der Datenverkehr mit sich selbst korreliert

[8] Bspw. mehrere Nutzer hinter einer öffentlichen IP (NAT), oder durch einen bereits kompromittierten Rechner.

Abbildung 5.12: Eine verschlüsselte Sitzung beginnt zunächst mit dem Verbindungsaufbau und der Authentisierungsphase (orange). Ist diese erfolgreich durchgeführt und vom Server bestätigt (grün), kann die Verbindung genutzt werden. Scheitert die Authentisierung, kann diese ggf. wiederholt werden, oder der Server beendet die Verbindung (rot). Da mehrere Sitzungen bzw. Verbindungsaufbauten parallel erfolgen können, müssen die dargestellten Fälle unterschieden werden.

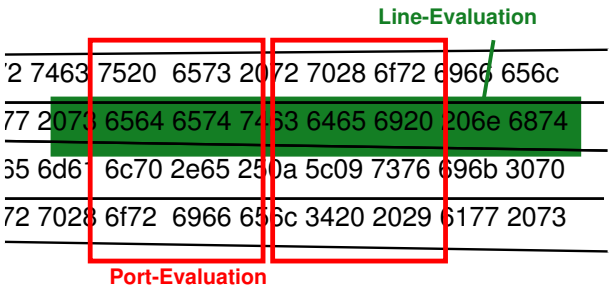


Abbildung 5.13: Verbindungs- und Portauswertung zur netzbasierten, schnellen Erkennung von Brute Force-Angriffen. Abhängig der erkannten Kommunikationssituation müssen die Abschnitte einer Verbindung miteinander korreliert werden (grüne Hervorhebung) oder die Daten verschiedener, paralleler Verbindungen (rote Hervorhebung).

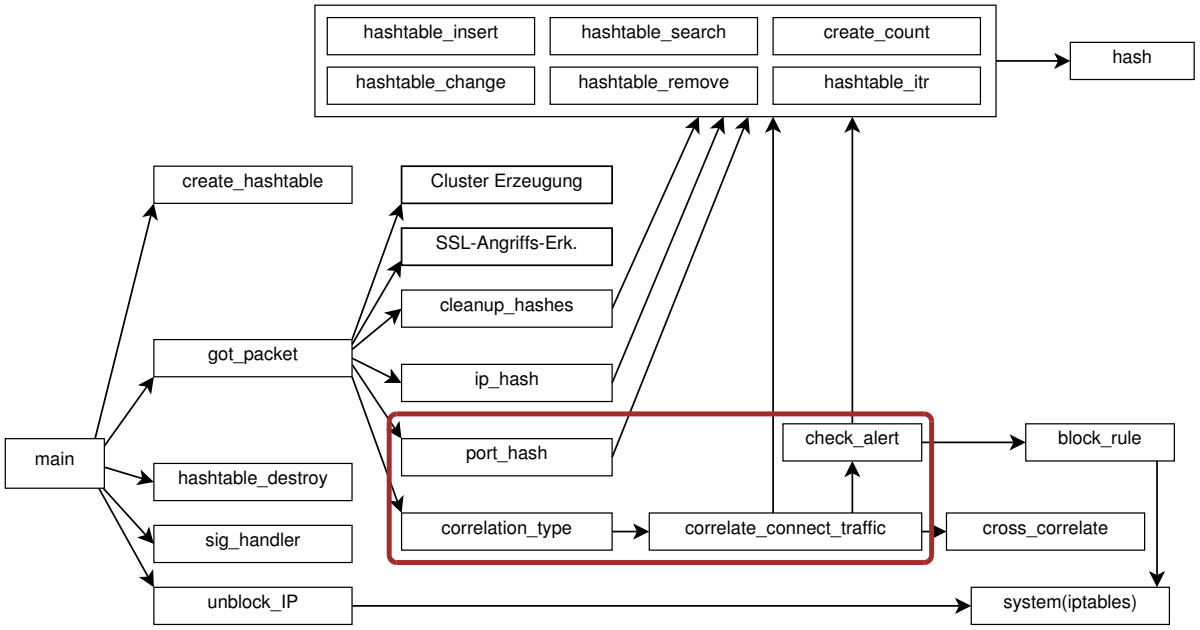


Abbildung 5.14: Aufbau des Moduls zur Brute Force-Erkennung und Integration in die Datensonde.

werden muss: Innerhalb einer Verbindung einer IP, zwischen mehreren Verbindungen einer IP oder zwischen und innerhalb der Verbindungen einer IP-Adresse. Mittels der Funktion `correlate_connect_traffic()` werden die eigentlichen Korrelationen der Paketserien durchgeführt und anschließend die Erfüllung der Alarmbedingungen geprüft (`check_alert()`). Hier kann bspw. auch angegeben werden, nach wieviel ausgelösten Alarmen die IP-Adresse des möglichen Angreifers gesperrt wird. Liegen die Bedingungen für eine Sperrung vor, wir dies entsprechend durch die Funktion `block_rule()` initiiert. Ebenfalls in Abbildung 5.14 ersichtlich ist die Integration der TLS-Angriffserkennung sowie der Clustererzeugung, die wiederum die Daten für die Module der Ausbruchserkennung bereitstellt.

Der Vorteil bei dieser Vorgehensweise liegt insbesondere auch darin, dass keine weiteren Informationen über den zugrunde liegenden Verschlüsselungsalgorithmus oder den genauen Aufbau des Sitzungsprotokolls benötigt werden, es kann somit eine im Netz integrierbare Detektion von allgemeinen Brute Force-Angriffen zum Schutz aller im Netz befindlichen Hosts umgesetzt werden. Die grundlegende Umsetzung ist in den Algorithmen 3 bis 5 in Kapitel F.3.11 ersichtlich.

Inter-Sitzungskorrelationen

Die Inter-Sitzungskorrelationen arbeiten ebenfalls direkt auf den durch die Datensonde gewonnenen Paketserien und somit auf den zugrundeliegenden, verschlüsselten Verbindungen. Im Gegensatz zu den *Intra*-Sitzungskorrelationen wird hier jedoch nicht die Kenntnis über bestimmte Abläufe des Protokolls wie bspw. den Authentifizierungsprozess betrachtet und für die Angriffsdetektion ausgehend von *einer* Adresse herangezogen, sondern die Korrelationen erfolgen zwischen mehreren Sitzungen verschiedener Adressen, also verschiedener Teilnehmer.

Der grundlegende Gedanke ist hierbei, dass bei angebotenen Diensten wie bspw. einem Webaufritt, die Anzahl der gutartigen Nutzer statistisch deutlich höher ist, als der Anteil der Angreifer, der lediglich einen geringen Prozentsatz aller Verbindungen darstellt. Dies kann wiederum zur Detektion von bösartigen Verbindungen genutzt werden, indem die Ähnlichkeit der Sitzungen, die mit dem Server aufgebaut werden, überprüft werden. Betrachtet man bspw. eine Internetseite mit einem Webshop, bietet dieser z.B. Funktionalitäten wie Produktseiten und -suche, Möglichkeiten zur Registrierung der Nutzer und Seiten für Bestellvorgänge, Gästebücher, etc. In typischen Nutzersitzungen werden hier z.B. Suchen nach Produkten durchgeführt, die gefundenen Artikel detailliert betrachtet, evtl. ein Bestellvorgang vorgenommen oder die Seite ohne Bestellung wieder verlassen. Wird jedoch ein Angriff auf die Webseite betrachtet, wie z.B. eine Structured Query Language (SQL)-Injection (vgl. Kapitel 6.2.2), wird diese sowohl von den Größen der Payloadserien, als auch vom Timing der Pakete typische Unterschiede zu normalen Nutzersitzungen aufweisen. So wird bspw. beim Anfordern einer Produktbeschreibung eine kurze Anfrage an den Webserver gesendet, welche durch eine erheblich größere Antwort beantwortet wird; im Falle des Versuchs einer SQL-Injection werden kleine Anfragen an den Server gesendet, die im Fehlerfalle typischerweise mit kleinen Paketgrößen beantwortet werden, im Erfolgsfalle jedoch eine umfangreiche Antwort generieren können. Bei

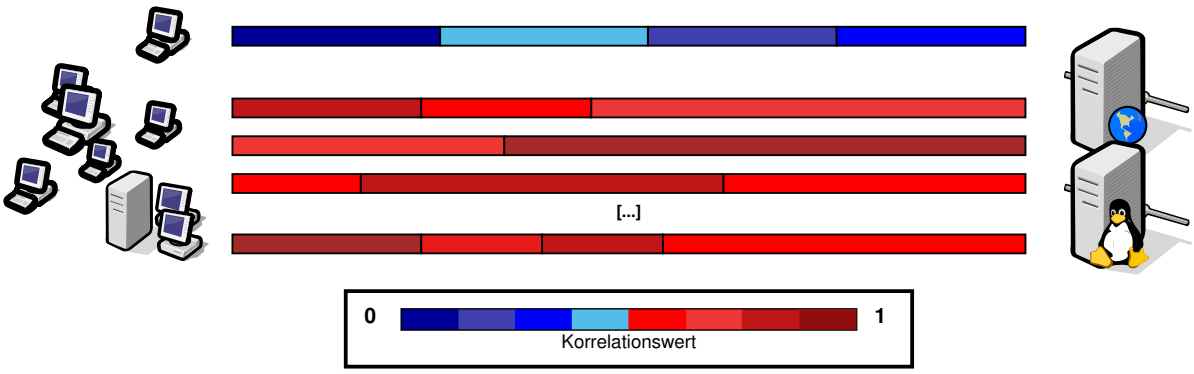


Abbildung 5.15: Bewertung von Verbindungen mittels Inter-Sitzungskorrelationen zur Anomaliendetektion im Datenverkehr. Während der Großteil der Nutzer gutartiger Natur ist, kommen mit einem geringen Anteil Angreifer vor. Die Verbindungen werden gegeneinander korreliert, somit sind die durchschnittlichen Korrelationswerte gutartiger Verbindungen hoch (rot dargestellt), während Verbindungen mit ungewöhnlichem Verhalten lediglich geringe durchschnittliche Werte erreichen (blau markiert).

der hier vorgeschlagenen Inter-Sitzungskorrelation wird jedoch im Gegensatz zu anderen wissenschaftlichen Arbeiten in diesem Gebiet (vgl. Kapitel 4.6.3) kein Profil des Servers benötigt, ebenso ist die Kenntnis des Aussehens der typischen, korrekten Verkehrscharakteristika nicht erforderlich: Das als gutartig zu bewertende Verhalten ergibt sich automatisch aus der statistisch größeren Gruppe der legitimen Nutzer[9]. Eine Anwendung des Systems ist in jeder Art von Umgebung möglich, wo ein entsprechender Datenverkehr bzw. Nutzerzahl vorliegt, unabhängig von der Art des Dienstes respektive der Anwendung oder der Verschlüsselung. Existierenden Arbeiten, welche statistische Evaluationen zur Detektion von Anomalien heranziehen, benötigen Lernphasen oder Kenntnis über das Kommunikationsverhalten der zu schützenden Systeme. Im Gegensatz dazu nutzt das hier vorgestellte Verfahren implizites Wissen aus, welches durch die Korrelation der verschiedenen, überwiegend gutartigen Verbindungen Anomalien in Echtzeit erkennen kann, ohne hierfür Wissen über das Aussehen einer gutartigen Verbindung zu benötigen.

Abbildung 5.15 zeigt schematisch die Funktionsweise der Inter-Sitzungskorrelation. Der Großteil der Nutzer, welcher mit den Diensten eines Servers im Internet arbeitet, ist gutartiger Natur. Werden die statistischen, charakteristischen Größen dieser Verbindungen untereinander korreliert und die Mittelwerte der Korrelationsergebnisse einer Verbindung mit den jeweils anderen, beobachtbaren Verbindungen erzeugt, ergeben diese hohe Korrelationswerte. Wird dahingegen eine bösartige Verbindung betrachtet und mit weiteren, typischerweise gutartigen Verbindungen zum Server korreliert, ergeben sich durch die Abweichungen der statistischen, charakteristischen Verbindungsparameter geringe Korrelationswerte. Wird ein Schwellwert der durchschnittlich minimalen Korrelationshöhe unterschritten, kann entsprechend ein Alarm generiert werden.

[9] Die Auswirkungen eines steigenden Anteils bösartiger Teilnehmer auf die Auswertungen des Sicherheitssystems werden im Kapitel 6.2.2 ebenfalls betrachtet und evaluiert.

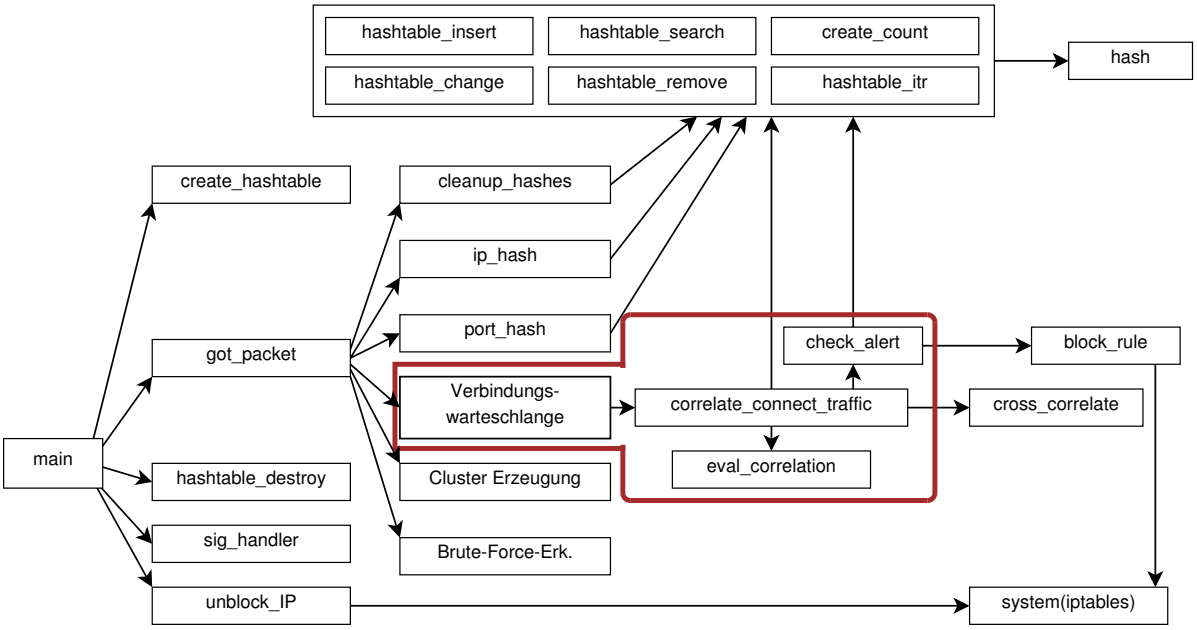


Abbildung 5.16: Aufbau des Moduls zur Anomaliendetektion auf Basis des Normalverhaltens von Nutzern und Integration in die Datensonde.

Ein entsprechendes Modul wird wie in Abbildung 5.16 veranschaulicht in das Gesamtsystem integriert.

Nachdem ein neues Netzpaket durch die Funktion `got_packet()` aufgezeichnet und die Werte mittels der Funktionen der Hash-Erzeugung in das System aufgenommen wurden (vgl. Kapitel 5.1.3), werden die Verbindungen zur Inter-Sitzungskorrelation an die `Verbindungswarteschlange` weitergegeben. Diese hat die Aufgabe, noch nicht betrachtete und bewertete Funktionen so lange vorzuhalten, bis genügend Korrelationspartner[10] vorhanden sind. Dieser Schritt ist insbesondere zum Start des Systems notwendig, wenn noch keine Verbindungen beobachtet wurden. Sind ausreichend Verbindungen vorhanden, erfolgt durch die Funktion `correlate_connect_traffic()` die abschnittsweise Kreuzkorrelation zwischen den statistischen Verbindungsdaten. Hierfür können alle Korrelationskombinationen berechnet werden, also jede neue Verbindung mit jeder bekannten Verbindung, oder es wird eine konfigurierbare Anzahl von Verbindungen zufällig aus den bereits vorhandenen ausgewählt, um sie mit einer neuen Verbindung zu korrelieren. Die weitere Auswertung, ob anhand der ermittelten Korrelationswerte eine Anomalie vorliegt und somit ein Alarm ausgelöst werden muss, erfolgt wiederum mittels der Funktion `check_alert()`. Kommt es zur Alarmierung, kann bei entsprechender Konfiguration einer Sperrung der verdächtigen IP-Adresse über die Funktion `block_rule()` erfolgen. Die Funktion `eval_correlation()` dient der weiteren statistischen Analyse der ermittelten Daten und wird für die Funktionalität des Moduls nicht benötigt. Mittels dieser Funktion werden bspw. die Matrizen für die Systemevaluation erzeugt und bereitgestellt.

[10] Die Bestimmung der optimalen Anzahl von Korrelationspartnern wird in Kapitel 6.2.2 behandelt.

Da die Detektion weder Parameter bzgl. der Server respektive Dienste, welche überwacht werden, benötigt, noch die gutartigen Charakteristika der Verbindungen bekannt sein müssen, sondern direkt und ohne Lernphase die aktiven Verbindungen ohne eine Entschlüsselung analysiert werden können, ist das Modul universell einsetzbar.

5.1.5 Module zur Ausbruchs- und Innentätererkennung

Im Gegensatz zur klassischen Einbruchserkennung sind die nachfolgend vorgestellten Systemteile darauf spezialisiert, Ausbrüche aus einer Umgebung sowie Handlungen von Innentätern festzustellen. Die besondere Schwierigkeit in diesem Kontext liegt darin, dass die Akteure hier bereits über Zugangsrechte zu einem System verfügen, da sie entweder bspw. als Mitarbeiter einen autorisierten Zugang haben oder bereits ein kompromittiertes Konto vorliegt (bspw. nach einem erfolgreichen R2L-Angriff), das von einem Angreifer für seine weiteren Schritte genutzt werden kann. Auch ein Bot, der ohne Wissen des Eigentümers eines Rechners auf diesem installiert ist, fällt bzgl. seines Verhaltens mit in diese Kategorie. Der legale bzw. beschaffte Systemzugang erschwert eine Detektion erheblich, zusätzlich ist das Agieren aus einem Netz heraus ins Internet typischerweise einfacher, als umgekehrt (vgl. Kapitel 2.3). Die Behandlung dieser Art von bösartigem Verhalten wird im weiteren Verlauf als *Ausbruchserkennung* zusammengefasst.

Um trotzdem eine Detektion im Bereich der Ausbruchserkennung zu ermöglichen, implementiert das System hierfür zwei Detektionsmodule, eines zur Erkennung und Evaluation eingegebener Befehle und eines für die Identifizierung des Nutzers einer verschlüsselten Verbindung.

Erkennung der Nutzerstrategie

Um bösartiges Verhalten bei legitimen Nutzern erkennen zu können ist es erforderlich, die Intention hinter einer Serie von Befehlen festzustellen. Detektionsmerkmale wie bspw. das Erkennen einer nicht-autorisierten Verbindung wie im Rahmen der Einbruchserkennung ist hier nur unzureichend möglich, da die Legitimierung des Nutzer bereits einen Zugang zum System, dem Umgang mit Daten, etc. ermöglicht. Vielmehr ist es daher erforderlich, Abweichungen vom normalen Verhaltensprofil des Nutzers zu erkennen, oder die Intention, die hinter einer Serie von durchgeführten Aktionen liegt, rechtzeitig zu erkennen. Im Kontext von verschlüsselten Umgebungen kommt hier jedoch zusätzlich die Herausforderung hinzu, anhand der eingeschränkten, statistisch beobachtbaren Daten auf bspw. die zugrunde liegenden Befehle schließen zu müssen.

Ausgangspunkt für eine entsprechende Analyse der Nutzerintention ist die Auswertung von Angriffsbäumen. Schneier beschreibt Angriffsbäume als eine Möglichkeit, systematisch die Sicherheit eines Systems zu beschreiben, Entscheidungen zu treffen und Auswirkungen, wie sich ein Angriff auf die Sicherheit eines Systems auswirken kann, zu untersuchen [338]. Ein Angriffsbaum beschreibt hier Angriffe und Gegenmaßnahmen in Form einer Baumstruktur; während die Wurzel das Ziel des Angriffs beschreibt, stellen die Blätter einzelne Angriffe dar. Die weiteren Knoten des Baumes stellen Zwischenziele dar, welche zur Durchführung des jeweiligen Angriffs erfüllt werden müssen.

In der Literatur wurden Angriffsbäume verschiedentlich zur Angriffsbeschreibung genutzt, bspw. von Byres et al., um Schwachstellen in SCADA-Netzen zu bewerten (vgl. [84]) oder zur Modellierung von Angriffen [286]. Nachfolgend werden sie genutzt, um die möglichen Vorgehensweisen zur Durchführung einzelner Angriffsschritte (vgl. Kapitel 2.3.3) zu eruieren und auf dieser Basis die hierbei genutzten Befehle abzuleiten. Abbildung 5.17 zeigt einen beispielhaften Angriffsbaum.

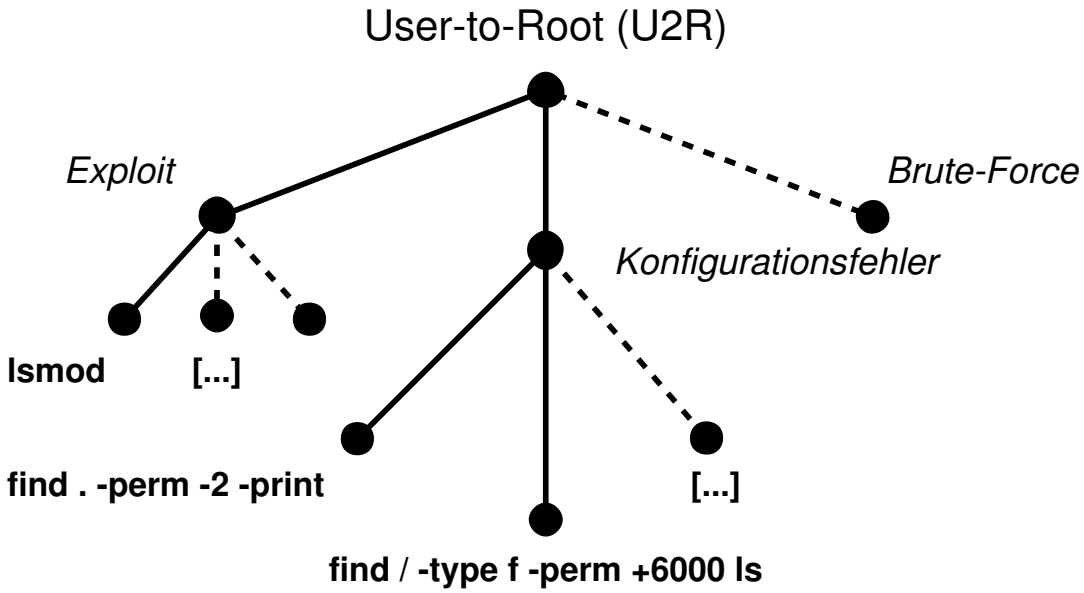


Abbildung 5.17: Beispielhafter Angriffsbaum.

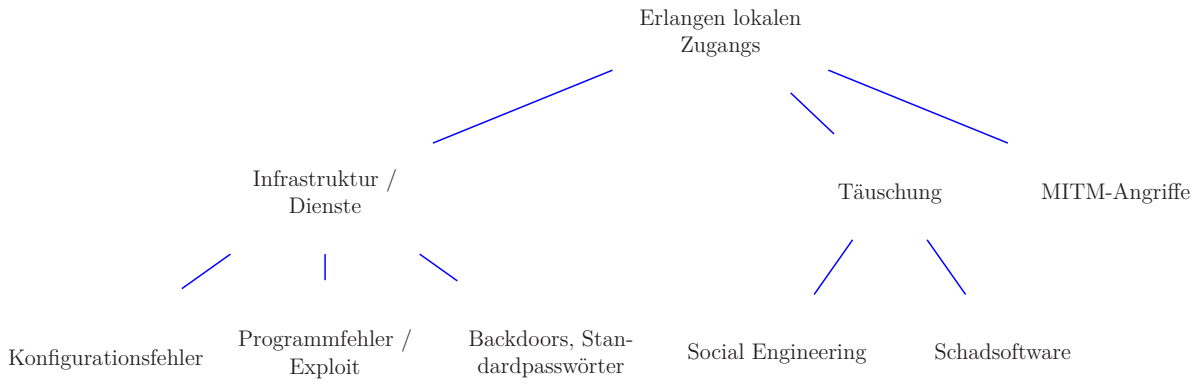
Hier werden Möglichkeiten aufgespannt, welche genutzt werden können, um einen U2R-Angriff durchzuführen, also den Versuch einer Erhöhung der Privilegien des Nutzers. Eine typische Vorgehensweise ist hier die Ausnutzung von Schwachstellen eines installierten Programms durch die Nutzung eines geeigneten Exploits. Um ein hierfür anfälliges Programm zu finden, können bspw. die aktiven Kernelmodule ausgegeben werden, um deren Versionen zu überprüfen und mit diesen Informationen nach der Verfügbarkeit von Exploits zu suchen. `lsmod` wäre in diesem Kontext daher ein nutzbarer Befehl, um die aktiven Kernelmodule auszugeben. Eine weitere Möglichkeit (angedeutet durch die gestrichelten Linien in Abbildung 5.17), ein entsprechend anfälliges Programm zu finden, könnte die Überprüfung der installierten Dienste sein, die auf dem System laufen. Hierzu können bspw. die Runlevel-Einträge des Systems geprüft und anschließend die zugehörigen Binärdateien analysiert werden. Neben dem Versuch einen Exploit anzuwenden, können auch allgemeine Konfigurationsfehler des Systems ausreichend sein, um eine Rechteeskalation durchzuführen. Ein Beispiel hierfür ist die Suche nach Dateien, die jeder Nutzer beschreiben kann (*world writable*). Diese können für das aktuelle Verzeichnis mit dem Befehl `find . -perm -2 -print` ausgegeben werden. Befindet sich darunter bspw. eine Konfigurationsdatei eines Dienstes, etc., kann diese beliebig durch den Angreifer geändert werden. Ein weiteres Beispiel ist die Suche nach

gesetzten SUID-Bits. Ist dieses Bit für eine Datei gesetzt, hat diese bei der Ausführung die Zugriffsrechte auf das System wie der Besitzer der Datei, nicht der Ersteller des Prozesses (der ausführende Nutzer). Dies bedeutet, dass wenn eine Datei ein `SUID root`-Berechtigung hat und durch einen normalen Nutzer ausgeführt wird, der Prozess bei der Ausführung `root`-Rechte erhält und somit umfassende Zugriffsmöglichkeiten auf das System. Dies stellt entsprechend eine besondere Gefährdung dar, zahlreiche Exploits nutzen Programme mit gesetztem SUID-Bit aus. Um alle Dateien im System, die das entsprechende Bit gesetzt haben aufzulisten, kann der Angreifer den Befehl `find / -type f -perm +6000` nutzen. Eine weitere Kategorie von Angriffen zur Erhöhung der Privilegien können Brute Force-Angriffe sein. Kann der Angreifer bspw. eine Passwortdatei des entsprechenden Systems kopieren, ist danach die Durchführung von Brute Force oder Hashtabellen-Angriffen möglich.

Wie hier bereits ersichtlich ist, entstehen zahlreiche Möglichkeiten, einen jeweiligen Angriffsschritt in der Praxis umzusetzen. Im weiteren Verlauf werden daher die Angriffsbäume für die Detektion relevanten Angriffsschritte zunächst in von konkreten Befehlen abstrahierter Form vorgestellt. Die Knoten der Bäume, die den Zwischenzielen entsprechen, werden anschließend mit den Befehls- und Verzeichnismengen gefüllt, durch welche die jeweiligen Zustände erreicht werden können.

Angriffsbaum Remote-to-Local

Die im Rahmen der Angriffsanalyse identifizierten Möglichkeiten der Erlangung eines lokalen Zugriffs sind im nachfolgenden Angriffsbaum abgebildet.

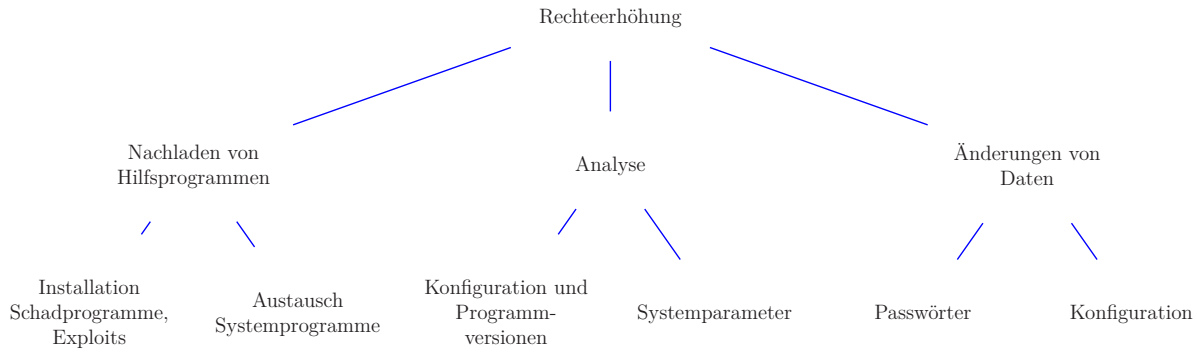


Die Auswahl des hierbei genutzten Verfahrens wird maßgeblich durch die Ergebnisse der Phasen der Analyse der Zielumgebung sowie der Identifizierung von Schwachstellen bestimmt. Als Möglichkeiten gibt es hier insbesondere Ansatzpunkte über die vorhandene Infrastruktur bzw. die im System zur Verfügung gestellten Dienste, Täuschungsversuche sowie MITM-Angriffe. Besonders kritisch erweist sich der Angriffsweg *Täuschung* für die Detektion, da die zugehörigen Social Engineering Verfahren nicht notwendigerweise

über das Netz ausgeführt werden. Bspw. kann die Zielperson des Angriffs telefonisch dazu gebracht werden, eine Aktion im Sinne des Angreifers durchzuführen, welche diesem einen anschließenden Zugang ermöglicht. Auch die Zusendung einer speziell angepassten Mail mit einem darin enthaltenen Schadprogramm kann genutzt werden, Zugang zum System zu erhalten. In diesen Fällen werden somit die ansonsten notwendigen Angriffsschritte *Analyse der Zielumgebung*, *Identifizieren von Schwachstellen* sowie *Erlangen von Fernzugriff* übersprungen (vgl. Kapitel 2.3.3). Somit ist eine Detektion lediglich in den höheren Angriffsschritten möglich, insbesondere bei der Rechteerhöhung oder dem Ausschleusen von Daten.

Angriffsbaum User-to-Root

Die unter den Angriffsbaum U2R subsumierten Techniken sind von besonderer Bedeutung, da diese den Kernpunkt der Detektion des Sicherheitssystems im Bereich der Ausbruchserkennung darstellen. Hier ist bereits der Zugriff vorhanden, entweder durch einen vorherigen, erfolgreichen R2L-Angriff, oder durch das Vorhandensein der entsprechenden Berechtigungen durch einen Innentäter. Gemäß der Analyse der Angriffsschritte in Kapitel 2.3.3 baut sich der Angriffsbaum wie folgt auf:



Werden die Angriffsbäume durch die entsprechenden Befehle komplettiert, ist erkennbar, dass ein Angriffsziel jeweils über verschiedene Zwischenziele erreicht werden kann, welche wiederum durch unterschiedliche Befehle erlangt werden können. Insbesondere ist hierbei auch die Reihenfolge der genutzten Befehle *nicht* zwingend vorgegeben. Für die Angriffsdetektion gilt daher, dass eine Gefährdung immer dann vorliegen kann, wenn ein Zwischenziel durch die eingegebenen Kommandos erreicht werden kann, unabhängig der genauen Reihenfolge und Vollständigkeit. Im Sinne eines traditionellen IDS würde die Alarmierung dann erfolgen, wenn der Angriff erkannt wurde. Hier können zwar auch noch Gegenmaßnahmen wie bspw. die Sperrung eines Ports oder der Abbau einer Verbindung ergriffen werden, jedoch ist der Angriff hier immer bereits im Gange. Die Besonderheit bei der Detektion der Angriffs-Zwischenziele liegt aber darin, dass bereits hier Schritte gegen den laufenden, aber noch nicht vollendeten Angriff initiiert werden

können. Umso früher das Detektionsstadium ist, umso höher ist jedoch auch die Wahrscheinlichkeit eines Fehlalarms, dementsprechend muss der früheste Reaktionszeitpunkt und die Art der zu ergreifenden Maßnahmen konfigurierbar sein.

Abbildung 5.18 zeigt den Ablauf der Auswertung einzelner Befehle mit der Identifikation der dahinter liegenden Nutzerstrategie. Im ersten Schritt werden die möglichen, in einer verschlüsselten Sitzung eingegebenen Befehle auf Basis der gefundenen Cluster analysiert. Das Ergebnis sind identifizierte Paketserien bzw. Korrelationswerte, je nach angewandter Auswertung. Zu den Angriffsbäumen zugehörig sind Mengen von Befehlen hinterlegt, die zur Erreichung des jeweiligen Teilaspekts herangezogen werden können. Die erkannten Befehle werden gegen diese Gruppen geprüft, umso mehr der möglichen Befehle auch erkannt wurden, umso höher ist die Wahrscheinlichkeit der Durchführung des entsprechenden Teilziels. In diesem Schritt kann auch eine Anpassung der Auswahl von Befehlen stattfinden: Wurden für einen Cluster bspw. drei mögliche Befehle mit ähnlichen, hohen Korrelationswerten festgestellt, kann deren Reihung im Sinne der zum Angriffsschritt gehörigen Befehle angepasst werden. Sind die Aspekte eines Teilziels erfüllt, kann eine Alarmierung erfolgen.

Modul zur Befehls-Evaluation*

Ist es möglich, die über eine verschlüsselte Verbindung übertragenen Befehle zu erkennen, kann anhand der Angriffsbäume eine bösartige Intention identifiziert werden: Eröffnen bspw. die eingebenen Befehle, ein Zwischenziel zur Durchführung einer Rechteerhöhung zu erreichen und wird dies erkannt, kann die entsprechende Verbindung abgebaut und die Ausgangsadresse gesperrt werden. Da eine Kommandoeingabe rein eine Abfolge von Paketen einer einheitlichen, charakteristischen Größe darstellt, reicht dies nicht für eine Evaluation des Kommandos aus. Die Hinzunahme der Serverantwort ist hier essentiell, da diese durch ihre Paketgrößen, der Anzahl der Pakete und somit der Antwortgrösse eine einfachere Ermittlung eines Fingerabdrucks ermöglicht.

Nachfolgend wird daher die Konzeptionierung eines Moduls zur Evaluation von Befehlen in verschlüsselten Umgebungen vorgestellt. Wie bereits oben beschrieben, können hierfür entweder die konkreten Paketserien, oder entsprechende Korrelationen betrachtet werden. Hierfür sind bzgl. der Auswertung der vorhandenen Daten verschiedene Vorgehensweisen denkbar:

- Clustergrenzen
  - Feste Client- und Servergrenzen
  - Nur feste Clientgrenzen
  - Freie Clustergrenzen
- Berücksichtigte Pakete

*Dieser Abschnitt enthält eine Zusammenfassung von Teilen des Artikels „Command Evaluation in Encrypted Remote Sessions“, Proceedings of the 4th International Conference on Network and System Security (NSS), IEEE, 2010.

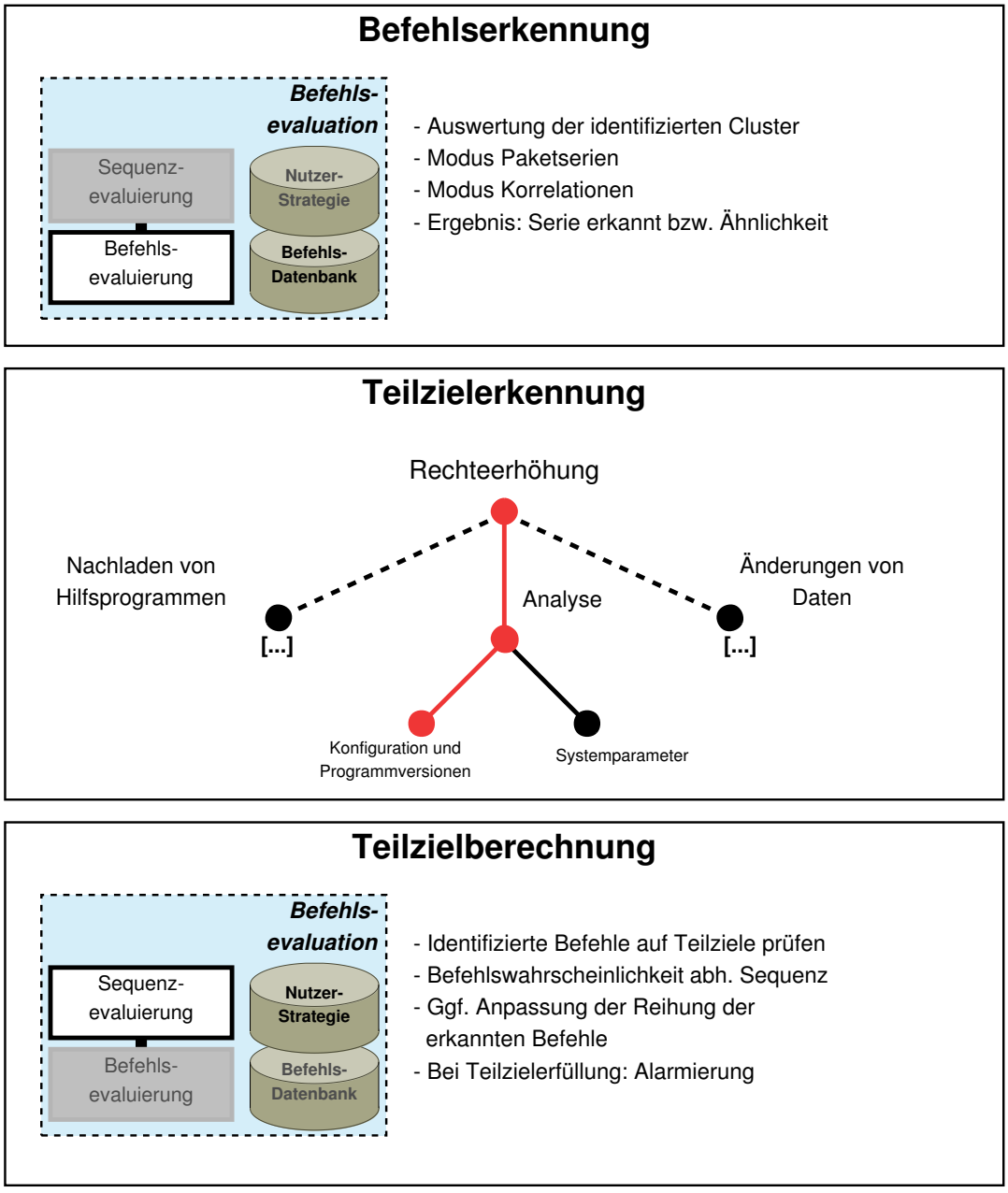


Abbildung 5.18: Teilschritte der Befehlsevaluation. Nach der Identifizierung einzelner Befehle werden diese mit den in den jeweiligen Teilschritten möglichen Befehlen verglichen, ist das Erreichen eines entsprechenden Teilziels möglich, erfolgt eine Alarmierung.

Tabelle 5.7: Übersicht der von Angreifern nach einer Systemkompromittierung genutzten Befehle.

	Befehle					
Software	`w`	`id`	`whoami`	`last`	`ps`	`cat /etc/*`
	`history`	`php -v`	`cat .bash_history`			
Installation	`tar`	`unzip`	`mv`	`rm`	`cp`	`chmod`
	`mkdir`					
Download	`wget`	`ftp`	`curl`	`lwp-download`		
Ausführung	`cround`	`httpd`	`kjournald`			
Passwort	`passwd`					
Hardware	`uptime`	`ifconfig`	`uname`	`cat /proc/cpuinfo`		
Konfiguration	`export`	`PATH=`	`kill`	`nano`	`pico`	`vi`
	`vim`	`sshd`	`useradd`	`userdel`		

- Client und Server
- Nur Server

Andere Kombinationen, bspw. *„Nur Client“* für die zu berücksichtigenden Pakete, werden nicht betrachtet, da diese leichter manipulierbar sind: Der Datenstrom vom Client zum Server kann maßgeblich durch den Angreifer beeinflusst werden (vgl. Kapitel 5.2.2), wohingegen die vom Server gesendeten Daten besser vor einer Manipulation geschützt sind.

Dies ist insbesondere auch zur Verbesserung der Detektionsergebnisse von Bedeutung, da zahlreiche Befehle bspw. auf Basis des aktuellen Systemverhaltens leicht veränderte Antwortpaketserien generieren können. Beispiele wären hier die Auflistung der aktiven Nutzer oder die Ausgabe der geladenen Kernelmodule.

Zunächst wird die Nutzung von kompletten Paketserien mittels direktem Vergleich betrachtet. Hierfür wird die Gruppe von Befehlen, welche im Sinne der Angriffsbäume im Rahmen der Durchführung böswilliger Aktivitäten charakteristisch sind, bestimmt. Eine Identifikation dieser im Rahmen der Zwischenziele möglichen bzw. erforderlichen Befehle ist schon daher vonnöten, da eine komplette Umsetzung *aller* Befehle im Rahmen der Auswertung des Sicherheitssystems einen enormen Aufwand darstellen würde: In diesem Fall müssen alle Befehle eines zu schützenden Betriebssystems erfasst und für alle auftretenden Verschlüsselungsvarianten analysiert werden. Entsprechend ist es notwendig, den für Angreifer relevanten Befehlssatz möglichst stark zu reduzieren; dieser wird auf Basis der eruierten Zwischenziele festgelegt. Die Selektion der im weiteren Verlauf zu betrachtenden Befehle erfolgt am Beispiel von Linux, weiterhin wird hierbei die Arbeit von Ramsbrock et al. herangezogen, der im Rahmen einer Untersuchung über das Verhalten von Angreifern die von diesen genutzen Befehle untersucht hat [320]. Tabelle 5.7 zeigt die Befehle nach Ramsbrock, aufgeteilt in sieben Kategorien.

Neben der Bedeutung dieser Befehle als Indizien für bösartige Aktivitäten, da diese im normalen Arbeitsbetrieb eines Nutzer typischerweise nur selten oder überhaupt

Tabelle 5.8: Entwicklung der Größe des verschlüsselten Payloads bei wachsender Originalnachricht (Nutzung von AES128-CBC und der bash, alle Größen in Bytes).

Originalausgabe	Payload-Größe
1	80
5	96
21	112
37	128
[...]	[...]

nicht vorkommen werden, sind sie auch daher relevant, da sie oftmals auch charakteristische Ausgaben erzeugen, welche für eine Erkennung herangezogen werden können. Dies sind insbesondere Befehle mit im Sinne von Netzpaketen konstanten Ausgabegrößen wie bspw. `uptime` oder `whoami`: Während die Abfrage der `uptime` zunächst sieben Pakete der charakteristischen Eingabegröße von Seiten des Clients erzeugt, generiert das Kommando eine Ausgabe von 12 Wörtern bzw. ca. 70 Bytes, abhängig von Faktoren wie der genauen Uptime, der Anzahl eingeloggter Nutzer, etc. Wird diese Antwort mit AES128-CBC verschlüsselt übertragen, ergibt dies einen Payload der Größe von 160 Bytes, der sich auch bei einer geringfügigen Änderung der Größe der Antwort (im Sinne der Anzahl der Zeichen) nicht ändert. Ruft man den Befehl `whoami` auf, wird der gerade aktive, mit der effektiven Nutzer-ID assoziierte Nutzername ausgegeben. Dies ergibt bei gleicher Verschlüsselung typischerweise eine Antwort der Payloadlänge von 96 Bytes, unabhängig der genauen Länge des Nutzernamens. Wächst die Länge der Ausgabe an, steigt in berechenbaren Schritten die Größe des verschlüsselten Payloads mit: Während eine einfache Textausgabe von bspw. 3 Bytes einen Payload von 80 Byte ergibt, beläuft dieser sich für einen Umfang von 10 Bytes Text auf 96 Bytes Payload. Dies liegt in der Nutzung der Blockgrößen begründet, die bei vorliegender AES-Verschlüsselung 16 Bytes betragen; demnach wird der nicht durch Information belegte Teil eines Block aufgefüllt, wird die Blockgröße überschritten, wird ein neuer Block erzeugt und übertragen (vgl. Tabelle 5.8).

Hier geht wie bereits oben erwähnt einerseits Information über die genaue Ausgabelänge des Originaltextes verloren, jedoch kann die Rekonstruktion von Befehlen andererseits einfacher werden[11], indem weniger mögliche Fälle kodiert werden müssen. Eine weitere, wichtige Beobachtung in diesem Kontext ist, dass die Größe der zurückgelieferten Antwort auch von der genutzten Shell abhängt: Hat ein Programm eine Textausgabe von bspw. 12 Bytes, entspricht dies nach Verschlüsselung einer Payloadgröße von 96 Bytes bei der Nutzung der `bash`, jedoch lediglich 64 Bytes bei `sh`. Auch Befehle wie `ifconfig` sind von besonderem Interesse, da deren Ausgaben typischerweise ebenfalls Ausgaben

[11] Inherent wird die Rekonstruktion durch den Informationsverlust jedoch ungenauer: Unterscheiden sich bspw. zwei im Sinne der einzugebenden Zeichen gleich lange Befehle nur durch zwei Zeichen in der Ausgabe, die jeweils mittels eines Blockes übertragen wird, sind die Befehle hierdurch nicht mehr unterscheidbar.

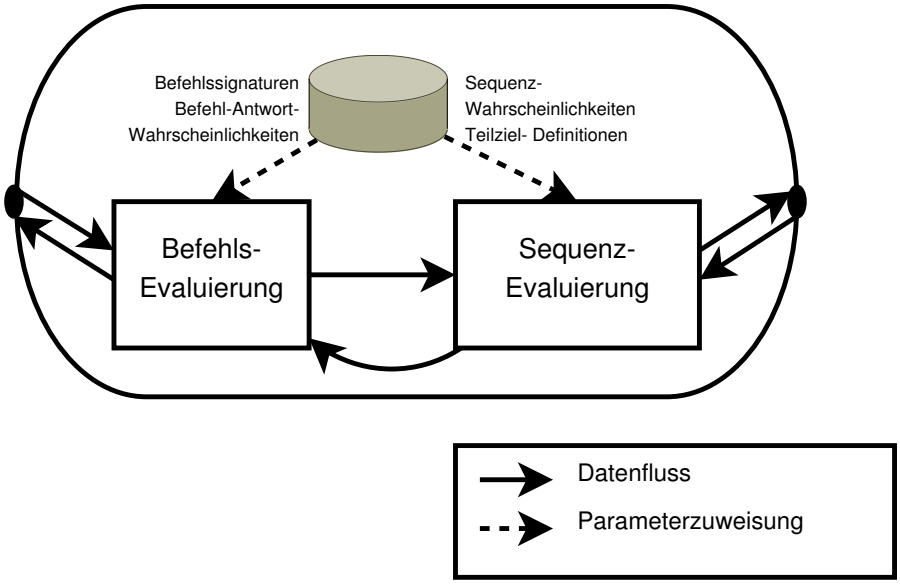


Abbildung 5.19: Teilschritte zur Befehlsevaluation im Schema des Sicherheitssystems S2E2. Von besonderer Bedeutung ist die Rückkopplung der Sequenz- auf die Befehlsevaluation. Diese bietet die Möglichkeit, die Auswahl einzelner Befehle der Befehlsevaluation anhand einer kompletten Befehlsserie zu korrigieren.

konstanter Größe (abhängig der Anzahl der Netzinterfaces) erzeugt.

Anhand einer SSH-Verbindung wird nun beispielhaft untersucht, wie sich die gewonnenen statistischen Daten nutzen lassen, um illegale Aktivitäten und Versuche einer Eskalation von Privilegien festzustellen.

Tabelle 5.9 zeigt beispielhafte Kombinationen der Muster von Payloadgrößen bei der Eingabe von Befehlen bzw. den zugehörigen Antworten des Servers. Da die Eingaben des Nutzers immer als einzelne Zeichen übertragen werden, haben die zugehörigen Pakete gleiche Größen; hier kann nur durch die Anzahl von Paketen und somit Tastaturanschlägen auf die Länge des jeweiligen Kommandos geschlossen werden, jedoch muss auch berücksichtigt werden, dass es durch Tippfehler, etc. zu einer Abweichung zwischen den tatsächlich übertragenen Paketen und der Länge des ausgeführten Befehls kommen kann. Somit ist insbesondere die Evaluation der vom Server zurückgegebenen Paketserien von entscheidender Bedeutung, da hieraus mehr Informationen ableitbar sind.

Tabelle 5.10 gibt die Paketserien einiger wichtiger Verzeichnisse einer Linux-Installation gemäß der File System Hierarchy (FSH) an. Von besonderer Bedeutung sind in diesem Kontext bspw. die Verzeichnisse der lokalen Konfigurationsdaten (`/etc`) oder ausführbarer Programme wie `/bin` (Programme für den Bootprozesses, Single-User Mode, Reparatur), `/sbin` (administrative Programme des Bootprozesses), `/usr/bin` (primäres Verzeichnis ausführbarer Programme) oder `/usr/sbin` (Programme zur Systemadmini-

[12] Die Tabulator-Taste kann insbesondere zur Vervollständigung von Befehlseingaben genutzt werden, bspw. in der Bash, und kann somit die Eingabesequenz maßgeblich beeinflussen.

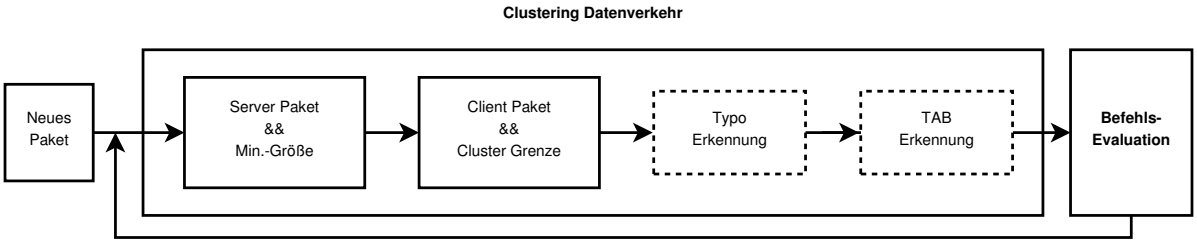


Abbildung 5.20: Stufen der Befehlserkennung aus dem Datenverkehr in verschlüsselten Umgebungen. Für die korrekte Erkennung eines Befehls sind insbesondere die Detektion des ersten und letzten jeweils zugehörigen Paketes entscheidend. Dies erfolgt insbesondere auf Basis der Paketgrößen und der beobachteten Zeiten. Zur Verbesserung der Detektionsergebnisse können Effekte wie Tippfehler oder bspw. die Nutzung der Tabulator-Taste[12]berücksichtigt werden. In der Implementierung des Prototypen wurde dies noch nicht berücksichtigt, weshalb die zugehörigen Komponenten gestrichelt eingezeichnet sind.

Tabelle 5.9: Beispiele von Mustern der Payloadgröße von Befehlen und den zugehörigen Antworten des Servers.

Eingabe beim Nutzer	
`df`	48, 48
`ls -l`	48, 48, 48, 48, 48
Login-Sequenz	39, 792, 24, 144, 16, 48, 64
Antwort des Servers	
`df`	576, 80
`ls -l`	96
Login-Sequenz	39, 784, 152, 720, 48

Tabelle 5.10: Beispiele der Serien von Payloadgrößen bei der Abfrage verschiedener Verzeichnisse.

Verzeichnis	Antwortsequenz
/	1448, 216
/etc	1448, 1448, 1248, 1448, 1448, 1448, 568, 1448, 1448, 1448, 1448, 1448, 1448, 848
/bin	1448, 1448, 1248, 1448, 1144, 1448, 792
/sbin	1448, 1448, 1248, 1448, 1048, 1448, 1448, 1248, 832
/boot	688

stration).

Neben diesen Größen können auch die Verarbeitungszeiten des Servers mittels der Zeitpunkte der Detektion der Pakete an der Datensonde berücksichtigt werden: Wird bspw. ein Verzeichnis mit einer großen Zahl von Dateien aufgelistet, dauert die Ausgabe der ersten Daten typischerweise deutlich länger, als dies bei Verzeichnissen mit einer geringen Anzahl von Dateien der Fall ist[13].

Für die Evaluation können daher entweder konkrete Paketserien genutzt werden, oder es kann eine Korrelation mit bekannten Serien erfolgen: Kann man konkrete Paketserien in einem direkten Vergleich prüfen, sinken entsprechend die Fehlerwahrscheinlichkeiten, andererseits muss ein umfangreicher Datensatz vorgehalten werden. Hierbei müssen nicht nur minimal sämtliche, durch die Angriffsbäume als wichtig eruierten Befehle aufgenommen werden, ggf. Schalter und Parameter berücksichtigt werden und je Befehl verschiedene Serverantworten vorgehalten werden (bspw. die im Kontext eines Angriffs wichtigsten Verzeichnisse für die Ausgabe von `ls -l`, bspw. `/etc`, `/sbin`, `/bin` oder `/boot`), sondern auch die zugrunde liegende Verschlüsselung berücksichtigt werden. Zusätzlich muss beachtet werden, dass bei der Nutzung von Blockchiffren unvollständige Informationsblöcke aufgefüllt werden und somit immer Blöcke gleicher Länge entstehen, wodurch mehr Informationen verloren gehen (vgl. Kapitel F.3.8). Hier kann kein exakter Rückschluss auf die Originallänge durch die Paketgrößen erfolgen, somit bieten sich ebenfalls wieder Korrelationsverfahren an.

Tabelle 5.11 fasst die Vor- und Nachteile der beiden Verfahren zusammen.

Zur jeweiligen Geschwindigkeit kann nur ein grober Richtwert gegeben werden. Die Ausführungsgeschwindigkeit bei der Nutzung eines direkten Vergleichs sinkt mit zunehmender Anzahl von zu prüfenden Serien; nutzt man Korrelationen, müssen insbesondere nicht die im Falle des direkten Vergleichs jeweiligen Kryptovarianten berücksichtigt werden, wodurch die Anzahl der durchzuführenden Operationen sinkt. Andererseits lassen sich bei direktem Vergleich bspw. feste Abstände zwischen den Blockgrößen nutzen, um die durchzuführenden Operationen zu reduzieren.

[13] Das exakte Verhalten ist weiterhin abhängig vom verwendeten Dateisystem und Funktionen wie bspw. dem integrierten Cache der Festplatte.

Tabelle 5.11: Vor- und Nachteile der Auswertung von Paketgrößen durch explizite Werte respektive durch Nutzung von Korrelationen.

	Explizite Paketgrößen	Korrelation
Fehlerrate	gering	mittel
Aufwand Umsetzung	sehr hoch	gering
Geschwindigkeit	gering	mittel

Abbildung 5.21 zeigt den Aufbau der Befehlsevaluation basierend auf dem direkten Vergleich bekannter Paketserien. Ein elementarer Schritt ist die Aufbereitung des Datenstroms in eine Folge von Clustern (vgl. Kapitel 5.1.3), welche jeweils die relevanten Client- und Server-Pakete eines Befehls beinhalten (Cluster-Erzeugung). Diese werden an die Funktion `evaluate_cmd()` gegeben, mittels der ein Vergleich der identifizierten Paketserien und den potentiell gefährlichen Paketserien der Datenbank erfolgt. Liegt eine Übereinstimmung vor, wird dem jeweiligen Befehl eine entsprechende Wahrscheinlichkeit für den Cluster zugewiesen. Sind mehrere, identische Paketserien verschiedener Befehle in der Datenbank vorhanden, werden diese zunächst alle als mögliche Kandidaten dem Cluster zugewiesen; eine Priorisierung erfolgt dann abhängig der weiteren Kommandos im Datenstrom. Hierzu wird insbesondere die Funktion `evaluate_seq` benötigt, deren Aufgabe es ist, die Wahrscheinlichkeit einzelner Befehle abhängig der Serie identifizierter Befehle eines Datenstroms zu prüfen und ggf. anzupassen. Hierfür werden die aufgrund der Korrelationsergebnisse mit den Referenzpaketserien ausgewählten Befehle mit den jeweiligen Befehlspools der Teilziele des zugehörigen Angriffsbaums verglichen. Jedem Befehl ist dabei ein Wahrscheinlichkeitswert und ein Gefährdungswert zugeordnet:

- Der **Wahrscheinlichkeitswert** korrigiert geringe Korrelationswerte im Sinne der Auswahlreihenfolge der identifizierten, möglichen Befehlskandidaten: Sind bereits mehrere Befehle aus einem Pool eines Teilziels identifiziert und kann ein weiterer Befehl zugeordnet werden, der als Kandidat erkannt wurde, jedoch geringere Ähnlichkeit aufweist als andere mögliche Kandidaten, wird dieser Befehl ausgewählt und zugeordnet, wenn er mit einem hohen Wahrscheinlichkeitswert im Pool vertreten ist. Am Beispiel: Es wird ein Befehl evaluiert und mögliche Kandidaten sind `uname` und `df -h` mit einem Korrelationswert von 0.693 respektive 0.971. Weiterhin sind bereits mehrere Befehle aus dem Befehlspool des Teilschritts Analyse identifiziert. In diesem Befehlspool liegt der Wahrscheinlichkeitswert für `df -h` bei 0.4, für `uname` jedoch bei 0.8. Dann wird im Rahmen der Evaluation der Befehlssequenz für das mögliche Teilziel der Befehl `uname` anstelle von `df -h` ausgewählt.

- Der **Gefährdungswert** beschreibt die Bedeutung eines Befehls für den Angriff im Rahmen des jeweiligen Befehlspools. Bspw. hat die Suche nach Dateien mit bestimmten Ausführungsrechten einen höheren Gefährdungswert, als das Auslesen eines Verzeichnisses.

Anhand der Wahrscheinlichkeitswerte kann somit zunächst die Reihung respektive Auswahl der Kandidaten eines Befehls angepasst werden. Sind ausreichend Befehle ei-

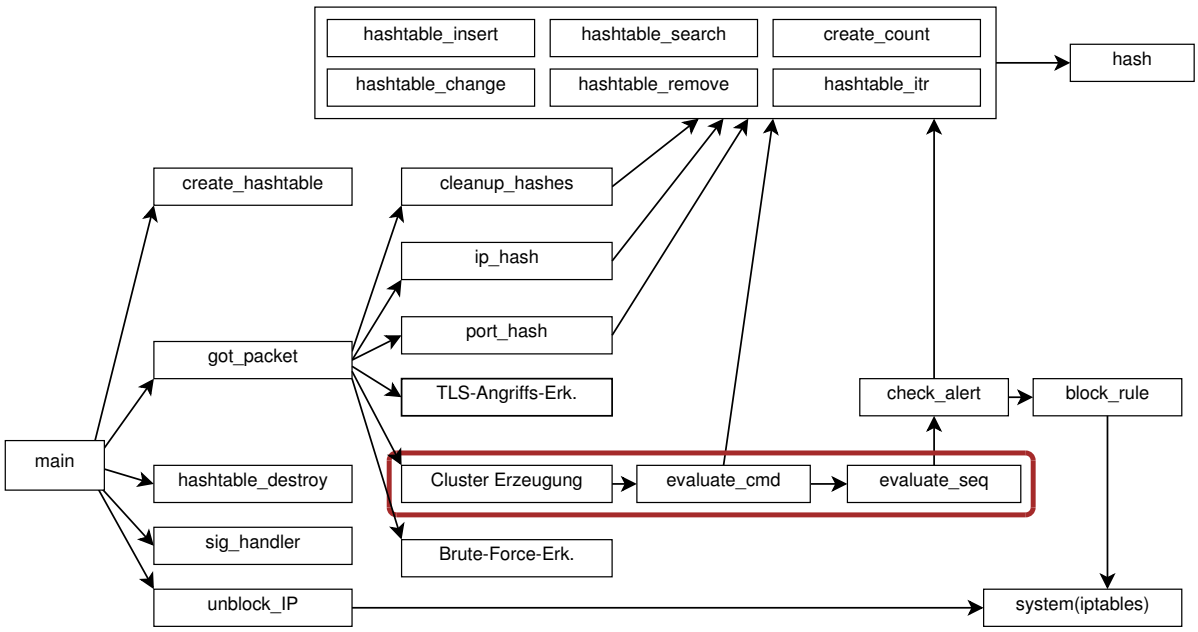


Abbildung 5.21: Aufbau des Moduls zur Befehlsevaluation und Integration in die Datensonde.

nes Teilziels identifiziert, kann ein Alarm generiert werden. Hierbei kann als Alarmierungskriterium einerseits die Anzahl notwendig zu identifizierender Befehle im jeweiligen Befehlspool als Schwellwert genutzt werden, andererseits kann die Summe der Gefährdungswerte der identifizierten Befehle als weitere Alarmierungschwelle herangezogen werden.

Ermöglicht eine Serie von Befehlen ein Erreichen eines Zwischenziels gemäß der aufgestellten Angriffsbäume, kann ein Alarm generiert werden, bzw. eine entsprechende Regel zur Sperrung der verdächtigen IP-Adresse erzeugt (Funktion `block_rule()`) und gesetzt (Funktionsaufruf `system(iptables)`) werden.

Der in Kapitel F.3.11 aufgeführte Algorithmus 6 zeigt die Umsetzung der Überprüfung auf Erreichung von Teilzielen der Angriffsbäume.

Da die Erzeugung und Pflege der notwendigen Datenbanken der Paketserien sehr aufwändig ist, wird für die Befehlsevaluation eine zweite Auswertetechnik integriert, basierend auf der Nutzung von Korrelationen. Hierbei werden komplette Sitzungen herangezogen, um die Ähnlichkeit der laufenden Verbindung zu ermitteln. Werden Korrelationen kompletter Sitzungen benutzt, können hierbei insbesondere auch die charakteristischen Eigenschaften der Nutzer mit eingehen, bspw. deren Nutzung der Tastatur, Funktionen wie Autovervollständigung oder der spezifische, von diesen genutzte Befehlssatz. Für die nun vorzunehmende Evaluation wird das Korrelationsmodul der Einbruchserkennung herangezogen, adaptiert und in die Befehlserkennung als alternative Auswertemöglichkeit integriert. Abbildung 5.22 zeigt die Integration der Auswertung im Rahmen des Gesamtsystems.

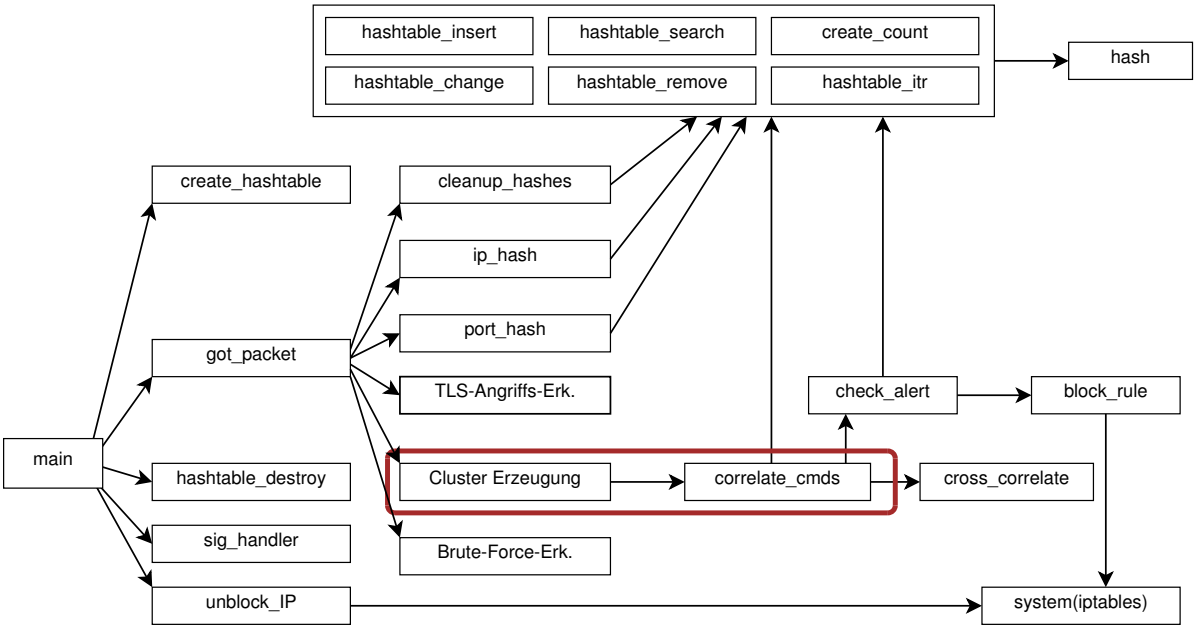


Abbildung 5.22: Aufbau des Moduls zur korrelationsbasierten Befehlsevaluation und Integration in die Datensonde.

Die Korrelation beginnt nach der Aufzeichnung einer minimalen Anzahl von Paketen. Dies kann anschließend bis zu einem Abbruchkriterium oder kontinuierlich fortgeführt werden, wobei bei letzterem entweder durchgehend alle aufgezeichneten Pakete genutzt werden, oder weitere Techniken wie bspw. *Sliding Windows* anwendbar sind und in künftige Versionen der Befehlserkennung integriert werden können. Für die Erprobung der Anwendbarkeit des Verfahrens wird ein Grenzwert festgelegt, nachdem der Prototyp die Evaluation abbricht und für die vorliegende Verbindung abschließt. Die Durchführung der Korrelationen stützt sich auf die bereits im Rahmen der Einbruchserkennung eingesetzte Funktion `cross_correlate()`, Alarmierungen und weitere Maßnahmen im Falle der Detektion einer bösartigen Verbindung laufen analog zur Befehlsevaluation bei Nutzung expliziter Paketserien.

Modul zur Nutzer-Identifizierung*

Konnte ein Angreifer den Zugang zu einem Nutzerkonto kompromittieren, ist der nächste Schritt eine Eskalation der Nutzerrechte, um die Kontrolle über den Rechner zu bekommen und ihn für die eigenen Zwecke einsetzen zu können. Ein Innentäter wiederum kann seine bereits bestehende Autorisierung ausnutzen, um bspw. unerlaubt Daten aus einem System zu kopieren. Um entscheiden zu können, ob eine jeweilige Aktion erlaubt oder illegal ist, ist daher die Kenntnis des Nutzers hinter einer verschlüsselten Verbindung

*Dieser Abschnitt enthält eine Zusammenfassung von Teilen des Artikels „User identification in encrypted network communications“, International Conference on Network and Service Management (CNSM), IEEE, 2010.

von besonderer Bedeutung. Um eine Nutzer-Identifikation während der Arbeit eines bereits legitimierten Benutzers zu eröffnen, ist eine fortlaufende Evaluation der von ihm erzeugten Daten möglich.

Die Untersuchung der Tipp-Eigenschaften ist ein intensiv behandelter Forschungsbereich, jedoch erfolgte bisher keine Umsetzung der Verfahren im Bereich verschlüsselter Datenübertragung. Eine mögliche Ursache hierfür ist die Art der Kommunikation auf entsprechenden Verbindungen, welche nicht alle charakteristischen, biometrischen Merkmale zur Verfügung stellen. Im Kontext des Sicherheitssystems wird eine Nutzer-Identifikation auf Basis einer bereits bestehenden Verbindung durchgeführt, indem biometrische Merkmale aus den übertragenen Datenpaketen zurückgewonnen werden.

Eine Identifikation eines Nutzers auf statistischer Basis lässt sich insbesondere mittels zweier Verfahren realisieren. Einerseits kann anhand von typischen charakteristischen Merkmalen der Verbindung wie z.B. der durchschnittliche Verbindungsdauer, übertragene Datenmengen, Uhrzeiten des Ein- und Ausloggens oder dem verwendeten Befehlssatz ein Verhaltensprofil für jeden Nutzer erstellt werden, gegen welches die erhobenen Daten einer neuen Verbindung abgeglichen werden können.

Eine andere Möglichkeit ist die Ausnutzung der Funktionsweise einer Remote Session: Um ein flüssiges und angenehmes Arbeiten zu ermöglichen, werden Nutzeraktionen wie Tastatureingaben oder Bewegungen der Maus unmittelbar an den Server übertragen. Im Falle der Tastatureingaben bedeutet dies, dass jedes Zeichen einzeln als ein Datenpaket charakteristischer Größe über das Netz übertragen wird. Charakteristisch bedeutet hier, dass abhängig des genutzten Chiffrier-Algorithmus und ggf. weiterer Faktoren wie bspw. der genutzten Konsole, das verschlüsselte Datenpaket eines eingegebenen Zeichens immer die gleiche Größe besitzt. Bei der Nutzung von SSH und dem Verschlüsselungsalgorithmus AES128-CBC beträgt diese bspw. 48 Bytes.

Da die einzelnen Zeichen unmittelbar übertragen werden, entspricht die Zwischenankunftszeit der Pakete der Verbindung an einem Beobachtungsort dem biometrischen Tippverhalten des Nutzers. Dieses Tippverhalten wird anhand der evaluierbaren statistischen Merkmale der Pakete durch das Modul zur Clustererzeugung wiedergewonnen und kann nachfolgend entsprechend zur biometrischen Analyse herangezogen werden.

Dadurch, dass die Zwischenankunftszeiten der Pakete evaluiert werden müssen, muss abhängig der Qualität und Auslastung des Netzes ein Fehlerintervall bei der Auswertung der registrierten Zeiten berücksichtigt werden; da eine Genauigkeit im Millisekundenbereich ausreichend ist, kann dies für heutige Netze bei Auswahl entsprechender Parameter jedoch in den meisten Fällen gewährleistet werden. Um eine sichere Missbrauchsdetektion vorzunehmen, müssen die verhaltensabhängigen Parameter einer Verbindung über einen Mindestzeitraum analysiert werden. Ist kein Nutzerprofil für das betroffene Konto bzw. passend zu einer laufenden Verbindung vorhanden, ist eine entsprechende Evaluation nicht möglich. Dies ist jedoch ebenfalls ein Indiz für eine *unerlaubte* Nutzung, da bei einer geeigneten Organisation gewährleistet werden kann, dass jeder legitime Nutzer über ein entsprechendes Profil im System verfügt.

Zahlreiche Arbeiten beschäftigen sich mit dem Themenfeld der Nutzeridentifizierung mittels Tastaturanschlägen. Sharif et al. [348] schlagen ein Verfahrenn zur Erweiterung der traditionellen Nutzer-/Passwortkombination um die biometrische Verfahren Tasta-

turanschläge und Muster, die mittels der Maus anzuklicken sind, vor. Für die Analyse der Tastaturanschläge nutzt Sharif fünf Eigenschaften:

- Das maximale Zeitintervall zwischen zwei bestimmten Anschlägen,
- das minimale Zeitintervall zwischen zwei anderen Anschlägen des gleichen Wortes,
- die Größe und Reihenfolge der Verzögerungen beim Tippen,
- die Gleichmäßigkeit der Anschläge eines Wortes sowie
- ein Klick-Muster des Nutzers in einem farbigen Quadrat mit vier Elementen.

Zur Verbesserung der Ergebnisse wird eine Fehlerschwelle genutzt, welche die Benutzer abhängig ihrer Fähigkeiten im Umgang mit der Tastatur in drei Gruppen einteilt, Experten, Standard-Nutzer und Anfänger.

Rein auf der Nutzung eines fest vorgegebenen Textes basiert die Arbeit von Rybnik et al. [332]. Hierbei werden sieben Eigenschaften der Tastenanschläge eines Nutzers identifiziert:

- Haltezeit: Dauer eines bestimmten Tastendrucks.
- Lauf: Die Pause zwischen zwei aufeinander folgenden Tastendrücken.
- Anschläge: Die durchschnittliche Anzahl von Tastendrücken in einem Zeitintervall.
- Überschneidung: Das Überlappen verschiedener Tastenkombinationen, bspw. durch schnelles Tippen oder Nutzung der *Shift*-Taste.
- Fehlerzahl: Die Anzahl der Tippfehler, detektierbar durch die Nutzung der *Delete*- und der *Backspace*-Taste.
- Fehlerkorrektur: Die Art und Weise, wie ein bestimmter Nutzer Tippfehler korrigiert.
- Nutzung der Cursor-Navigation: Die Art und Weise, wie ein bestimmter Nutzer die Cursor-Tasten zur Navigation einsetzt.

Rybnik unterscheidet weiterhin zwischen dem Drücken einer bestimmten und einer beliebigen Taste. Weiterhin sind die Auswertung von Zeichen-Eingabefolgen, welche insbesondere von der jeweiligen Sprache abhängen, vorgeschlagen.

Angemerkt sei, dass mehrere Veröffentlichungen die Umsetzbarkeit eines Systems zur Authentisierung *rein* auf der Basis des Tippverhaltens aufgrund der Detektions- und Fehlwahrscheinlichkeiten für nicht realisierbar halten[14]. Dahingegen bietet bspw. die Firma Psylock[15] ein System zur Nutzerauthentisierung an, das lediglich auf der Analyse

[14] In Europa werden die Grenzwerte für die maximalen Fehlerzahlen von Authentisierungssystemen im Europäischen Standard EN 50133-1 [18] festgelegt, die in Deutschland national in der DIN EN 50133-1 / VDE 0830-8-1:2003-09 umgesetzt ist.

[15] Die Firma hat im März 2011 Insolvenz angemeldet.

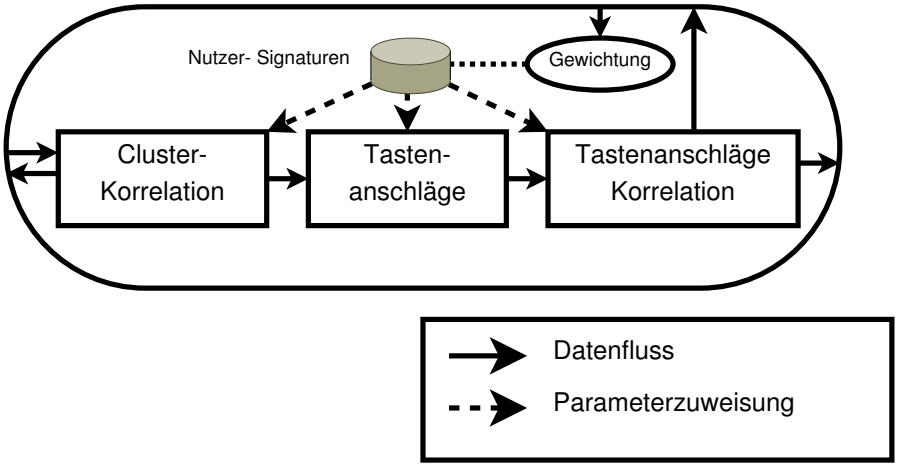


Abbildung 5.23: Teilverfahren zur Nutzeridentifikation in der S2E2-Architektur.

der Eingabe eines kurzen, festgelegten Textes basiert [165]. Das Verfahren basiert auf der Auswertung der folgenden sieben Eigenschaften: Anschläge, Rhythmus, Agilität, Kontinuität, Tippfehler, die Inkoheränz des Tippens sowie die Nutzung der *Shift*-Taste [16] und ist bspw. auch für die Nutzung für Online-Banking zugelassen.

Entsprechend wird ein neues Verfahren zur Nutzer-Identifikation als Modul integriert, welches aus den Komponenten gem. Abbildung 5.23 besteht.

Anhand der in die einzelnen Cluster aufgeteilten Pakete kann das Timing der Tastenanschläge des Nutzers identifiziert werden. Hierfür werden die Unix- Zeitstempel für jedes Paket aufgezeichnet, welches den Beobachtungspunkt im Netz passiert. Die Genauigkeit der zugehörigen Funktion `gettimeofday()` beträgt eine Mikrosekunde, was eine ausreichende Genauigkeit für sowohl die Netzumgebung als auch den Anforderungen bzgl. einer Auswertung der Tastenanschläge ergibt. Basierend auf den aufgezeichneten Zeitstempeln und der Auswahl der zu einem Cluster gehörigen Paketen, werden im nächsten Schritt mehrere statistische Analysen durchgeführt, u.a. Kreuzkorrelationen zwischen den ermittelten Zeiten und den Werten der Nutzerprofile in der Datenbank, die Bestimmung von Verzögerungen, etc.

Durch das Vorliegen von verschlüsselten Datenverkehr lassen sich nicht alle charakteristischen Eigenschaften, die in einem unverschlüsselten Umfeld zur Verfügung stehen, auswerten. Tabelle 5.12 gibt einen Überblick der wichtigen Eigenschaften.

Wie in der Tabelle aufgeführt, können nicht alle charakteristischen Eigenschaften der Tippbiometrie genutzt werden. Insbesondere markante Parameter wie die Haltezeit und die Überschneidung von Eingaben, bspw. bei speziellen Tastenkombinationen aber auch, wenn sich durch schnelles Tippen die Betätigungszeiten von beliebigen verschiedenen Tasten überschneiden, gehen bei der Übertragung verloren. Direkt anhand der detektierten, verschlüsselten Netzpakete sind somit lediglich die Parameter *Lauf* und *Anschläge* regenerierbar. Diese können anhand der Auswertung der Zeitstempel der jeweiligen Pakete

[16] Aufgrund der proprietären Software stehen keine detaillierten Informationen zur Verfügung.

Tabelle 5.12: In verschlüsselten Umgebungen verfügbare Eigenschaften von Tastaturanschlägen. Im Gegensatz zu unverschlüsseltem Datenverkehr können mehrere Parameter nicht ausgewertet werden.

Eigenschaft	Nutzbarkeit
Haltezeit: Dauer eines Tastendrucks	✗
Lauf: Pause zwischen Tastendrücken	✓
Anschläge: Durchschnittliche Zahl von Tastendrücken	✓
Überschneidung: Spezielle Tastenkombinationen	✗
Fehler: Tippfehler	(✓)
Korrekturen: Art der Fehlerkorrektur	✗
Cursor-Nutzung: Navigation mittels Cursor-Tasten	✗

bestimmt werden, nachdem der Beginn eines Clusters detektiert wurde: Der Lauf entspricht dem Zeitunterschied zwischen zwei aufeinanderfolgenden Paketen des Clients an den Server mit jeweils charakteristischer Größe:

$$t_{Lauf}(n) = t_{n+1} - t_n \tag{5.5}$$

Für die Anschläge gilt, dass diese zunächst für jeden eingegebenen Befehl ermittelt werden können:

$$n_{Ansch}(t) = |command(t)| \tag{5.6}$$

Anschließend können die Anschläge eines Intervalls $[t_1, t_2]$ abhängig der bereits eingegebenen Befehle ermittelt werden:

$$n_{Ansch} = \frac{\sum_{t=t_1}^{t_2} n_{Ansch}(t)}{t_2 - t_1} \tag{5.7}$$

Nachfolgend ist die Funktionsweise der hierfür genutzten Evaluationsstufen (vgl. Abbildung 5.24) dargelegt.

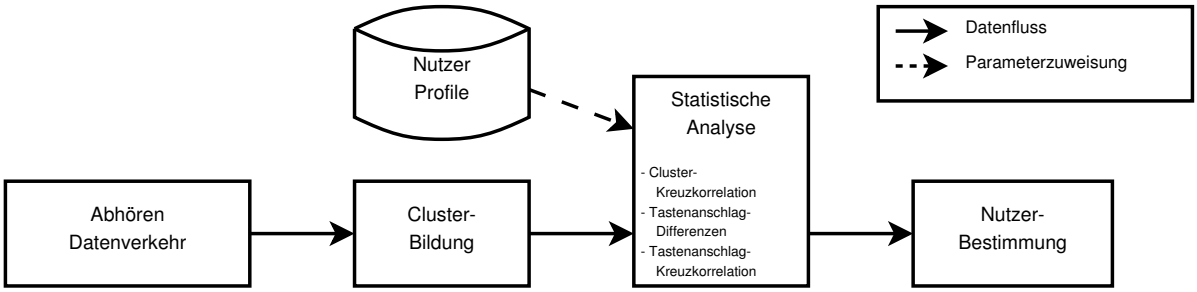


Abbildung 5.24: Architektur zur Nutzererkennung in verschlüsselten Umgebungen. Anhand mehrerer statistischer Analysen werden die mittels der Netzpakete erzeugten und aufbereiteten Daten mit den Profilen der Datenbank verglichen und der Nutzer einer Verbindung evaluiert.

Für die Analyse der aufgezeichneten Daten wird ebenfalls die Clustererzeugung herangezogen. Dies bietet den Vorteil, dass die zur Übertragung der Tastendrücke zugehörigen Pakete bereits in der für die Gewinnung der statistischen Daten erforderlichen Form vorliegen, insbesondere sind administrative und Echo-Pakete bereits entfernt. Voraussetzung für die Identifikation eines Nutzers ist das Vorhandensein eines entsprechenden Profils in der Datenbank. Dies kann in der Praxis generell in zwei Varianten erfolgen:

- Durch eine explizite Lernphase, bspw. mittels eines expliziten Tools und der Eingabe mehrerer Sätze zur Identifikation der relevanten tippbiometrischen Parameter jedes Nutzers, oder
- mittels eines Agenten, welcher die benötigten Parameter in einer laufenden Sitzung extrahiert.

Die statistische Analyse macht sich mehrere Aspekte der Tippbiometrie zunutze, welche auf verschiedene Arten genutzt und verarbeitet werden. Hierfür werden folgende Operationen durchgeführt:

- Kreuzkorrelationen zwischen den Zwischenankunftszeiten der aufgezeichneten Paketserie und den Profilen, somit Berücksichtigung der Parameter *Anschläge* und *Lauf*.
- Absolute Differenz zwischen dem Mittel der Anschläge der aufgezeichneten Paketserie und dem Mittel der Anschläge der jeweiligen Profile.
- Absolute Differenzen der maximalen und minimalen Verzögerungen zwischen zwei Tastendrücken der Aufzeichnung und den Profilen.

Anfänglich konnte ein Profil in der Auswertung regelmäßig nicht erkannt werden (vgl. Kapitel 6.3.2), weshalb die zur Verfügung stehenden Parameter nochmals betrachtet wurden und um eine Reihe weiterer Bewertungsaspekt ergänzt wurden: Die Differenzen zwischen den absoluten Beträgen verschiedener Parameter gehen sowohl direkt in die Bewertung der einzelnen Verbindungen ein, als auch in der Form von Korrelationen über eine Serie von mehreren ausgewerteten Befehlen hinweg. Eine weitere, hierbei eingeführte Analyse wird im weiteren Verlauf als *Profilverzögerung* bezeichnet und wie folgt bestimmt: Alle aufeinanderfolgenden Zwischenankunftszeiten eines Befehls werden aufsteigend sortiert, die Reihenfolge der so sortierten Positionen ergibt die für weitere Korrelationen genutzte Sequenz. Als Veranschaulichung wird eine Eingabe des Befehls `mount` mit beispielhaften Zeiten demonstriert. Somit ergibt sich aus den Zwischenankunftszeiten (Angaben beispielhaft in gerundeten Millisekunden) von

$$m\ (10)\ o\ (70)\ u\ (40)\ n\ (30)\ t\ (80)\ \textit{Enter}$$

eine Profilverzögerung von

$$1 - 4 - 3 - 2 - 5$$

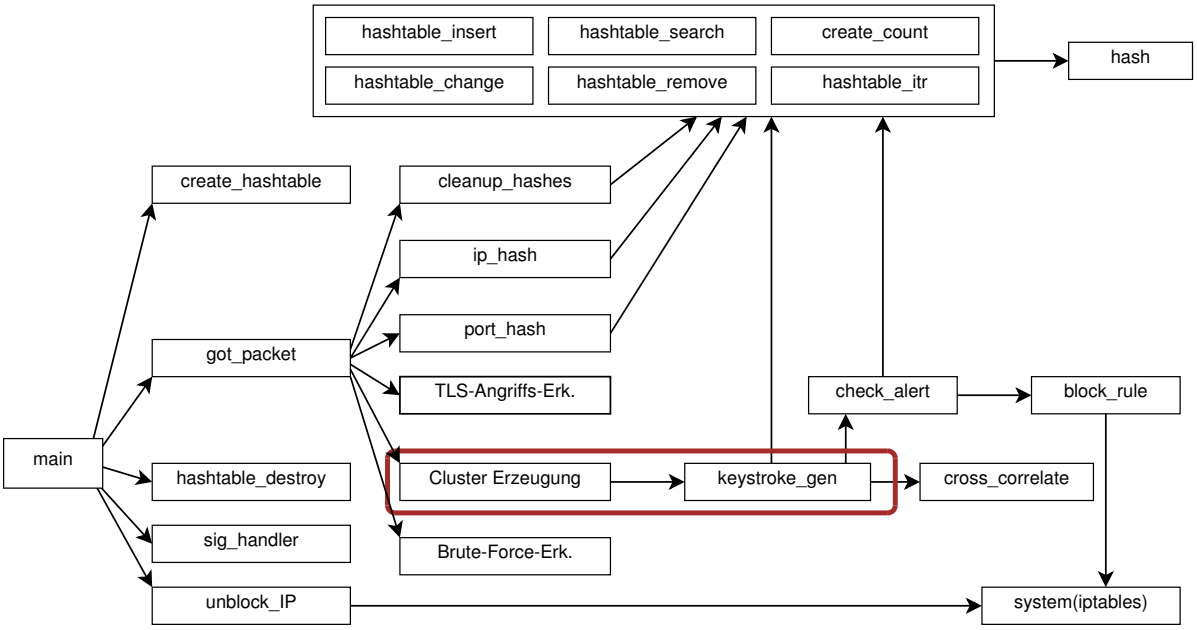


Abbildung 5.25: Aufbau des Moduls zur Nutzererkennung und Integration in die Datensonde.

Entsprechend wird nachfolgend eine Korrelation mit den jeweiligen Werten der verschiedenen Profile der Datenbank durchgeführt.

Abbildung 5.25 zeigt die Umsetzung der Funktionalitäten und deren Integration in die Datensonde des Sicherheitssystems.

Die zur Auswertung erforderlichen Cluster stehen durch die vorgestellte Funktion (vgl. Kapitel 5.1.3) zur Verfügung. Diese werden an die Funktion `keystroke_gen()` weitergegeben, welche die beschriebenen Parameter der Tippbiometrie erzeugt und die erforderlichen Rechnungen und Gewichtungen durchführt. Für Serien, welche durch Korrelationen verarbeitet werden, wird wiederum die Funktion `cross_correlate()` genutzt.

Da im Falle eines Innentäters eine Prüfung auf das Vorhandensein eines passenden, biometrischen Profils nicht ausreichend ist, muss zusätzlich eine entsprechende Evaluation der jeweils für einen Nutzer gültigen Richtlinien erfolgen. Dies erfolgt mittels der Autorisierungsevaluation (vgl. Abbildung 5.9). Hier werden anhand des festgestellten Nutzers dessen Befugnisse überprüft, wofür die Adressen der Kommunikation verwendet werden und ebenfalls die identifizierten Eingaben der Befehlsevaluation herangezogen werden können.

5.1.6 Skalierbarkeit und Datenstromaufteilung

Mit Hinblick auf die immer steigenden Bandbreiten und Datenmengen ist eine effiziente Detektion von Angreifern von entscheidender Bedeutung. Hierfür werden zum einen

Hashtabellen genutzt, um den Verarbeitungsprozess zu beschleunigen. Verwendet werden hierbei FNV-1a Hashes, welche für hohe Geschwindigkeiten und geringe Kollisionsraten konzipiert sind und durch eine hohe Streuung insbesondere auch für das Hashen ähnlicher Strings wie bspw. IP-Adressen geeignet sind (vgl. Anhang F.3.4). Zusätzlich und insbesondere ermöglicht das Design und die Funktionsweise des Sicherheitssystems eine hochgradige Parallelisierung:

- Im Rahmen der Datengewinnung- und Aufbereitung erfolgt eine verbindungsweise Analyse. Diese lässt sich bspw. anhand von Adressbereichen auf verschiedene Auswerteinstanzen aufteilen, bis hinunter zu einer Zuweisung von einer Verbindung pro Instanz.
- Die Einbruchserkennung nutzt neben Intra-Kreuzkorrelationen auch Korrelationen zwischen Verbindungen verschiedener Quellen. Da die Auswahl der jeweiligen Korrelationspartner auf statistischem Wege erfolgt, kann eine entsprechende Aufteilung auf mehrere Auswerteinstanzen problemlos erfolgen.

Dadurch, dass für bestimmte Operationen eine minimale Anzahl von Verbindungen miteinander korreliert werden müssen, um einen repräsentativen, mittleren Wert zu erhalten, gilt für die maximale Anzahl möglicher Prozesse n_{proc} bei einer Anzahl von n_{con} Verbindungen sowie n_{min} minimaler Anzahl von Verbindungen, welche miteinander korreliert werden müssen, für die Parallelisierung:

$$n_{proc} = \frac{n_{con} - m}{n_{min}} \tag{5.8}$$

Die restlichen m Verbindungen können hierbei beliebig auf die vorhandenen Prozesse aufgeteilt werden. Abhängig des Parameters n_{min} ermöglicht dies eine extrem hohe Parallelisierung der Evaluation, falls hohe Lastbedingungen und Ressourcenknappheit auf einer Auswerteinstanz dies erfordern. Die Bestimmung des Parameters n_{min} erfolgt im Rahmen der Evaluation, vgl. Kapitel 6.2.2.

5.2 Gegenmaßnahmen und Reaktionen

Um die Resistenz des Sicherheitssystems gegen Manipulationen oder Täuschungsversuche abzuschätzen, erfolgt nachfolgend eine Analyse möglicher Angriffsvektoren gegen die vorgestellte Architektur. Das Detektionsverfahren für die Einbruchserkennung in verschlüsselten Umgebungen kann in mehreren Punkten manipuliert werden, wenn ein Angreifer Kenntnis über die Nutzung eines entsprechenden Detektionsverfahrens in der Zielumgebung hat.

5.2.1 Einbruchserkennung

Eine Gefährdung und Manipulationsmöglichkeit entsteht bei der Einbruchserkennung bei der Durchführung der Inter-Sitzungskorrelationen, wenn die Angreifer einen zu hohen Anteil bzgl. des gesamten Datenaufkommens eines Servers haben. Da die Detektion

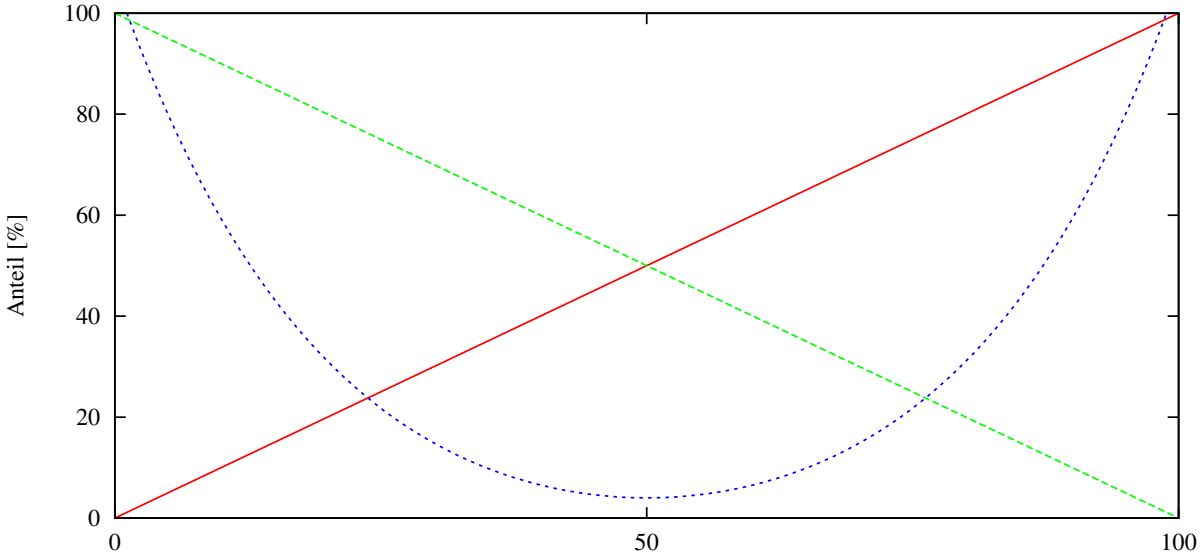


Abbildung 5.26: Zu erwartender Verlauf der kombinierten Alarmrate (blaue Kurve) unter steigendem Einfluss von bösartigem Verhalten (rote Linie). Die grüne Linie stellt den Verlauf der gutartigen Verbindungen dar, die entsprechend dem Zuwachs der bösartigen Verbindungen sinken. Die besten Detektionsraten können erreicht werden, wenn nur ein sehr geringer Anteil der Verbindungen bösartig ist, da diese eindeutig mittels der Korrelationen erkannt werden können. Bei steigendem Anteil bösartiger Verbindungen sinkt die Detektionsrate, da zunehmend auch bösartige mit bösartigen Verbindungen korreliert werden. Sind gut- und bösartige Verbindungen gleich vertreten, kann keine Detektion durchgeführt werden, bzw. entspricht einer zufälligen Bestimmung. Steigt der bösartige Anteil weiter, steigt die Detektionsrate wieder, jedoch werden in diesem Falle die geringer vertretenen, gutartigen Verbindungen durch das System gemeldet.

von Angriffen auf der Erkennung der Ähnlichkeit der Verbindungen beruht und bösartige Sitzungen durch deren Abweichungen in den Paketserien erkannt werden, kann ein Angreifer versuchen, durch eine Erhöhung der bösartigen Verbindungen eine entsprechende Dominanz der Verbindungen zu erzwingen und somit zu maskieren oder als gutartig darzustellen. Prinzipiell entspricht das Vorgehen hierfür also der Durchführung eines klassischen DoS-Angriffs. Hierbei muss jedoch berücksichtigt werden, dass dies wiederum auffällige bzw. detektierbare Charakteristika aufweisen kann: Insbesondere verlagert sich durch steigende Angriffe die Detektion von anormalen Verhalten hin zu den anteilig betrachtet abnehmenden, gutartigen Verbindungen. Folglich werden diese dann, durch das geänderte Verhältnis, als bösartig erkannt und gemeldet. Hier wird zwar im Sinne der Detektion ein Fehlalarm (False Positive) ausgelöst, jedoch kann dadurch der bzgl. den nicht gemeldeten Angriffen zweite, gleichzeitig vorliegende Fehlalarm (False Negative) mittels einer Betrachtung des Systemzustands entdeckt werden.

Abbildung 5.26 veranschaulicht die zu erwartenden Zusammenhänge der Angreiferrate auf die Detektionsfähigkeit des Systems.

Hierbei wird der Anteil der Angreifer von 0 auf 100 Prozent gesteigert, entsprechend sinkt der Anteil gutartiger Verbindungen im gleichen Maße ab. Mit einer steigenden Anzahl bösartiger Verbindungen wird die Wahrscheinlichkeit geringer, dass die jeweilige Verbindung nur mit zufällig ausgewählten, gutartigen Verbindungen korreliert wird. Da für die Bestimmung, ob eine Verbindung guter oder böser Natur ist, das Mittel über die verschiedenen Korrelationen der jeweiligen Verbindung herangezogen wird, steigt dieser Wert, sobald eine bösartige Verbindung mit einer anderen bösartigen Verbindung korreliert wird. Da Angriffe anhand von *geringen* Korrelationswerten erkannt werden, sinkt die Detektionswahrscheinlichkeit mit einer zunehmenden Zahl von Angriffen. Steigt der Angriffsanteil sogar über den der gutartigen Verbindungen hinaus, wird die Wahrscheinlichkeit dafür größer, dass mehr und mehr bösartige Verbindungen miteinander korreliert werden und immer weniger gutartige untereinander: Somit ergeben die Verbindungen der Angreifer hohe mittlere Korrelationswerte, gutartige Verbindungen geringe, wodurch für letztere Alarme generiert werden, jedoch nicht für die Angriffe. Anhand der durchschnittlichen, berechneten Korrelationswerte aller Verbindungen sollte sich dieses Verhalten jedoch erkennen lassen, so dass geeignete Maßnahmen ergriffen werden können. Das genau Systemverhalten muss im Rahmen der Evaluation analysiert werden (vgl. Kapitel 6).

5.2.2 Ausbruchs- und Innentätererkennung

Die Manipulationsmöglichkeiten im Rahmen der Ausbruchserkennung müssen für die beiden Module *Befehlsevaluation* und *Nutzeridentifizierung* betrachtet werden. Beiden gemeinsam ist die Nutzung der in Form von Clustern aufbereiteten Daten; werden zusätzliche Pakete durch den Angreifer in den Datenfluss eingeschleust, kann dies zu einer Erschwerung der jeweiligen Detektionsverfahren führen.

Bereits durch die Nutzung von *keep-alive* Nachrichten motiviert, lassen sich zunächst folgende Einflussmöglichkeiten feststellen:

- Absichtliche Tippfehler
- Erzwingen eines konstanten bzw. zusätzlichen Paketstromes
- Nutzung von Skripten
- Einspielen gutartiger Befehle

Diese Maßnahmen sind geeignet, den zu analysierenden Paketstrom nachhaltig zu verändern: Tippfehler beeinflussen die übertragene Anzahl von Paketen charakteristischer Größe. Eine Erkennung dieser ist im Kontext der Auswertung der statistischen Paketdaten jedoch nur begrenzt möglich. Ist ein Nutzerprofil vorhanden und konnte dies anhand der Datenbank identifizierten werden, können die Eigenschaften der benutzten Korrekturverfahren des jeweiligen Nutzers einbezogen werden, um Korrekturmaßnahmen des Nutzers zu erkennen und damit wiederum auf die korrekte Anzahl der Zeichen eines Befehls zu schließen. Andererseits bieten moderne Konsolen wie die bash eine automatische

Vervollständigung von Befehlen, bspw. durch die Nutzung der Tabulator (TAB)-Taste an. Dies verbirgt zum einen die korrekte Anzahl der Zeichen eines Befehls, zum anderen liegt keine ggf. identifizierbare Korrektur des Nutzers vor. Die Einführung absichtlicher Tippfehler (welche wiederum unmittelbar korrigiert werden) sorgt dafür, dass die ausgewerteten clientseitigen Informationen nicht mehr zu den entsprechenden Mustern der verschiedenen Befehle passen. Auch ein absichtliches Erzwingen eines konstanten bzw. kontinuierlichen Paketstromes stellt eine Möglichkeit dar, den Paketfluss nachhaltig zu verändern. Neben einer entsprechenden Konfiguration bspw. der keep-alive Nachrichten kann z.B. auch ein Skript genutzt werden, um zusätzliche Datenpakete zu übertragen und somit die Evaluation zu verfälschen (vgl. Kapitel F.3.9).

Den Verfahren ist zunächst gemein, dass sich hierdurch lediglich die von Clientseite gesendeten Informationen manipulieren lassen, die Serverantworten und deren spezifische Reaktionszeiten bleiben hiervon jedoch unbeeinflußt. Eine Möglichkeit besteht daher darin, dass im Rahmen der Befehlsevaluation lediglich die dem Server zugehörigen Anteile eines jeden Clusters ausgewertet werden, die insbesondere den höheren Informationswert im Vergleich zu den vom Client gesendeten Paketen haben. Der Detektionsmechanismus für die Auswertung der Kommandos bietet daher zwei Betriebsmodi an, eine Analyse basierend auf Client- und Serverpaketen und eine Detektion rein auf Basis der Server-Informationen. Für die Nutzeridentifizierung lässt sich dieses Verfahren jedoch nicht anwenden, da hier das Timing der Clientpakete von entscheidender Bedeutung ist; andererseits ist für die Identifizierung die Kenntnis bzw. Zuordnung zu einem Befehl nicht notwendig, da die erforderlichen Parameter aus dem Paketstrom der Tastenanschläge gewonnen werden[17]. Hierbei gilt, dass durch ein bewusstes, verändertes Bedienen der Tastatur zwar die Detektion eines vorhandenen Profils verhindert werden kann, es jedoch nicht möglich ist, das Profil eines anderen Nutzers zu imitieren (vgl. z.B. [251]). Da im Falle eines nicht erkennbaren Nutzers bzw. dem Fehlen eines passenden Profils ein Alarm generiert wird, kann somit hier durch die bewusste Änderung des eigenen Tippverhaltens die Detektion nicht umgangen werden (Ausbruchserkennung). Auf der anderen Seite kann nach der korrekten Erkennung eines Nutzers nicht nur bei der detektierten Eingabe von Befehlen, die zu Angriffen führen können, ein Alarm ausgelöst werden, sondern auch bei einem hohen Grad insgesamt nicht auswertbarer Befehle, da dies ebenfalls entsprechend verdächtig ist. Ist eine Detektion von einem Nutzer und dessen Behlen möglich, lässt sich der vorgesehene Weg des Systems gehen und die Nutzeraktionen können gegen die gemäß den festgelegten Richtlinien für ihn erlaubten geprüft werden.

Dadurch, dass der Angreifer im Rahmen der Ausbruchs- bzw. Innentätererkennung bereits Zugang zum überwachten System hat, kann er dort auch beliebige Pakete Richtung Client erzeugen, um die Analyse ebenfalls zu stören (vgl. Kapitel F.3.9). Da hier die Kommunikationsrichtung vom Server zum Client beeinträchtigt wird, betrifft dies nur die Evaluation der Befehlsauswertung; die Manipulation beeinträchtigt eine Evaluation der Befehle maßgeblich, wodurch das wiederholte Scheitern der Auswertung ein

[17] Ausnahme ist hierbei der Parameter *Profilverzögerung*, für dessen Bestimmung die Erkennung der Clustergrenze erforderlich ist.

Tabelle 5.13: Manipulationsmöglichkeiten des Sicherheitssystems und deren Auswirkungen auf die Detektion sowie mögliche Verteidigungs- und Schutzmaßnahmen.

Verfahren	Auswirkung	Verteidigung
Parallele Angriffe	Änderung Normalverhalten	Alarm bei Änderung des Detektionslevels
Paketinjektion	Störung Befehlsevaluation	Reine Server-Auswertung Alarm bei kontinuierl. Scheitern
Veränderung Tippverhalten	Störung Nutzeridentifikation	Alarm bei unbek. Nutzern
Nutzung von Skripten	Störung Ausbruchserkennung	Alarm bei nicht-autor. Verbindungen
Einspielen gutartiger Befehle	Störung Befehlsevaluation	Langfristige Beobachtung bösartiger Befehle

Anzeichen für einen entsprechenden Angriff ist.

Tabelle 5.13 fasst die Manipulationsmöglichkeiten und deren Auswirkungen für das Sicherheitssystem zusammen.

Von besonderer Bedeutung ist hier, dass die möglichen Angriffsverfahren, die gegen eine Detektion durch das Sicherheitssystem bei Kenntnis über dessen Vorhandensein gerichtet werden können, zu einem evaluierbaren Detektionsereignis führen: Es wird zwar in den oben aufgeführten Manipulationsversuchen *kein* korrekter Alarm bzgl. eines Angriffes erzeugt, jedoch liegen jeweils untypische Verhaltensweisen des Sicherheitssystems bzw. Evaluationsergebnisse außerhalb der normalen Betriebsschwellen vor. Daher können diese Ereignisse ebenfalls genutzt werden, um eine Alarmierung durchzuführen und somit den Manipulationsversuch zu vereiteln.

Diese Möglichkeit kann bspw. einfach in Form eines im Hintergrund ausgeführten Skripts umgesetzt werden, welches ein Zeichen auf die Verbindung schreibt und anschließend sofort wieder löscht. Dieses erfüllt somit die Aufgabe, zusätzliche künstliche Tastendrücke zu erzeugen; um die normale Funktion sicherzustellen, wird hier jedoch unmittelbar an jedes übertragene Zeichen ein *Backspace* übertragen und die künstlich hinzugefügte, eigentlich unerwünschte Eingabe korrigiert (vgl. z.B. Anhang F.3.9). Somit ist eine normale Befehlsausführung möglich, die Client-seitige Paketauswertung ist jedoch erheblich verfälscht. Um die Detektionsmöglichkeiten für einen solchen Angriff noch zu erschweren, können auch konstante Paketserien erzeugt werden; in diesem Fall sinkt jedoch die Agilität der SSH-Sitzung, der Angreifer muss dann mit höheren Verzögerungen bei der Befehlsübertragung und somit -verarbeitung rechnen.

Eine andere Möglichkeit besteht darin, dass der Angreifer eine hohe Zahl von unverdächtigen Befehlen verwendet, um die Nutzung der als gefährlich eingestuften Befehle vor dem System zu maskieren: Da eine Evaluation immer eine Reihe von Befehlen betrachten muss, um auf eine Angriffsintention schließen zu können, kann somit versucht werden, Schwellwerte des Systems zu unterwandern. Dieses Verfahren ist jedoch durch

drei Punkte maßgeblich charakterisiert:

- Ein derartiges Vorgehen wird typischerweise ein anderes Verhalten in der Auswertung zeigen, als eine gutartige Nutzersitzung.
- Die Schwellwerte und Reaktionen des Systems sowie die implementierten Befehle müssen bekannt sein.
- Der Angreifer muss seine Aktivitäten über einen sehr langen und im Sinne der erforderlichen Befehlsstreuung nicht abschätzbaren Zeitraum verteilen.

Zahlreiche andere Aspekte können zusätzlich aufgegriffen werden, um einen Nutzer anhand seiner charakteristischen Merkmale zu identifizieren, welche noch als zusätzliche Module in das Sicherheitssystem integrierbar sind (vgl. z.B. [397]). Von besonderer Bedeutung kann hier z.B. auch die Betrachtung der von einem Nutzer verwendeten Befehle sein (Befehlswörterbuch). Beispiele sind hier ebenfalls die Auswertung von Pausephasen des Nutzers (vgl. [91]) oder die Evaluation der zu einem Nutzerdatenstrom erzeugten Flowdaten. Melnikov nutzt bspw. auf dem Browsing-Verhalten von Nutzern basierende Flowdaten zur Identifizierung des Urhebers einer Sitzung [277].

Wichtig ist, dass es hier auch ausreichen kann, die Effizienz eines Angriffs durch die für eine erfolgreiche Manipulation oder Täuschung des Sicherheitssystems notwendigen Maßnahmen so sehr zu senken, dass der Wert eines Angriffs für den Angreifer nicht mehr in einem lohnenden Verhältnis zum Aufwand steht.

5.3 Zusammenfassung

Das Kapitel stellt die Architektur zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen vor. Hierfür werden zunächst die Möglichkeiten, welche in einem verschlüsselten Datenstrom noch zur Analyse zur Verfügung stehen, ausgewertet. Auf dieser Basis wird die Gesamtarchitektur vorgestellt, anschließend erfolgt eine detaillierte Diskussion der Teilkomponenten, welche sich in die Bereiche Datengewinnung, Einbruchs- sowie Ausbruchs- und Innentätererkennung aufteilen. Hierbei werden mehrere Aspekte betrachtet, in welcher Art und Weise die statistisch erhebbaren Daten einer Verbindung auf der einen Seite zur Erkennung von Angriffen von Außen ohne die Nutzung von Signaturen oder Lernphasen verwendet werden können, andererseits dient die Erkennung von Innentätern und kompromittierten Accounts dem Schutz vor Datenausschleusung, Aktivitäten von Schadsoftware u.a. Eine Diskussion über mögliche Stör- bzw. Täuschmaßnahmen gegen die neue Architektur des Sicherheitssystems sowie mögliche Gegenmaßnahmen und eine Betrachtung der Parallelisierbarkeit des Systems schließen das Kapitel ab.

Ich habe meinen Schülern beigebracht, dass sie mir kein Wort glauben dürfen, sondern alles selbst nachprüfen müssen.

Konrad Lorenz

6 Evaluation

Die vorgestellte Architektur eines Sicherheitssystems für verschlüsselte Umgebungen wird im nachfolgenden Kapitel validiert (vgl. Abbildung 6.1).

Nach der Vorstellung der Vorgehensweise werden die verschiedenen Module des Sicherheitssystems bzgl. ihrer Leistungs- und Detektionsfähigkeit analysiert. Ziel ist es, die Leistungsfähigkeit der neuen Architektur bzgl. der Erkennung von Angriffen nachzuweisen. Hier wird zunächst das Modul der Datengewinnung selbst, anschließend die Module der Einbruchserkennung (Brute Force-Erkennung, TLS-Angriffsdetektion) respektive die Module der Ausbruchserkennung (Befehlsevaluation, Nutzeridentifizierung) untersucht. Der Technische Anhang in Kapitel F.4.1 gibt zusätzliche Informationen bzgl. der genutzten Evaluationsumgebung, sowie weitere, detaillierte Ausgaben der implementierten Prototypen.

6.1 Modul zur Datengewinnung

Abbildung 6.2 zeigt den Ressourcenbedarf der Datensonde bei der Analyse eines 100 Mbps Datenlinks[1]. Die Erzeugung der statistischen Daten zur Weitergabe an die nachfolgenden Module wurde anhand einer produktiven Verbindung gemessen, hierbei wurde als Hardwarebasis ein älterer Rechner mit einem mit 3 Gigahertz (GHz) getakteten Pentium-4 Prozessor und 498 MB Arbeitsspeicher genutzt. Nach Beginn des Datenverkehrs und Start der Aufbereitung der statistischen Daten ist zunächst ein deutlicher Anstieg der Speichernutzung zu verzeichnen, welche von der Generierung der initialen Hashtabellen herrührt (vgl. Abbildung 6.2a). Die Speichernutzung ist dann zunächst konstant; im Falle, dass die möglichen Einträge der Hashtabelle ausgehen, wird diese vergrößert, was sich wiederum als Sprung in der Speichernutzung darstellt. Das Auftreten einer Vergrößerung der Tabellen ist abhängig von der Anzahl der unter Beobachtung befindlichen Verbindungen und von deren Dauer. Durch die Entfernung der Daten abgeschlossener Verbindungen aus den Hashtabellen, kann der Speicherbedarf somit gering gehalten und das Anwachsen der Hashtabellen reduziert werden. Auf einem 1 Gbps-Link, der ebenfalls über einen längeren Zeitraum unter verschiedenen Lastbedingungen analysiert wurde, belief sich der durchschnittliche Speicherbedarf auf ca. 20 MB.

Die CPU-Nutzung unter Lastbedingungen des 100 Mbps-Links ist ebenfalls in Abbildung 6.2 ersichtlich. Gut erkennbar ist, dass die durchschnittliche Nutzung zwischen 20 und 40 Prozent liegt, wenn die Datensonde im Sinne der Generierung statistischer

[1] Angemerkt sei, dass bei diesen Messungen bereits das Modul zur Brute Force-Erkennung mit integriert ist; die alleinige Erzeugung der statistischen Daten bedarf entsprechend geringerer Ressourcen.

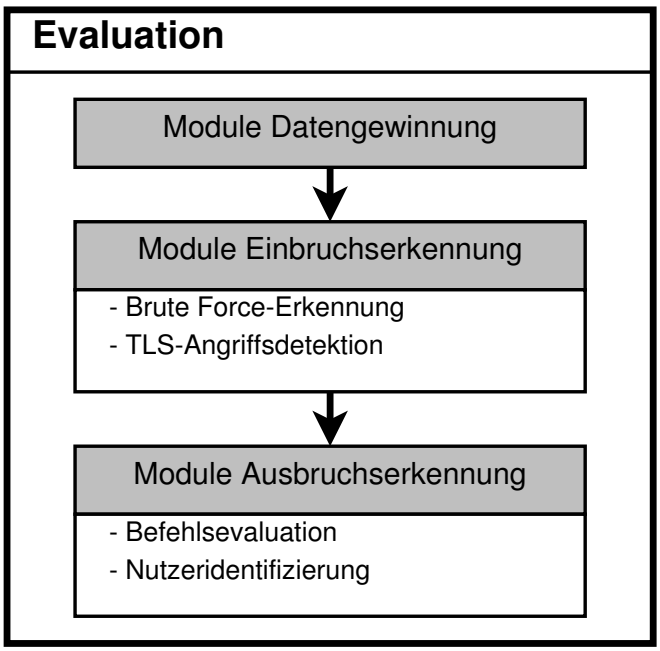


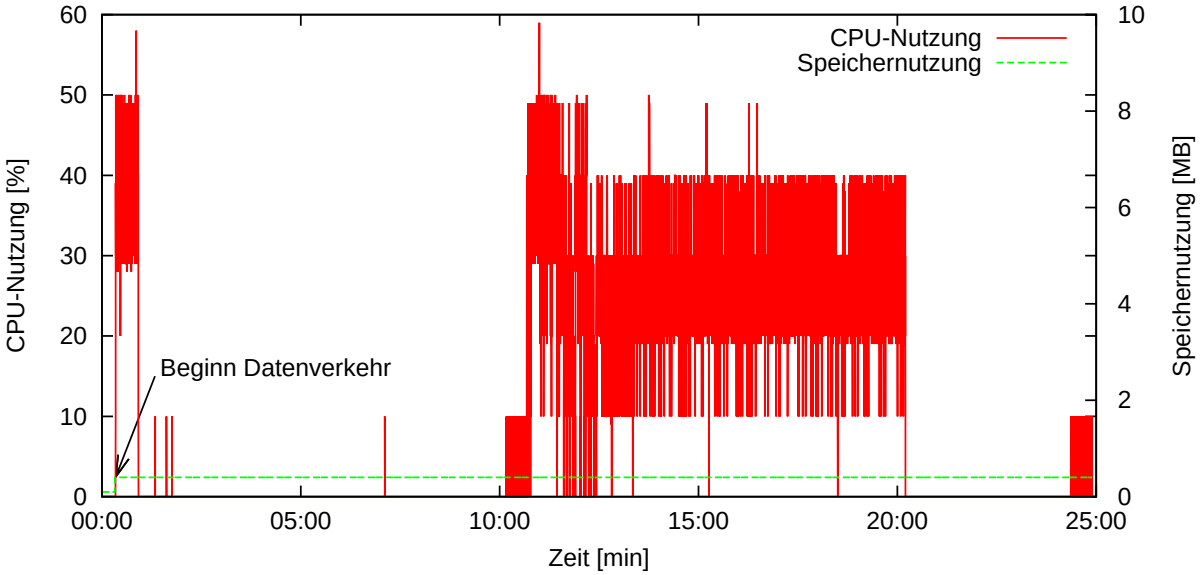
Abbildung 6.1: Aufbau von Kapitel 6.

Daten aktiv ist; liegt kein Datenverkehr auf dem überwachten Netz vor, benötigt die Sonde entsprechend keine CPU-Ressourcen.

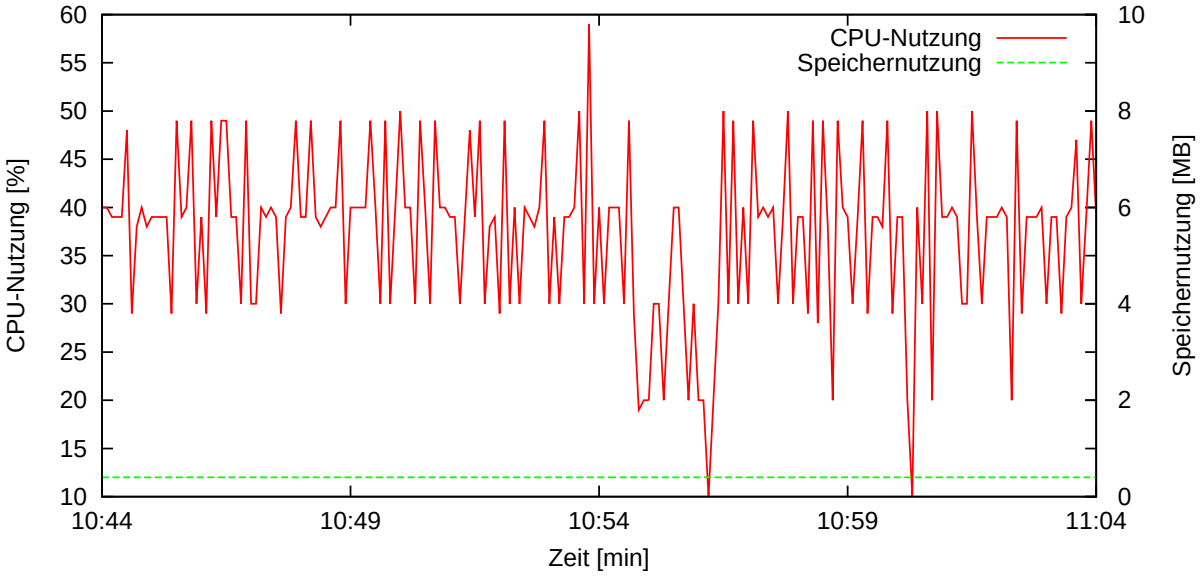
Abbildung 6.3 zeigt einen detaillierteren Ausschnitt der Ressourcenmessung der Datensonde. Eine relativ konstante, durchschnittliche Last von 40 Prozent CPU-Nutzung ist erkennbar und entspricht dem erwarteten Verhalten der Sonde beim Vorliegen eines konstanten Paketstroms auf dem untersuchten Link.

Die Generierung der Daten erzeugt somit eine sehr geringe Systemlast, auch bei hoher Verkehrsaktivität. Somit ist auch unter steigenden Datenraten eine vollständige Verarbeitung der Verbindungen möglich. Die verschiedenen Teilmodule der Ein- und Ausbruchserkennung sind wiederum in extra Threads bzw. Prozesse ausgelagert worden, um eine Parallelisierung des Systems zur ermöglichen. Eine Skalierung durch die Datensonde ist insbesondere auch dadurch gewährleistet, dass sich die statistischen Daten für jede Verbindung einzeln und unabhängig zu den jeweils anderen Verbindungen erheben lassen, somit ist eine Verteilung ohne relevanten Kommunikationsoverhead möglich. Hierdurch können die eingehenden Daten beliebig auf verschiedene Systeme verteilt werden, jedoch sind auch Gbps-Links aufgrund der geringen Last problemlos mittels einer Sonde analysierbar.

Abbildung 6.3 zeigt ein weiteres Beispiel eines voll ausgelasteten 100 Mbps-Links. Zunächst ist keine Kommunikation auf dem Netz vorhanden, nach der mehrminütigen Ruhephase beginnt der Datenverkehr. Die durchschnittliche CPU-Auslastung liegt hierbei bei 44 Prozent, der durchschnittliche Speicherbedarf bei 0.3 MB. Für die Größe der Hashtabellen wurde die Anzahl der zu Beginn vorliegenden Einträge auf 3000 festge-



(a) Nach Initialisierung der Datensonde steigt der Speicherbedarf zunächst an, um die notwendigen Datenstrukturen zu allokieren. Die Nutzung der CPU bleibt auch bei Auslastung des Datenlinks regelmäßig deutlich unter 50 Prozent.



(b) Ausschnitt der Ressourcenmessung. Gut erkennbar ist der relativ konstante Bedarf an Arbeitsspeicher sowie die stabile, durchschnittliche CPU-Nutzung von ca. 40 Prozent unter Lastbedingungen.

Abbildung 6.2: Durchführung von Messungen zur Evaluation der Ressourcenanforderungen der Datensonde.

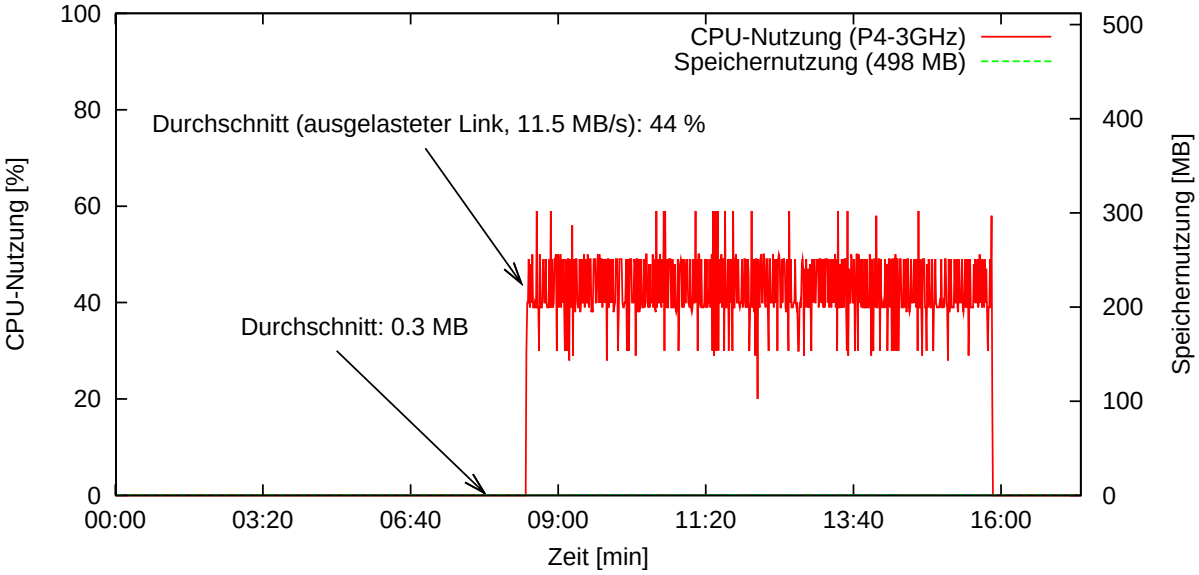


Abbildung 6.3: Ressourcenbedarf der Datensonde unter Lastbedingungen bei Nutzung eines 100 Mbps-Links sowie eines älteren P4-Rechners.

legt[2], was im untersuchten Produktivnetz ausreichend war (vgl. auch Abbildung 6.5b). Steigt der Bedarf an Tabelleneinträgen über eine Belegung von 65 Prozent an, wird die darunter liegende Struktur automatisch vergrößert[3]. Eine effiziente Ausführung ist somit auch auf geringen Hardwarekapazitäten möglich, wobei zusätzlich eine einfache Verteilung erfolgen kann.

6.2 Module zur Einbruchserkennung

Im Rahmen der Einbruchserkennung wurden Module zur Intra- und Intersitzungskorrelation implementiert, welche im weiteren Verlauf erprobt und analysiert werden. Inter-Sitzungskorrelationen werden für die Detektion von Angriffen im Rahmen der verschlüsselten Kommunikation einer Vielzahl von Nutzern mit einem Dienst durchgeführt, bspw. Verbindungen zu einem Webshop, welche über TLS abgesichert werden. Die Intra-Sitzungskorrelation dient der Erkennung von Angriffen innerhalb einer Verbindung, bspw. bei einer Remotesession eines Nutzers mittels SSH. Hierbei liegt noch kein Nutzerzugang vor, d.h. der Nutzer ist noch nicht authentifiziert; ein typischer Angriff dieser Kategorie ist somit die Durchführung eines Brute Force-Angriffes, wobei Angriffe *nach* einer

[2] 3000 entspricht dem initialen Wert im Programm; während des Aufbaus der Hashtabellen wird dieser Wert aus Effizienzgründen des Zugriffs durch die nächsthöhere, intern festgesetzt Primzahl ausgetauscht, falls der initiale Wert nicht selbst prim ist. Die exakte, interne Tabellengröße zum Programmstart entspricht in diesem Falle also 3079.

[3] Die Anzahl der Einträge wird hierbei ungefähr verdoppelt, im Falle von 3079 Einträgen ist die nächste Größe 6151.

erfolgreichen Authentifizierung des Nutzer zum Aufgabenbereich des Moduls zur Befehlsevaluation gehören (vgl. Kapitel 6.3.1).

6.2.1 Schnelle Brute Force-Erkennung

Die schnelle Brute Force-Erkennung dient im Rahmen der Einbruchserkennung der zentralen, netzbasierten Detektion und Verhinderung von Brute Force-Angriffen auf verschlüsselte Dienste in einem zu schützenden Netz. Um das hierfür implementierte Modul zu erproben und auszuwerten, wurden zwei Arten von Evaluationen durchgeführt, ein rein synthetischer Ansatz sowie eine Auswertung in einer produktiven Umgebung. Dieses Vorgehen ist erforderlich, um einerseits die Detektionsleistung des Systems bzgl. der Angriffe messbar festzustellen, andererseits aber auch die Leistungsfähigkeit des Systems in einer produktiven Umgebung zu erproben. Da im Produktivnetz im Allgemeinen nicht mit Bestimmtheit entschieden werden kann, ob sämtliche vorhandenen Angriffe detektiert wurden und somit eine korrekte Bewertung aller Datensätze erfolgt ist (es kann also keine absolute Aussage über die Ground Truth getroffen werden), muss dies mit einem entsprechenden, markierten Datensatz erfolgen.

Auf der anderen Seite ist es schwer möglich, die genauen Gegebenheiten einer Produktivumgebung hinreichend mittels einer Simulation oder der Nutzung aufgezeichneter Daten abzubilden (vgl. Kapitel 4.4). Als weitere Evaluation wird die Datensonde daher in einem Produktivnetz eingesetzt und kontrollierte Angriffe auf ein sich darin befindliches Zielsystem durchgeführt. Hiermit sind diese Angriffe bekannt und können zur Bewertung des Sicherheitssystems herangezogen werden; in diesem Fall kann jedoch nicht ausgeschlossen werden, dass ein während der Durchführung ebenfalls stattfindender, realer Angriff vom System nicht erkannt und für die Bewertung des Systems fälschlicherweise nicht mit einbezogen wird. Diese Gefahr kann jedoch durch eine entsprechende Absicherung des Produktivnetzes und mehrmalige Versuchsdurchführungen minimiert werden.

Um alle Detektionsaspekte der Brute Force-Erkennung zu betrachten (vgl. Kapitel 5.1.5), müssen folgende Abläufe nach dem Aufbau einer Verbindung abgebildet werden:

- Erfolgreiche Authentifizierung (Erwartung: Keine Detektion)
- Wiederholt fehlerhafte Authentifizierungen (Erwartung: Detektion)
- Parallele Verbindungen und Authentifizierungsversuche (Erwartung: Detektion)
- Authentifizierte Sitzung und parallel fehlerhafte Authentifizierungsversuche (Erwartung: Detektion)
- Parallele, authentifizierte Sitzungen (Erwartung: Keine Detektion)

Bewertet werden kann insbesondere, welcher Anteil der Gesamtdaten korrekt klassifiziert wurde, wie hoch die Detektionswahrscheinlichkeit eines durchgeführten Angriffs ist, welches Fehlalarmverhältnis sich einstellt und wie hoch die Fehlalarmrate ist (vgl. Kapitel 4.4).

Tabelle 6.1: Verschlüsselungsalgorithmen bei SSH, Protokollversion 2.

Algorithmus
AES128-CTR
AES192-CTR
AES256-CTR
ARCFOUR256
ARCFOUR128
AES128-CBC
3DES-CBC
BLOWFISH-CBC
CAST128-CBC
AES192-CBC
AES256-CBC
ARCFOUR

Für die Durchführung der Verbindungsversuche im Sinne eines Angreifers wurden zum einen manuelle Eingaben genutzt, zum anderen spezielle Tools zur Durchführung von Brute Force-Angriffen wie bspw. `brutessh` und `sshatter`. Bei den manuellen Eingaben wurden insbesondere verschiedene Verschlüsselungsalgorithmen genutzt, um deren Einfluss auf die Detektion zu prüfen. Tabelle 6.1 zeigt die für das SSH-Protokoll Version 2 verfügbaren Verschlüsselungsalgorithmen. Eine Auswahl des zu nutzenden Algorithmus kann einfach mittels einer Parameterübergabe erfolgen, bspw. für die Nutzung von Blowfish bei einer Verbindung zu einem Server mit der IP-Adresse `192.168.1.10` mittels nachfolgendem Aufruf:

```
$ ssh -o Ciphers=blowfish-cbc 192.168.1.10
```

Die verschiedenen Programme für die Durchführung von Brute Force-Angriffen implementieren verschiedene Strategien bzw. sind typischerweise mittels Kommandozeilenparameter konfigurierbar, bspw. in der Anzahl paralleler Verbindungen. Abbildung 6.4 stellt einen Durchlauf der Evaluation mittels synthetisch erzeugter Datenströme dar. Die gutartigen Verbindungen wurden hierbei durch mehrere Skripte dargestellt, welche Sitzungen auf- und abbauen und verschiedene Aktionen durchführen. Der Verlauf der Anzahl von Sitzungen über eine Testdauer von ca. 35 Minuten ist anhand der Abbildung ersichtlich. Während des Tests wurden wiederholt Angriffe eingespielt, Alarmmeldungen sind entsprechend ihrem Auftretenszeitpunkt eingetragen. Hierbei ist zwischen korrekten Alarmen auf Basis erkannter Angriffe (*True Positives*) und Fehlalarmen auf Basis einer fehlerhaften Bewertung gutartiger Verbindungen (*False Positives*) zu unterscheiden. Die richtigen Bewertungen gutartiger Verbindungen (*True Negatives*) sind zur Übersichtlichkeit nicht in der Grafik dargestellt; diese werden nicht durch das Sicherheitssystem gemeldet, um einer unerwünschten Informationsflut vorzubeugen, sondern werden intern nach gutartiger Bewertung nicht weiter untersucht und bei Abschluss der Verbindung

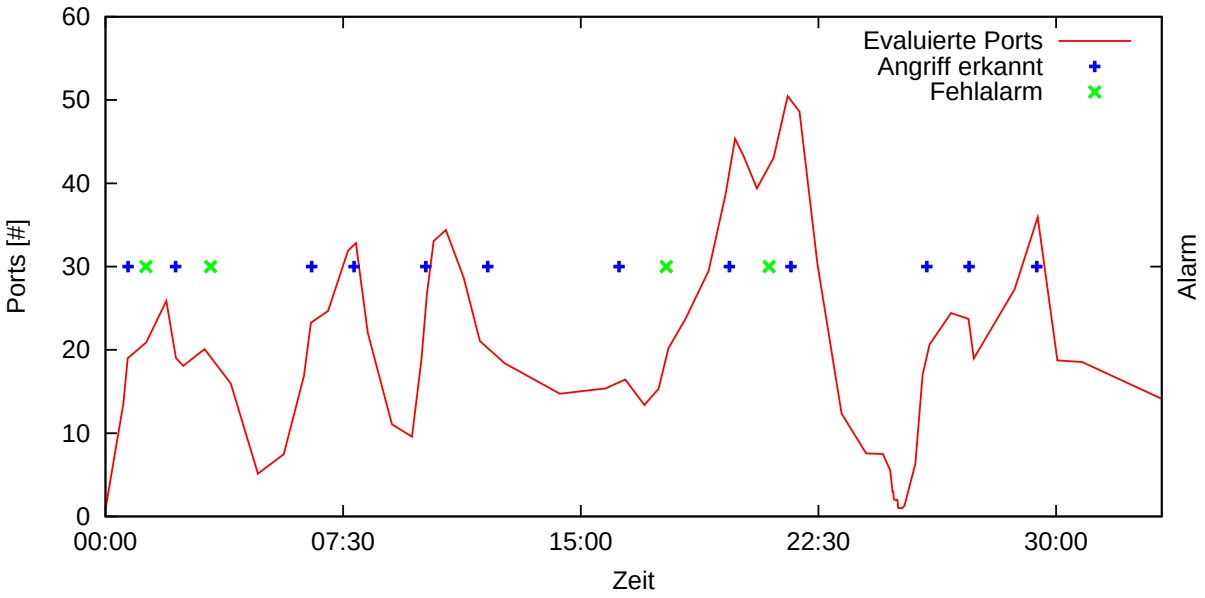


Abbildung 6.4: Brute Force-Erkennung in synthetisch erzeugten Verbindungen. Die Darstellung von guten Nutzersitzungen wurde mittels skriptbasierten Sitzungen sowie manuell eingespielt, Angriffe wurden auf Skriptbasis unter Nutzung von Brute Force-Tools und manuell durchgeführt.

aus den jeweiligen Tabellen gelöscht. Aufgrund der künstlichen Zusammenstellung des gut- und bösartigen Datenverkehrs sind sämtliche Angriffe bekannt, daher kann hier auch der Anteil an fälschlich als gut bewerteten, bösartigen Verbindungen (*False Negatives*) angegeben werden. In der entsprechenden Kategorie wurden über alle Testläufe hin keine Kandidaten festgestellt. Über die Testläufe hinweg gemittelt ergibt sich eine Detektionsrate von 98.68 Prozent, die Fehlalarmrate liegt bei 0.84 Prozent.

Nach der Evaluation mittels der synthetischen Daten wurden die Versuche in einem Produktivnetz wiederholt. Hierbei wurden zunächst *alle* vorhandenen Verbindungen durch das Sicherheitssystem ausgewertet, unabhängig des zugrunde liegenden Ports bzw. Dienstes. Somit sind hier bspw. auch unverschlüsselte Verbindungen betrachtet und bewertet. Abbildung 6.5 zeigt das Ergebnis einer Auswertung, wobei kontrollierte Angriffe auf einen hierfür temporär im Produktivnetz abgestellten Server durchgeführt wurden bzw. in einer weiteren Variante die Test- und Evaluationsumgebung genutzt wurde (vgl. Kapitel F.4.2), um auf Basis der Produktivdaten in einer gesicherten Testumgebung Angriffe durchzuführen. Die Erkennung eines Angriffes zeigt sich wie im Detail der Abbildung 6.5a dargestellt. Im Gegensatz zu den Messdurchführungen in der Simulationsumgebung, bei der die identifizierten Adressen bösartiger Verbindungen auch direkt blockiert werden können, wurden in der Evaluation in der Produktivumgebung keine Adressen real gesperrt, sondern das Firewall-Regelwerk eines Rechners hinter der Datendiode manipuliert (vgl. Kapitel F.4.1). Somit kann die Sperrung der detektierten IP durch das prototypische Sicherheitssystem simuliert ohne Beeinflussung des Produk-

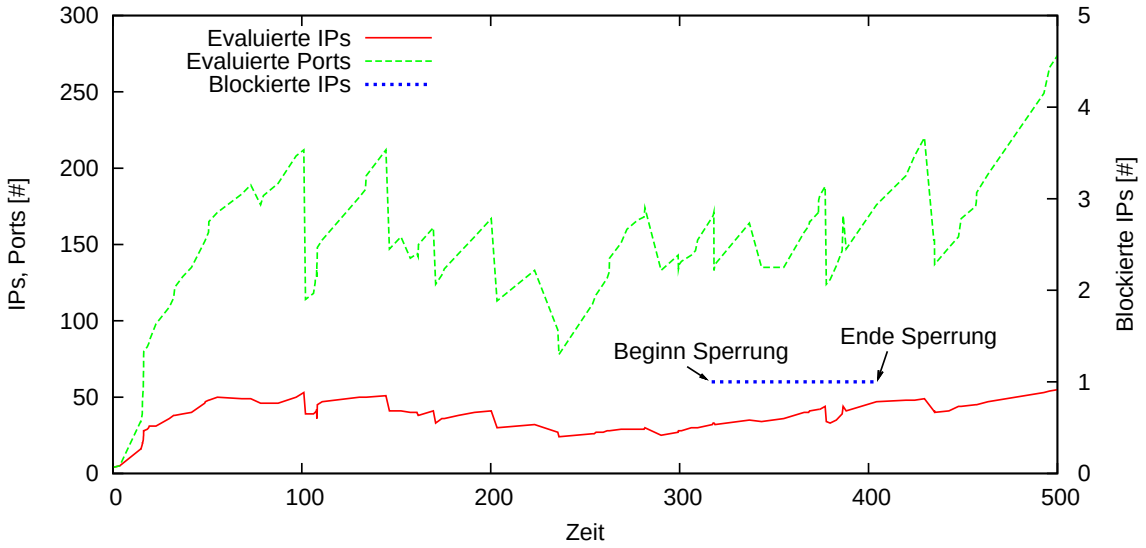
tivnetzes erfolgen und einfach asugewertet werden. Die Dauer der Sperrung einer IP wird im Prototypen durch einen konfigurierbaren Wert festgelegt, welcher für die Versuche auf 60 Sekunden festgesetzt wurde. Entsprechend ist aus Abbildung 6.5 der Beginn der Sperrung einer detektierten, bösartigen IP nach ca. 320 Sekunden festzustellen, nach 60 Sekunden wird die IP-Adresse wieder freigegeben.

Für die Darstellung eines längeren Auswertungszeitraumes sind zur Übersichtlichkeit lediglich die Startpunkte der Sperrung einer IP-Adresse angegeben (vgl. Abbildung 6.5b). Betrachtet man den Umfang der im analysierten Zeitraum gesehenen und ausgewerteten IP-Adressen, sind Datenverbindungen zwischen durchschnittlich 56 Kommunikationspartnern vorhanden, wobei im Schnitt 190 Ports genutzt wurden. Die Sperrungen von Adressen aufgrund der Detektion von Angriffen ist aus dem Graphen ersichtlich. Da diese Auswertungen in realen Umgebungen durchgeführt wurden, ist keine sichere Aussage bzgl. den *False Negatives* möglich; läuft ein Angriff unbemerkt im Netz ab und kann dieser weder vom Sicherheitssystem noch durch dritte Maßnahmen erkannt werden, geht dies entsprechend falsch in die Statistik ein.

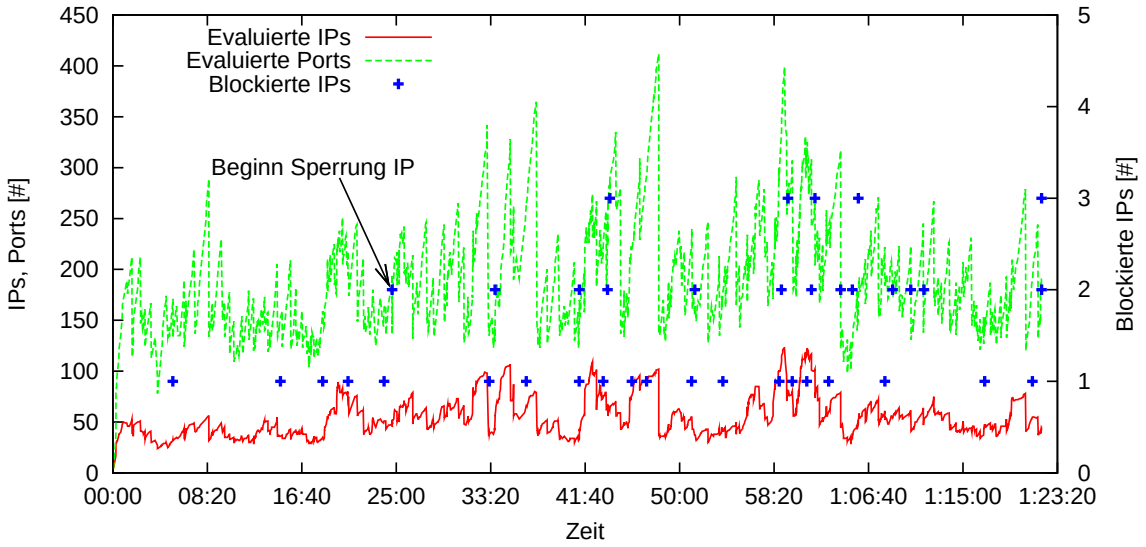
Für die Durchführung der Messungen wurden wie bereits im synthetischen Fall auf Basis von Tools und Skripten Angriffe eingespielt, da deren (Nicht-)Detektion direkt ausgewertet und bewertet werden kann. Um eine Klassifizierung für weitere, nicht durch die eigenen Angriffe aufgetretenen Alarme vornehmen zu können, wurden entsprechende IP-Adressen, welche als bösartig identifiziert wurden, sowie deren Kommunikationspartner betrachtet. Während mehrerer Versuchsdurchführungen wurde eine durchschnittliche Wahrscheinlichkeit einer korrekten Klassifizierung von 99.53 Prozent festgestellt. Somit wird nur ein sehr geringer Anteil der Verbindungen nicht richtig bewertet. Betrachtet man das durchschnittliches Fehlalarmverhältnis, ergibt sich hierfür zunächst ein Wert von 0.5179. Dies bedeutet, dass bei einer Auslösung eines Alarms in knapp 48 Prozent der Fälle auch tatsächlich ein Angriff vorliegt. Dieses Verhältnis beruht insbesondere auf der Auslösung von Fehlalarmen; betrachtet man die Ergebnisse des Sicherheitssystems genauer stellt man fest, dass Fehlalarme im Sinne von *False Positives* nahezu ausschließlich für Verbindungen erzeugt werden, welche *keine* verschlüsselten Sitzungen repräsentieren. Nimmt man die unverschlüsselten Verbindungen aus der Evaluation aus, sind kaum Fehlalarme detektierbar und das Fehlalarmverhältnis sinkt auf nahezu 0 ab. Eine Einschränkung der Auswertung auf entsprechende Ports (insbesondere Port 22 für SSH) bzw. die Nutzung einer Protokollerkennung zur Identifizierung des zu analysierenden Datenverkehrs kann somit die Leistungsfähigkeit des Systems deutlich erhöhen. Eine Protokollerkennung kann in diesem Kontext bspw. als zusätzliches Modul in das Sicherheitssystem integriert werden, hierbei können verschiedenen Verfahren aus der Literatur aufgegriffen und genutzt werden (vgl z.B. [55, 285, 392]).

Die Detektion der unterschiedlichen, zu berücksichtigenden Fälle bei der Untersuchung von Authentifizierungsversuchen ist in Tabelle 6.2 ersichtlich. Alle identifizierten Konstellationen wurden erprobt und korrekt behandelt.

Tabelle 6.3 fasst die Ergebnisse sowohl der synthetischen Evaluation als auch der Evaluation im produktiven Netz zusammen. Zu erkennen ist, dass die Ergebnisse der synthetischen Evaluation leicht geringer als die Ergebnisse der Evaluation im Produktivnetz sind. Dies lässt sich dadurch erklären, dass im Produktivnetz sämtlicher Datenverkehr



(a) Detektion einer bösartigen IP-Adresse und deren Blockade für 60 Sekunden.



(b) Detektion von Angriffen über einen längeren Zeitraum und wiederholter Durchführung von Angriffen. Zur Übersichtlichkeit wurden lediglich die Startzeitpunkte der Sperrung, jedoch nicht die Endpunkte eingezeichnet.

Abbildung 6.5: Durchführung von Messungen zur Brute Force-Angriffsdetektion. Beschriftet ist der evaluierte Zeitraum von 1 Stunde 23 Minuten, nicht die tatsächliche Uhrzeit (Beginn um 09:18:13 Uhr am 24.01.11).

Tabelle 6.2: Detektionsarten des Moduls für schnelle Brute Force-Erkennung. Alle Fälle wurden korrekt durch das Modul umgesetzt, bei rot markierten Symbolen erfolgt eine Maßnahme (Sperrung), bei grün markierten Fällen erfolgt keine Aktion.

Detektionsart	
Erfolgreiche Authentifizierung	✓
Wiederholt fehlerhafte Authentifizierungen	✓
Parallele Verbindungen und Authentifizierungsversuche	✓
Authentifizierte Sitzung und parallel fehlerhafte Authentifizierungsversuche	✓
Parallele, authentifizierte Sitzungen	✓

Tabelle 6.3: Detektionsraten des Moduls für schnelle Brute Force-Erkennung der Einbruchsdetektion. Die Klassifizierung gibt den Anteil aller korrekt bewerteten Verbindungen an, die Detektionswahrscheinlichkeit gibt die Wahrscheinlichkeit an, dass ein realer Angriff erkannt und korrekt klassifiziert wird, die Fehlalarmrate gibt die Wahrscheinlichkeit einer falschen Detektion an. Das Fehlalarmverhältnis gibt an, welcher Anteil der Alarme fehlerhaft war bzw. die Meldung eines echten Angriffs. 4 Prozent bedeutet hier, dass 96 Prozent der Alarme aufgrund von echten Angriffen enstanden sind. Die Werte in Klammern geben die im Falle der Berücksichtigung der Ports resultierenden Ergebnisse an (Angaben in Prozent).

	Klassifizierung	Detektionswahr.	Fehlalarmverhältnis	Fehlalarmrate
Synthetisch	98.68	92.0	4.0	0.84
Produktivnetz	99.53	98.41	51.79 (1.59)	0.48
Gesamt	99.11	95.21	27.90 (2.80)	0.66

berücksichtigt wurde und eine deutlich höhere Datenmenge bzw. Verbindungszahl vorliegt, als im synthetischen Fall, der nur verschlüsselte Verbindungen darstellt. Ebenso liegt im synthetischen Fall ein deutlich höherer Anteil von Angriffen bezogen auf alle Verbindungen vor.

Gut zu erkennen ist, dass in allen Fällen eine sehr hohe korrekte Klassifizierung erreicht wird. Die Detektionswahrscheinlichkeiten für einen realen Angriff sind ebenfalls sehr hoch; fehlerhafte Alarme im Sinne von als bösartig bewertetem, gutartigen Datenverkehr treten insbesondere bei der kompletten Betrachtung des Datenverkehrs auf verschiedenen, nicht-verschlüsselten Verbindungen auf. Nimmt man diese aus der Evaluation aus, ist nur noch ein sehr geringer Anteil der Alarme fehlerhafter Natur. Die Brute Force-Erkennung eignet sich somit für eine schnelle, zentralisierte Angriffsdetektion und -verhinderung zum Schutz aller im Netz vorhandenen entsprechender Dienste, ohne dass Konfigurationen erforderlich sind.

6.2.2 TLS-Angriffsdetektion

Um eine Evaluation der Angriffserkennung auf SSL/TLS-Verbindungen durchführen zu können, jedoch ohne einen produktiven Webserver zu gefährden, muss eine andere Ver-

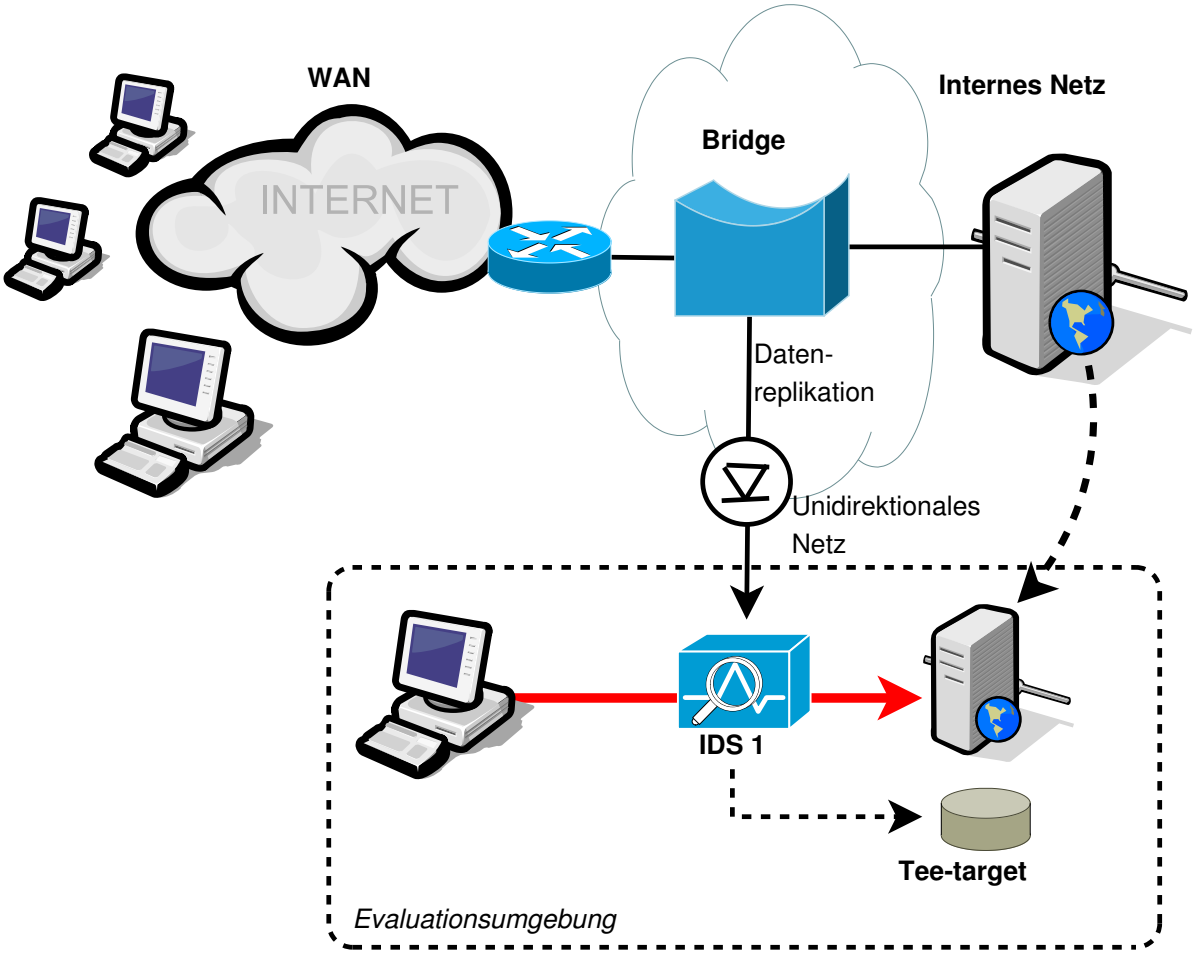


Abbildung 6.6: Aufbau für die Evaluation des Sicherheitssystems mit Daten des Produktivnetzes in einer abgeschotteten Umgebung.

fahrensweise gewählt werden. Prinzipiell lässt sich auch hier die vorgestellte Evaluationsumgebung nutzen, indem der reale Datenverkehr zwischen einem Server und den jeweiligen Benutzern mittels des TEE-Targets in ein Evaluationsnetz kopiert wird. Das Einspielen der Angriffe erfolgt dann im Evaluationsnetz auf einem dort installierten, gleichartigen Server. Abbildung 6.6 zeigt einen entsprechenden Messaufbau.

Dieses Verfahren ermöglicht die Nutzung der realen, im Produktivnetz vorliegenden Daten für die Evaluation des Sicherheitssystems, während die Angriffe kontrolliert in der abgeschotteten Evaluationsumgebung durchgeführt und ausgewertet werden können. Hierbei muss jedoch beachtet werden, dass ein reines Kopieren und Weiterleiten des Datenverkehrs mittels des TEE-Targets (vgl. Anhang F.4.2) hier *nicht* ausreicht: Wird der Verkehr auf diese Weise weitergeleitet, bleiben die Adressen der Pakete unverändert, insbesondere die Zieladressen. Da die NICs der Netzbrücke eindeutig sein müssen, und vor allem nicht in *zwei verschiedene* angeschlossene Subnetze (Internes Produktivnetz und Evaluationsnetz) mit *gleichem* Adressraum weitergeleitet werden kann, muss der Klon

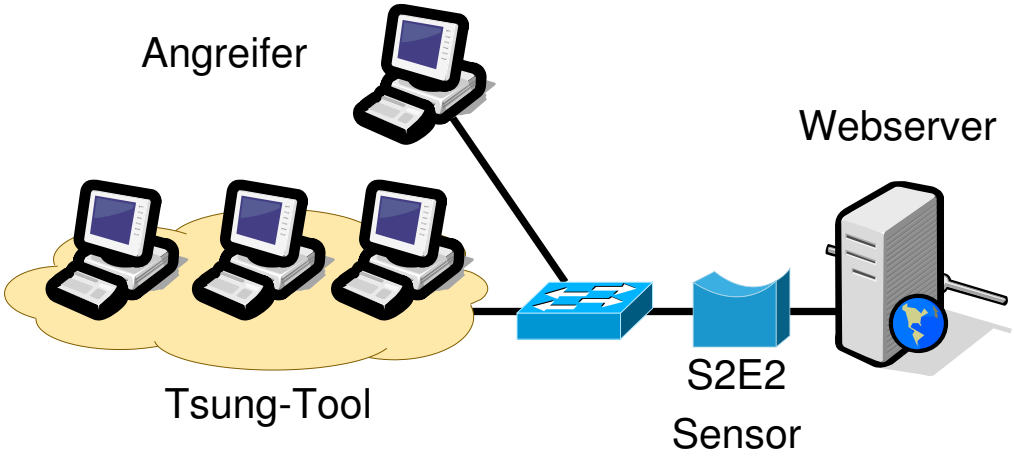


Abbildung 6.7: Evaluationsumgebung zur Auswertung der Leistungsfähigkeit der SSL/TLS-Angriffsdetektion. Die Sitzungen der Nutzer werden mittels des Benchmarking-Tools Tsung erzeugt, Angriffe werden von einem zusätzlichen System eingespielt.

des Webservers im Evaluationsnetz mit einer anderen IP-Adresse ausgestattet werden. Da der kopierte und in das Evaluationsnetz weitergeleitete Datenverkehr nun nicht mehr an die Zieladresse angelangen kann, werden die Pakete nach Ankunft am TEE-Target verworfen. Um dies zu verhindern und den Datenverkehr dem Klon zuzuführen, müssen die Zieladressen der Pakete angepasst werden, was mit einer weiteren iptables-Regel erfolgen kann:

```
$ iptables -A PREROUTING -t nat -p tcp -d 137.193.63.100 --dport 443 -j
    DNAT --to 192.168.1.10:443
```

Auf diese Weise ist eine Evaluation und Leistungsanalyse mit Produktivdaten in einer sicheren, abgeschotteten Umgebung möglich. Weiterhin ist es aber ebenfalls schwer, auf einem Webserver im Netz bspw. der Fakultät ausreichend reale Last zu erzeugen, bzw. eine erforderliche Anzahl von Nutzern für die Messungen zu finden und zu koordinieren, da hier typischerweise eine sehr viel geringere Zahl von Sitzungen vorliegt, als dies bspw. bei einem Webshop im Internet der Fall ist. Da das vorgestellte Verfahren zur Identifikation von bösartigen Verbindungen maßgeblich auf der Auswertung des Verhaltens von Nutzergruppen basiert, muss daher eine ausreichende Zahl von Nutzern für die Evaluation vorhanden sein.

Um dennoch eine geeignete Evaluation durchführen zu können, wurde daher eine Umgebung zur Analyse gem. Abbildung 6.7 aufgebaut. Der Systemaufbau besteht maßgeblich aus vier Komponenten, die nachfolgend kurz beschrieben sind:

- Netzsonde zum Abhören des Datenverkehrs und Durchführung der SSL/ TLS-Angriffsdetektion (S2E2)
- Tsung-Benchmarking-Tool zur Erzeugung der Nutzerlast

- Tsung-Benchmarking-Tool zur Erzeugung von Angriffen
- Angreifbarer Webshop „BadStore“

Die Netzsonde des Sicherheitssystems S2E2 wird wie bereits zuvor wieder als transparente Brücke in die Netzverbindung zwischen dem Webserver und den Nutzern integriert. Um einen entsprechenden Dienst zur Verfügung zu stellen, der angreifbare und somit beobachtbare Schwachstellen aufweist, wurde ein Image des Webshops *BadStore* als Webserver aufgesetzt, der bewusst Angriffspunkte implementiert. Für die Darstellung der erforderlichen Nutzer wurde das Benchmarking-Tool *Tsung* herangezogen. Dieses zeichnet sich dadurch aus, dass es eine hohe Zahl von Nutzern simulieren kann, um die Skalierbarkeit und Leistungsfähigkeit von IP-basierten Client-Server-Applikationen zu testen. Es ist protokollunabhängig in der Programmiersprache Erlang[4] implementiert und unterstützt derzeit u.a. HTTP und SSL. Die Konfiguration von Tsung erfolgt anhand einer Extensible Markup Language (XML)-Datei, in welcher die Aktionen der zu simulierenden Nutzer detailliert festgelegt werden können. Hierbei können sowohl die maximale Anzahl von gleichzeitigen Nutzern, als auch das Auftretensintervall (periodisch oder stochastisch) und der Pool genutzter IP-Adressen angegeben werden. Die Nutzung unterschiedlicher Adressen macht es insbesondere möglich, eine einfache Unterscheidung zwischen normalen Nutzern und Angreifern für die Verifikation der Evaluationsergebnisse des Systems vorzunehmen: Während der Datenverkehr legitimer Nutzer auf den Adressbereich (letztes Oktett) von `100` bis `199` verteilt wird, wird den Angreifern der Bereich von `30` bis `99` zugewiesen. Somit kann die Evaluation des Systems, ob eine Verbindung einen Angriff repräsentiert oder nicht, einfach anhand der Einstufung der jeweiligen IP-Adresse erfolgen. Eine Beeinflussung oder Verfälschung der Evaluationsergebnisse des Systems durch die feste Zuweisung von IP-Bereichen für gutes bzw. bösartiges Verhalten erfolgt hierbei jedoch nicht[5], da diese Adressen während der Verarbeitung *nicht* genutzt bzw. einbezogen werden.

Die konfigurierten IP-Adressen müssen im System vorhanden sein, daher bietet sich deren Bereitstellung als virtuelle NICs an, da eine entsprechend hohe, erforderliche Anzahl nur schwer durch reale Interfaces erfolgen kann:

```
#!/bin/sh
for (( i=$START; i<=$END; i++ ))
do
ifconfig eth0:$i 192.168.1.$i netmask 255.255.255.0
done
```

Neben dem Auftreten der Nutzer können auch deren Aktionen detailliert in den Tsung-Konfigurationsdateien bestimmt werden. Hierzu können Warte- bzw. Bedenkzeiten konfiguriert und verschiedene Sitzungen und Transaktionen festgelegt werden, die mit angegebenen Wahrscheinlichkeiten während der Simulationsausführung aufgerufen werden. Bei den Aktionen können bspw. auch POST-Werte übertragen werden, so dass das Einloggen von bekannten Nutzern, Such- und Bestellaktionen, etc. einfach erzeugt und

[4] Erlang und Erlang Open Transaction Platform (OTP), siehe [384].
[5] Vgl. Kapitel 5.1.4.

durchgeführt werden können. Zwei Rechner mit jeweiligen Tsung-Instanzen wurden für die Evaluation genutzt, ein Rechner zur Simulation der legitimen Webshop-Nutzer, das andere System für die Durchführung der Angriffe.

Das Detektionssystem wurde anhand der Einspielung zwei verschiedener Angriffsarten getestet: Zum einen wurde ein Wörterbuchangriff auf den Nutzer-Login des Webshops gefahren, bei dem verschiedene Kombinationen von Nutzernamen und Passwörtern systematisch ausprobiert werden, um Zugriff zu einem vorhandenen Nutzerkonto zu erhalten. Als zweiter Angriff wurde eine SQL-Injection durchgeführt. Hierbei handelt es sich um die Ausnutzung einer fehlenden oder fehlerhaften Überprüfung der Nutzereingabe, bevor diese an die Datenbank weitergegeben wird. Dies ermöglicht es einem Angreifer, Metazeichen in die Anfrage mit einzubauen und somit Befehle in die Datenbank einzuschleusen, die einem Außenstehenden sonst nicht zur Verfügung stehen. Auf diese Weise können bspw. Inhalte der Datenbank ausgelesen werden. Trotz des Bekanntheitsgrades der hierdurch entstehenden Gefährdung und den in der Vergangenheit zahlreichen, negativen Pressemeldungen über entsprechende Vorfälle ist dies auch heute noch ein sehr bedeutender und wiederholt erfolgreich ausgenutzter Angriffsvektor.

Für die Analyse der Verbindungen wurde jede neu eingehende Verbindung mit jeweils 12 anderen, zufällig ausgewählten Verbindungen anhand der Paketgrößen der Paketserien von Server und Client korreliert. Da das Korrelationsergebnis normiert wurde, ergibt sich für jede Verbindung ein Wert im Bereich $[0..1]$, welcher die Ähnlichkeit des Datenverkehrs in Bezug auf den weiter vorhandenen Datenverkehr des Servers ausdrückt. Je höher der Wert, umso ähnlicher sind die beobachtbaren Parameter (insbesondere Timing und Paketgrößen, wobei die TLS-Evaluation maßgeblich auf den Paketgrößen beruht) der Datenverbindung zu den weiteren Verbindungen. Für die Einteilung einer Verbindung in eine der zwei grundlegenden Klassen *„gutartig“* und *„bösartig“* muss eine entsprechende Teilung erfolgen; hierfür bestehen folgende zwei Möglichkeiten:

- Festlegung *einer* Grenze im Wertebereich, die zwischen *gutartigen* und *bösartigen* Verbindungen trennt, oder
- Nutzung *zweier* Grenzen, welche den Wertebereich in ein *gutartiges*, ein *bösartiges* sowie ein *unbestimmtes* Intervall unterteilen. Die Aufgabe des unbestimmten Intervalls ist hierbei insbesondere, Verbindungen, für deren Evaluation noch nicht hinreichend Daten vorliegen und somit keine eindeutige Entscheidung getroffen werden kann, für eine weitere Evaluation vorzuhalten.

Entsprechend muss eine bzw. zwei Grenzen gewählt werden, um die erforderlichen Intervalle einzuteilen: Eine entsprechende Festlegung kann zunächst zufällig oder logisch (bspw. die Einteilung in gleich große Teilabschnitte) erfolgen. Durch Beobachtungen der Detektionsresultate können die jeweiligen Parameter nachfolgend empirisch optimiert werden. Wurde der Korrelations-Wertebereich in *drei* Intervalle unterteilt, gilt für den Korrelationswert s als Ausgang:

- $s \in [0..0.4]$: Bösartige Verbindung erkannt.
- $s \in]0.4..0.6]$: Verbindung unter Beobachtung, weitere Korrelationen.

- $s \in]0.6..1]$: Gutartige Verbindung erkannt.

Für die Unterteilung in *zwei* Intervalle wird die Grenze auf 0.4 festgelegt; diese ist - im Vergleich zur zunächst naheliegenden Grenze von 0.5 - zum einen dadurch motiviert, dass das Angriffsniveau im vorliegenden Szenario *deutlich* geringer anzusetzen ist, als das gutartige Normalverhalten, zum anderen wurde diese durch die empirische Evaluation bestätigt.

Die Evaluation der einzelnen Verbindungen beruht maßgeblich auf den Charakteristika der Nutzung eines jeweiligen Dienstes; bspw. sind die Datenmengen im Download bei einer typischen Websitzung deutlich höher, als die des Uploads. Wird im Vergleich dazu ein Angriff betrachtet, ändern sich die jeweils übertragenen Datenmengen oft erheblich hinsichtlich der Paketgrößen und/oder des Timings. Abbildungen 6.8 und 6.9 zeigen den beispielhaften Verlauf der gleichzeitigen Nutzer und übertragenen Datenmengen einer Sitzung von knapp 200 Sekunden. Maximal sind hierbei 62 gutartige Nutzer auf den Seiten des Webshops aktiv. Die übertragenen Datenmengen zeigen den typischen Verlauf von Websitzungen, bei denen die zum Server übermittelten Datenmengen deutlich geringer sind, als die Datenmengen die vom Server zu den einzelnen Nutzern transferiert werden.

Für eine umfassende Evaluation des Moduls wurden im weiteren Verlauf Verbindungszahlen zwischen 50 und 600 Nutzern simuliert; für die Auswertung des Systems ist jedoch nur das *Verhältnis* zwischen gut- und bösartigen Verbindungen von Bedeutung, weshalb im weiteren Verlauf nur diese Entwicklung dargestellt wird. Nachfolgend werden die Auswertungen mehrerer Sitzungen vorgestellt.

Bei der Durchführung wurden zunächst verschiedene Sitzungselemente und Aktionen für Nutzer des Webshops generiert, bspw. das Anlegen eines neuen Nutzerkontos, das Ein- und Ausloggen, zufällige Bestellvorgänge, das Aufrufen verschiedener Angebote, die Nutzung des Gästebuchs, etc. Verschiedene Sitzungen und Aktionen werden mit bestimmten Wahrscheinlichkeiten und zufälligen Wartezeiten durchgeführt, um eine umfassende Nutzung des Webshops darzustellen. In einer weiteren Instanz des Benchmarking-Tools wurde das Angriffsverhalten definiert, wobei hier ebenfalls typische Schritte eines gutartigen Nutzers mit aufgenommen wurden, bspw. ein Aufrufen des Warenangebotes. Zusätzlich wurden hier die beiden Angriffe Brute-Force-Nutzerlogin und SQL-Injection integriert.

Für die Evaluation der Verbindungen wird jede neue Verbindung mit 12 anderen, zufällig ausgewählten Verbindungen korreliert. Sind noch nicht genug Verbindungen vorhanden, wartet das System auf die Eröffnung neuer Verbindungen, jedoch lässt sich die Evaluation nach einer festgelegten Wartezeit Δ_t auch mit einer geringeren Zahl von Verbindungen starten; die durch einen Parameter festgesetzten n letzten Verbindungen werden durch das Modul zwischengespeichert, um zum einen eine schnelle Auswertung auch im Falle von wenigen aktuellen Verbindungen zu ermöglichen, andererseits ermöglicht eine Anpassung dieses Parameters eine Einstellung der Sensitivität des Systems hinsichtlich sich änderndem Nutzerverhalten: Umso weniger Verbindungen aus der Vergangenheit gespeichert und für die Korrelationen genutzt werden, umso weniger stark wirken sich Abweichungen des aktuellen Nutzerverhaltens aus.

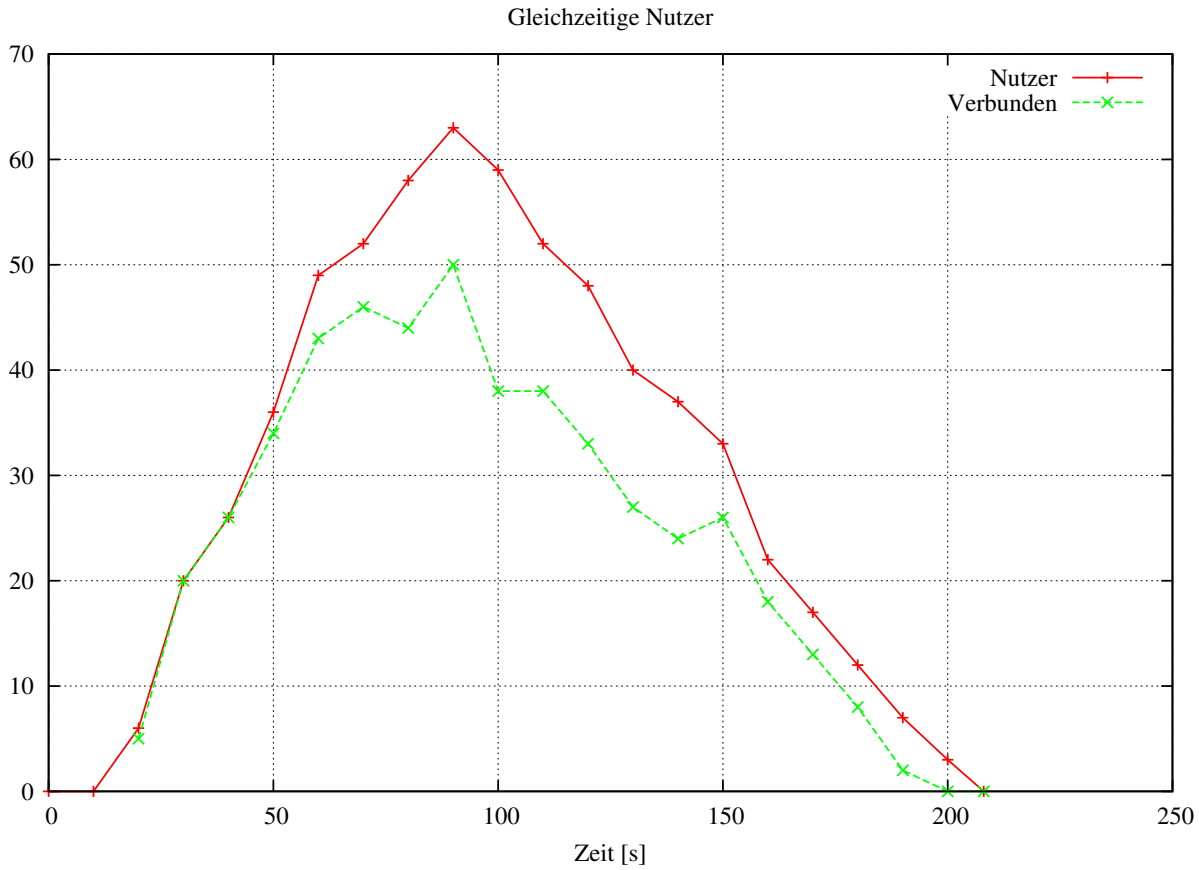


Abbildung 6.8: Verlauf der Anzahl simulierter, gleichzeitig auf den Webshop zugreifender Nutzer.

Tabelle 6.4 zeigt das Evaluationsergebnis der Überwachung eines Webservers mit zahlreichen Verbindungen. Aufgrund der Übersichtlichkeit wurde die Matrix stark gekürzt und zeigt nur die ersten neun der insgesamt aufgetretenen Verbindungsadressen an. Diese sind aufsteigend nach IP-Adresse sortiert, wodurch erreicht wird, dass die Verbindungen der Angreifer immer an erster Stelle aufgeführt werden, da diese im Adressbereich von `30` bis `99` liegen. Da weiterhin die Anzahl der bösartigen Verbindungen im vorliegenden Szenario typischerweise gering im Vergleich zur Anzahl der Gesamtverbindungen ist, sind mehrere gutartige Verbindungen als Vergleichsoption ersichtlich.

In der Tabelle sind die letzten Oktetts von zwei IP-Adressen (`35` und `36`) zu erkennen, welche Verbindungen von Angreifern repräsentieren. Die Adressen von `122` bis `136` gehören zu gutartigen Sitzungen, entsprechen also normalem bzw. dem gewünschten Nutzerverhalten. Aus Platzgründen wurden weitere Verbindungen nicht in die Darstellung mit aufgenommen. Die vorletzte Spalte der Tabelle gibt die jeweilige Summe aller berechneten Korrelationen für die jeweilige Verbindung an, die letzte Spalte den ermittelten, durchschnittlichen Korrelationswert einer Verbindung mit Hinblick auf andere Verbindungen. Anhand der jeweiligen Höhe der Korrelation und dem somit zugehören-

Tabelle 6.4: Ausschnitt der Korrelationsmatrix von TLS-Verbindungen zu einem Server. Dargestellt ist jeweils das letzte Oktett der IP-Adresse, wobei alle Adressen ab `100` gutartiger Natur sind, Angreifer haben zur leichteren Identifizierung Adressen zwischen `30` und `99`. Rote Ergebnisse zeigen Angriffsmeldungen, grüne Ergebnisse als gutartig ausgewertete Verbindungen, orange Werte werden weiter analysiert.

	IP-Adresse (letztes Oktett)										
	35	36	122	130	131	132	133	134	136	Sum.	Avg.
35	0.000	0.000	0.000	0.000	0.373	0.000	0.337	0.000	0.030	**5.886**	**0.218**
36	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**8.676**	**0.280**
122	0.000	0.000	0.000	0.999	0.000	0.789	0.530	0.000	0.000	**10.880**	**0.604**
130	0.000	0.000	0.999	0.000	0.000	0.000	0.476	0.000	0.000	**13.832**	**0.629**
131	0.373	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	**14.461**	**0.556**
132	0.000	0.000	0.789	0.000	0.000	0.000	0.000	0.000	0.000	**8.038**	**0.574**
133	0.337	0.000	0.530	0.476	0.000	0.000	0.000	0.000	0.000	**11.764**	**0.560**
134	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**12.050**	**0.709**
136	0.030	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	**11.524**	**0.640**

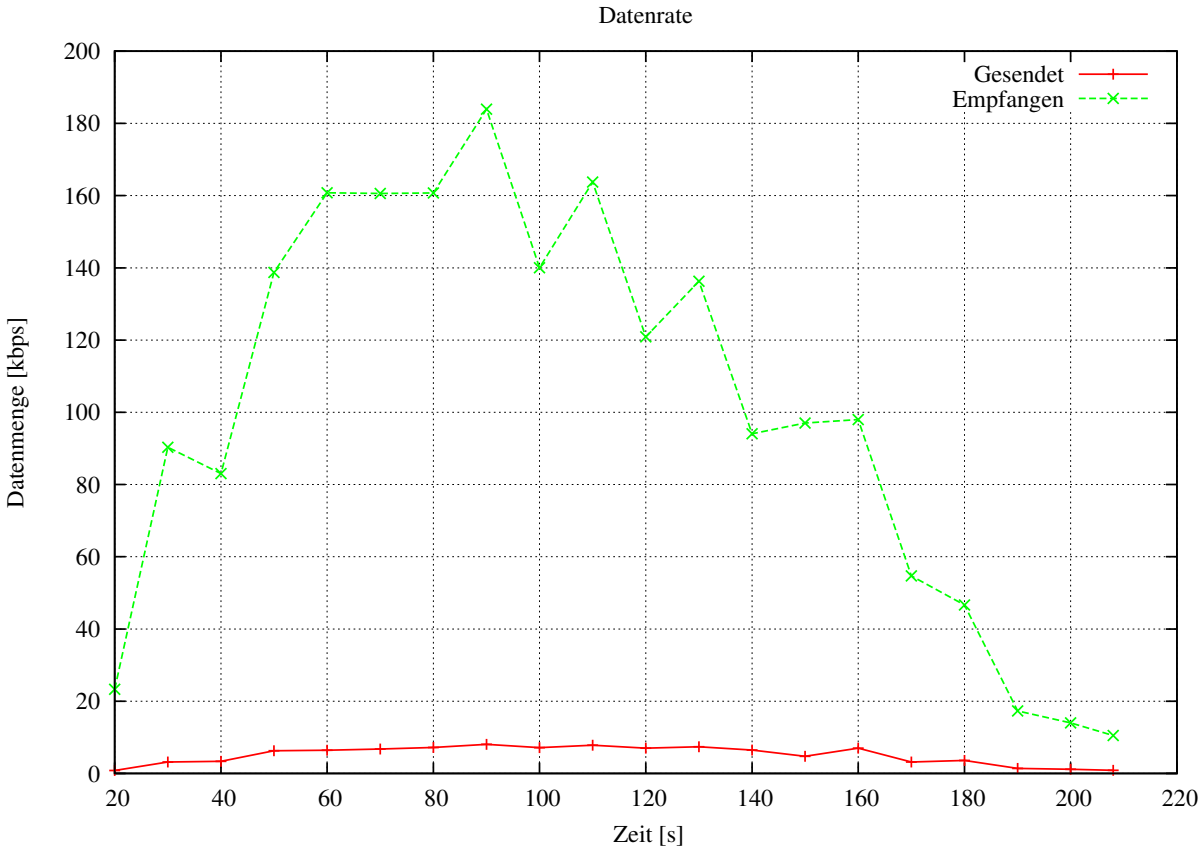


Abbildung 6.9: Verlauf der zwischen Webshop und Nutzern übertragenen Datenmengen bei einem Simulationslauf mittels des Tsung-Benchmark-Tools.

den Intervall ([0..0.4],]0.4..0.6],]0.6..1]) wurden die Ergebnisse mit einer repräsentativen Farbe eingefärbt. Rot markierte Durchschnittswerte sind somit vom Detektionssystem gemeldete, erkannte Angriffe. Wie gut erkennbar ist, sind beide Angreiferadressen `35` und `36` aufgrund ihrer geringen Korrelationswerte zu weiteren Verbindungen vom System als bösartig erkannt und gemeldet. Im weiteren Verlauf der Evaluation wird das Systemverhalten bzgl. einer steigenden Anzahl von Angriffen *(Angriffsniveau)* evaluiert, was mit Hinblick auf die Übersichtlichkeit eine graphische Darstellung erfordert, die entsprechend vorgestellt wird.

Definition (Angriffsniveau)**.** *Das Angriffsniveau bezeichnet den Anteil bösartiger Verbindungen an der Gesamtzahl aller Verbindungen.*

Für eine Minimierung der Fehlalarmrate wurde die Anzahl der Korrelationspartner empirisch ermittelt und auf den Wert 12 festgesetzt. Abbildung 6.10 zeigt die Entwicklung der Detektions- und Fehlalarmraten für verschiedene Anzahlen von Korrelationen pro Verbindung. Die Festlegung des Parameters hat so zu erfolgen, dass zum einen eine möglichst geringe Fehlalarmrate vorhanden ist, zum anderen muss die Beobachtungsgruppe (bei der Nutzung von drei Intervallen zur Angriffsdetektion) möglichst klein

sein: In dieser sind noch nicht auswertbare bzw. eindeutig klassifizierbare Verbindungen enthalten, somit verzögert sich deren Bestimmung, bis mehr Daten zur Verfügung stehen.

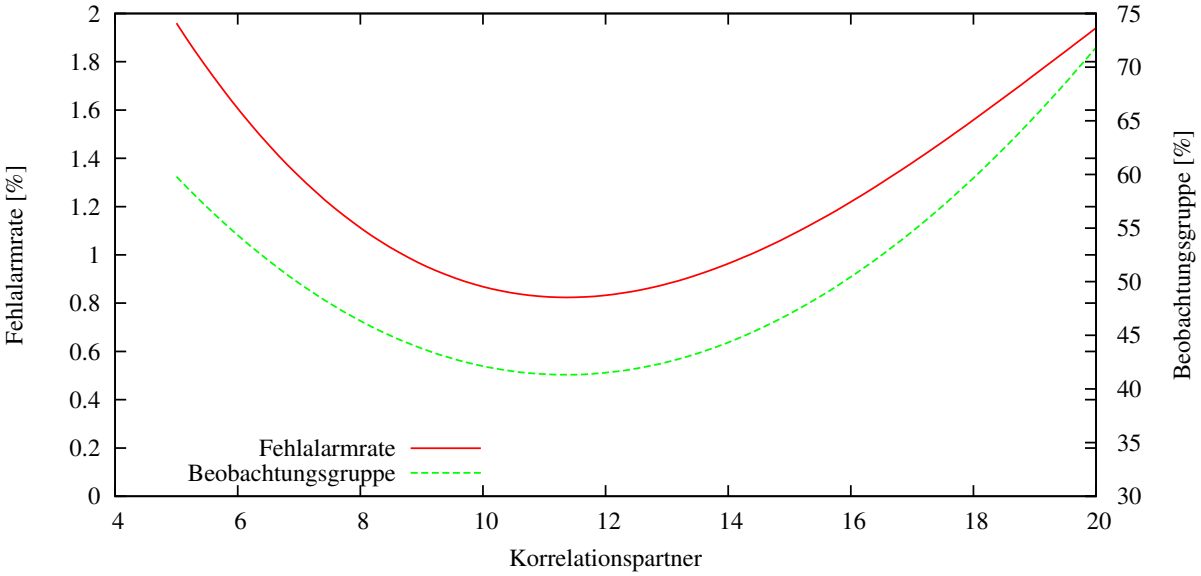
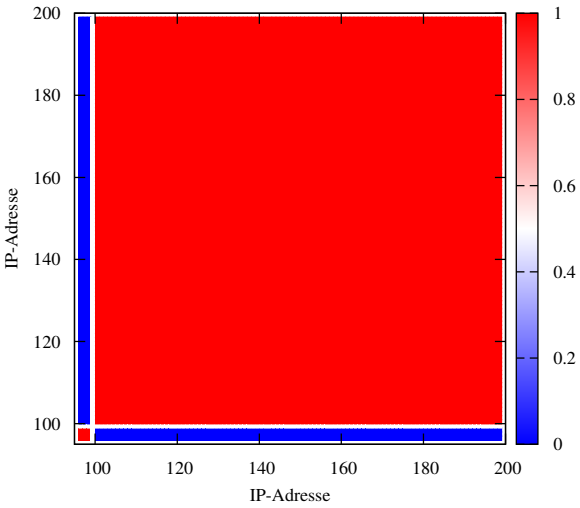


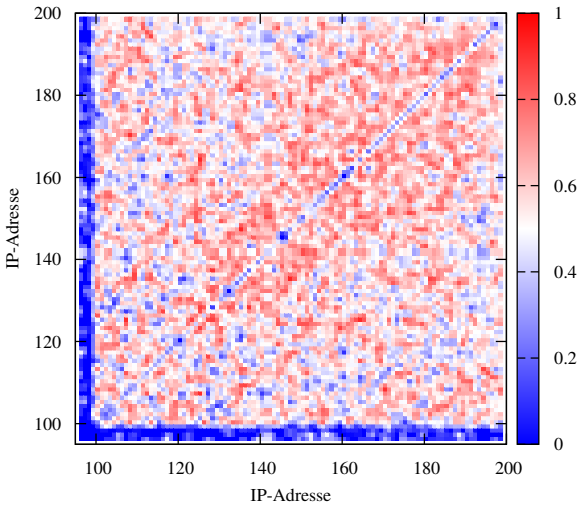
Abbildung 6.10: Auswirkung der Anzahl von Korrelationspartnern auf die Ergebnisqualität.

Die mittels des Moduls durchgeführte Detektion eines Angriffs basiert maßgeblich auf der Tatsache, dass der größte Teil der Verbindungen, die von Nutzern zu einem typischen, betrachteten Server aufgebaut werden, gutartiger Natur sind. Durch die Kreuzkorrelation der verschiedenen Verbindungen untereinander lassen sich Angreifer anhand der mehrfach auftretenden, geringen Korrelationswerte zu anderen, beliebig ausgewählten Verbindungen erkennen. Dies bedeutet jedoch auch, dass mit einer steigenden Anzahl von Angriffen das bösartige Verhalten durch die mehrfache Beobachtung immer stärker als *normale* Nutzerhandlung gewertet wird. Für eine Bewertung der Resistenz gegenüber einer steigenden Anzahl von Angriffen wurde daher der Anteil der Angriffe an den Gesamtverbindungen in mehreren Schritten gesteigert und anschließend analysiert. Die Abbildungen 6.11 bis 6.13 zeigen die Auswertungen der TLS-Angriffsanalyse visualisiert in Form von Farbkarten. Dargestellt werden hierbei die Korrelationswerte der Paketserien zwischen zwei jeweiligen Verbindungen, wobei geringe Korrelationswerte durch dunkle Blautöne, hohe Korrelationen durch Rottöne repräsentiert werden.

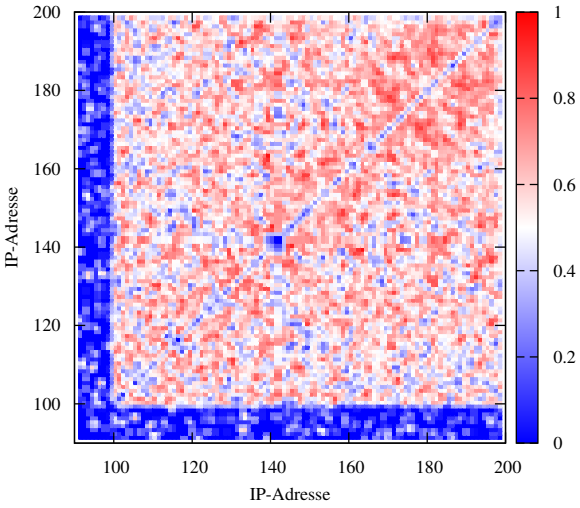
Die Entwicklung der Korrelationsergebnisse bei einem steigenden Anteil von Angreiferverbindungen lässt sich anhand der Entwicklung der Farbflächen nachvollziehen. Das *ideale*, theoretische Ergebnis bei einer Detektionsrate von 100 Prozent und der kompletten Abwesenheit von Fehlalarmen ist in Abbildung 6.11a dargestellt. Da zur besseren Auswertung der IP-Adressbereich zwischen Angreifern und normalen, gutartigen Nutzern getrennt ist, ergeben sich folgende markanten Flächenelemente in der Visualisierung:



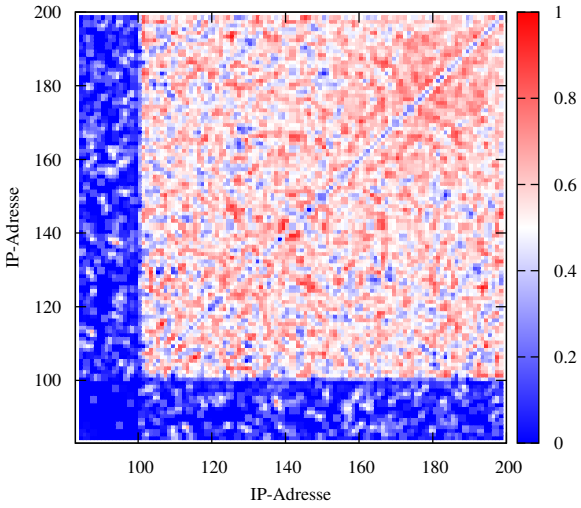
(a) Visualisierung einer theoretischen, optimalen Detektion. Der mit Abstand grösste Teil der Verbindungen ist gutartig und erzeugt maximale Korrelationswerte (rot, 1), während die Angriffsverbindungen nicht korrelieren (blau, 0).



(b) Auswertung von Verbindungen bei einem Angriffsniveau von 1%. Die Detektionsrate liegt bei 99.3%.

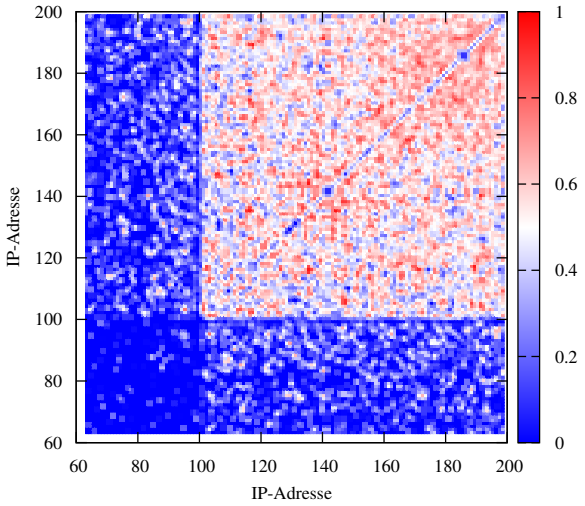


(c) Änderung der Korrelationsergebnisse bei einer steigenden Anzahl von Angreifern. Die Anzahl geringer Korrelationswerte sinkt durch die zusätzlichen, bösartigen Verbindungen (Angriffsniveau 1.5%). Die Qualität der Auswertung bleibt auf hohem Niveau.

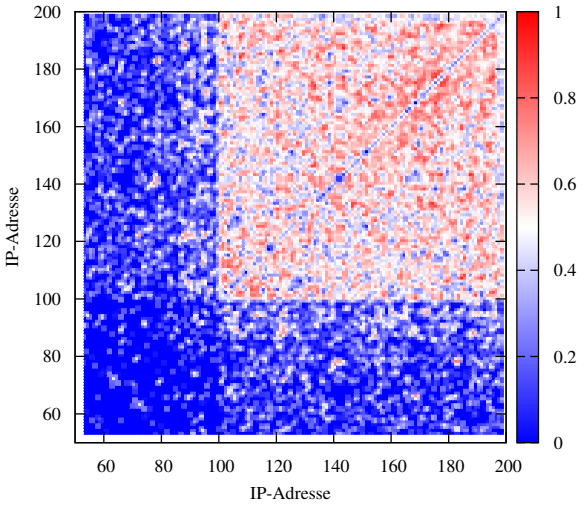


(d) Weitere Erhöhung des Angriffsniveaus auf 2.7%. Die Detektionsrate bleibt bei 97.3%, sowohl gutartige als auch bösartige Verbindungen können mit hoher Wahrscheinlichkeit identifiziert werden.

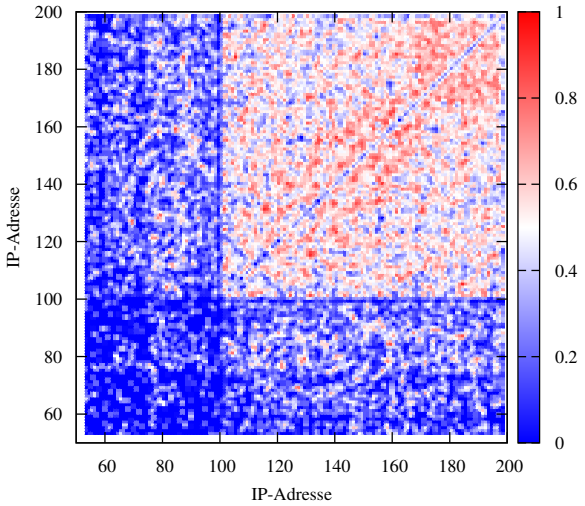
Abbildung 6.11: Visualisierung der TLS-Angriffsdetektion. Ergibt sich in der Evaluation ein geringer Korrelationswert, handelt es sich mit hoher Wahrscheinlichkeit um einen Angriffsversuch.



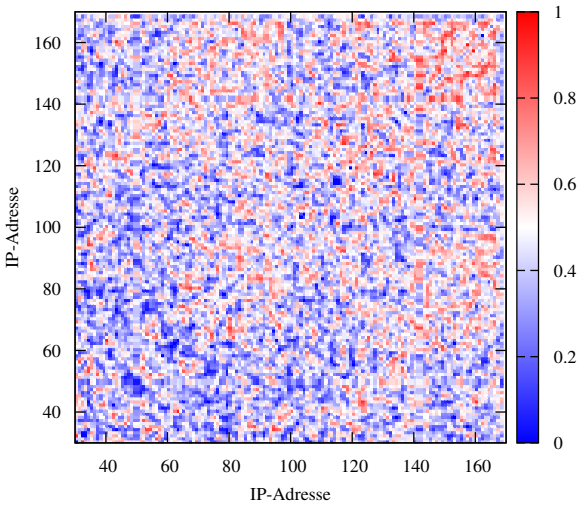
(a) Korrelationsergebnisse bei einem Angriffsniveau von 6.8%.



(b) Erhöhung des Angriffsniveaus auf 10.8%.



(c) Erhöhung des Angriffsniveaus auf 17.1%. Die unterschiedlichen Korrelationsgrade von gut- bzw. bösartigen Verbindungen sind noch gut zu erkennen, jedoch steigt der Anteil unscharfer (im Beobachtungs-Status befindlicher) Auswertungen stark an, wodurch die Detektionrate gesenkt wird (66.7%).



(d) Korrelationsevaluation bei einem Angriffsniveau von 49.1%. Durch das ausgewogene Verhältnis zwischen guten und bösartigen Verbindungen sinken die durchschnittlichen Korrelationswerte, Ausreißer zur Identifikation von gut- bzw. bösartigen Verbindungen sind kaum mehr erkennbar.

Abbildung 6.12: Visualisierung der TLS-Angriffsdetektion, steigende Angriffsanteile.

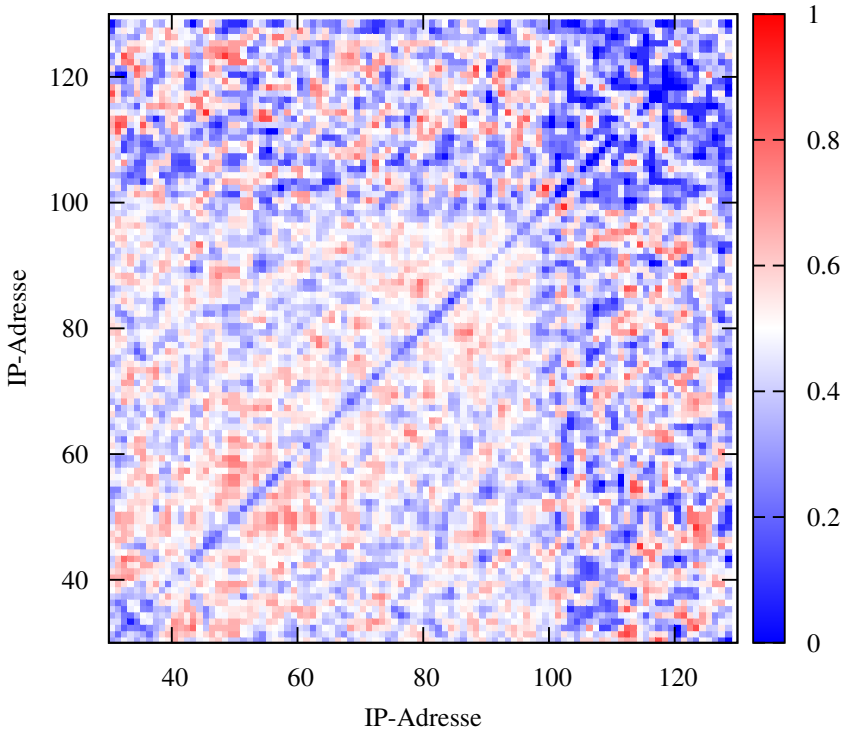


Abbildung 6.13: Evaluationsergebnis bei einem überwiegenden Anteil von bösartigen Verbindungen (Angriffsniveau 83.1%). Die Korrelationen der bösartigen Verbindungen dominieren, gutartige Verbindungen weisen geringe Korrelationswerte auf.

- Der dominante Anteil der gutartigen Verbindungen weist gleiche Eigenschaften der Datenübertragung (Up- und Downstream, Payload-Größen) auf und ergibt bei einer Vielzahl von Korrelationen zwischen jeweils zwei Verbindungen hohe, d.h. sehr ähnliche Korrelationsergebnisse. Diese werden in Rot dargestellt und repräsentieren gutartige Verbindungen; das System löst keinen Alarm aus.
- Die sehr geringe Anzahl bösartiger Verbindungen wird mit diversen gutartigen Verbindungen korreliert und weist maßgeblich unterschiedliche Eigenschaften auf: Es enstehen sehr geringe (blau dargestellte) Korrelationswerte.

Entsprechend ergeben sich im Idealfall eine große Fläche maximaler Korrelationen der gutartigen Verbindungen, sowie minimale Werte für Korrelationen bösartiger mit normalen Verbindungen. Werden für die Auswertung zufällig bösartige Verbindungen untereinander ausgewählt, spiegelt sich dies ebenfalls in hohen Korrelationen wieder (kleiner roter Bereich für Adressen kleiner 100 in Abbildung 6.11a). Für die Auswertung bedeutet dies, dass eine entsprechende Mindestanzahl von Verbindungen betrachtet werden muss, um diese als Ausreißer zu erkennen. Dieser Wert wurde wie bereits zuvor beschrieben auf 12 optimiert.

Die weiteren Abbildungen stellen die Entwicklung der Korrelationsergebnisse bei kontinuierlich steigendem Angriffsniveau dar. Liegt ein geringes Angriffsniveau vor, welches

typisch im entsprechenden Szenario ist, lassen sich hohe Detektionsraten durch das Verfahren erreichen. Abbildungen 6.11b, 6.11c sowie 6.11d zeigen die Korrelationsergebnisse der verschiedenen Verbindungen bei einem Anteil von 1, 1.5 und 2.7 Prozent bösartigem Datenverkehr. Wird der Anteil der Angreifer weiter verstärkt, lösen sich die gut abgrenzbaren Korrelationsregionen zunehmend auf. Erreichen die Angriffe einen Anteil von 50 Prozent, lassen sich keine ausreichenden Erkenntnisse mehr für eine Klassifikation aus den berechneten Werten gewinnen (vgl. Abbildung 6.12d).

Für einen Extremfall des Angriffsniveaus wie einer umfangreichen DDoS-Attacke, welche die Mehrzahl der Verbindungen zu einem Server repräsentiert, bedeutet dies, dass dieses Angriffsverhalten als gutartig bewertet wird, während die eigentlich gutartigen Verbindungen durch die geringen Korrelationen als bösartig eingestuft werden. Abbildung 6.13 zeigt eine entsprechende Situation, bei der 83.1 Prozent der vorliegenden Verbindungen bösartiger Natur sind. Dieser Effekt lässt sich jedoch reduzieren, wenn bösartige Verbindungen sofort bei Detektion von der Nutzung für weitere Korrelationen ausgeschlossen werden. Nimmt die Zahl der bösartigen Verbindungen über einen gewissen Zeitraum zu, bspw. durch die Nutzung eines Bot-Netzes für den Angriff, können somit die bösartigen Verbindungen rechtzeitig erkannt werden und nehmen keinen Einfluss auf die weitere Evaluation. Kann der Angriff jedoch hinreichend synchronisiert gestartet werden, ist dies nicht ohne weiteres möglich.

Die analysierte Entwicklung der Detektionsraten in Abhängigkeit des Anteils bösartiger Verbindungen ist in den Abbildungen 6.14 und 6.15 dargestellt. Während im ersten Fall die Evaluation mittels dreier Intervalle genutzt wird, sind beim Zweiteren lediglich die Gruppen gut- und bösartiger Daten vorhanden. Gut zu erkennen ist, dass im Falle der Nutzung von drei Intervallen auch bei steigender Zahl von Angriffen eine hohe, korrekte Klassifikationsrate erzielt werden kann. Dies erfolgt jedoch auf Kosten einer wachsenden Gruppe von noch nicht klassifizierbaren Verbindungen; für diese ist anhand der berechneten Korrelationen noch keine Entscheidung möglich. Insbesondere bei einem starken Anstieg des Angriffspotentials nimmt diese Gruppe erheblich zu (vgl. Abbildung 6.16), so dass eine Evaluation von Verbindungen zu lange verzögert werden kann oder aufgrund mangelnder Daten nicht möglich ist.

Wird dahingegen keine Intervalleinteilung vorgenommen, existiert keine Beobachtungsgruppe, sondern alle Verbindungen werden sofort in eine der beiden Klassen, gut- und bösartig eingeteilt. Abbildung 6.15 zeigt die hierbei erzielbaren Detektionsraten mit Hinblick auf einen steigenden Anteil von Angriffen. Liegt ein geringes Angriffsniveau vor, ist auch hier eine hohe Detektionsrate erreichbar. Diese sinkt bei zunehmenden Angriffen schneller ab als bei der Nutzung von Intervallen, jedoch sind hierbei sämtliche Daten bewertet und es kommt nicht zu noch nicht evaluierbaren Verbindungen.

Lässt sich bei der Nutzung von Intervallen keine sofortige Klassifizierung vornehmen, kann jedoch auch das stetige Anwachsen der Beobachtungsgruppe als Detektionsmerkmal für einen Angriff oder anormalen Datenverkehr herangezogen werden, da eine entsprechende Entwicklung unter normalen Betriebsparametern nicht entsteht.

Tabelle 6.5 fasst die Evaluationsergebnisse bei der Nutzung von drei Auswertungsintervallen zusammen. Gut zu erkennen ist die Entwicklung der Detektionswahrscheinlichkeit eines Angriffs und der Fehlalarmraten bei der Meldung von Angriffen in Abhängigkeit

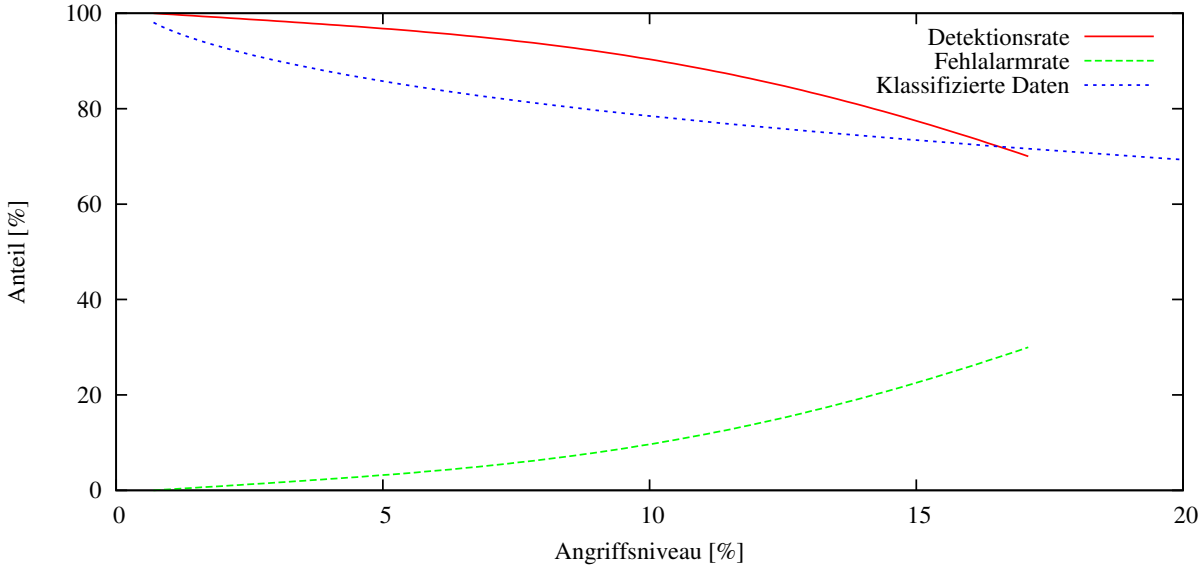


Abbildung 6.14: Detektionsraten der SSL/TLS Angriffsdetektion bei einem steigenden Anteil von bösartigen Verbindungen.

Tabelle 6.5: Entwicklung der korrekten Klassifizierung und der Detektionswahrscheinlichkeiten für die Korrelation von **Einzelverbindungen** im Modul der TLS-Angriffsdetektion bei Nutzung von *drei* Intervallen, Angaben in Prozent.

Angriffsniv.	Klassif.	Detektionswahr.	Fehlalarmverh.	Fehlalarmrate
1.0	74.39	84.70	91.14	25.92
2.7	72.05	70.14	83.59	27.80
6.8	64.19	62.83	77.24	35.83
17.1	60.88	54.80	69.02	37.26
49.1	53.70	43.57	44.27	35.81

des Angriffsniveaus. Nutzt man zur Evaluation nur zwei Intervalle, also eine Grenze im Wertebereich $[0, .., 1]$ zur Einteilung zwischen gut- und bösartigen Verbindungen und verzichtet auf die Einführung eines Intervalls für unter Beobachtung stehende Verbindungen, ergeben sich die Werte gem. Tabelle 6.6.

Interessant ist hierbei, dass die Wahrscheinlichkeit einer korrekten Klassifizierung bei geringen Angriffsleveln ansteigt und die Fehlalarmraten besser liegen. Dies lässt sich dadurch erklären, dass der Schwellwert für die Unterteilung zwischen gut- und bösartig bei 0.4 liegt, was der unteren Grenze bei der Einteilung in drei Intervallen entspricht. Somit verschiebt sich bei der Reduzierung der Intervalle auf zwei die obere Grenze zur unteren hin, wobei alle Werte des vormaligen Beobachtungsbereiches in den Bereich für gutartige Verbindungen fallen. Hierdurch steigen nur die Werte für *True Negatives* und *False Negatives* an, während die anderen Werte konstant bleiben, was die vorliegende Entwicklung der Wahrscheinlichkeiten erklärt. Während sich somit durch den überwie-

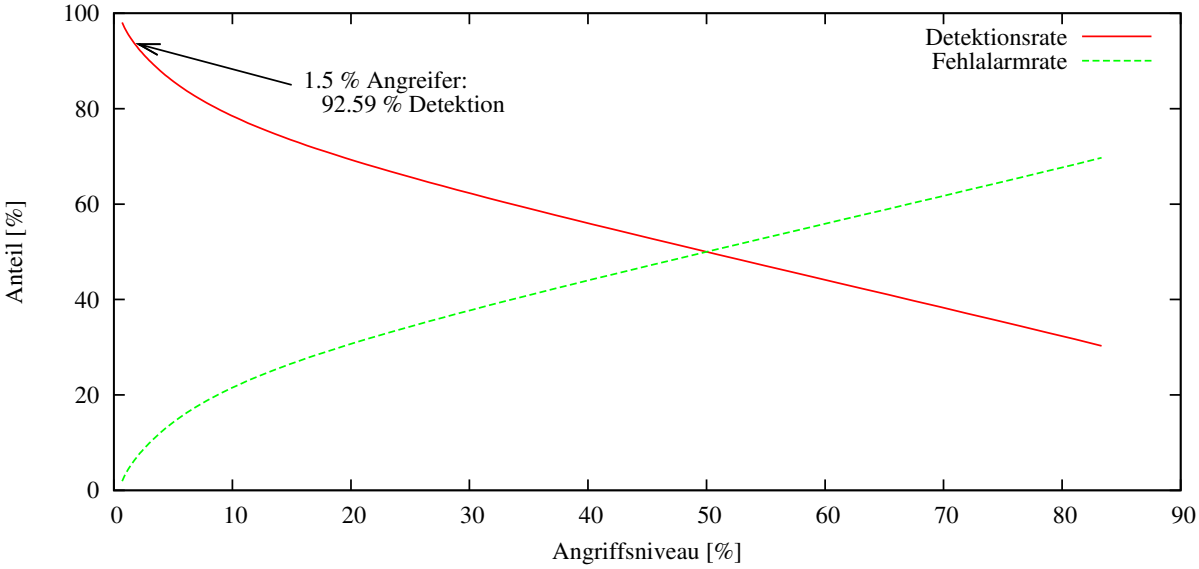


Abbildung 6.15: Detektionsraten der SSL/TLS Angriffsdetektion bei einem steigenden Anteil von bösartigen Verbindungen, wenn kein Status *„unter Beobachtung“* eingeführt wird.

Tabelle 6.6: Entwicklung der korrekten Klassifizierung und der Detektionswahrscheinlichkeiten für die Korrelation von **Einzelverbindungen** im Modul der TLS-Angriffsdetektion bei Nutzung von *zwei* Intervallen, Angaben in Prozent.

Angriffsniv.	Klassif.	Detektionswahr.	Fehlalarmverh.	Fehlalarmrate
1.0	82.50	74.88	91.14	17.33
2.7	79.61	60.35	83.59	19.16
6.8	72.15	52.11	77.24	25.01
17.1	68.57	44.21	69.02	25.25
49.1	51.64	32.20	44.27	26.15

genden Teil gutartiger Verbindungen bei geringen Angriffswerten der Anteil korrekter Klassifizierungen erhöht, sinkt die Detektionswahrscheinlichkeit eines Angriffs, da bösartige Verbindungen, die zuvor im Beobachtungsfenster lagen und ggf. noch als bösartig erkannt werden konnten, nun fehlerhaft als gutartig eingestuft werden. Um in diesem Kontext den Einfluss der Auswahl der Grenze zwischen gut- und bösartigen Verbindungen zu prüfen, wurden entsprechende Schwellwerte gem. Tabelle 6.7 eruiert. Gut erkennbar ist, dass bei einer Anhebung der Schwelle die Detektionswahrscheinlichkeit von Angriffen wie erwartet ansteigt, andererseits steigt jedoch auch die Fehlalarmrate entsprechend an. Die etwas geringere Detektionswahrscheinlichkeit eines Angriffes bei Nutzung der Schwelle 40/60 wird hier im Sinne einer deutlich besseren Fehlalarmrate präferiert: Die Wahrscheinlichkeit, dass hierdurch ein Alarm übersehen wird, sinkt durch das Auswahlverfahren, da jeweils die Korrelationsergebnisse mit 12 weiteren Verbindun-

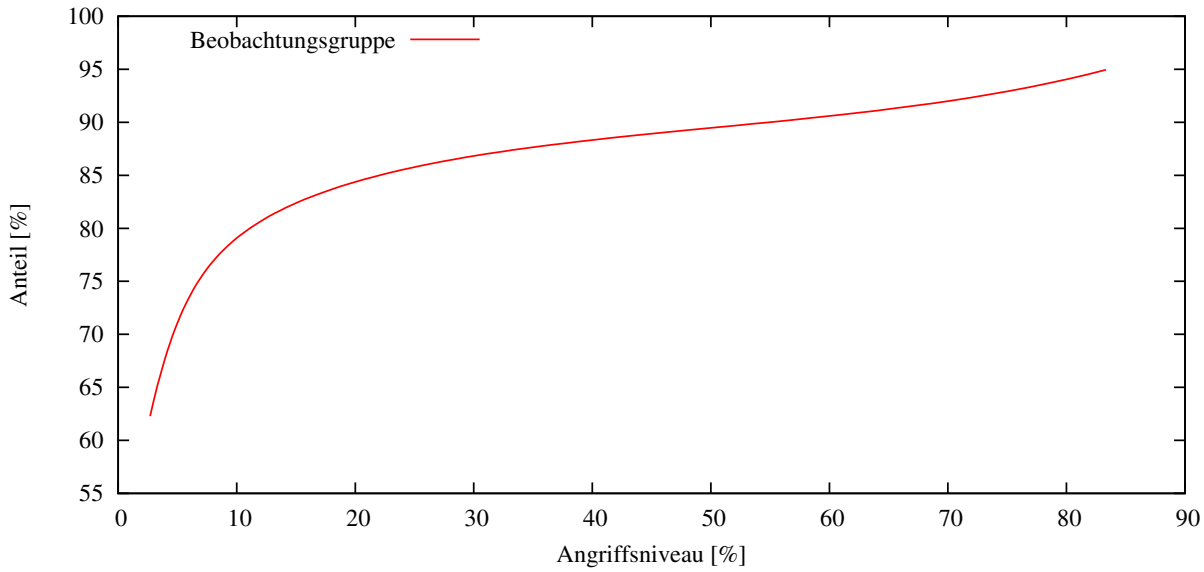


Abbildung 6.16: Entwicklung der Anzahl der Verbindungen im Status *„unter Beobachtung“* bei einem steigenden Anteil von bösartigen Verbindungen.

gen gemittelt werden (vgl. auch Abbildung 6.10) und das Ergebnis zur Einstufung der Verbindung dient.

Beachtet werden muss, dass die in den Tabellen 6.5 bis 6.7 aufgeführten Ergebnisse die Wahrscheinlichkeiten für einzelne Verbindungskorrelationen darstellen, wobei die endgültige Bestimmung der Verbindung auf gut- oder bösartig durch die Betrachtung der Ergebnisse mehrerer Korrelationspartner mit der jeweiligen Verbindung eruiert wird. Hierdurch wird bei einem Angriffsniveau von 1.5 Prozent eine durchschnittliche Detektionsrate von rund 93 Prozent erreicht. Somit lassen sich bösartige Verbindungen bei einem normalen Angriffsniveau mit hoher Wahrscheinlichkeit und geringen Fehlalarmraten detektieren.

Tabelle 6.7: Bedeutung der Intervallunterteilung zur Festlegung von gut- bzw. bösartigen Korrelationswerten: Eine Anhebung auf 0.5 findet geringfügig mehr bösartige Verbindungen, senkt jedoch die korrekten Klassifizierungen allgemein. Da diese Betrachtung für einzelne Korrelationen gilt und die finale Festsetzung einer Verbindung auf gut- oder bösartig anhand des Mittels über mehrere Korrelationen erfolgt, ist die Beibehaltung der Schwelle von 0.4 im Sinne der Detektions- und Fehlerraten vorzuziehen.

Intervall	Klassifizierung	Detektionswahr.	Fehlalarmverh.	Fehlalarmrate
30/70	88.70	69.57	84.16	69.57
40/60	82.50	74.88	91.14	17.33
50/50	69.39	83.09	94.30	30.92
60/40	69.47	86.47	94.08	69.08

6.3 Module zur Ausbruchs- und Innentätererkennung

Nachfolgend werden die Module zur Ausbruchs- und Innentätererkennung evaluiert, hierzu gehören das Modul zur Befehlsevaluation sowie das Modul zur Nutzeridentifizierung.

6.3.1 Befehlsevaluation

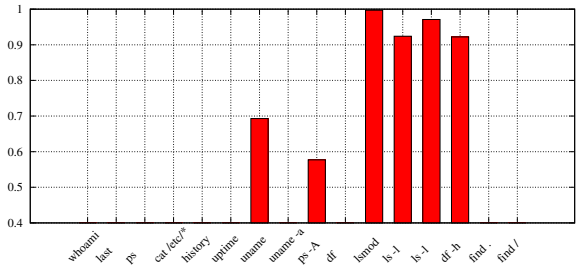
Die Durchführung der Evaluation der Befehlserkennung wird anhand von realen Nutzersitzungen durchgeführt. Hierfür werden Sitzungen mit unterschiedlichen Befehlsserien ausgeführt und die Auswertung des Systems mit den ursprünglich eingegebenen Befehlen verglichen. Eine Entscheidung über das korrekte Verhalten des Systems bzw. die richtige Auswertung ist somit gegeben.

Zunächst wird die paketsequenzbasierte Evaluation für die Durchführung der Messungen herangezogen. Im Gegensatz zur ersten prototypischen Implementierung der Befehlsevaluation, welche einen exakten Vergleich der Paketserien zur Identifizierung von Befehlen vornahm, werden im weiteren Verlauf die Ergebnisse des aktuellen Systems vorgestellt, in welchem die aufgezeichneten und aus den Clustern gewonnenen Paketserien mit den Referenzserien korreliert werden. Dieses Verfahren wurde eingeführt, da bei einem exakten Vergleich bereits geringste Änderungen der Serverantwort zu einer Nichterkennung des Befehls führen. Während das Verfahren des exakten Vergleichs im Sinne der korrekten Identifizierung bei einem *genau bekannten System* geringe falsche Auswertungen ermöglicht, ist die Nutzung von Korrelationen durch die hohe Dynamik von Rechnersystemen besser in der Praxis anwendbar.

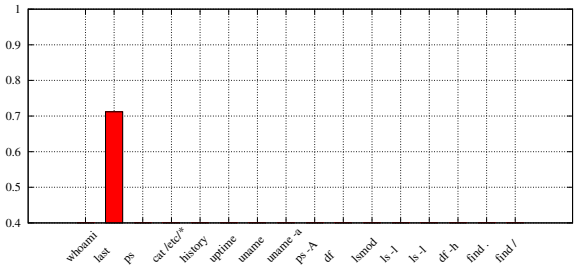
Auf Basis der in Kapitel 5.1.5 zusammengestellten Befehle, welche für einen Angreifer von besonderer Bedeutung sind, wurden die zugehörigen Referenzdaten in den Prototypen integriert. Für die Erzeugung der im Rahmen der Befehle notwendigen Antwortsequenzen des Servers wurde *Ubuntu* [257] in den Versionen 10.10 (Maverick Meerkat) und 11.04 (Natty Narwhal) eingesetzt. Aufgrund des hohen Aufwands, alle möglichen Befehle umfassend in die Detektion aufzunehmen, erfolgt eine explizite Suche nach Befehlen, die typische Angriffsmuster repräsentieren können. Andererseits werden Befehle, die in diesem Sinne gutartig sind, nicht implementiert und werden nach einer negativen Prüfung gegen die vorhandenen, risikoreichen Befehle nicht weiter bewertet, bzw. stehen für eine gutartige Verbindung.

Zunächst wird eine Evaluation einzelner eingegebener Befehle durchgeführt (vgl. Abbildung 6.17). Hierfür werden der Reihe nach die Kommandos `lsmod`, `last`, `uname`, `cd /etc` sowie `ls -l` aufgerufen. Angemerkt sei auch, dass der Befehl zum Wechseln des Verzeichnisses (`cd`) nicht im Prototypen integriert ist, also auch nicht erkannt werden kann.

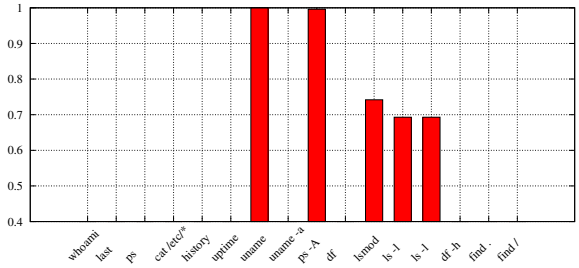
Abbildung 6.17a zeigt das Ergebnis des Sicherheitssystems, nachdem die Datenpakete des Clusters (vgl. Kapitel 5.1.3) des Befehls `lsmod` und der zugehörigen Serverantwort ausgewertet wurden. Hierbei berücksichtigt das Modul insbesondere auch die Länge der vom Server zurückgegebenen Antworten: Stimmen diese nicht genau mit der im System hinterlegten Anzahl von Paketen überein, erfolgt trotzdem eine Korrelation, solange die Bedingungen für die Clustergrenzen erfüllt sind (maximale und minimale Paketzahlen).



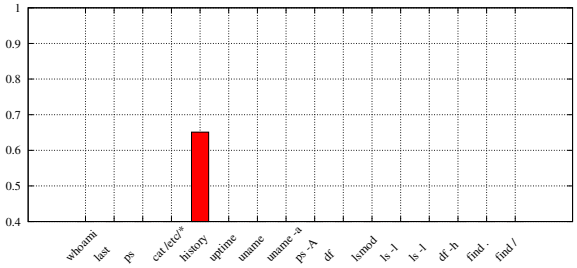
(a) Auswertung des Befehls `lsmod`. Der Befehl wurde korrekt erkannt, wobei insbesondere die Befehle `ls -l` und `df -h` ebenfalls hohe Ähnlichkeiten aufweisen.



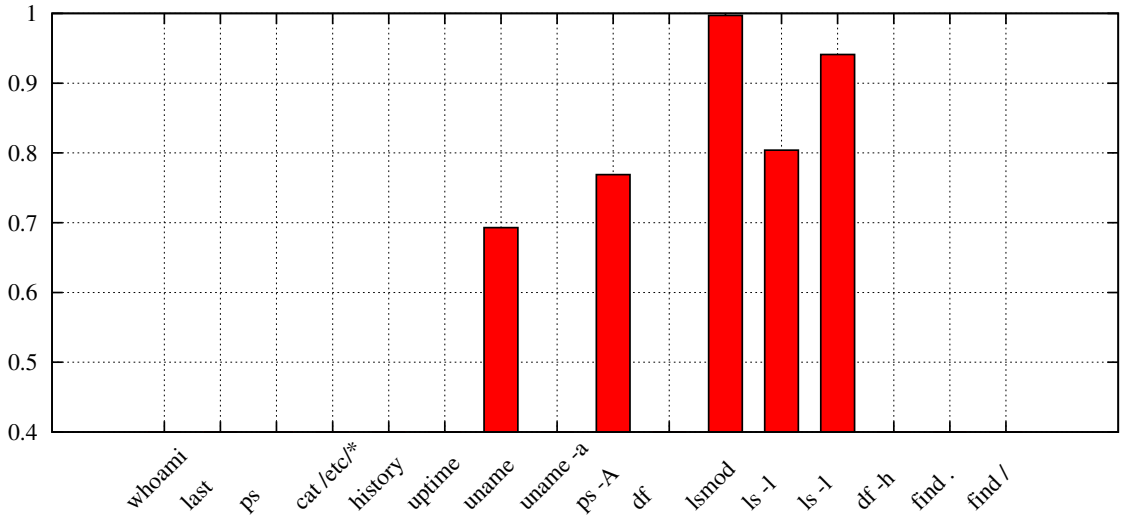
(b) Erkennung des Befehls `last`. Der Befehl wurde als einziger Kandidat identifiziert und mit einer Ähnlichkeit von 0.712 bewertet.



(c) Analyse der Eingabe von `uname`. Der richtige Befehl korreliert mit 1.0, `ps -A` weist eine ebenfalls hohe Ähnlichkeit von 0.996 auf.



(d) Fehlerhafte Interpretation eines nicht im System vorhandenen Befehls (`cd /etc`) als `history`.



(e) Auswertung von `ls -l` im Verzeichnis `/etc`. Der richtige Befehl weist eine Korrelation von 0.941 auf und liegt damit knapp unter dem Kandidaten `lsmod`.

Abbildung 6.17: Befehlsevaluation verschlüsselter Verbindungen.

Dies ist erforderlich, da die Antworten des Servers leicht variieren können, bspw. bei Auflistung der geladenen Kernelmodule, nachdem zusätzliche Module geladen wurden oder durch eine zum Referenzsystem unterschiedliche Hardwarebasis und somit unterschiedlicher Anzahl von geladenen Treibern. In diesem Rahmen wurden vom System fünf mögliche Befehle selektiert und deren Ähnlichkeitswerte berechnet. Hierbei lassen sich zwei Gruppen identifizieren, Werte im Bereich von ca. 0.6 bis 0.7 sowie Werte über 0.9. Der korrekte, eingegebene Befehl ist `lsmod`, welche in der Auswertung den höchsten Korrelationswert erzielt hat (0.997); ihm nachfolgend erreicht der Befehl `ls -l` jeweils 0.924 und 0.971 für die Verzeichnisse `/` bzw. `/etc`. Auf Basis dieser Auswertung wurde der Befehl korrekt identifiziert und kann für die weiteren Betrachtungen herangezogen werden. Hierbei werden auch die weiteren Befehle, welche mit sehr hoher Ähnlichkeit als mögliche Kandidaten identifiziert wurden, weiter berücksichtigt. Das Vorgehen hierbei wird in der an die Auswertung einzelner Befehle anschließende Evaluation kompletter Sitzungen gezeigt. Der nächste betrachtete Befehl ist `last`, welcher eine Liste der zuletzt eingeloggten Nutzer zurück gibt. Wie in Abbildung 6.17b präsentiert, findet das Modul nur einen möglichen Befehlskandidaten, nämlich korrekt den Befehl `last` und bewertet diesen mit 0.712. Abbildung 6.17c zeigt den dritten ausgeführten Befehl, `uname`. Dieser wird genutzt, um Systeminformationen auszugeben, bspw. die Version des laufenden Kernels oder den Namen des Systems im Netz. Da dieser Befehl oftmals mit dem Parameter `-a` aufgerufen wird, um alle Systeminformationen anzuzeigen, wurde er in zwei Varianten in das Sicherheitssystem integriert, als Aufruf mit und ohne Parameter. Die Ausführung des Kommandos ergibt nach Beurteilung durch das Sicherheitssystem eine Ähnlichkeit von 1.0, wobei bei der genutzen Systemkonfiguration und dem zum Zeitpunkt der Ausführung des Befehls aktuellen Systemzustand auch der Befehl `ps -A` mit einem Korrelationswert von 0.996 sehr hoch bewertet wird. Eine weitere Gruppe von drei Befehlen folgt mit einigem Abstand und Korrelationswerten von 0.693 bis 0.742. Auch hier wurde der richtige Befehl identifiziert, wobei der Unterschied zu `ps -A` in diesem Falle nur marginal ist. Dadurch, dass letzterer jedoch auch in der Gruppe der unter Beobachtung stehenden Befehlen ist, muss dies noch nicht zu einer Fehlentscheidung des Systems führen (vgl. die Auswertung der Teilziele ab Seite 226). Andererseits kann die fehlerhafte Interpretation eines eigentlich unverdächtigen Kommands als gefährliches Kommando wiederum die Auswertung verfälschen. Abbildung 6.17d zeigt einen entsprechenden Fall. Das hier genutzte Kommando `cd /etc` ist nicht im System hinterlegt, wird jedoch fehlerhaft als eine Eingabe von `history` mit einer Ähnlichkeit von 0.651 interpretiert. Wie anhand der Auswertung längerer Sitzungen im weiteren Verlauf gezeigt, werden jedoch nicht im System befindliche (gutartige) Befehle oft korrekterweise *nicht* erkannt und sind somit ein Indiz für eine gutartige Sitzung. In Abbildung 6.17e wird ein weiteres Beispiel der Eingabe des Befehls `ls -l` gezeigt, eingegeben im Verzeichnis `/etc`. Für `ls -l` sind mehrere Payload-Serien für wichtige Zielverzeichnisse im Sicherheitssystem hinterlegt. Die Auswertung bewertet hierbei das Verzeichnis `/` mit 0.804, das korrekte Verzeichnis `/etc` mit 0.941. Der Kandidat `lsmod` erzielt fälschlicherweise einen etwas höheren Korrelationswert.

Die Evaluation der Eingabe einzelner Befehle hat gezeigt, dass die korrekten Befehle mit hohen Wahrscheinlichkeiten richtig identifiziert werden können, oder Teil einer klei-

nen Auswahlgruppe sind. Aufgrund der Generierung der Referenzpaketserien anhand der oben genannten Ubuntu-Installation muss nun weiterhin geprüft werden, wie sich die Detektionsresultate verhalten, wenn verschiedenen Zielsysteme verwendet werden.

Entsprechend werden für diese Untersuchung folgende Distributionen verwendet:

- Ubuntu 11.04, `i686`, Kernel `2.6.38-8-generic`
- SuSE 11.4, `i686`, Kernel `2.6.37.1-1.2-default`
- Mandriva 2010.2, `i686`, Kernel `2.6.33.7`

Abbildung 6.18 zeigt den Vergleich der Evaluation des Kommandos `lsmod` durch das Sicherheitssystem.

Gut zu erkennen ist, dass in allen Fällen das richtige, ursprünglich eingegebenen Kommando `lsmod` identifiziert wird. Die exakten Korrelationswerte differieren wie erwartet leicht zwischen den verschiedenen Systemen, basierend auf Abweichungen in den Paketserien, bspw. durch eine unterschiedliche Anzahl geladener Kernelmodule im Falle von `lsmod` oder einer anderen Anzahl von Dateien und Unterverzeichnissen im Konfigurationsverzeichnis `/etc` bei einer Auflistung des Verzeichnisses mit `ls`. Ein interessanter Aspekt beim Vergleich der Auswertungsergebnisse zwischen den Systemen ist, dass nicht die Ubuntu-Distribution das ausgeprägteste Ergebnis liefert[6], sondern die Korrelation auf der genutzten SuSE-Distribution: Obwohl die ursprünglichen Referenzwerte von einem Ubuntu-System gewonnen wurden, sind diese nicht auf dieses System beschränkt, sondern lassen sich generell für Linux-Systeme nutzen. Hierdurch wird eine beträchtliche Einsparung von in der Datenbank zu hinterlegenden Mustern von gefährlichen Befehlen erreicht, die im anderen Falle notwendig geworden wäre.

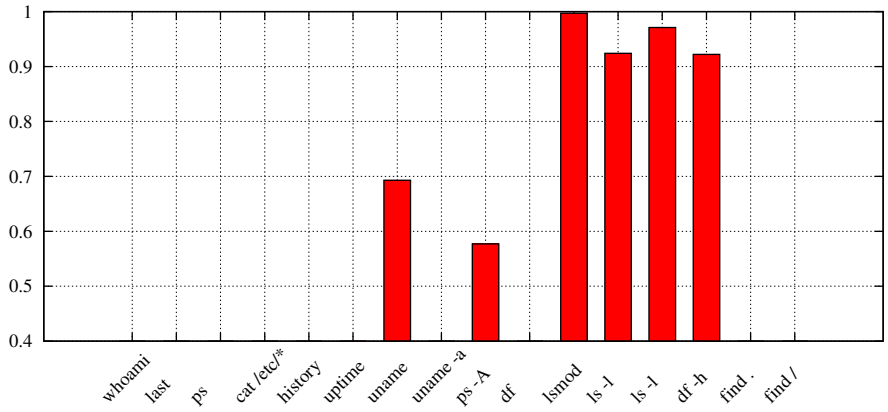
Mittels der auszugsweise vorgestellten Evaluation von Befehlen in verschiedenen Umgebungen ist ersichtlich, dass eine Identifizierung der eingegebenen Befehle mittels der in der vorliegenden Arbeit vorgestellten Methode möglich ist. Da in der Praxis jedoch nur ein Teil aller möglichen Befehle und deren Parameterkombinationen umgesetzt werden können, verbleiben zahlreiche Eingaben, welche nicht korrekt erkannt werden können. Zusätzlich können Änderungen am System zur fehlerhaften Erkennung einzelner Befehle führen.

Nachfolgend wird daher die Fähigkeit des Systems untersucht, Angriffe auf Basis einer Sitzung, also durch eine längere Serie von Befehlen zu erkennen. Hierfür werden die Ergebnisse der Befehlsevaluation der einzelnen Cluster an die Sequenzevaluation des Moduls weitergegeben (vgl. Kapitel 5.1.5).

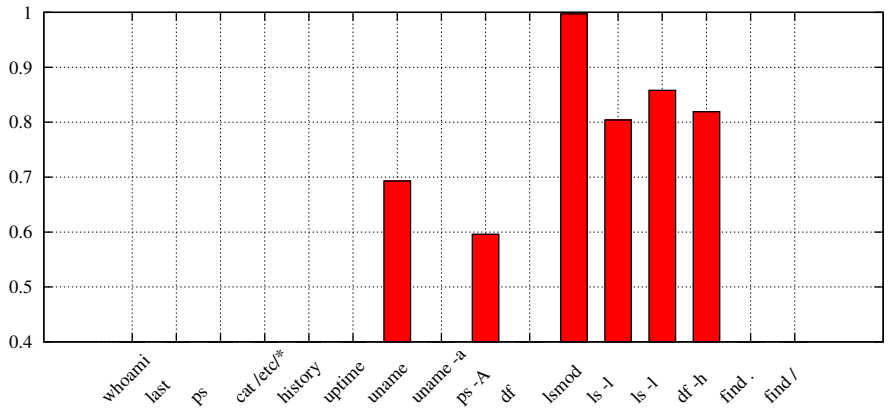
Für eine Evaluation des Verfahrens werden verschiedene Remotesitzungen initiiert, welche auf Basis folgender Auswahl bösartiges bzw. gutartiges Verhalten dargestellen:

- Nutzersitzung 1: Bewegung im Nutzerverzeichnis, Abfrage des freien Speichers, Editieren einer Datei, Auflisten des Verzeichnisses.

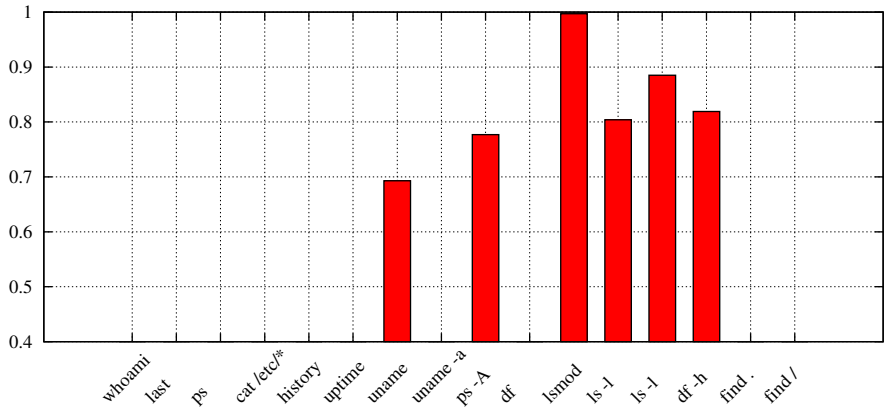
[6] Dies lässt sich durch den zeitlichen Abstand zwischen der Gewinnung der Paketserie und der Durchführung der vorliegenden Evaluation erklären. Zwischenzeitliche, kleine Änderungen am System sorgen hier für eine Abweichung, die im Laufe der Zeit jedoch auch wieder geringer werden kann.



(a) Auswertung der Antwort eines Systems mit Ubuntu 11.04.



(b) Auswertung der Antwort eines Systems mit SuSE 11.4.



(c) Auswertung der Antwort eines Systems mit Mandriva 2010.2.

Abbildung 6.18: Vergleich der Evaluation bei verschiedenen Systemen am Beispiel des Befehls `lsmod`.

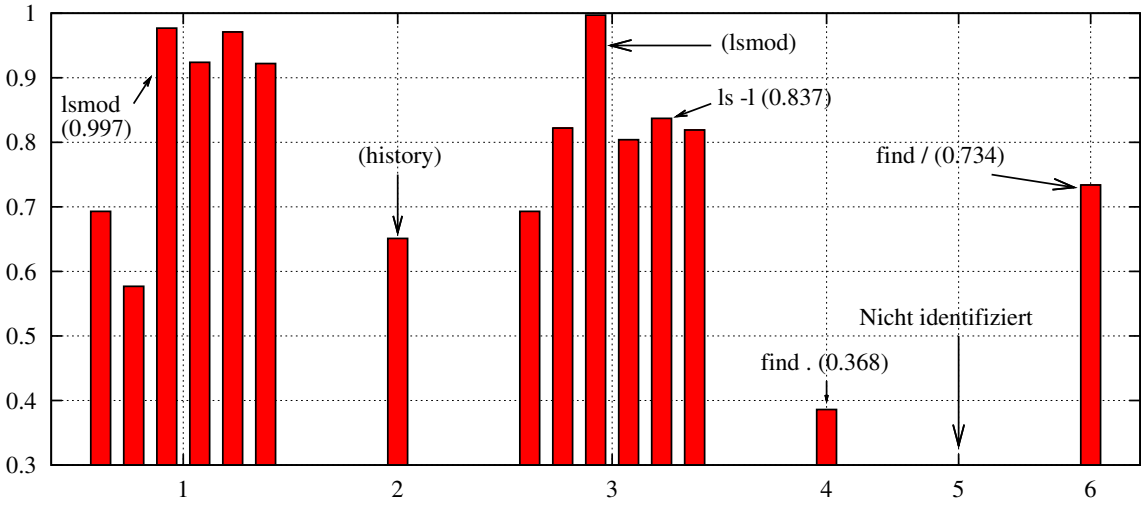


Abbildung 6.19: Analyse einer bösartigen Sitzung mit dem Ziel einer Rechteeskalation (U2R). Mehrere gefährliche Befehle können durch das System identifiziert werden. Insbesondere die fehlerhafte Evaluation von Befehl 2 beeinträchtigt nicht das Ergebnis (Erkennung von `history` anstelle von `cd /etc`, was nicht in der Datenbank vorhanden ist). Das nicht identifizierte Kommando 5 ist korrekt, hierbei handelt es sich um ein fehlerhaft eingegebenes Kommando. Die Sitzung kann korrekt als bösartig erkannt werden.

- Nutzersitzung 2: Anlegen einer Datei, Kopieren von Daten, Bewegung durch den Verzeichnisbaum, Auflisten verschiedener Verzeichnisse.
- Angreifer 1: Abfrage der Nutzer-ID, Auflisten der Kernelmodule, Auflisten der Kernelversion, Suche nach privilegierten Dateien.
- Angreifer 2: Suche im Konfigurationsverzeichnis, Nachladen von Programmen, Abfrage der Nutzer-ID.

Tabelle 6.8 fasst die involvierten Befehle der beiden Detektionskategorien *Nutzer* und *Angreifer* zusammen.

Die durchgeführten und durch das System ausgewerteten Sitzungen werden nun analysiert.

Abbildung 6.19 zeigt die Ergebnisse der Auswertung eines Ausschnitts einer bösartigen Sitzung. Hierbei handelt es sich um den Versuch eines Angreifers, der bereits Zugang zum Zielsystem mit eingeschränkten Rechten hat, seine Privilegien zu erhöhen (U2R, vgl. Kapitel 2.3.3).

Zunächst identifiziert das System eine Auflistung der geladenen Kernelmodule, was dem eingegebenen Befehl des Angreifers entspricht. Der nächste eingegebene Befehl wird als Eingabe von `history` erkannt, was jedoch nicht mit der realen Eingabe übereinstimmt; der hier eigentlich genutzte Befehl ist `cd /etc`. Im nächsten Schritt erfolgt eine

Tabelle 6.8: Im Rahmen der Erprobung der Befehlserkennung genutzte Sitzungsbefehle (ohne eventuelle Parameter) nach den Kategorien *Angreifer* und gutartiger *Nutzer*. Unter *System* markierte Befehle werden im Rahmen des Sicherheitssystems evaluiert. Weitere Befehle, die jedoch nur durch gutartige Nutzer verwendet und nicht im System integriert sind, werden nicht aufgeführt (Symbolisiert mittels [...]).

Befehl	Angreifer	Nutzer	System
`cat`	✓	✓	
`cd`	✓	✓	
`chmod`	✓	✓	✓
`df`	✓	✓	✓
`find`	✓	✓	✓
`history`	✓	✓	✓
`last`	✓		✓
`ls`	✓	✓	✓
`lsmod`	✓		✓
`mkdir`	✓	✓	
`mv`	✓	✓	
`passwd`	✓		✓
`ps`	✓		✓
`rm`	✓		
`uname`	✓		✓
`uptime`	✓		✓
`vi`	✓	✓	
`whoami`	✓		✓
[...]		✓	

Eingabe von `ls -l`. Das System identifiziert die zugehörige Paketserie als die Eingabe von `lsmod`, findet jedoch den korrekten Befehl mit einer ebenfalls hohen Ähnlichkeit von 0.837, welcher mit in die weitere Evaluation einfließt (s.u.). Der vierte eingegebene Befehl wird als die Eingabe von `find . -perm 2 -print` identifiziert, was korrekt ist. Trotz der im Vergleich zu anderen Befehlen geringeren Korrelationshöhe des Ergebnisses läuft dieser Befehl entsprechend in die weitere Verarbeitung ein, da er der einzige, mögliche Kandidat ist. Befehl Nummer fünf kann durch das Sicherheitssystem nicht zugeordnet werden und wird daher nicht für die weitere Evaluation herangezogen. Dies ist korrekt, da der hinter den Datenpaketen stehende Befehl einen Tippfehler enthielt und somit lediglich eine Fehlermeldung zurückliefert, jedoch keine ggf. bösartige Paketsequenz darstellt. Der letzte im gezeigten Ausschnitt ausgewertete Befehl wird korrekt als `find / -type f -perm +0600` erkannt und stellt den einzigen möglichen Kandidaten mit einem Korrelationswert von 0.734 dar. Der deutlich bessere Korrelationswert der Suche auf dem gesamten Verzeichnisbaum (`/`) im Vergleich zur deutlich geringeren Ähnlichkeit der Suche in einem lokalen Verzeichnis (`.`, befindlich in `/etc`) lässt sich dadurch erklären, dass kleine Abweichungen im lokal gewählten Verzeichnis hier größeren Einfluss auf das Gesamtergebnis im Sinne der zurückgelieferten Paketserien haben, als entsprechende Änderungen bezogen auf das gesamte Dateisystem. Auch große Nutzerverzeichnisse mit privaten Daten (typischerweise unter `/home`) werden hier keinen starken, negativen Einfluss auf die Evaluation haben, da sich hier typischerweise nur geringfügig Dateien mit dem gesuchten Rechtemuster 0600 befinden[7].

Im nächsten Schritt erfolgt die Überprüfung, ob mittels einer Befehlssequenz ein Teilziel eines Angriffsbaumes erreicht werden kann. Hierbei wird für jedes Teilziel der zugehörige Befehlspool mit den identifizierten Kandidaten verglichen; jedem Befehl aus dem Pool sind dabei ein Wahrscheinlichkeitswert und ein Gefährdungswert zugeordnet, welche für die Anpassung der Auswahlreihenfolge zwischen den möglichen Kandidaten eines Befehls und der Berechnung der von Befehlen ausgehenden Gefährdung genutzt werden (vgl. Kapitel 5.1.5). Tabelle 6.9 zeigt Auszüge der Gefährdungswerte verschiedener Befehle im Rahmen des Angriffsbaums Rechteerhöhung. Die Werte wurden auf Basis der Angriffsanalyse in Kapitel 2.3, den Ergebnissen gem. Ramsbrock [320] sowie empirischen Untersuchungen anhand des Sicherheitssystems S2E2 eruiert.

In der vorliegenden Auswertung des Sicherheitssystems werden entsprechend vier der sechs Befehle identifiziert und bzgl. des Teilziels ausgewertet. Der Angreifer wählt hier also ein Vorgehen, fehlerhaft konfigurierte Programme und Dateien zu finden, um diese zur Rechteerhöhung zu missbrauchen. Die dabei genutzten Befehle finden sich entsprechend im Befehlspool des zugehörigen Astes im Angriffsbaum der Rechteerhöhung wieder (vgl. Abbildung 6.20). Der Angriffsbaum repräsentiert die notwendigen Teilziele, welche im Rahmen einer Eskalation der Nutzerrechte erforderlich sind (vgl. Kapitel 2.3). Die jeweiligen Teilziele können auf verschiedene Arten erreicht werden, bspw. kann die

[7] Ob ein Nutzer mit den vorliegenden, eingeschränkten Rechten überhaupt die Dateien der Nutzerverzeichnisse durchsuchen kann, hängt maßgeblich von der Systemkonfiguration bzw. der Voreinstellung der genutzten Distribution ab. Typischerweise ist diese auf 0755 festgesetzt, aus Sicherheitsaspekten finden sich insbesondere Änderung auf 0700. Für Ubuntu kann diese bspw. mittels des Parameters `DIR_MODE` in der Datei `/etc/adduser.conf` festgelegt werden.

Tabelle 6.9: Wahrscheinlichkeits- und Gefährdungswerte für Befehle im Rahmen der Erfüllung des Teilziels Analyse im Angriffsbaum Rechteerhöhung.

Befehl	Wahrscheinlichkeit	Gefährdung
`cat /etc/*`	0.5	0.5
`find . -perm -2 -print`	0.9	0.9
`find / -type f -perm +6000`	1.0	1.0
`ls -l` *im Verzeichnis* `/etc`	0.9	0.4
`lsmod`	0.9	0.6
`uname`	0.8	0.3
`uname -a`	0.9	0.4

erforderliche Analyse durch Auswertung von Konfigurationen oder durch die Untersuchung von Systemparametern erfolgen. Beispielhaft sind in der Abbildung eine Reihe von Befehlen angegeben, welche im Kontext der Konfigurationsanalyse genutzt werden können.

Bei der Evaluation mittels Angriffsbaum ergeben sich vier zutreffende Befehle, welche mit einer Gefährdungssumme von 3.1 gewichtet werden. Die Schwellwerte für eine Alarmierung wurden im Prototypen auf fünf Befehle respektive einer Gefährdungssumme von 3.0 festgesetzt, entsprechend wird ein Alarm bzgl. der möglichen Erreichung eines Teilziels des Angriffsschrittes U2R generiert. Von besonderer Bedeutung ist hier, dass der Angriff noch in seiner Durchführung, genauer gesagt der Analysephase ist und insbesondere die Ausführung der eigentlich zur Rechteerhöhung führenden Aktion noch nicht stattgefunden hat. Die entsprechende Alarmierung ermöglicht es also, den Angriff vor Eintreten eines schadhaften Ereignisses zu stoppen.

Abbildung 6.21 zeigt die Auswertung zweier gutartiger Nutzersitzungen durch das Sicherheitssystem. Im Falle der in Abbildung 6.21a gezeigten Evaluation identifiziert die Befehlsauswertung fälschlicherweise das Vorkommen des Befehls `lsmod`, wobei der Nutzer der betroffenen Sitzung den freien Festplattenspeicher ausgelesen hat (`df -h`). Die weitere Serie der vom Nutzer eingegebenen Befehle kann nicht identifiziert werden: Die entsprechenden Clustergrenzen ergeben keine auszuwählenden Korrelationspartner, bzw. sind ggf. vorhandene Korrelationen unter dem minimalen Schwellwert, welcher in der prototypischen Implementierung auf 0.2 festgesetzt ist. Hier durchgeführte Nutzeraktionen sind unter anderem die Navigation durch die Nutzerverzeichnisse und das Bearbeiten von Textdateien mittels Editor. Aufgrund der fehlenden Befehlskandidaten kann keine Evaluation von Angriffsteilzielen erfolgen; dahingegen wird anhand der Serie nicht identifizierter Eingaben, welche als nicht integrierte Befehle ein Indiz für gutartige Kommandos sind, eine Bewertung der Sitzung als gutartig vorgenommen.

Die Auswertung der unter Abbildung 6.21b ersichtlichen Nutzersitzung ergibt eine Identifikation von drei Befehlen mit mehreren möglichen Kandidaten. Hierbei handelt es sich zunächst um zwei fehlerhafte Analysen, während das dritte Kommando korrekt

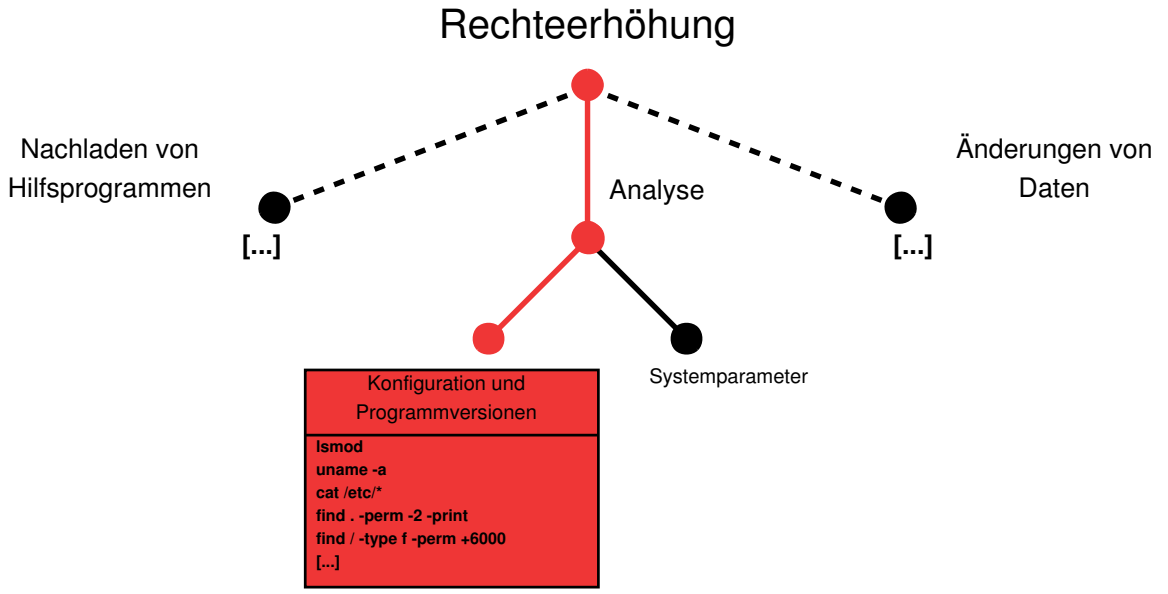


Abbildung 6.20: Erfüllung des Teilziels Analyse im Angriffsbaum der Rechteerhöhung.

als Ausgabe eines Verzeichnisses in Langform erkannt wird. Die weiteren Eingaben des Nutzers können nicht durch das System erkannt werden; nicht identifizierte Befehle stellen wiederum ein Indiz einer gutartigen Verbindung dar. Auch in diesem Fall werden keine Schwellwerte überschritten oder Teilziele erkannt, es erfolgt keine Alarmierung.

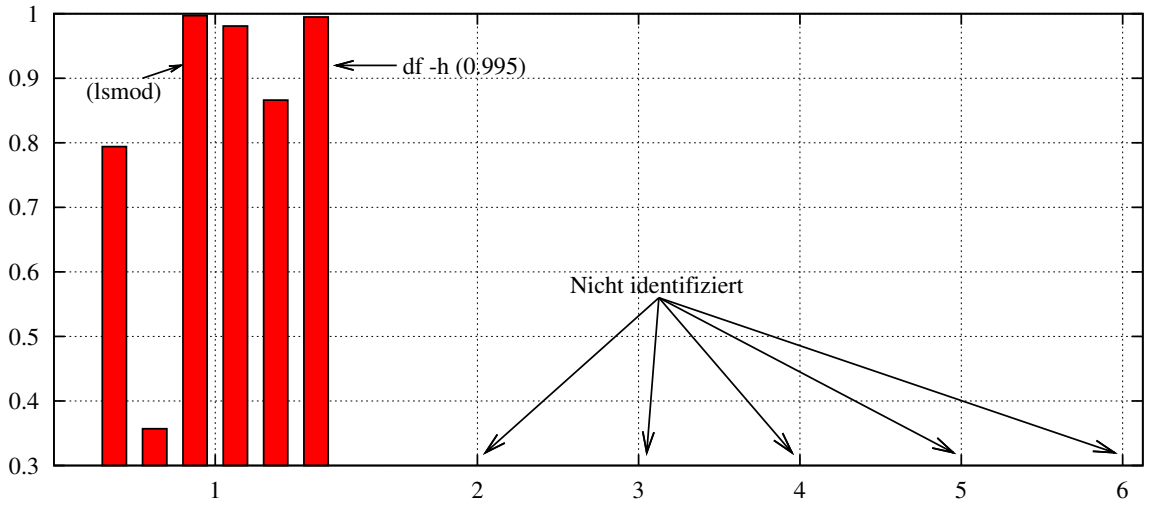
Da die Durchführung der paketsequenzbasierten Evaluation auch im Falle der vorgenommen Reduktion der Befehle mit einem großen Aufwand zur Umsetzung der benötigten Werte für die jeweiligen Systeme verbunden ist[8], wurde weiterhin eine komplett korrelationsgestützte Auswertung auf Basis der Module der Einbruchserkennung durchgeführt.

Um dies umzusetzen, wurden zunächst verschiedene Nutzersitzungen über längere Dauer aufgezeichnet, die maßgeblichen, statistischen Daten extrahiert und entsprechende Sitzungsprototypen für gutartige und bösartige Sitzungen in das Sicherheitssystem integriert.

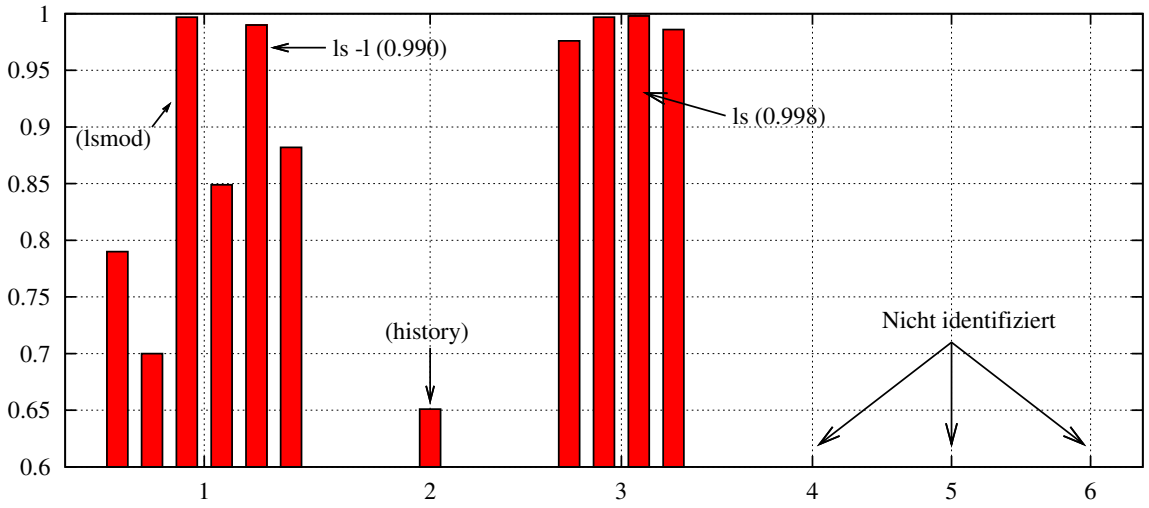
Nachfolgend werden Ergebnisse der Analyse verschiedener Sitzungen vorgestellt. Abbildung 6.22 zeigt die Entwicklung der Korrelationswerte einer bösartigen Sitzung.

Nachdem die minimale Anzahl von Paketen aufgezeichnet wurde, beginnt die Korrelation mit den im System hinterlegten, gut- und bösartigen Sitzungsprototypen. Die Grafik zeigt die kontinuierliche Entwicklung der Korrelationsergebnisse mit jeweils einer gut- und einer bösartigen Sitzung. Gut zu erkennen ist, dass die Startwerte der Korrelationen zunächst eine gewisse Dauer (im Sinne von übertragenen Datenpaketen) konstant sind, wobei eine klare Trennung zwischen den Ähnlichkeitswerten mit den gut- bzw. bösartigen Referenzen vorhanden ist. Nachdem die Sitzung weiter durchgeführt wird, sinken die Korrelationswerte beider Verbindungstypen ab; dies lässt sich in Form

[8] Insbesondere auch bei der erforderlichen Berücksichtigung der möglichen Parameter diverser Befehle.



(a) Untersuchung einer Nutzersitzung. Eine Serie nicht identifizierter Kommandos stellt ein Indiz für eine gutartige Verbindung dar, da das Sicherheitssystem insbesondere bösartige Kommandos in seiner Datenbank beinhaltet.



(b) Ausschnitt einer weiteren Nutzersitzung. Die Befehlserkennung identifiziert mehrere Eingaben, wodurch jedoch keine Schwellwerte für eine Alarmierung überschritten werden. Die nicht identifizierten Befehle stellen wiederum ein Indiz einer gutartigen Verbindung dar.

Abbildung 6.21: Analyse gutartiger Nutzersitzungen durch die Befehls- und Sequenzevaluation.

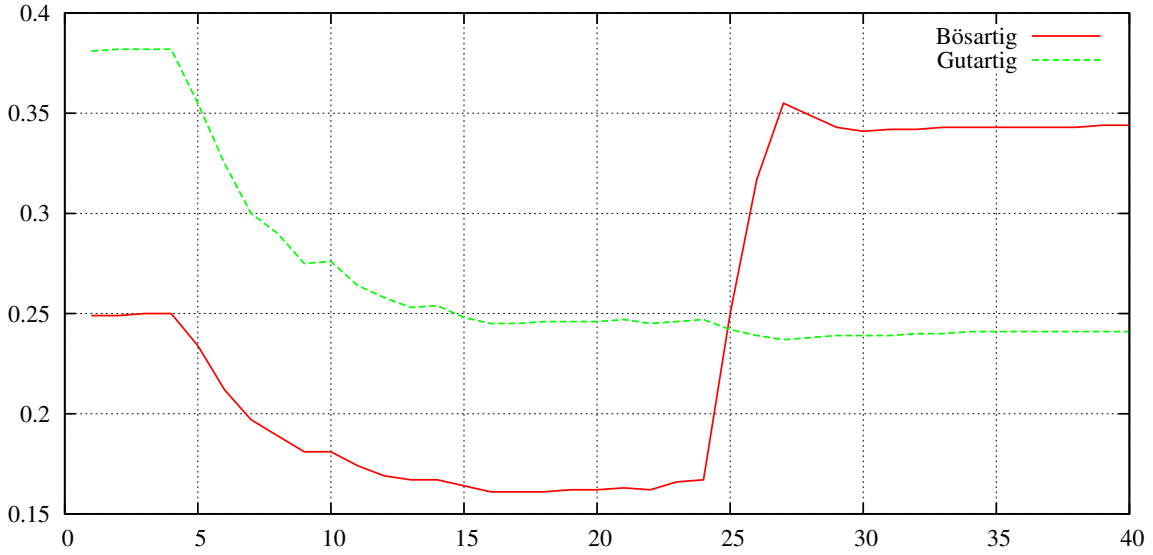


Abbildung 6.22: Befehlsevaluation mittels der Korrelation mit gut- bzw. bösartigen, prototypischen Sitzungen. Nachdem sich die Werte stabilisiert haben, erfolgt eine korrekte Klassifizierung als bösartige Verbindung. Von besonderer Bedeutung ist hier auch, dass die jeweiligen Korrelationen gut separierbar sind.

eines Einschwingvorgangs erklären, da noch nicht viele Pakete vorliegen und die Charakteristika der Verbindung somit noch nicht ausreichend ausgeprägt sind. Während sich die Korrelationswerte mit der gutartigen Referenzverbindung auf ein stabiles Niveau mit einem Ähnlichkeitswert von ca. 0.245 einpendeln, steigt der Korrelationswert mit dem bösartigen Prototypen nach einiger Zeit rapide an, was auf die Übereinstimmung eines oder mehrerer, für die bösartige Verbindung charakteristischer Befehle hinweist. Nach dem starken Anstieg des Korrelationswertes mit der Angreiferverbindung stabilisiert sich auch dieser Wert, auf einem Niveau von 0.345. Dies entspricht der korrekten Bestimmung der analysierten Datenverbindung, bei der es sich um die Durchführung eines Angriffs handelt.

Abbildung 6.23a zeigt ein weiteres Beispiel der Auswertung einer bösartigen Sitzung, wobei hier ein anderes Korrelationsverfahren, welches ebenfalls in das Modul zur Befehlserkennung integriert wurde, genutzt wird. Nachdem die ersten Korrelationswerte vorliegen, befinden sich die Ähnlichkeitswerte mit der gutartigen Referenzverbindung bereits auf einem sehr geringen Niveau, welches im weiteren Verlauf noch leicht absinkt und sich dann wiederum auf einem konstanten Wert einpendelt. Im Vergleich hierzu zeigt die Entwicklung der Korrelationen mit dem bösartigen Prototypen einen Verlauf, der ähnlich dem bereits bekannten einer bösartigen Verbindung ist (vgl. Abbildung 6.22). Auch hier startet der Korrelationswert zunächst unterhalb des Wertes der gutartigen Verbindung und steigt nach einiger Zeit stark an; die beiden Kurven sind auch hier wieder von Beginn an deutlich separiert. Nach dem Anstieg stabilisiert sich die Korrelation mit der bösartigen Verbindung auf einen Wert von ca. 0.12, was dem korrekten

Sitzungstyp entspricht.

Abbildung 6.23b zeigt zusätzlich die Entwicklung der am Ende der Evaluation ausgegebenen Korrelationen bei Nutzung einer entsprechender Verschiebung. Das Ergebnis der vorherigen Evaluation ändert sich hierdurch nicht: Gut erkennbar ist, dass der Korrelationswert mit der bösartigen Verbindung bei optimaler zeitlicher Verschiebung der Paketserien zueinander sogar noch weiter ansteigt. Insbesondere sind die Korrelationswerte mit der gutartigen Verbindung jederzeit unterhalb der von der bösartigen Verbindung stammenden Werten.

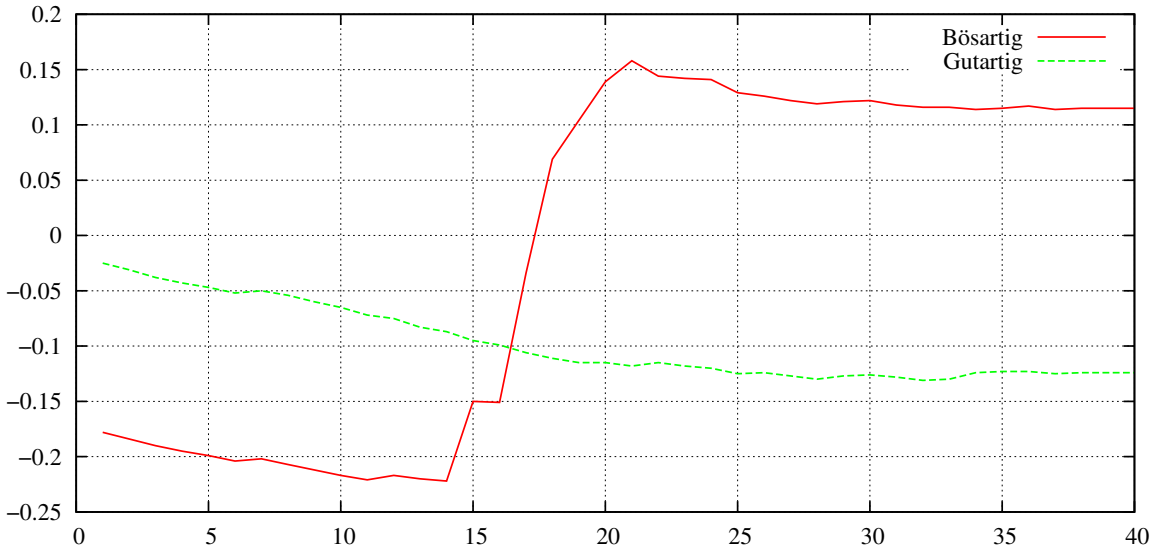
Der Verlauf der Analyse beim Vorliegen einer gutartigen Nutzerverbindung ist in Abbildung 6.24a dargestellt. Von Beginn an ist eine sehr starke, konstant verlaufende Separierung zwischen den Ähnlichkeitswerten mit der gut- bzw. bösartigen Verbindung erkennbar. Während die Evaluation mit einem bösartigen Prototypen Werte unter 0 ergibt, liegen die Werte für die gutartige Verbindung konstant über 0.6. Betrachtet man die Entwicklung der Korrelationswerte bei Nutzung einer zeitlichen Verschiebung zeigt sich, dass die Lage bereits optimal ist; trotz deutlich geringerer Abstände bleiben auch hier die Werte der bösartigen Verbindung fast vollständig unterhalb denen der gutartigen Verbindung (vgl. Abbildung 6.24b).

Die Evaluation einer weiteren gutartigen Sitzung ergibt den in Abbildung 6.25a dargestellten Verlauf. Auch hier starten die Korrelationswerte deutlich separiert und verlaufen konstant, wobei sich der Abstand zwischen den Werteverläufen sogar noch etwas vergrößert. Das Ergebnis der Evaluation ordnet die untersuchte Sitzung eindeutig und korrekt als gutartige Nutzerverbindung ein. Zu beachten ist hierbei jedoch, dass der absolute Unterschied zwischen den Werten der beiden Kurven sehr viel geringer ist, als dies im vorherigen Beispiel der Fall war (vgl. Abbildung 6.24a). Entsprechend muss wiederum die Entwicklung der Korrelationswerte bei Verschiebung der Sitzungsdaten betrachtet werden (vgl. Abbildung 6.25b). Hier zeigt sich, dass die Korrelationswerte insbesondere der guten Verbindung bei der richtigen Lage deutlich ansteigen, von 0.03 als Evaluationsergebnis in der Ausgangslage auf knapp 0.7. Auch hier befindet sich die Korrelation mit der Angriffsverbindung nahezu ausschließlich unterhalb der gutartigen Korrelation und weist deutlich geringere Maximalwerte auf. Die Überprüfung der Verschiebung unterstreicht also auch in diesem Fall die korrekte Klassifizierung der analysierten Sitzung.

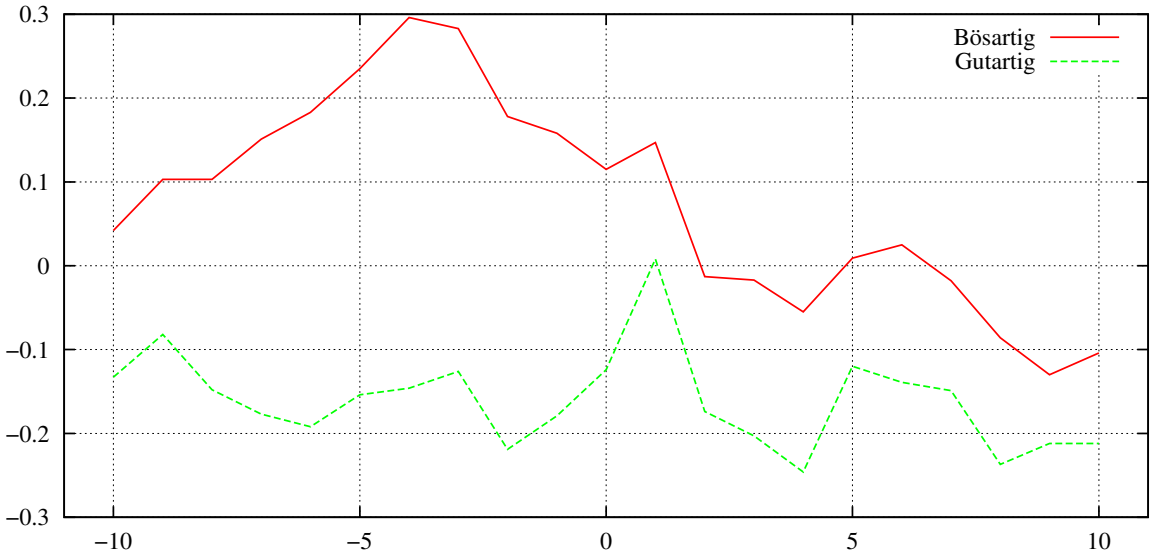
Tabelle 6.10 fasst die Ergebnisse der Befehlsidentifizierung zusammen. Hier muss angemerkt werden, dass im Gegensatz zu den Auswertungen der anderen Module, hier nur Richtwerte angegeben werden können. Die Erkennung der genutzten Befehle ist Grundlage für die Bestimmung des Sitzungscharakters mittels der Prüfung der Erreichbarkeit von Teilzielen in den jeweiligen Angriffsbäumen.

Die Detektionsraten einzelner Befehle sowie insbesondere die Fehlalarmraten hängen stark vom jeweils umgesetzten bzw. nicht berücksichtigten Satz an Befehlen ab.

Die in der vorliegenden Arbeit identifizierte Auswahl von Befehlen gibt einen möglichen Befehlspool für im Rahmen der Durchführung eines Angriffs wichtiger Befehle vor. Basierend auf den Evaluationen zeigt sich, dass eine eingeschränkte Befehlsimplementierung für eine Angriffsdetektion ausreichend sein kann, einzelne, weniger charakteristische Befehle wie bspw. `ls -l` jedoch schwer identifizierbar sein können und nur im Rahmen der Betrachtung ganzer Befehlssequenzen korrekt interpretierbar sind. Insbe-

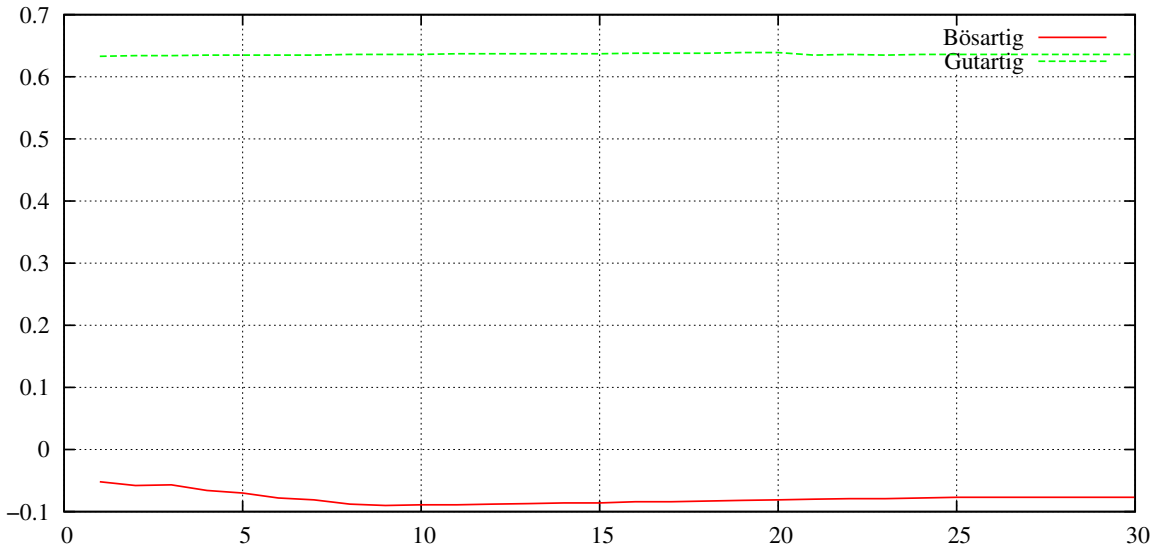


(a) Verlauf der Korrelationswerte mit den prototypischen Sitzungen. Es erfolgt eine korrekte Klassifizierung als bösartige Verbindung.

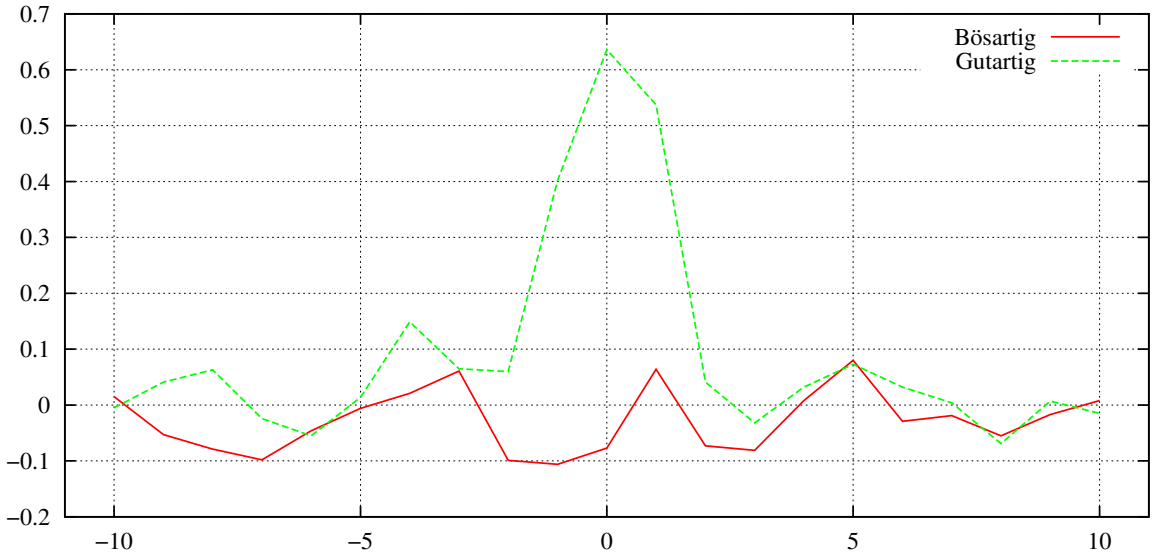


(b) Entwicklung der Korrelationswerte nach Stabilisierung bei Anwendung einer zeitlichen Verschiebung der Paketserien. Unter optimaler Lage steigt der Korrelationswert mit dem Angreiferprototypen stark an, die Werte für die gutartigen Korrelationen liegen stets deutlich unterhalb.

Abbildung 6.23: Befehlsevaluation mittels Sitzungskorrelation unter Nutzung eines anderen Korrelationsverfahrens.

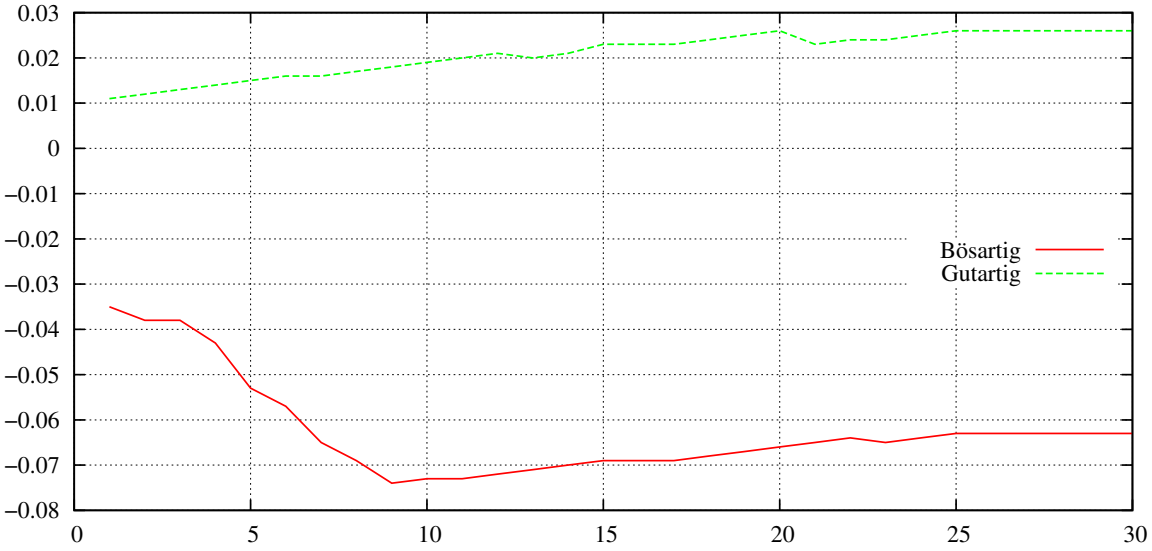


(a) Die Korrelationswerte sind von Beginn an stark separiert und stabil, die Klassifizierung als gutartige Sitzung ist korrekt.

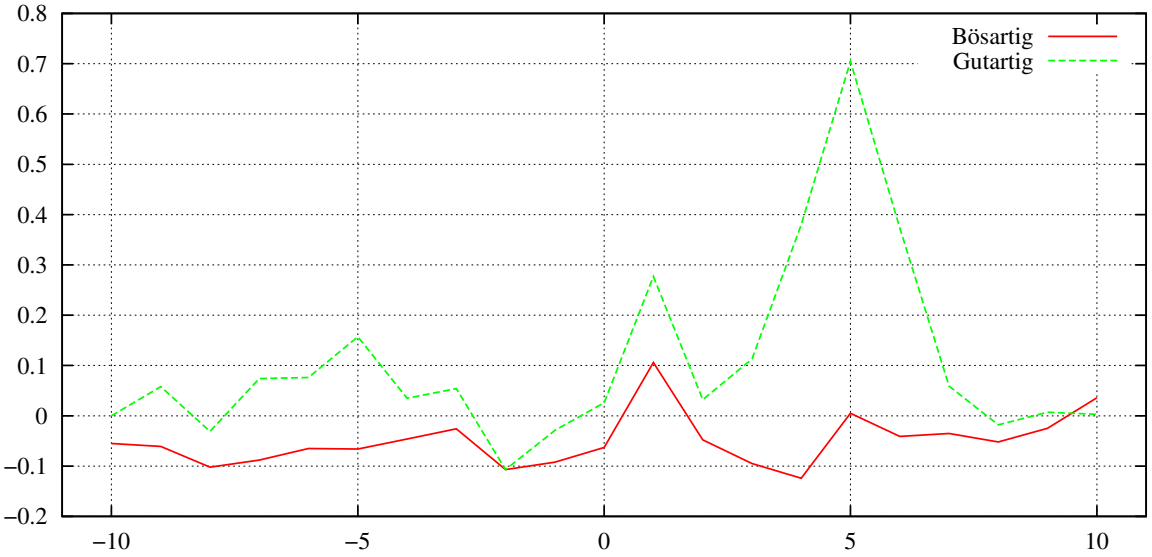


(b) Die Verschiebung der Sitzungsdaten zeigt, dass die Lage bereits optimal ist. Da die bösartigen Werte auch bei Verschiebung fast permanent unterhalb der gutartigen Verbindung liegen, stärkt dies die Klassifizierung als gutartige Sitzung nochmals.

Abbildung 6.24: Durchführung der Sitzungskorrelation bei Vorliegen einer gutartigen Nutzersitzung.



(a) Korrelation einer gutartigen Verbindung. Wieder zeigt sich das charakteristische Bild und eine scharfe Trennung zwischen den beiden Kurven. Die Klassifizierung erfolgt wiederum korrekt. Zu beachten sind jedoch die geringen absoluten Abstände der beiden Kurven, welche aus der Ausgangslage der Paketserien zueinander resultieren.



(b) Bei optimaler Lage der Paketserien wird die Einordnung als gutartige Nutzersitzung unterstrichen: Die Werte der Korrelation steigen stark an, während diejenigen mit der bösartigen Verbindung auf konstant geringem Niveau verbleiben.

Abbildung 6.25: Weiteres Beispiel einer gutartigen Sitzungskorrelation.

Tabelle 6.10: Detektions- und Fehlalarmraten bei der Befehlsevaluation auf Basis des genutzten Befehlssatzes und mehrerer Angriffs- und Nutzersitzungen. Bei den Nutzersitzungen sind lediglich 25 Prozent der verwendeten Befehle im Detektionssystem bekannt, von den durch Angreifer genutzten Befehlen 70 Prozent. Angegeben ist, welcher Anteil der Befehle korrekt erkannt wurde, fehlerhaft gemeldet wurde oder ohne Ergebnis verworfen wurde. Angaben in Prozent.

	Angriffsbef.	Erkannt	Fehlerhafte	Nicht Ident.
Angreifer	70	60 (Top 3: 80)	20	20
Nutzersitzung	25	15	35	50

sondere kann im Rahmen eines reduzierten Befehlssatzes ein Befehl, dessen Verwendung auch für einen normalen Nutzer typisch ist (wie bspw. `ls`), eher weggelassen als in die Befehlsdatenbank aufgenommen werden, da eine fehlerhafte Interpretation nachteiliger ist, als dies durch einen nicht identifizierten Befehl der Fall ist. Auf der anderen Seite hingegen müssen charakteristische Befehle wie bspw. die Suche nach Dateien mit bestimmten Rechten umfassend berücksichtigt werden. Durch die Vielzahl an Parametern, die manche Befehle zur Verfügung stellen, ist dies entsprechend aufwändig. Die gezeigte Nutzung von Referenzsitzungen stellt hier eine mögliche Lösung dar.

6.3.2 Nutzeridentifizierung

Nachfolgend werden Beispiele der Auswertung der verschiedenen genutzten Parameter der Tippbiometrie sowie Analyseergebnisse von Nutzersitzungen vorgestellt. Auf Basis der Datenpakete, welche die Sonde im Netz passieren, werden mittels des Timings und der charakteristischen Paketgrößen (vgl. Kapitel 5.1.3) die benötigten, biometrischen Tippmerkmale des Nutzers der Verbindung gewonnen.

Abbildung 6.26 zeigt das Ergebnis der Kreuzkorrelation des Timings der aufgezeichneten Daten mit dem Profil eines falschen Nutzers. Ausgewertet werden hier die Zeiten des tippbiometrischen Merkmals *Tippgeschwindigkeit*. Gut zu erkennen sind die geringen Korrelationswerte von meist unter 0.2, ebenso fallen die unregelmäßigen Korrelationsmaxima bei der Verschiebung Δ_t auf. Die entsprechenden, geringen Korrelationswerte gehen als einer der nutzbaren Parameter der Tippbiometrie in die Gesamtwertung ein.

Das Ergebnis einer Korrelation des selben Parameters (Tippgeschwindigkeit) mit dem korrekten Profil des tatsächlichen Nutzers der Verbindung ergibt hohe Korrelationswerte wie in Abbildung 6.27 ersichtlich. Gut zu erkennen ist ein hoher Korrelationswert bei der korrekten zeitlichen Verschiebung, das Maximum wird hier bei $t = 0$ erreicht. Neben diesem hohen Korrelationswert liegen auch in weiteren Punkten der zeitlichen Verschiebung hohe Werte vor. Von besonderer Bedeutung ist ebenfalls die regelmäßige Struktur, die sich für die Korrelationswerte mit dem korrekten Profil während der gesamten zeitlichen Verschiebung ergibt. Wie bereits gesehen entstehen hier unregelmäßige Strukturen, wenn ein falsches Nutzerprofil für die Korrelation herangezogen wird.

Weitere Korrelationen mit verschiedenen Nutzerprofilen sind in Abbildung 6.28 er-

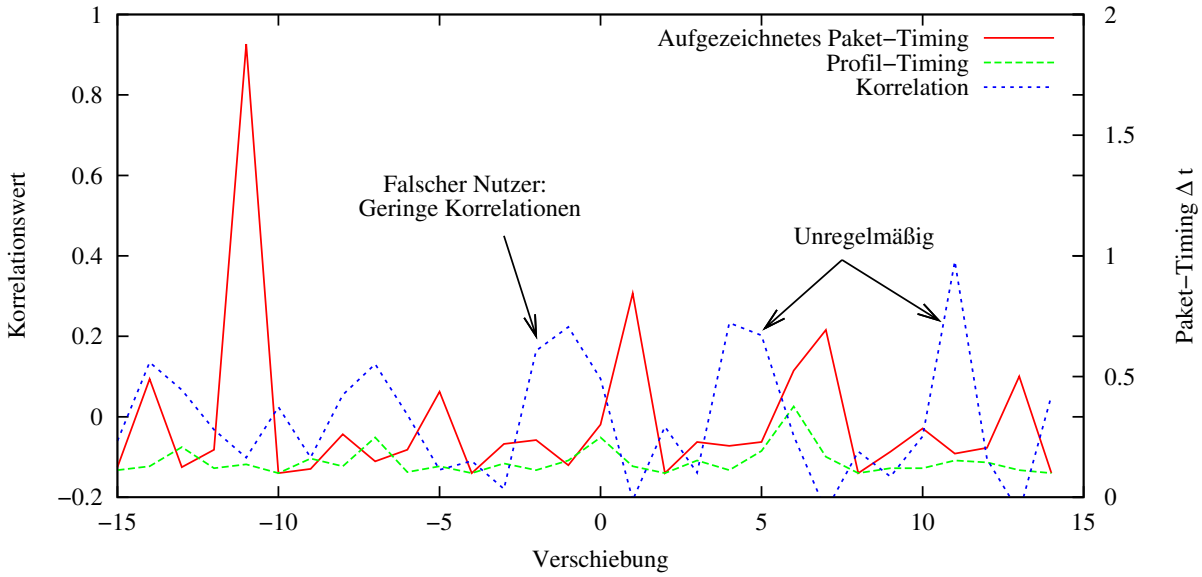


Abbildung 6.26: Korrelation der ermittelten Paket-Zeiten mit dem Nutzer-Profil eines Unbeteiligten. Die Korrelationswerte sind hauptsächlich gering und ergeben ein unregelmäßiges Ergebnis.

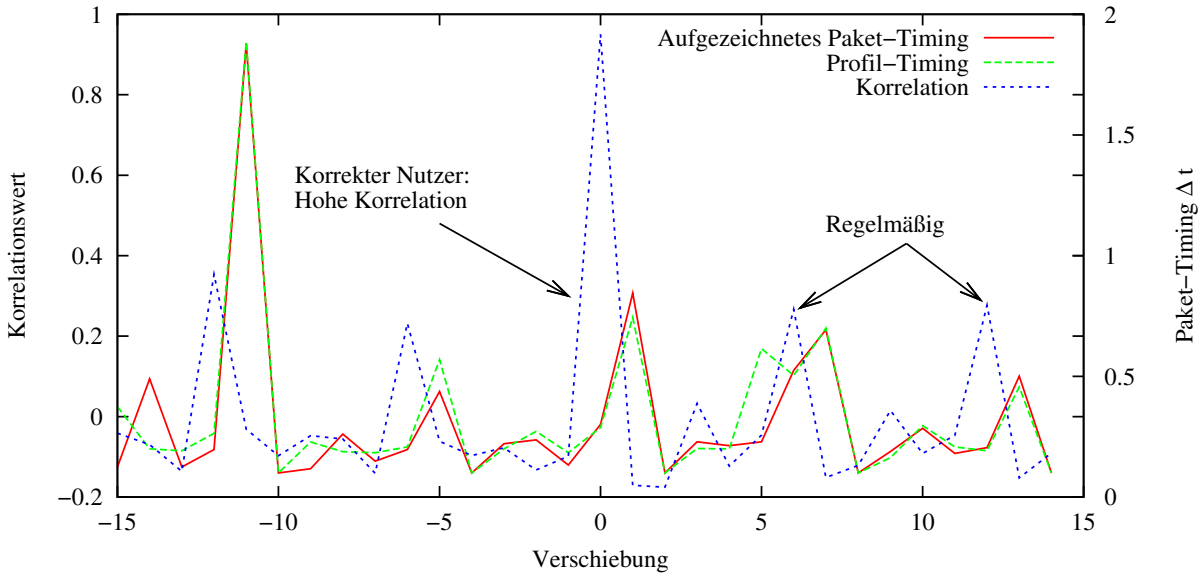


Abbildung 6.27: Korrelation der ermittelten Paket-Zeiten mit dem korrekten Nutzer-Profil. Die Korrelationswerte sind bei der optimalen Verschiebung hoch und ergeben über die gesamte Verschiebung ein regelmäßiges Profil.

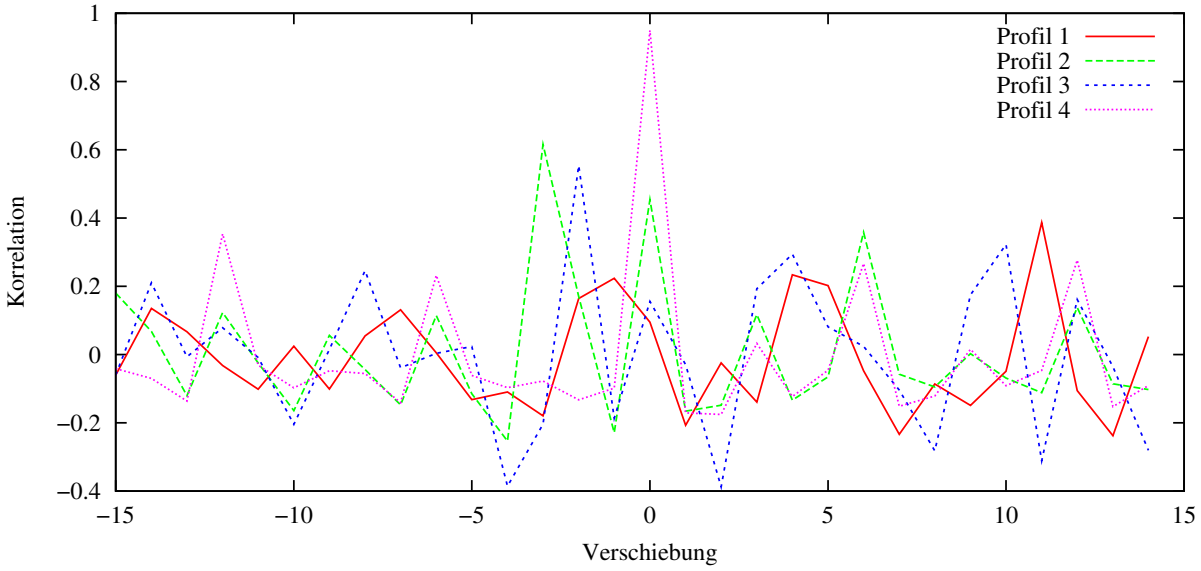


Abbildung 6.28: Korrelation der ermittelten Paket-Zeiten mit verschiedenen Nutzer-Profilen. Das Profil des tatsächlichen Nutzers zeigt bei der Verschiebung wiederholt hohe Korrelationswerte und ein regelmäßiges Profil, weiterhin erreicht es bei optimaler Verschiebung mit Abstand den höchsten Korrelationswert.

sichtlich. Zur Gegenüberstellung werden hier das korrekte, bereits bekannte Profil (in der Abbildung als Nummer vier gekennzeichnet), sowie drei nicht zutreffende Nutzer evaluiert. Die gute Unterscheidbarkeit des Parameters ist für alle Profile gut erkennbar, das korrekte Profil hat mit Abstand die höchsten Korrelationswerte bei der optimalen zeitlichen Verschiebung und weist insbesondere auch als einziges Profil eine regelmäßige Struktur bei der zeitlichen Verschiebung auf. Bei nicht-optimalen Verschiebungen können die Korrelationswerte nicht-korrekter Nutzer diejenigen des zutreffenden Profils leicht übertreffen. Um trotzdem eine korrekte Identifikation des Nutzer zu erreichen, werden in den Evaluationsprozess die während der Beschreibung des Moduls in Kapitel 5.1.5 vorgestellten, weiteren tippbiometrischen Parameter einbezogen.

Dies ist insbesondere auch deshalb erforderlich, da die Auswertung eines einzelnen oder einer zu geringen Zahl tippbiometrischen Parameter mit einer steigenden Anzahl von Nutzern nicht skaliert, sondern in sinkenden Erkennungsraten resultiert. Aus diesem Grunde werden zusätzliche Parameter ausgewertet: In Abbildung 6.29 werden die Evaluationen der maximalen und minimalen Verzögerungen, die durchschnittlichen Anschläge sowie der Profilverzögerung dargestellt. Die Notwendigkeit der Nutzung mehrerer, verschiedener Parameter wird hier nochmals klar: Während der korrekte Nutzer der Verbindung (Profil vier) in den meisten Fällen richtig identifiziert wird, ergibt die Auswertung des Parameters *Maximale Verzögerung* ein umgekehrtes Bild und eruiert fälschlicherweise das Profil eins als Nutzer. Insbesondere erreicht das Profil eins bei mehreren anderen Parametern deutlich geringere Werte im Vergleich zu den weiteren Profilen. Dies zeigt, dass einzelne Parameter der Tippbiometrie nicht ausreichend sind,

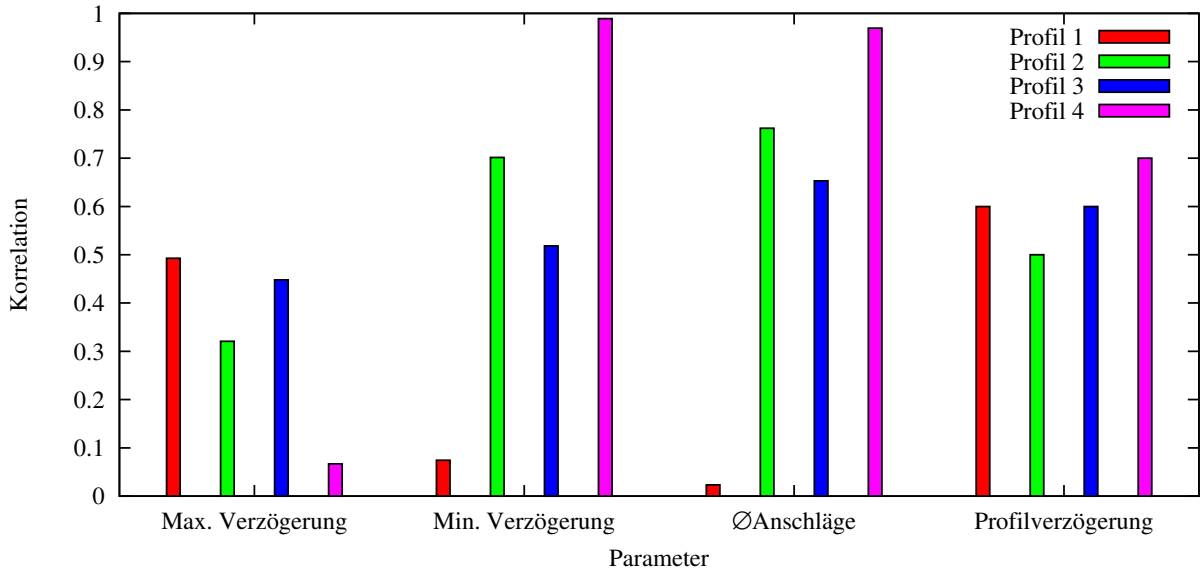


Abbildung 6.29: Ergebnisse der Korrelationen von Tippverhaltens-Parametern für verschiedene Nutzerprofile.

eine eindeutige Klassifizierung von Nutzern durchzuführen. Wird dahingegen eine geeignete Kombination mehrerer Parameter wie gezeigt gewählt, können entsprechend hohe Klassifizierungsraten erreicht werden.

Neben der Korrelation der aufgezeichneten Werteserien der Datensonde mit den durch die Profile gespeicherten Daten werden weiterhin absolute Differenzen berechnet, um den Identifikationsprozess zu verbessern. Die jeweiligen Ergebnisse der Parameter maximale und minimale Verzögerungen sowie der durchschnittlichen Anschläge sind in Abbildung 6.30 gezeigt. Da es sich bei diesem Evaluationsschritt um die Betrachtung absoulter Differenzen zwischen registrierten und aufgezeichneten Werten handelt, stellen kleine Werte ein höheres Indiz für die Zugehörigkeit eines Profils dar, als große Werte. Erst durch die Kombination aller genannten Parameter kann eine korrekte Identifizierung mit geringen Fehldetektionswahrscheinlichkeiten erfolgen.

Die Detektionswahrscheinlichkeiten bei der Evaluation von Abschnitten eines knapp über 30 Sekunden langen Ausschnitts einer verschlüsselten Remote-Verbindung sind in Tabelle 6.11 dargestellt. Gut zu erkennen ist zum einen die deutliche Verbesserung, welche durch die Hinzunahme der zusätzlichen Parameter in Bezug auf die Erkennungsraten der Nutzer erreicht wird. Clusterverifikation bezeichnet hier die Durchführung einer Prüfung der korrekten Grenzen der Cluster vor den entsprechenden Korrelationen und Berechnungen. Dies führt nochmals zu einer erheblichen Verbesserung der Daten und ist wie folgt begründet: Kommt es durch Tippfehler beim Nutzer zu fehlerhaften Eingaben, kann ggf. der korrekte Cluster nicht mehr erkannt werden (vgl. Kapitel 5.1.5), was zu einer Verschiebung von Grenzen und dem Ausschluß von Paketserien führen kann. Ist jedoch eine Detektion des bzw. der Tippfehler möglich und können somit die korrekten Grenzen des Cluster gefunden werden, bleiben die Tippfehler des Nutzers mit in den

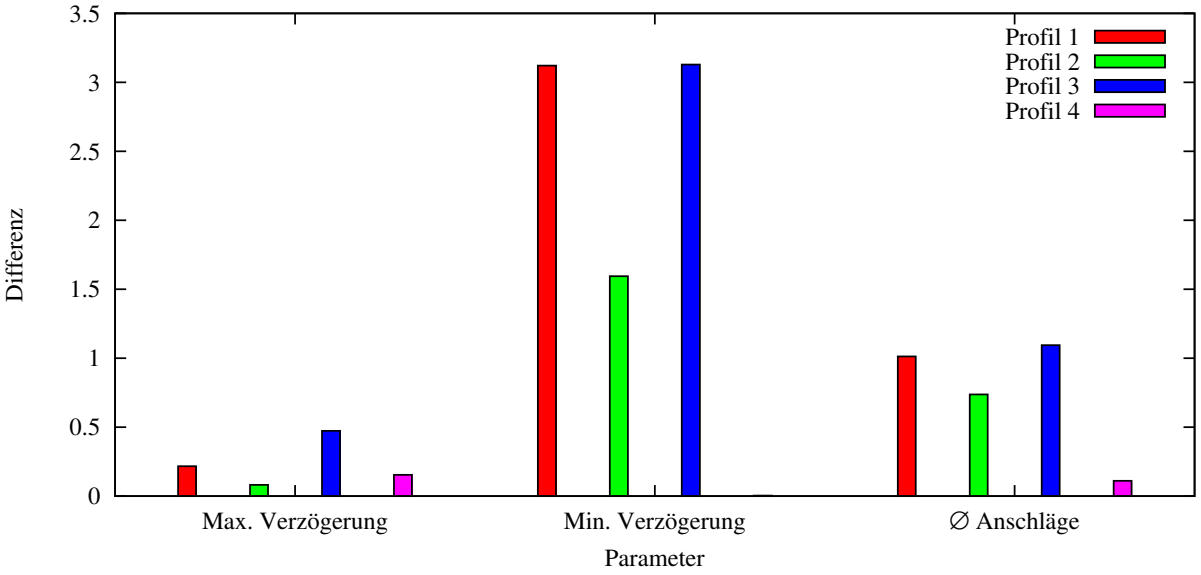


Abbildung 6.30: Absolute Differenzen der beobachteten zu den registrierten Werten verschiedener Tippverhaltens-Parameter.

Tabelle 6.11: Klassifizierungsleistung des S2E2 Modul-Prototypen zur Identifikation von Nutzern. Profile von vier Nutzern wurden herangezogen und 16 Sitzungen ausgewertet.

		Anschläge
	Cluster	und Differenzen
Ohne Clusterverifikation	50 %	87,5 %
Mit Clusterverifikation	75 %	93,75 %

jeweils für die Korrelationen verwendeten Paketsequenzen erhalten. Dies bedeutet, dass die biometrischen Informationen in Bezug auf Tippfehler und deren Korrekturen nicht verloren gehen und somit ausgewertet werden können, was zu einer besseren Deckung mit den Profilen in der Datenbank führt.

Da die Auswertung weiterhin fortlaufend auf dem Datenstrom einer bestehenden, authentisierten Verbindung erfolgt, steigt die Detektionswahrscheinlichkeit mit der Länge des betrachteten Sitzungsintervalls. Dies kann insbesondere genutzt werden, um Unsicherheiten bei ähnlichen Korrelationswerten zu verbessern.

6.4 Zusammenfassung

Das Kapitel führt eine Evaluation der prototypischen Implementierung des Sicherheitssystem, das auf der im Kapitel 5 vorgestellten Architektur beruht, durch. Hierfür wird

zunächst die Leistungsfähigkeit der Datensonde zur Aufbereitung des Datenstroms und Gewinnung der statistischen Daten analysiert. Anschließend werden die Module zur Einbruchserkennung, das Modul zur schnellen Brute Force-Erkennung sowie zur TLS-Angriffsdetektion evaluiert, gefolgt von den Modulen zur Ausbruchserkennung (Befehlsevaluation und Nutzeridentifizierung). Die hierbei genutzten Verfahren werden vorgestellt und diskutiert. Es wird gezeigt, dass die vorgestellten Module zur Ein- und Ausbruchserkennung in der Lage sind, Angriffe in verschlüsselten Umgebungen sowie Innentäter ohne die Notwendigkeit einer Entschlüsselung des Datenverkehrs zu erkennen.

Am schwersten sind die Entscheidungen, deren Auswirkungen in der Zukunft liegen. Und das sind alle.

Thomas M. Scheerer

7 Zusammenfassung und Ausblick

Das vorliegende Kapitel fasst die Ergebnisse der Arbeit zusammen (Kapitel 7.1) wobei die Eignung der präsentierten Architektur mit Hinblick auf den Anforderungskatalog beurteilt wird (Kapitel 7.2). Die im Rahmen der Motivation eröffneten Fragestellungen werden aufgegriffen und zusammenfassend beantwortet (Kapitel 7.3). Abschließend gibt Kapitel 7.4 mögliche, künftige Arbeitsrichtungen im Gebiet der verschlüsselten Einbruchserkennung an.

7.1 Einordnung der wissenschaftlichen Fragestellungen

Die Einbruchserkennung in Rechnernetzen ist seit über 30 Jahre in der Forschung und zahlreiche Systeme sind auf dem Markt verfügbar. Trotzdem steigt die Anzahl von Sicherheitsvorfällen kontinuierlich an. Fast täglich wird über entsprechende Vorfälle berichtet, im Rhythmus von wenigen Monaten sind hierbei immer wieder aufsehenerregende Fälle, wie bspw. die jüngsten Vorfälle von Angriffen und Datendiebstahl bei RSA und den damit ermöglichten Angriffen auf Lockheed Martin, Northrop Grumman und weiteren Unternehmen der amerikanischen Verteidigungsindustrie. Um zu ermitteln, warum derzeitige Sicherheitssysteme regelmäßig nicht in der Lage sind, Angriffe und Einbrüche zu verhindern, wurden in Kapitel 2 zunächst die Bedrohungen für die Kommunikation anhand eines Unternehmensszenarios untersucht. Die Rolle des Innentäters wurde explizit betrachtet, da dieser eine erhebliche Auswirkung auf ein Sicherheitssystem im vorliegenden Szenario hat: Mittels der Autorisierungen, die er für das Zielsystem besitzt, kann ein Zugriff und die Durchführung unerwünschter Operationen einfach erfolgen. Andererseits stellt sich auch ein externer Angreifer, der bspw. mittels Social Engineering-Techniken Zugriff auf ein Netz erlangt, im Sinne der Angriffserkennung auf die gleiche Position wie ein Innentäter. Eine anschließende, detaillierte Angriffsanalyse untersuchte das Vorgehen und die zu einem Angriff gehörenden Teilschritte (vgl. Kapitel 2.3). Dies ist zum einen erforderlich, um Detektionsmöglichkeiten für ein Sicherheitssystem zu evaluieren, andererseits lässt sich anhand des aufgestellten Modells der Angriffsschritte ableiten, wieso heutige Sicherheitssysteme kein probates Mittel für eine entsprechende Detektion sind: Die zunehmende Nutzung von zielgerichteten Angriffen und der Anwendung von Social Engineering-Techniken vereiteln eine Erkennung durch heutige Systeme. Selbst State-of-the-Art Systeme, welche auf verhaltenbasierten Detektionsschemata basieren, können unter diesen Bedingungen keinen ausreichenden Schutz gewährleisten. Anhand der Bedrohungen und den Folgerungen aus der Angriffsanalyse wurde in Kapitel 3 ein Anforderungskatalog an ein System zur Ein- und Ausbruchserkennung der nächsten Generation aufgestellt. Die Leistungsfähigkeit bestehender State-of-the-Art Systeme sowie

aktueller Forschungsarbeiten aus dem Bereich der Einbruchserkennung wurde untersucht und mittels des Kriterienkataloges in Kapitel 4 beurteilt. Die Analyse zeigt, dass die Systeme und Verfahren insbesondere nicht geeignet sind, Innentäter oder Angriffe in verschlüsselten Umgebungen zu detektieren. Eine ausführliche Untersuchung der hierbei festgestellten, offenen Punkte der jetzigen Verfahren wurde anschließend vorgenommen und vorhandene Ansätze der Forschung, welche entsprechende Lücken adressieren, vorgestellt. Anhand der Anforderungen wurde in Kapitel 5 eine neue Architektur zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen vorgestellt, welche die identifizierten Lücken bestehender Systeme schließt. Hierbei werden verschiedene Verfahren entwickelt, um bösartige Verbindungen erkennen zu können. Module für die Einbruchserkennung setzen hierfür Korrelationen innerhalb der statistischen Daten einer Verbindung ein, bzw. korrelieren die Daten verschiedener Nutzersitzungen miteinander. Bei der Ausbruchserkennung werden Befehle hinter einer verschlüsselten Verbindung identifiziert und anhand von Angriffsbäumen die Nutzerstrategie abgeleitet. Die Angriffsbäume sind das Ergebnis der Angriffsanalyse und beinhalten Sätze von Befehlen, mit denen entsprechende Angriffsteilziele durchgeführt werden können; kann ein Teilziel erreicht werden, erfolgt eine Alarmierung. Der Prozess der Befehlsevaluation wird durch eine tippbiometrische Analyse des Nutzers unterstützt. Möglichkeiten, wie mittels des Wissens um das Vorhandensein des neuen Sicherheitssystems durch einen Angreifer versucht werden kann, dieses zu unterwandern, wurden in Kapitel 5.2 untersucht. Von besonderer Bedeutung war hierbei die Feststellung, dass eine versuchte Unterminierung des Systems typischerweise zu wiederum detektierbaren Ereignissen führt: Zwar wird in solchen Fällen ggf. kein korrekter Alarm generiert, wie er eigentlich vorgesehen ist, das Verhalten des Angreifers erzeugt jedoch gleichzeitig Anomalien in der Auswertung des Sicherheitssystems, die für eine Alarmierung herangezogen werden können. Die neue Architektur zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen wurde anschließend in Kapitel 6 evaluiert. Die hierbei genutzten Verfahren, zum einen Simulationen, andererseits auch Messungen in einem produktiven Netz wurden vorgestellt und jedes Teilmodul für sich detailliert erprobt und ausgewertet. Anhand der Evaluation wurde gezeigt, dass die neue Architektur in der Lage ist, Angriffe und Innentäter zu identifizieren. Hierbei werden keine Verhaltensmodelle oder Lernphasen benötigt: Die Verfahren arbeiten direkt auf dem aktuellen Daten der Verbindungen, bzw. sind alle von manchen Modulen benötigten Referenzdaten im Vorfeld in einer sicheren Umgebung erzeugt worden. Dies ermöglicht einen direkten Start des System ohne unsichere Lernphasen.

7.2 Bewertung der Architektur zur Ein- und Ausbruchserkennung (S2E2)

Nachfolgend werden die einzelnen Punkte des Anforderungskataloges aus Kapitel 3 aufgenommen und die vorgestellte Architektur auf deren Erfüllung hin untersucht.

Detektion von Angriffen Die Detektion von Angriffen ist die grundlegende Forderung für die Architektur zur Ein- und Ausbruchserkennung. Hierfür wurden zwei, auf unterschiedlichen Verfahren beruhende Teilstränge der Architektur vorgestellt und implementiert. Die anschließend angestellten Evaluationen haben gezeigt, dass die Architektur in der Lage ist, Angriffe mit hohen Detektionsraten zu erkennen; dies gilt sowohl für die Ein- als auch die Ausbruchserkennung. Von besonderer Bedeutung ist hier, dass der gezielte Versuch, der Detektion zu entgehen, zu sekundären Anomalien führt, die wiederum als Indiz für einen Angriff herangezogen werden können.

Unterbinden von Angriffen Sowohl aus Gründen der Effizienz bzw. Machbarkeit, als auch wegen der erforderlichen, umgehenden Reaktion im Falle eines Angriffs ist eine automatische Reaktion erforderlich. Dies wurde im Rahmen der vorgestellten Architektur sichergestellt, indem Funktionen zur Generierung und Verwaltung von Firewallregeln für den *iptables*-Paketfilter implementiert wurden. Diese können auf Anforderung eines Moduls bei Generierung eines Alarms eine entsprechende Regel zur Blockierung einer IP-Adresse erzeugen und an die Firewall weitergeben. Durch die Ausführung des Sicherheitssystems als transparente Brücke, konnte die Firewall direkt mit integriert werden.

Erkennen von Innentätern Die Bedeutung des Innentäters wurde im Laufe der Arbeit herausgestellt, was die Forderung nach einer entsprechenden Detektierbarkeit nach sich zieht. In der Architektur wurde dies durch die Integration der Module für die Ausbruchserkennung realisiert. Da Innentäter mittels ihrer Autorisierung des Systemzugangs typischerweise nicht anhand von Signaturen und nur eingeschränkt anhand einer Verhaltensevaluation detektiert werden können, wurden für diese Aufgabe zwei autarke Verfahren integriert: Die Intention eines Nutzers wird anhand dessen Eingaben rekonstruiert, hierbei dienen Angriffsbäume zur Identifikation von bösartigen Arbeitsschritten. Als zweites Verfahren wurde eine Nutzeridentifikation auf Basis der Analyse von tippbiometrischen Eigenschaften, welche aus dem verschlüsselten Datenverkehr wiedergewonnen werden, implementiert.

Transparente Integration Die Forderung der transparenten Integration wurde mittels der Umsetzung des Sicherheitssystems als transparente Brücke vorgenommen. Dies ermöglicht die Integration in bestehende Netze ohne Anpassung oder Veränderung von Netz- oder Systemstrukturen: Das Sicherheitssystem kann beliebig in eine bestehende Verbindung integriert werden, bevorzugt dicht am Netzübergang zur externen Verbindung.

Wartungsarmes System Das Sicherheitssystem kann direkt und ohne Konfiguration eingesetzt werden. Weiterhin sind weder Lernphasen noch die regelmäßige Pflege von Signaturdatenbanken erforderlich; sämtliche erforderlichen Daten werden vom System mitgebracht bzw. im Betrieb selbst generiert. Hierfür sind keine Konfigurationen, weder bzgl. der Netzumgebung noch bzgl. der Installation, notwendig. Im Falle der Notwendigkeit einer Aktualisierung der Datenbank, bspw. bei Veröffentlichung eines neuen

Betriebssystems, genügt ein einmaliges Update dieser, welches auch automatisch durchgeführt werden kann.

Nahe-Echtzeitauswertung Eine umgehende Reaktion bei Erkennung eines Angriffes ist erforderlich, um einen möglichen Schaden zu verhindern bzw. mindestens so weit wie möglich zu beschränken. Entsprechend darf eine Evaluation der zu bewertenden Daten keine aufwändigen Algorithmen erfordern, sondern muss möglichst echtzeitfähig sein. Die Evaluation hat gezeigt, dass selbst auf älteren Pentium 4 (P4)-Rechnern eine durchgehende Analyse eines 100 Mbps-Links möglich ist.

Erweiterbarkeit Aufgrund der kontinuierlich und schnell ansteigenden Datenmengen muss ein Sicherheitssystem derart gestaltet werden, dass eine Erweiterbarkeit ohne große Aufwände möglich ist. Insbesondere soll ein bestehendes System ergänzt werden können, ohne jeweils die Komponenten tauschen zu müssen. Dies wird durch die Art und Weise der Verarbeitung der statistischen Daten der jeweiligen Verbindungen erreicht: Diese können beliebig zwischen Rechnerinstanzen verteilt werden; hierbei gilt, dass jede Verbindung der Ausbruchserkennung sowie der Brute Force-Angriffserkennung einzeln analysiert werden kann und Verbindungen der TLS-Angriffserkennung auf beliebige Gruppen mit einer angestrebten Zahl von 12 Teilnehmern (vgl. Kapitel 6.2.2) verteilt werden können.

Verschlüsselte Netze Die Einsetzbarkeit der entwickelten Architektur in verschlüsselten Netzen ist eine zentrale Anforderung an das Sicherheitssystem, da bestehende Systeme und Forschungsarbeiten hier bisher keinen ausreichenden Schutz anbieten können. Durch die alleinige Nutzung von statistischen Daten, welche aus den beobachtbaren Parametern verschlüsselter Verbindungen gewonnen werden können, wir diese Anforderung erfüllt. Die Detektionsfähigkeit von Angriffen innerhalb verschlüsselten Datenverkehrs durch das Sicherheitssystem wurde in der Evaluation validiert.

Rechtskonformer Einsatz Um in der Praxis anwendbar zu sein, ist ein rechtskonformer Einsatz insbesondere hinsichtlich der Einhaltung der Datenschutzgesetze erforderlich. Die hierfür geltenden Restriktionen und Bestimmungen wurden im Laufe der Arbeit analysiert und bei der Entwicklung der Architektur berücksichtigt. Der gesamte Detektionsablauf kann mittels der Nutzung von Hashtabellen komplett anonymisiert ablaufen; lediglich im Falle, dass das Sicherheitssystem aktiv in den Datenverkehr eingreifen soll, d.h. als bösartig erkannte IP-Adressen mittels Firewallregeln geblockt werden, muss die IP des externen Kommunikationspartners vorgehalten werden (äußere Verbindungsdaten). Im Falle einer erlaubten Privatnutzung ist diesbezüglich eine entsprechende Nutzervereinbarung anzuvisieren; ist keine Privatnutzung gestattet, ist dies nicht erforderlich.

Verhaltensbasierte Evaluation Die Arbeitsweise des Sicherheitssystems macht sich verhaltensbasierte Ansätze zu nutzen: Abhängig des jeweiligen Moduls wird das Verhalten der Nutzer während einer Sitzung gegeneinander untersucht bzw. das Verhalten von

Referenzsitzungen mit einzelnen Nutzersitzungen verglichen. Weiterhin wurde mit der Befehls- und Sequenzevaluation eine Identifizierung des Nutzerverhaltens anhand von der teilweisen Erfüllung von Angriffsbäumen implementiert. Alle grundlegend erforderlichen Daten stehen somit von Beginn an zur Verfügung, es ist weder die Pflege einer Signaturdatenbank erforderlich, noch muss eine Lernphase zu Beginn des Systemeinsatzes durchgeführt werden.

Verzicht auf eine Lernphase Die verhaltensbasierte Evaluation wertet Verhaltensmuster von Nutzern aus, welche direkt durch die Interaktion mit dem Zielsystem entstehen. Hierbei wird der Fakt zu nutze gemacht, dass der Anteil der Angreifer regelmäßig sehr viel geringer ist, als der Anteil der regulären, gutartigen Nutzer. Verhaltensweisen, die bereits per se für die Evaluation bekannt sein müssen, werden in Form von Referenzsitzungen und Angriffsbäumen in das System integriert; somit kann komplett auf eine Lernphase in der produktiven Umgebung verzichtet werden.

Verzicht auf DPI Der aus mehreren Gründen, insbesondere dem Nichtvorhandenseins des Payloads durch Verschlüsselung motivierten Forderung des Verzichts auf eine DPI wird durch das System Rechnung getragen, indem lediglich statistische, von außen beobachtbare Informationen einer Verbindung verarbeitet werden.

Tabelle 7.1 fasst die Ergebnisse der Beurteilung der Architektur zusammen.

7.3 Beantwortung der Fragestellungen

Nachfolgend werden die Eingangs aufgestellten Fragestellungen rezitiert und kurz beantwortet.

Fragestellung 1: Ist eine verhaltensbasierte Ein- und Ausbruchserkennung ohne Lernphasen sowie ohne genaue Kenntnis des Kommunikationsverhaltens der geschützten Systeme sowie ohne Nutzung von DPI möglich? Wie bereits in der Einleitung in Kapitel 1 motiviert, reichen wissensbasierte Detektionsansätze nicht mehr aus, den heutigen Angriffsverfahren ausreichend zu begegnen. Hieraus erwächst der Bedarf der Nutzung von verhaltensbasierten Systemen, die jedoch andere Herausforderungen aufwerfen. Insbesondere benötigen diese Systeme regelmäßig eine Lernphase oder erfordern eine Kenntnis über das Kommunikationsverhalten des Systems. Dies zeigt sich auch bei der Betrachtung von State-of-the-Art Systemen, welche in Kapitel 4 vorgenommen wurde. Nach einer Untersuchung der im Rahmen von verschlüsselten Verbindungen noch verfügbaren Informationen wurden daher in Kapitel 5 Verfahren entwickelt, wie diese zur Ein- und Ausbruchserkennung ohne die Notwendigkeit einer Lernphase sowie ohne der Erfordernis einer DPI genutzt werden können. Hierbei wurden auf Basis von verschiedenen Korrelationen eine Reihe von Detektionsmodulen entwickelt, welche eine Angriffs- bzw. Ausbruchsdetektion ermöglichen. Eine Lernphase konnte hierbei komplett

Tabelle 7.1: Bewertung der Architektur anhand des Anforderungskataloges aus Nutzer- und Architektursicht von Kapitel 3.

	Verhaltensbasiert	Steuerung iptables	Angriffsstrategie	Nutzererkennung	Transparente Brücke	Keine Konfiguration	Leichtgewichtige Sensorik	Parallelisierbarkeit	Durchführung Hashing	Statistische Evaluation
Detektion von Angriffen	✓									✓
Unterbinden von Angriffen		✓								
Erkennen von Innentätern	✓		✓	✓						✓
Transparente Integration					✓					
Wartungsarmes System	✓					✓				
Nahe-Echtzeitauswertung							✓			
Erweiterbarkeit							✓	✓	✓	
Verschlüsselte Netze										✓
Rechtskonformer Einsatz									✓	
Verhaltensbasierte Eval.	✓		✓						✓	
Verzicht auf Lernphase			✓						✓	
Verzicht auf DPI	✓								✓	

eliminiert werden, da die Module entweder rein auf Basis des inherent durch die untersuchten Verbindungen vorhandenen Wissens arbeiten oder sämtliche, für den sofortigen Betrieb erforderlichen Daten einmalig vorkonfiguriert beinhalten. Weiterhin werden hierbei keine Informationen des Kommunikationsverhaltens der zu schützenden Systeme zu Konfigurationszwecken o.ä. benötigt, da dies ebenfalls aus den laufenden Verbindungen während des Betriebs erkannt wird. Die Module zur Einbruchserkennung nutzen hierbei aus, dass sich der Anteil der Angreifer im regulären Betrieb eines Servers im Vergleich zu den gutartigen Zugriffen als gering darstellt und nutzt Auswertungen der Ähnlichkeit von Verbindungen, um bösartige Verbindungen bzw. Angriffe zu detektieren. Dies erfolgt direkt und in nahe-Echtzeit anhand der vorliegenden Verbindungen, ohne dass eine Lernphase erforderlich ist. Detektionsmöglichkeiten, um Manipulationsversuche seitens des Angreifers erkennen zu können, wie bspw. das Erhöhen des Anteils bösartiger Verbindungen, um diese zu maskieren, wurden ebenfalls vorgestellt. Das System arbeitet dabei direkt auf den beobachtbaren Parametern verschlüsselter Verbindungen, ohne auf eine DPI oder andere Form der Manipulation oder Einflussnahme der verschlüsselten Daten angewiesen zu sein. Die Module zur Ausbruchserkennung realisieren eine verhaltensbasierte Nutzererkennung anhand der verschlüsselten Verbindungen bzw. ermöglichen die Bewertung einer Verbindung anhand der Detektion der darin eingegebenen Befehle durch eine Evaluation von Paketsequenzen. Auch hier werden lediglich beobachtbare Parameter der Verbindung ausgenutzt, ohne dass eine DPI erforderlich ist. Sämtliche Referenzdaten, die zur Korrelation und Erkennung der Befehle benötigt werden, werden einmalig vorkonfiguriert bereitgehalten. Ein weiteres Modul zur Ausbruchserkennung nutzt die Verfahren der Einbruchserkennung aus und benötigt keine Befehlsdatenbank; hierbei werden komplette Referenzsitzungen mit den aktuellen Sitzungen korreliert, um eine Bewertung vorzunehmen.

Sämtliche Module der Ein- und Ausbruchserkennung arbeiten rein verhaltensbasiert und erfordern weder eine Lernphase, noch die vorherige Kenntnis über das Kommunikationsverhalten des Systems oder eine DPI. Die Evaluation der Module hat gezeigt, dass hierdurch eine Erkennung von Angriffen und Innentätern ermöglicht wird.

Fragestellung 2: Wie kann eine Architektur zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen aussehen? Um eine Architektur im Rahmen der vorliegenden Restriktionen zur Ein- und Ausbruchserkennung zu entwickeln (vgl. Kapitel 5), wurde zunächst untersucht, welche Informationen und Parameter anhand der verschlüsselten Verbindung noch genutzt werden können. Hierbei sind insbesondere die Größen des Payloads der übertragenen Pakete von Interesse, sowie deren Auftretenszeitpunkt am Beobachtungsort. Eine anschließende Betrachtung, wie diese Daten weiter verarbeitet und zur Angriffserkennung genutzt werden können, identifiziert Korrelationsverfahren als geeignete Vorgehensweise. Darauf basierend wurden mehrere Module, jeweils spezialisiert für die Ein- bzw. Ausbruchserkennung, entwickelt. Abhängig ihrer genauen Spezialisierung, nutzen die Module Korrelationen *innerhalb einer* Verbindung, *zwischen verschiedenen* Verbindungen, mit *Paketsequenzen* einer Datenbank oder mit *Referenzsitzungen* einer Datenbank. Die verschiedenen Module mit ihren Spezialisierun-

gen garantieren dabei, dass sowohl eine umfassende Ein- wie auch Ausbruchserkennung bzw. Innentätererkennung ermöglicht wird.

Die vorgestellte Architektur basiert maßgeblich auf der Auswertung von Paketgrößen und Auftretenszeitpunkten. Diese immer verfügbaren Informationen genügen als Grundlage, um verschiedenste Module, bspw. zur Nutzer- oder zur Befehlsidentifikation zu realisieren. Die Nutzung von Kreuzkorrelationen zeigt sich als geeignetes Mittel, Angriffe sowie Innentäter durch entsprechenden Vergleich von Verbindungen bzw. deren Teilsequenzen zu erkennen.

Fragestellung 3: Müssen Innentäter in einem Sicherheitssystem adressiert werden? Für ein System zur Ein- und Ausbruchserkennung stellt die Gefahr eines Innentäters besondere Herausforderungen bzgl. dessen Detektion dar. Da dieser durch seine vorhandene Autorisierung legalen Zugriff auf das System hat, ist ein Missbrauch nur schwer erkennbar. Die Anforderung einer entsprechenden Detektion kann somit zu einer deutlich höheren Komplexität eines Sicherheitssystems führen. Entsprechend ist eine Analyse erforderlich, inwieweit eine Gefahr von Innentätern ausgeht und ob diese entsprechend im Rahmen eines Sicherheitssystems betrachtet werden müssen.

Um diese Frage beantworten zu können, erfolgte in Kapitel 2.2.2 eine detaillierte Untersuchung der in diesem Bereich verfügbaren Statistiken und Berichte, die teils stark kontroverse Ergebnisse präsentieren. Nach Auswahl der entsprechenden Literatur wurden die Ursachen für die unterschiedlichen Ergebnisse analysiert. Hierfür wurden sowohl die Methodologien, als auch die jeweils ausgewertet Datensätze berücksichtigt. Als maßgebliche Faktoren haben sich hier die Beachtung bzw. das Vernachlässigen von Dunkelziffern, die nicht-Anzeige von Fällen aufgrund der Angst vor Reputationsverlust sowie unterschiedlich betrachtete bzw. eingeteilte Unternehmensgrößen herausgestellt. Es zeigt sich insbesondere auch, dass Delikte von Innentätern regelmäßig lediglich rein zufällig entdeckt werden. Auch die hohe Gefährdung von insbesondere kleinen und mittelständischen, innovativen Unternehmen wird gezeigt. Bezogen auf die Anzahl von Sicherheitsvorfällen insgesamt, ergeben sich hierbei Zahlen von ca. 20 bis 50 Prozent Innentäteranteil. Die geringeren Zahlen treten hier insbesondere bei Studien auf, die lediglich den *tatsächlich festgestellten* Datenabfluss auswerten.

Die Analyse der Innentäterbedrohung zeigt, dass diese Gefahr maßgeblich im Rahmen eines Sicherheitssystems betrachtet werden muss. Weiterhin steigt die Gefahr durch Innentäter heutzutage durch *unbewusste* Innentäter, bspw. in Form von durch infizierte Botrechner ausgeführten Aktionen. Entsprechend muss eine Komponente für die Ausbruchsdetektion vorgesehen werden.

Fragestellung 4: Ist eine Ein- und Ausbruchserkennung in verschlüsselten Umgebungen unter Einhaltung der rechtlichen Anforderungen möglich? Um ein in der Praxis nutzbares System zu erhalten, müssen die jeweiligen rechtlichen Rahmenbedingungen eingehalten werden. Im Hinblick auf ein System zur Ein- und Ausbruchserkennung sind dies maßgeblich die Anforderungen gem. den einschlägigen Datenschutzbestimmungen. Eine Analyse der hieraus entstehenden Anforderungen wurde in Kapitel 4.6.5 gegeben.

Grundsätzlich betreffen die Datenschutzgesetze nur personenbezogene Daten, weiter ist zwischen erlaubter und nicht-erlaubter Privatnutzung zu unterscheiden. Ist diese nicht erlaubt, können die erforderlichen, äußeren Verbindungsdaten im Rahmen des Verhältnismäßigkeitsprinzips ausgewertet werden. Ist eine Privatnutzung erlaubt, dürfen diese Daten nicht per se erhoben werden, eine Auswertung muss in einer Nutzervereinbarung geregelt werden. Die in Kapitel 5 vorgestellte Architektur basiert auf der Nutzung rein statistischer Daten, namentlich den Größen des Payloads der übertragenen Pakete sowie den Beobachtungszeitpunkten. Sämtliche Verbindungsdaten können bei der Analyse anonymisiert werden. Nur im Falle einer erlaubten Privatnutzung muss eine Nutzervereinbarung getroffen werden, falls das Sicherheitssystem aktiv IP-Adressen sperren soll, da hierfür die äußeren Verbindungsdaten benötigt werden.

Das vorgestellte System zur Ein- und Ausbruchserkennung ermöglicht somit einen rechtskonformen Einsatz. Dies gilt für ungesicherte als auch für verschlüsselte Umgebungen.

7.4 Zukünftige Forschungsgebiete

Die im Rahmen der vorliegenden Arbeit entwickelte Architektur ermöglicht die Realisierung eines System zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen und wurde anhand der umgesetzten Prototypen erfolgreich getestet.

Mehrere Richtungen können für künftige Forschungsarbeiten weiter verfolgt werden:

- Die in der Arbeit genutzten Verfahren wurden anhand verschiedener Protokolle auf ihre Leistungsfähigkeit hin analysiert, bspw. TLS und SSH. Es werden jedoch keine spezifischen Eigenschaften dieser Protokolle genutzt, so dass eine Anwendung des Systems auf andere Protokolle einfach möglich sein sollte. Dies ist erstrebenswert, um eine umfassende Evaluation aller möglichen Verbindungen durchzuführen.

- Die Nutzung von Paketsequenzen im Rahmen der Befehlsevaluation hat gezeigt, dass diese Herangehensweise sehr aufwändig ist. Entsprechend wurden weiterhin Mechanismen konzipiert und implementiert, welche eine Verbindungsbewertung auch ohne die Evaluation einzelner Befehle ermöglicht, sondern mittels Referenzsitzungen arbeitet. Die optimale Kombination dieser Verfahren sowie eine Untersuchung, welche Befehle als Paketsequenzen minimal und maximal genutzt werden sollten, kann zur Verbesserung der Fehlalarmraten durchgeführt werden.

- Während der Evaluation der Befehlserkennung wurde festgestellt, dass die korrelierten Befehlskandidaten charakteristische Unterschiede bzgl. verschiedener Distributionen aufweisen. Hier kann untersucht werden, ob dies zur Identifizierung des darunter liegenden Systems herangezogen werden kann. Ist dies der Fall, kann das Ergebnis wiederum genutzt werden, um den Auswahlprozess der einzelnen Befehlskandidaten zu verbessern. Weiterhin kann ein entsprechendes Ergebnis verwendet werden, um ein neues, passives Fingerprinting-Tool zur Betriebssystem-Identifizierung zu entwickeln.

- Die vorgestellten und in verschiedenen Analysemodulen umgesetzten Korrelationsverfahren haben sich in der Evaluation als vielversprechende Detektionsmöglichkeit von bösartigem Verhalten erwiesen. Dies eröffnet die Möglichkeit einer komplett neuen Familie von Sicherheitssystemen der nächsten Generation welche darauf basieren, dass das überwiegende Systemverhalten zahlreicher Nutzer gutartiger Natur ist und der Anteil der Angreifer dahingegen deutlich geringer ist. Der Verzicht auf Signaturdatenbanken und Lernphasen sowie die Echtzeitverarbeitung eröffnen umfassende Detektionsmöglichkeiten, welche auch gutartige Effekte wir Flash Crowds korrekt erkennen können. Mögliche, zu untersuchende Felder sind hier bspw. leichtgewichtige Sicherheitssysteme für Handys und Smartphones, künftige Car2X-Anwendungen oder für neue Sicherheitssysteme in der Cloud. Insbesondere in der Cloud können traditionelle und gerade wissensbasierte Detektionsverfahren nicht mehr effizient eingesetzt werden, da die umfangreichen, hochdynamischen Daten und Dienste keine in der Genauigkeit hinreichenden Profile für eine entsprechende Evaluation ermöglichen. Die in dieser Arbeit entwickelten, verhaltensbasierten Korrelationsverfahren ohne Lernphase und insbesondere ohne Notwendigkeit des Wissens über die Kommunikationseigenschaften der zu sichernden Systeme können hier Startpunkt für neue und leistungsfähige IDS-Agenten sein, welche die Kommunikationsbeziehungen in der Cloud automatisch und in nahe-Echtzeit auswerten und Angriffe auf Basis der inherenten Informationen der Verbindungen detektieren. Auch in Bereichen wie bspw. Grids oder industriellen Steuerungen wie bspw. SCADA-Netzen kann die Anwendbarkeit der vorgestellten Verfahren geprüft werden. Es sei hier nochmals darauf hingewiesen, dass sich der Versuch des Angreifers, das Eigenverhalten mittels bspw. einer Vielzahl bösartiger Verbindungen als gutartig gegenüber dem Sicherheitssystem auszusehen zu lassen, regelmäßig durch andere, dadurch auftretende Anomalien erkennen lässt. Ähnliche Effekte sind auch im Falle der Übertragung auf andere Bereiche zu erwarten.

Die Verfahren und Algorithmen der vorliegenden Arbeit eröffnen die Möglichkeit, eine neue Generation von IDSs zu entwickeln. Die präsentierte Architektur ermöglicht eine Ein- und Ausbruchserkennung in verschlüsselten Umgebungen. Im Gegensatz zu bisher verfügbaren, verhaltensbasierten Systemen wird das notwendige Wissen zur Klassifizierung der Verbindungen implizit aus der Gesamtheit der Verbindungen gewonnen. Die neuen Verfahren zur Erkennung von Innentätern können weiterhin genutzt werden, der wachsenden Gefahr von Datenverlust zu begegnen. Die für das Sicherheitssystem entwickelten und genutzten Verfahren können jedoch auch in weiteren Gebieten genutzt werden, neue und leistungsfähige Sicherheitssysteme zu entwickeln.

Q.E.D.

A Literaturverzeichnis

[1] Back Orifice BO2K. Website. http://www.bo2k.com/index2.shtml, aufgerufen am 11. März 2011.

[2] Emerging Threats Rule Downloads. Website. http://www.emergingthreats.net/index.php?option=com_content&view=article&id=16&Itemid=38, aufgerufen am 12. April 2011.

[3] Free Rainbow Tables. Website. http://www.freerainbowtables.com/, aufgerufen am 8. März 2011.

[4] gnuplot homepage. Website. http://www.gnuplot.info, aufgerufen am 24. August 2011.

[5] HASHCRACK.COM. Website. http://hashcrack.com/, aufgerufen am 9. März 2011.

[6] Hashkiller. Website. http://hashkiller.com/, aufgerufen am 9. März 2011.

[7] History to date. Website. http://www.w3.org/History/19921103-hypertext/hypertext/WWW/History.html, aufgerufen am 3. Februar 2011.

[8] The libpcap project. Website. http://sourceforge.net/projects/libpcap/, aufgerufen am 25. Mai 2011.

[9] MAWI Working Group Traffic Archive. Website. http://tracer.csl.sony.co.jp/mawi/, aufgerufen am 11. August 2011.

[10] Online Hash Crack. Website. http://www.onlinehashcrack.com/, aufgerufen am 9. März 2011.

[11] Strict Transport Security. Website. http://www.chromium.org/sts, aufgerufen am 16. Februar 2011.

[12] TCP/IP fingerprinting methods supported by nmap. Website. http://nmap.org/book/osdetect-methods.html, aufgerufen am 8. März 2011.

[13] The Hacker Defender Project. Website. http://www.aboutus.org/Hxdef.org, aufgerufen am 11. März 2011.

[14] The Linux Kernel Archives. Website. http://www.kernel.org/, aufgerufen am 8. März 2011.

[15] The Open Web Application Security Project. Website. http://www.owasp.org/index.php/Main_Page, aufgerufen am 9. März 2011.

[16] Top 500 Supercomputer Sites. Website. http://www.top500.org/stats/list/36/osfam, aufgerufen am 1. Februar 2011.

[17] Understanding an Nmap Fingerprint. Website. http://nmap.org/book/osdetect-fingerprint-format.html, aufgerufen am 8. März 2011.

[18] European Standard EN 50133-1:1996/A1:2002, Alarm Systems. Access control systems for use in security applications. Part 1: System requirements., 2002.

[19] RainbowCrack Project. Website, 2003. http://www.project-rainbowcrack.com/, aufgerufen am 9. März 2011.

[20] TOTEM Datasets. Website, 2008. http://totem.info.ucl.ac.be/dataset.html, aufgerufen am 11. August 2011.

[21] Welt Online: Onlinehandel erstmals vor Katalog und Telefon. Website, November 2009. Aufgerufen am 7. Oktober 2010.

[22] Internet World Stats Usage and Population Statistics. Website, Oktober 2010. Aufgerufen am 6. Oktober 2010.

[23] Internethandel-Umsätze zum ersten Mal vor den stationären Handelszahlen. Website, Februar 2010. Aufgerufen am 7. Oktober 2010.

[24] RFC 789. Vulnerabilities of Network Control Protocols: An Example. RFC. http://rfc-ref.org/RFC-TEXTS/789/chapter1.html, aufgerufen am 23. August 2011.

[25] A.A. Abimbola, J.M. Munoz, and W.J. Buchanan. NetHost-Sensor: Investigating the capture of end-to-end encrypted intrusive data. *Computers & Security*, 25(6):445 – 451, 2006. DOI: 10.1016/j.cose.2006.04.001.

[26] Leonard Adleman. An Abstract Theory of Computer Viruses. In Shafi Goldwasser, editor, *Advances in Cryptology — CRYPTO' 88*, volume 403 of *Lecture Notes in Computer Science*, pages 354–374. Springer Berlin / Heidelberg, 1990. 10.1007/0-387-34799-2_28.

[27] Nadezda Agapova. Statistiken und Prognosen: Unternehmensbedürfnisse bezüglich der notwendigen Bandbreiten im Internet heute und in der Zukunft. Technical report, Fachhochschule Gelsenkirchen, Fachbereich Informatik, 2006.

[28] National Security Agency. Security-Enhanced Linux. Website, 2009. http://www.nsa.gov/research/selinux/, aufgerufen am 5. April 2011.

[29] Siddhant Ahuja. Correlation based similarity measures. Website, 2010. http://siddhantahuja.wordpress.com/2010/04/11/correlation-based-similarity-measures-summary/.

[30] Francois Ajenstat. Get Virtual Now - Virtualization and "Green IT". msdn, September 2008.

[31] Abdulrahman Alharby and Hideki Imai. IDS False Alarm Reduction Using Continuous and Discontinuous Patterns. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security*, volume 3531 of *Lecture Notes in Computer Science*, pages 423–442. Springer Berlin / Heidelberg, 2005. ISBN: 978-3-540-26223-7.

[32] Magnus Almgren and Erland Jonsson. Using Active Learning in Intrusion Detection. In *IEEE Computer Security Foundations Workshop*, volume 17 of *CSFW 04*. IEEE, 2004. 1063-6900/04.

[33] R. Alshammari and N. Zincir-Heywood. Generalization of signatures for ssh encrypted traffic identification. In *Computational Intelligence in Cyber Security, 2009. CICS'09. IEEE Symposium on*, pages 167–174. IEEE, 2009.

[34] James P. Anderson. Computer Security Technology Planning Study. Technical report, Hanscom Airforce Base Electronic Systems Divison, Bedford Massachusetts 01730, Oktober 1972.

[35] James P. Anderson. Computer Security Threat Monitoring and Surveillance. James P. Anderson Co., Box 42 Fort Washington, Pa. 19034, 215 646-4706, Contract 79F296400, Februar 1980.

[36] Nicholas Athanasiades, Randal Abler, John Levine, Henry Owen, and George Riley. Intrusion Detection Testing and Benchmarking Methodologies. In *International Workshop on Information Assurance*, volume 1 of *IWIA 03*. IEEE, IEEE, 2003. 0-7695-1886-9/03.

[37] AV-Test. Website, 2010. http://www.av-test.org/numbers.php, aufgerufen im Oktober 2010.

[38] Gildas Avoine, Pascal Junod, and Philippe Oechslin. Characterization and Improvement of Time-Memory Trade-Off Based on Perfect Tables. *ACM Trans. Inf. Syst. Secur.*, 11(4):17:1–17:22, Juli 2008. DOI: http://doi.acm.org/10.1145/1380564.1380565.

[39] Stefan Axelsson. The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans. Inf. Syst. Secur.*, 3(3):186–205, August 2000. DOI: http://doi.acm.org/10.1145/357830.357849.

[40] BackupTechnology. Data loss incident affects NASA. Website, Dezember 2010. http://www.backup-technology.com/5451/data-loss-incident-affects-nasa/, aufgerufen am 7. März 2011.

[41] M. Bailey, E. Cooke, F. Jahanian, Yunjing Xu, and M. Karir. A Survey of Botnet Technology and Defenses. In *Conference For Homeland Security, 2009. CATCH '09. Cybersecurity Applications Technology*, pages 299–304, März 2009. DOI: 10.1109/CATCH.2009.40.

[42] Wade Baker, Alexander Hutton, C. David Hylender, Joseph Parnula, Christopher Porter, and Marc Spitler. 2011 Data Breach Investigation Report. Technical Report 8, Verizon RISK Team, 2011.

[43] Wade et al. Baker. 2008 Data Breach Investigations Report. Technical report, Verizon RISK Team, 2008.

[44] Wade et al. Baker. 2009 Data Breach Investigations Report. Technical report, Verizon RISK Team, 2009.

[45] Wade et al. Baker. 2010 Data Breach Investigations Report. Technical report, Verizon RISK Team, 2010.

[46] Wade H. Baker, C. David Hylender, and J. Andrew Valentine. 2008 Data Breach Investigation Report. Technical report, Verizon Business, 2008. www.verizonbusiness.com, aufgerufen am 4. Februar 2011.

[47] Zachary K. Baker and Viktor K. Prasanna. High-throughput Linked-Pattern Matching for Intrusion Detection Systems. In *ANCS 05*. ACM, Oktober 2005. ACM 1-59593-082-5/05/0010.

[48] Marco Balduzzi, Christian Platzer, Thorsten Holz, Engin Kirda, Davide Balzarotti, and Christopher Kruegel. Abusing social networks for automated user profiling. In *Proceedings of the 13th international conference on Recent advances in intrusion detection*, RAID'10, pages 422–441, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15511-1, 978-3-642-15511-6.

[49] Roni Bar Yanai, Michael Langberg, David Peleg, and Liam Roditty. [realtime classification for encrypted traffic.

[50] Paul Baran. On Distributed Communications. Technical report, RAND Corporation, 1962. http://www.rand.org/about/history/baran-list.html aufgerufen am 21. Januar 2011.

[51] David Barroso. Botnets – The Silent Threat. Technical report, ENISA European Network and Information Security Agency, November 2007. ENISA Position Paper No. 3.

[52] Johannes M. Bauer. ITU Study on the Financial Aspects of Network Security: Malware and Spam. Technical report, ITU International Telecommunications Union, 2008.

[53] Kathrin Beckert. Sicherheitstipp: Wirtschaftspionage per USB-Stick. Website, 2010. https://www.it-sicherheit.de/ratgeber/it_sicherheitstipps/tipp/sicherheitstipp-wirtschaftspionage-per-usb-stick/, aufgerufen am 18. April 2011.

[54] Richard Bejtlich. Insider Threat Myth Documentation. Website, Mai 2009. http://taosecurity.blogspot.com/2009/05/insider-threat-myth-documentation.html, aufgerufen am 4. Februar 2011.

[55] Laurent Bernaille, Renata Teixeira, Ismael Akodkenou, Augustin Soule, and Kave Salamatian. Traffic classification on the fly. *SIGCOMM Comput. Commun. Rev.*, 36(2):23–26, April 2006. http://doi.acm.org/10.1145/1129582.1129589.

[56] G. Bissias, M. Liberatore, D. Jensen, and B. Levine. Privacy vulnerabilities in encrypted http streams. In *Privacy Enhancing Technologies*, pages 1–11. Springer, 2006.

[57] C. Bitter, D.A. Elizondo, and T. Watson. Application of artificial neural networks and related techniques to intrusion detection. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8. IEEE, 2010. DOI 10.1109/IJCNN.2010.5596532.

[58] Dion Blazakis. Interpreter Exploitation: Pointer Interference and JIT Spraying. Blackhat 2010, 2010.

[59] N. Boggs, S. Hiremagalore, A. Stavrou, and S.J. Stolfo. Experimental results of cross-site exchange of web content Anomaly Detector alerts. In *Technologies for Homeland Security (HST), 2010 IEEE International Conference on*, pages 8 – 14. IEEE, November 2010. DOI: 10.1109/THS.2010.5655103.

[60] D. Bolzoni and S. Etalle. Approaches in anomaly-based network intrusion detection systems. *Intrusion Detection Systems*, pages 1–15, 2008. Springer.

[61] Damiano Bolzoni. *Revisiting Anomaly-based Network Intrusion Detection Systems*. PhD thesis, University of Twente, 2009. DOI: 10.3990/1.9789036528535.

[62] Tony Bradley. Zero Day Exploits. Website. http://netsecurity.about.com/od/newsandeditorial1/a/aazeroday.htm, aufgerufen am 9. März 2011.

[63] Viola Bräuer. Erkennungs-Dienst. Linux-Magazin, Website, März 2002. http://www.linux-magazin.de/layout/set/print/content/view/full/8839.

[64] Peter Bright. RSA finally comes clean: SecurID is compromised. Website, Juni 2011. http://arstechnica.com/security/news/2011/06/rsa-finally-comes-clean-securid-is-compromised.ars, aufgerufen am 23. August 2011.

[65] Felix Brosius. *SPSS 8*, chapter 27, pages 671–690. International Thomsen Publishing, 2005.

[66] Terry Brugger. KDD Cup '99 dataset (Network Intrusion) considered harmful. Website, September 2007. http://www.kdnuggets.com/news/2007/n18/4i.html, aufgerufen am 4. April 2011.

[67] Erik Buchanan, Ryan Roemer, Stefan Savage, and Hovav Shacham. Return-oriented Programming: Exploitation without Code Injection. Blackhat 2008, 2008. University of California, San Diego.

[68] Bundesamt für Sicherheit in der Informationstechnik. Technical report.

[69] Bundesamt für Sicherheit in der Informationstechnik. IT-Grundschutz. Website. Aufgerufen am 7. Oktober 2010.

[70] Bundesamt für Sicherheit in der Informationstechnik. Netze. Website. https://www.bsi.bund.de/cln_183/ContentBSI/grundschutz/kataloge/baust/b04/b04.html, aufgerufen am 17. Februar 2011.

[71] Bundesamt für Sicherheit in der Informationstechnik. Übergreifende Aspekte. Website. https://www.bsi.bund.de/cln_183/ContentBSI/grundschutz/kataloge/baust/b01/b01.html, aufgerufen am 17. Februar 2011.

[72] Bundesamt für Sicherheit in der Informationstechnik. *IT-Grundschutz-Kataloge*, volume 11. BSI, November 2009. https://www.bsi.bund.de/cln_183/DE/Themen/weitereThemen/ITGrundschutzKataloge/Download/download_node.html.

[73] Bundeskriminalamt. Bundeskriminalamt und BITKOM teilen mit: Online-Kriminelle gehen immer raffinierter vor. Website, September 2010. http://bka.de/pressemitteilungen/2010/pm100906.html, aufgerufen am 4. Februar 2011.

[74] Bundeskriminalamt. IuK-Kriminalität Bundeslagebild 2009, 2010. http://www.bka.de/lageberichte/iuk/bundeslagebild_iuk_2009.pdf.

[75] Bundeskriminalamt. *Polizeiliche Kriminalstatistik 2009*, volume 57. Bundeskriminalamt, Kriminalistisches Institut, Fachbereich KI 12, 65173 Wiesbaden, 2010. http://www.bka.de/pks/pks2009/download/pks-jb_2009_bka.pdf.

[76] Bundeskriminalamt. Wirtschaftskriminalität Bundeslagebild 2009. Referat SO 51, Zentrale Lage, Früherkennung und OK-Analyse, 65173 Wiesbaden, 2010.

[77] Bundesministerium des Inneren. Polizeiliche Kriminalstatistik 2009, 2009.

[78] Bundesministerium des Inneren. *Verfassungsschutzbericht 2009*. Bundesministerium des Inneren, 2010.

[79] Bundesministerium für Wirtschaft und Technologie. Antwort der Bundesregierung auf die Kleine Anfrage: Stand des Breitbandausbaus und Strategie der Bundesregierung zur Breitbandversorgung in Deutschland, Dezember 2010. Drucksache 17/4348 vom 29.12.2010.

[80] Bundesnetzagentur. Jahresbericht 2009. Technical report, Bundesnetzagentur, 2010. http://www.bundesnetzagentur.de.

[81] Bryan et al. Burns. *Security Power Tools*. O'Reilly Media, August 2007. ISBN 9780596009632, http://amazon.com/o/ASIN/0596009631/.

[82] Kai-D. et al. Bussmann. Wirtschaftskriminalität 2009. Technical report, Pricewaterhouse-Coopers, Martin-Luther-Universität Halle-Wittenberg, September 2009.

[83] Eric Butler. Firesheep. Website, Oktober 2010. http://codebutler.github.com/firesheep/, aufgerufen am 16. Februar 2011.

[84] Eric J. Byres, Matthew Franz, and Darrin Miller. The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems. In *International Infrastructure Survivability Workshop (IISW)*. IEEE, 2004.

[85] A.A. C'ardenas, S. Amin, Z.S. Lin, Y.L. Huang, C.Y. Huang, and S. Sastry. Attacks Against Process Control Systems: Risk Assessment, Detection, and Response. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS'11)*. ACM, 2011.

[86] Alberto Carrascal, Jorge Couchet, Enrique Ferreira, and Daniel Manrique. Anomaly Detection using prior knowledge: application to TCP/IP traffic. In Max Bramer, editor, *Artificial Intelligence in Theory and Practice*, volume 217 of *IFIP International Federation for Information Processing*, pages 139–148. Springer Boston, 2006. http://dx.doi.org/10.1007/978-0-387-34747-9_15.

[87] Pedro Casas, Johan Mazel, and Philippe Owezarski. Steps Towards Autonomous Network Security: Unsupervised Detection of Network Attacks. In *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*. IEEE, 2011. DOI 10.1109/NTMS.2011.5721067.

[88] Imperva Application Defense Center. Consumer Password Worst Practices. Whitepaper, 2010. http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf, aufgerufen am 11. August 2011.

[89] SANS Internet Storm Center. Infocon. Website. http://isc.sans.edu/infocon.html, aufgerufen am 4. April 2011.

[90] Terrence Champion and Mary L. Denz. A Benchmark Evaluation of Network Intrusion Detection Systems. In *Aerospace Conference*, volume 6 of *Aerospace Conference*, pages 2705–2712. IEEE, IEEE, 2001.

[91] Kuan-Ta Chen and Li-Wen Hong. User identification based on game-play activity patterns. In *Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games*, NetGames '07, pages 7–12, New York, NY, USA, 2007. ACM. DOI: http://doi.acm.org/10.1145/1326257.1326259.

[92] Anton Chuvakin and Cyrus Peikari. *Kenne deinen Feind*. O'Reilly Vlg. GmbH & Co., Juli 2004. ISBN: 9783897213760, http://amazon.com/o/ASIN/3897213761/.

[93] Daniel B. Cid. Information about the illogic Rootkit. Website, 2003. http://www.ossec.net/rootkits/illogic.php, aufgerufen am 11. März 2011.

[94] Cisco. Visual Networking Index - VNI Forecast. Website. http://www.cisco.com/en/US/netsol/ns827/networking_solutions_sub_solution.html#~forecast, aufgerufen am 31. Januar 2011.

[95] Cisco. Third annual broadband study shows global broadband quality improves by 24% in one year. Website, Dezember 2010. http://newsroom.cisco.com/dlls/2010/prod_101710.html.

[96] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010-2015. Technical report, Cisco Systems Inc., Februar 2011.

[97] Christopher Clark. Hashtable. Website, 2004. http://www.cl.cam.ac.uk/~cwc22/hashtable.

[98] cnet News. Microsoft details new security plan. Website, Oktober 2003. http://news.cnet.com/2100-1002-5088846.html, aufgerufen am 7. März 2011.

[99] F.A.B.S. Collie, H.A. Command, and R.A.A. Force. Intrusion Investigation and Post-Intrusion Computer Forensic Analysis. *Headquarter Air Command, Royal Australian Air Force*, 2006. http://www.mirrors.wiretapped.net/security/info/papers/law-enforcement/intrusion-investigation-andpost-intrusion-forensic-analysis.pdf.

[100] European Commission. Industrial competitiveness. Website. http://ec.europa.eu/enterprise/policies/industrial-competitiveness/, aufgerufen am 6. Februar 2011.

[101] ConSecur GmbH. Einführung von Intrusion-Detection-Systemen Rechtliche Aspekte. Technical report, Bundesamt für Sicherheit in der Informationstechnik, Oktober 2002.

[102] Car2Car Communication Consortium. Mission & Objectives. Website. http://www.car-to-car.org/, aufgerufen am 18. April 2011.

[103] Cooperative Association for Internet Data Analysis. IPv4 WHOIS Map. Website, 2007. http://www.caida.org/research/id-consumption/whois-map/, aufgerufen am 16. August 2011.

[104] Corporate Trust. Studie: Industriespionage - Die Schäden durch Spionage in der Deutschen Wirtschaft. Technical report, Corporate Trust - Business Risk and Crisis Management GmbH, 2007.

[105] MITRE Corporation. Common Computer Vulnerabilities and Exposures. Website. http://cve.mitre.org/cve/, aufgerufen am 9. März 2011.

[106] Symantec Corporation. Symantec DeepSight Threat Management System. Website. https://tms.symantec.com/, aufgerufen am 15. August 2011.

[107] Symantec Corporation. Symantec DeepSight Early Warning Services. Website, 2008. http://eval.symantec.com/mktginfo/enterprise/brochures/b-brochure/_symc_deepsight_early_warning_services_06-2008.en-us.pdf, aufgerufen am 5. April 2011.

[108] Symantec Corporation. Symantec Intelligence Quarterly April - June 2010. Technical report, Symantec Corporation, 2010. http://www.symantec.com/business/theme.jsp?themeid=threatreport, aufgerufen am 4. Februar 2011.

[109] Symantec Corporation. Symantec Intelligence Quarterly January - March 2010. Technical report, Symantec Corporation, 2010. http://www.symantec.com/business/theme.jsp?themeid=threatreport, aufgerufen am 4. Februar 2011.

[110] R. K. Cunningham, R. P. Lippmann, D. J. Fried, S. L. Garfinkel, I. Graf, K. R. Kendall, S. E. Webster, D. Wyschogrod, and M. A. Zissman. Evaluating Intrusion Detection Systems without Attacking your Friends: The 1998 DARPA Intrusion Detection Evaluation. Technical report, Lincoln Laboratory MIT, 1998.

[111] Dirk Dahlhaus and Herbert Lindenborn. Leitfaden für kommunale Entscheidungsträger und Unternehmen zur Versorgung ländlicher Bereiche mit Breitband-Kommunikationsverbindungen: Zugangstechnologien für den Endkunden. Technical report, University of Kassel, Dezember 2008.

[112] M. Dass, J. Cannady, and W.D. Potter. LIDS: Learning Intrusion Detection System. In *FLAIRS 2003*, 2003.

[113] Noah Davids. Initial TTL Values. Website, Januar 2009. http://members.cox.net/~ndav1/self_published/TTL_values.html.

[114] Willem de Bruijn, Asia Slowinska, Kees van Reeuwijk, Tomas Hruby, Li Xu, and Herbert Bos. SafeCard: A Gigabit IPS on the Network Card. In Diego Zamboni and Christopher Kruegel, editors, *Recent Advances in Intrusion Detection*, volume 4219 of *Lecture Notes in Computer Science*, pages 311–330. Springer Berlin / Heidelberg, 2006. http://dx.doi.org/10.1007/11856214_16.

[115] Herve Debar, Marc Dacier, and Andreas Wespi. A Revised Taxonomy for Intrusion-Detection Systems. Technical report, IBM Research, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland, 1999.

[116] Hervé Debar, Marc Dacier, and Andreas Wespi. Towards a taxonomy of intrusion-detection systems. *Computer Networks*, 31(8):805 – 822, 1999. DOI: 10.1016/S1389-1286(98)00017-6.

[117] Hervé Debar, Marc Dacier, and Andreas Wespi. A revised taxonomy for intrusion-detection systems. *Annals of Telecommunications*, 55(7):361–378, 2000. DOI 10.1007/BF02994844.

[118] Lori L. DeLooze. Classification of Computer Attacks using a Self-Organizing Map. In *Proceedings of the 2004 IEEE Workshop on Information Assurance*, pages 365–369. IEEE, 2004.

[119] Guillaume Delugré. Closer to metal: Reverse engineering the Broadcom NetExtreme's firmware. Presentation HACK.LU 2010, Luxembourg, November 2010.

[120] DUDEN Neues Wörterbuch der Szenensprache. Zero-Day-Exploit. Website, 2009. http://szenesprachenwiki.de/definition/zero-day-exploit/, aufgerufen am 9. März 2011.

[121] Verfassungsschutzbehörden des Bundes und der Länder. Wirtschaftsspionage - Risiko für Ihr Unternehmen, Juni 2008.

[122] Nicolas Deschamps. Worldwide Observatory of Malicious Behaviors and Attack Threats. Website, Dezember 2010. http://www.wombat-project.eu/, aufgerufen am 5. April 2011.

[123] Deutscher Bundestag. Kleine Anfrage: Stand des Breitbandausbaus und Strategie der Bundesregierung zur Breitbandversorgung in Deutschland, Dezember 2010. Drucksache 17/4211 vom 14.12.2010.

[124] Verizon Business Deutschland. Dediziertes Internet. Website, 2011. http://www.verizonbusiness.com/de/Products/networking/internet/dedicated/, aufgerufen am 10. Februar 2011.

[125] Nishant Doshi, Ashwin Athaley, and Eric Chien. Pay-Per-Install: The New Malware Distribution Network. Technical report, Symantec Corporation, 2010.

[126] Trading Economics. Indicator historical data chart Internet users (per 100 people) in Germany. Website. http://www.tradingeconomics.com/germany/internet-users-per-100-people-wb-data.html, aufgerufen am 2. Februar 2011.

[127] Edge-Security. BruteSSH. Website. http://www.edge-security.com/brutessh.php, aufgerufen am 10. März 2011.

[128] Jon Erickson. *Hacking: The Art of Exploitation*. No Starch Press, 2003. ISBN 1-59327-007-0, http://amazon.com/o/ASIN/B002WSY37O/.

[129] Amsterdam Internet Exchange. sFlow Stats. Website, März 2011. http://www.ams-ix.net/sflow-stats/ether/, aufgerufen am 16. März 2011.

[130] The Measurement Factory. MAPS OF ISI LANDER CENSUS DATA. Website. http://maps.measurement-factory.com/gallery/USC-LANDER-Census/, aufgerufen am 16. April 2011.

[131] Z.M. Fadlullah, T. Taleb, N. Ansari, K. Hashimoto, Y. Miyake, Y. Nemoto, and N. Kato. Combating against attacks on encrypted protocols. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 1211–1216. IEEE, 2007. 1424403537.

[132] Weiwei Fang, Bingru Yang, Zheng Peng, and ZhiGang Tang. Research and Realization of Trusted Computing Pla. In *Proceedings to the Second International Symposium on Electronic Commerce and Security*, pages 43–46. IEEE Computer Society, 2009. DOI 10.1109/ISECS.2009.146.

[133] Finanz-lexikon.de. Dotcom-Blase. http://www.finanz-lexikon.de/dotcom-blase_1449.html, aufgerufen am 3. Februar 2011.

[134] Vahid Aghaei Foroushani, Fazlollah Adibnia, and Elham Hojati. Intrusion Detection in Encrypted Accesses with SSH Protocol to Network Public Servers. In *Proceedings of the International Conference on Computer and Communication Engineering*, pages 314–318. IEEE, Mai 2008. 978-1-4244-1692-9/08.

[135] S. Forrest, S.A. Hofmeyr, and A. Somayaji. Computer immunology. *Communications of the ACM*, 40(10):88–96, 1997.

[136] JustLinux Forums. gethostbyname error. Website, 2001. http://www.justlinux.com/forum/archive/index.php/t-9656.html, aufgerufen am 11. März 2011.

[137] Marc Fossi. Symantec Global Internet Security Threat Report. Technical Report XV, Symantec Corporation, April 2010.

[138] Marc Fossi. Symantec Intelligence Quarterly July - September 2010. Technical report, Symantec Corporation, 2010.

[139] Marc Fossi. Symantec Report on Attack Kits and Malicious Websites. Technical report, Symantec Corporation, 2010.

[140] Marc et al. Fossi. Symantec Report on the Underground Economy July 07 - June 08. Technical report, Symantec Corporation, November 2008.

[141] Marc et al. Fossi. Symantec Global Internet Security Threat Report Trends for 2009. Technical Report XV, Symantec Enterprise Security, April 2010.

[142] Marc et al. Fossi. Symantec Internet Security Report Trends for 2010. Technical report, Symantec Corporation, 350 Ellis Street, Mountain View, CA 94043 USA, April 2011.

[143] Shadowserver Foundation. Botnets. Website. http://www.shadowserver.org/wiki/pmwiki.php/Stats/Botnets, aufgerufen am 18. April 2011.

[144] Shadowserver Foundation. Statistics. Website. http://www.shadowserver.org/wiki/pmwiki.php/Stats/Statistics, aufgerufen am 06. März 2011.

[145] Wireshark Foundation. Wireshark Go deep. Website. http://www.wireshark.org, aufgerufen am 23. August 2011.

[146] Dirk Fox and Frank Schaefer. Passwörter – fünf Mythen und fünf Versäumnisse. *Datenschutz und Datensicherheit - DuD*, 33(7):425–429, 2009. DOI: http://dx.doi.org/10.1007/s11623-009-0109-0.

[147] Metasploit Framework. ms11_xxx_createsizeddibsection.rb. Website, Framework, Januar 2011. https://dev.metasploit.com/redmine/projects/framework/repository/revisions/11466/entry/modules/exploits/windows/fileformat/ms11_xxx_createsizeddibsection.rb, aufgerufen am 15. April 2011.

[148] Manuel Fuchs. Globalisierungs-Fakten. Website. http://www.globalisierung-fakten.de/globalisierung/definition-globalisierung.html, aufgerufen am 15. Februar 2011.

[149] Der Bundesbeauftragte für den Datenschutz und die Informationsfreiheit. Bundesdatenschutzgesetz (BDSG), Dezember 1990. Stand vom 11. Juni 2010.

[150] Bundesamt für Sicherheit in der Informationstechnik. Glossar / Begriffe. Website. https://www.bsi.bund.de/cln_156/ContentBSI/Themen/Internet_Sicherheit/Glossar/glossarbegriffe.html, aufgerufen am 13. März 2011.

[151] Bundesamt für Verfassungsschutz. Elektronische Attacken auf Informations- und Kommunikationstechnik. Flyer, August 2010. http://www.verfassungsschutz.de.

[152] Bundesamt für Verfassungsschutz. Schrankenlose Offenheit - soziale Netzwerke im Web. Flyer, August 2010. http://www.verfassungsschutz.de.

[153] Bundesamt für Verfassungsschutz. Sicherheitslücke Mensch - Der Innentäter als grösste Bedrohung für die Unternehmen. Flyer, August 2010. http://www.verfassungsschutz.de.

[154] Bundesamt für Verfassungsschutz. Verfassungsschutz - Ihr Ansprechpartner für Wirtschaftsschutz. Flyer, August 2010. http://www.verfassungsschutz.de.

[155] Bundesamt für Verfassungsschutz. Wissenschaftsspionage - Gefahr für Forschung und Lehre. Flyer, August 2010. http://www.verfassungsschutz.de.

[156] Bundesministerium für Wirtschaft und Technologie. Breitbandstrategie der Bundesregierung. Technical report, Bundesministerium für Wirtschaft und Technologie, Februar 2009. http://www.bmwi.de.

[157] Fyodor. Remote OS detection via TCP/IP Stack FingerPrinting. Website, Oktober 1998. http://nmap.org/nmap-fingerprinting-article.txt, aufgerufen am 7. März 2011.

[158] M. Gao, K. Zhang, and J. Lu. Efficient packet matching for gigabit network intrusion detection using TCAMs. In *Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)*. IEEE Computer Society, 2006.

[159] Fachhochschule Gelsenkirchen. Sybil Angriff. Website. https://www.internet-sicherheit.de/service/glossar/eintrag/eintrag-detail/sybil-angriff/, aufgerufen am 28. Februar 2011.

[160] GeNUA. VPN-Appliance Hardware. Website. http://www.genua.de/produkte/vpn/varianten/index.html, aufgerufen am 8. Februar 2011.

[161] Anup K. Ghosh, Christoph Michael, and Michael Schatz. A Real-Time Intrusion Detection System Based on Learning Program Behavior. In H. Debar, L. Me, and F. Wu, editors, *LNCS 1907*, RAID 2000, pages 93–109. Springer-Verlag Berlin Heidelberg, 2000.

[162] Giovanni Giacobbi. The GNU Netcat project. Website. http://netcat.sourceforge.net/, aufgerufen am 11. März 2011.

[163] Consecur GmbH. Einführung von Intrusion-Detection-Systemen Grundlagen. Technical report, Bundesamt für Sicherheit in der Informationstechnik, Oktober 2002.

[164] DE-CIX Management GmbH. DE-CIX Traffic Statistics. Website. http://www.de-cix.net/content/network/Traffic-Statistics.html, aufgerufen am 31. Januar 2011.

[165] Psylock GmbH. Tipp-Biometrie. http://www.psylock.com/tippbiometrie, Psylock GmbH, Galgenberstr. 25, 93053 Regensburg.

[166] T-Systems Enterprise Services GmbH. Sicherheitseigenschaften von Standleitungstechnologien. Technical report, Bundesamt für Sicherheit in der Informationstechnik, 2007.

[167] Vik Tor Goh. *Intrusion Detection Framework for Encrypted Networks*. PhD thesis, Queensland University of Technology, November 2010.

[168] Vik Tor Goh, Jacob Zimmermann, and Mark Looi. Towards Intrusion Detection for Encrypted Networks. *Availability, Reliability and Security, International Conference on*, 0:540–545, 2009.

[169] Vik Tor Goh, Jacob Zimmermann, and Mark Looi. Experimenting with an intrusion detection system for encrypted networks. In *International Journal of Business Intelligence and Data Mining*, volume 5(2), pages 172–191. Interscience Publishers, 2010. http://eprints.qut.edu.au.

[170] Vik Tor Goh, Jacob Zimmermann, and Mark Looi. Intrusion Detection System for Encrypted Networks using Secret-Sharing-Schemes. In *International Journal of Cryptology Research 2010*. QUT Digital Repository, Juli 2010.

[171] Maya Gokhale, Dave Dubois, Andy Dubois, Mike Boorman, Steve Poole, and Vic Hogsett. Granidt: Towards Gigabit Rate Network Intrusion Detection Technology. In Manfred Glesner, Peter Zipf, and Michel Renovell, editors, *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream*, volume 2438 of *Lecture Notes in Computer Science*, pages 47–61. Springer Berlin / Heidelberg, 2002. http://dx.doi.org/10.1007/3-540-46117-5_43.

[172] D.G. G'omez. Receive-only UTP cables and Network Taps. Citeseer, 2003.

[173] Ralph Gross and Alessandro Acquisti. Information Revelation and Privacy in Online Social Networks. In *WPES '05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, New York, NY, USA, 2005. ACM. ISBN 1-59593-228-3.

[174] Miniwatts Marketing Group. Internet Usage in Europe, Juni 2010. http://www.internetworldstats.com/stats4.htm.

[175] Netowrk Working Group. Cryptographic Suites for IPsec. RFC 4308, Dezember 2005. http://tools.ietf.org/html/rfc4308, aufgerufen am 10. Februar 2011.

[176] Network Working Group. Security Architecture for the Internet Protocol. RFC 2401, November 1998. http://www.ietf.org/rfc/rfc2401.txt, aufgerufen am 10. Februar 2011.

[177] Network Working Group. BGP/MPLS VPNs. RFC 2547, März 1999. http://www.faqs.org/rfcs/rfc2547.html, aufgerufen am 10. Februar 2011.

[178] Network Working Group. Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework. RFC 2576, März 2000.

[179] Network Working Group. InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks. RFC 3176, September 2001.

[180] Network Working Group. Management Information Base (MIB) for the Simple Network Management Protocol (SNMP). RFC 3418, Dezember 2002.

[181] Network Working Group. Cisco Systems NetFlow Services Export Version 9. RFC 3954, Oktober 2004.

[182] Network Working Group. Requirements for IP Flow Information Export (IPFIX). RFC 3917, Oktober 2004.

[183] Network Working Group. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. RFC 5101, Januar 2008.

[184] Peter Gutmann. Secure Deletion of Data from Magnetic and Solid-State Memory. Website. http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html, aufgerufen am 12. März 2011.

[185] Peter Gutmann. Data Remanence in Semiconductor Devices. Technical report, IBM T.J. Watson Research Center, 2001. Usenix Security '01.

[186] E. Haimerl. Ähnlichkeitsmasse. Website. http://ald.sbg.ac.at/dm/germ/Theorie/Aehnlichkeitsmasse.htm.

[187] J. W. Haines, R. P. Lippmann, D. J. Fried, M. A. Zissmann, E. Tran, and S. B. Boswell. 1999 DARPA Intrusion Detection Evaluation: Design and Procedures. Technical Report 1062, Lincoln Laboratory MIT, Februar 2001. ESC-TR-99-061.

[188] Joshua W. Haines, Lee M. Rossey, and Richard P. Lippmann. Extending the DARPA Off-Line Intrusion Detection Evaluations. In *unk*, 1999. Submitted to DISCEX-II.

[189] Simon Hansman and Ray Hunt. A taxonomy of network and computer attacks. In *Computer & Security*, volume 2004. Elsevier Ltd., 2004. doi:10.1016/j.cose.2004.06.011.

[190] John A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition, 1975. ISBN 047135645X.

[191] Universitätsbibliothek Heidelberg. Datenbank-Glossar. Website. http://www.ub.uni-heidelberg.de/helios/epubl/info/daba/glossardb.html#systematik, aufgerufen am 25. Februar 2011.

[192] heise. IPv4-Adressen: Abschiedsgrüsse, Mahnungen und Pappschilder. Website, Februar 2011. http://www.heise.de/netze/meldung/IPv4-Adressen-Abschiedsgruesse-Mahnungen-und-Pappschilder-1183204.html, aufgerufen am 16. März 2011.

[193] heise Security. Adobe schließt 23 Lücken in Reader und Acrobat – und gelobt Besserung. Website, Oktober 2010. http://www.heise.de/security/meldung/Adobe-schliesst-23-Luecken-in-Reader-und-Acrobat-und-gelobt-Besserung-1102390.html?, aufgerufen am 7. März 2011.

[194] heise Security. SAP führt Patchday ein. Website, September 2010. http://www.heise.de/security/meldung/SAP-fuehrt-Patchday-ein-1079757.html, aufgerufen am 7. März 2011.

[195] heise Security. Update für Adobe Reader schließt 19 Lücken. Website, November 2010. http://www.heise.de/security/meldung/Update-fuer-Adobe-Reader-schliesst-19-Luecken-1137609.html, aufgerufen am 7. März 2011.

[196] heise Security. Wettrüsten beim Cookie-Klau-Tool Firesheep. Website, November 2010. http://www.heise.de/security/meldung/Wettruesten-beim-Cookie-Klau-Tool-Firesheep-Update-1132720.html?view=print, aufgerufen am 16. Februar 2011.

[197] heise Security. Studie: Stuxnet befällt deutsche Energieversorger. Website, April 2011. http://heise.de/-1229240, aufgerufen am 18. April 2011.

[198] Cormac Herley. So long, and no thanks for the externalities: the rational rejection of security advice by users. In *Proceedings of the 2009 workshop on New security paradigms workshop*, NSPW '09, pages 133–144, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-845-2.

[199] John D. Howard. *An Analysis Of Security Incidents On The Internet 1989 - 1995*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 USA, April 1997.

[200] John D. Howard and Thomas A. Longstaff. A Common Language for Computer Security Incidents. Technical report, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, 1998.

[201] W. Hu, W. Hu, and S. Maybank. Adaboost-based algorithm for network intrusion detection. In *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, volume 38, pages 577–583. IEEE, 2008. DOI 10.1109/TSMCB.2007.914695.

[202] Bradley Huffaker. IPv4 BGP Geopolitical Analysis. Website, 2008. http://www.caida.org/research/policy/geopolitical/bgp2country/, aufgerufen am 18. April 2011.

[203] Liu Hui and Cao Yonghui. Research Intrusion Detection Techniques from the Perspective of Machine Learning. In *Multimedia and Information Technology (MMIT), 2010 Second International Conference on*, volume 1, pages 166–168. IEEE, April 2010. DOI 10.1109/MMIT.2010.161.

[204] Ralf Hund, Thorsten Holz, and Felix C. Freiling. Return-Oriented Rootkits: Bypassing Kernel Code Integrity Protection Mechanisms. Slides USENIX Security Symposium '09, August 2009. University of Mannheim, Laboratory for Dependable Distributed Systems.

[205] IBM Corporation. IBM Internet Security Systems. Website. http://www.iss.net/, aufgerufen am 15. August 2011.

[206] Vinay M. Igure and Ronald D. Williams. Taxonomies of Attacks and Vulnerabilities in Computer Systems. In *IEEE Communication Surveys*, volume 10 of *IEEE Communication Surveys and Tutorials*, pages 6–19. IEEE, 2008.

[207] K. Ilgun. USTAT: A real-time intrusion detection system for UNIX. *Published by the IEEE Computer Society*, 1993.

[208] Immunity. CANVAS. Website. http://www.immunitysec.com/products-canvas.shtml, aufgerufen am 9. März 2011.

[209] Cisco Systems Inc. Botnets: The New Threat Landscape. Whitepaper, 2007.

[210] Cisco Systems Inc. Hyperconnectivity and the Apporaching Zettabyte Era. Whitepaper, Juni 2010. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.pdf, aufgerufen am 18. April 2011.

[211] Code Green Networks Inc. Code Green Networks TrueDLP. Website. http://www.codegreennetworks.com/products/index.htm, aufgerufen am 5. April 2011.

[212] Finjan Inc. Cybercrime Intelligence Report. Technical Report 3, Finjan Malicious Code Research Center, September 2009.

[213] Lancope Inc. Flow-based Network Behavior Analysis. Website. http://www.lancope.com/solutions/security-operations/network-behavior-analysis/, aufgerufen am 16. April 2011.

[214] InformationWeek. Blended Web Attacks Hitting More Websites. Website, Juli 2011.

[215] SANS Institute. Intrusion Detection Systems: Definition, Need and Challenges. Technical report, SANS Institute, 2001. http://www.sans.org/reading_room/whitepapers/detection/intrusion-detection-systems-definition-challenges_343 aufgerufen am 06. Februar 2011.

[216] SRI International. Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD). Website, 2000. http://www.csl.sri.com/projects/emerald/, aufgerufen am 5. April 2011.

[217] Internet2. The Internet2 Observatory Data Collections. Website. http://www.internet2.edu/observatory/archive/data-collections.html, aufgerufen am 11. August 2011.

[218] ITWissen. DPI (deep packet inspection). Website. http://www.itwissen.info/definition/lexikon/DPI-deep-packet-inspection.html, aufgerufen am 18. April 2011.

[219] N. D. Jayaram and P. L. R. Morse. Network security - a Taxonomic View. In *European Conference on Security and Detection, Conference Publication No. 437*, pages 124–127, 1997.

[220] Shuyuan Jin, Yong Wang, Xiang Cui, and Xiaochun Yun. A Review of Classification Methods for Network Vulnerability. In *Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 1171–1175. IEEE, Oktober 2009.

[221] S.P. Joglekar and S.R. Tate. ProtoMon: embedded monitors for cryptographic protocol intrusion detection and prevention. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 1, pages 81–88, April 2004. DOI: 10.1109/ITCC.2004.1286430.

[222] Y. Frank Jou, Shyhtsun Felix Wu, Y. Frank, Jou Shyhtsun, Felix Wu, Fengmin Gong, W. Rance Cleaveland, and Chandru Sargor. Architecture design of a scalable intrusion detection system for the emerging network infrastructure. Technical report, MCNC Information Technologies Division and North Carolina State University Dep. of Computer Science, 1997.

[223] JuraForum. Verkehrssicherungspflicht. Website. http://www.juraforum.de/lexikon/verkehrssicherungspflicht, aufgerufen am 22. Februar 2011.

[224] Michael Kassner. Ransomware: Extortion via the Internet. Website. http://www.techrepublic.com/blog/security/ransomware-extortion-via-the-internet/2976, aufgerufen am 3. Februar 2011.

[225] H.G. Kayacik and Nur Zinicir-Heywood. Generating Representative Traffic for Intrusion Detection System Benchmarking. In *Annual Communication Networks and Services Research Conference*, volume 3 of *CNSR 05*. IEEE, IEEE, 2005. 0-7695-2333-1/05.

[226] Joseph Migga Kizza. *Computer Network Security*, chapter Kapitel 12. Springer Science+Business Media, Inc., 2005. ISBN: 0-387-20473-3.

[227] Johannes Köbler. *Kryptologie I*, chapter Kryptoanalyse der klassischen Verfahren, pages 28–42. Köbler, 2005.

[228] R. Koch. Changing Network Behavior. In *Network and System Security, 2009. NSS '09. Third International Conference on*, pages 60–66. IEEE, Oktober 2009. DOI: 10.1109/NSS.2009.55.

[229] Robert Koch and Björn Stelte. Vorstudie HeviSens. Wissenschaftliche studie, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85579 Neubiberg, 2009. Unterauftrag für Secunet. BSI-Studie HeviSens.

[230] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. SOM PAK: The self-organizing map program package. *Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science*, 1996.

[231] Chao Kong, Bo Yang, Zhiping Jia, and Zhenxiang Chen. A Common On-board Hardware Architecture for Intrusion Detection System. In *Multimedia Information Networking and Security*. IEEE Computer Society, IEEE, 2009. 978-0-7695-3843-3/09.

[232] Jan Korenek and Petr Kobiersky. Intrusion Detection System Intended for Multi-gigabit Networks. In *unk*. IEEE, 2007. 1-4244-1161-0/07.

[233] Daniel Kotschate. TTL Werte von Betriebssystemen. Website, September 2007. http://www.epyx-online.de/2007-09-20/ttl-werte-von-betriebssystemen/.

[234] KPMG. e-Crime-Studie 2010, 2010. http://www.kpmg.de/Themen/21481.htm.

[235] Claudia Krauß. Internet am Arbeitsplatz. Website, 2004. http://www.jurpc.de/aufsatz/20040014.htm, aufgerufen am 28. Januar 2011.

[236] Sandeep Kumar. IDIOT Intrusion Detection In Our Time. Website. http://www.cerias.purdue.edu/about/history/coast/projects/, aufgerufen am 13. März 2011.

[237] Sandeep Kumar and Eugene H. Spafford. An Application of Pattern Matching in Intrusion Detection. Technical report, Department of Computer Science, Purdue University, West Lafayette, IN 47907-1398, Juni 1994. Technical Report CSD-TR-94-013.

[238] NSS Labls. Network Intrusion Prevention Systems Test Methodology V6.1. Website, 2010. http://www.nsslabs.com/assets/Methodologies/nsslabsipsgrouptestmethodologyv6.1.pdf, aufgerufen am 4. April 2011.

[239] Lawrence Berkeley National Laboratory. Bro Intrusion Detection System. Website. http://www.bro-ids.org/, aufgerufen am 4. April 2011.

[240] AKMA Labs. FlowMatrix Network Behavior Analysis System. Website. http://www.akmalabs.com/flowmatrix.php, aufgerufen am 7. April 2011.

[241] NSS Labs. Network Intrusion Prevention Systems Test Methodology V6. Website, 2009. http://www.nsslabs.com/assets/Methodologies/NSSLabs_IPS_GroupTestMethodology_v6.pdf, aufgerufen am 4. April 2011.

[242] NSS Labs. Network Intrusion Prevention Systems Individual Product Test Results McAffee Network Security Platform M-8000. Technical report, NSS Labs, September 2010.

[243] Lancope. NetFlow versions. http://netflowv7.com/, aufgerufen am 16. März 2011.

[244] Mary Landesman. Jerusalem virus. Website. http://antivirus.about.com/cs/virusencyclopedia/p/jerusalem.htm, aufgerufen am 24. Januar 2011.

[245] Kwok Law and Lam Kwok. IDS False Alarm Filtering Using KNN Classifier. In Chae Lim and Moti Yung, editors, *Information Security Applications*, volume 3325 of *Lecture Notes in Computer Science*, pages 114–121. Springer Berlin / Heidelberg, 2005. ISBN: 978-3-540-24015-0.

[246] George Lawton. New Technology Prevents Data Leakage. *Journal Computer*, 41(9):14–17, September 2008. doi: 10.1109/MC.2008.394.

[247] Elizabeth B. Lennon. Testing intrusion detection systems. In *iTL Bulletin*. Information Technology Laboratory ITL, National Institute of Standards and Technology NIST, Juli 2003.

[248] Yang Li, Binxing Fang, Li Guo, and You Chen. Network anomaly detection based on TCM-KNN algorithm. In *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, ASIACCS '07, pages 13–19, New York, NY, USA, 2007. ACM. ISBN 1-59593-574-6.

[249] DI Management Services Pty Limited. Using Padding in Encryption. Website, August 2010.

[250] Cheng-Hung Lin and Shih-Chieh Chang. Efficient Pattern Matching Algorithm for Memory Architecture. In *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, volume 19. IEEE, Januar 2011. DOI 10.1109/TVLSI.2009.2028346.

[251] Daw-Tung Lin. Computer-access authentication with neural network based keystroke identity verification. In *Neural Networks,1997., International Conference on*, volume 1, pages 174–178, Juni 1997. DOI 10.1109/ICNN.1997.611659.

[252] Richard Lippmann, David Fried, Keith Piwowarski, and William Streilein. Passive Operating System Identification From TCP/IP Packet Headers. In *unk*, 224 Wood Street, Lexington, MA 02173, USA, 2003. MIT Lincoln Laboratory.

[253] Richard Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, and Kumar Das. The 1999 DARPA Off-Line Intrusion Detection Evaluation. Technical report, Lincoln Laboratory MIT, 244 Wood Street, Lexington, MA, 2000.

[254] Richard Lippmann, Seth Webster, and Douglas Stetson. The effect of identifying vulnerabilities and patching software on the utility of network intrusion detection. In *Proceedings of the 5th international conference on Recent advances in intrusion detection*, RAID'02, pages 307–326, Berlin, Heidelberg, 2002. ACM, Springer-Verlag. http://portal.acm.org/citation.cfm?id=1754701.1754725.

[255] Richard P. Lippmann, David J. Fried, Isaac Graf, Joshua W. Haines, and David Kendall, Kristopher R. amd McClung. Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation. In *unk*. IEEE, IEEE, 1999. 0-7695-0490-6/99.

[256] Rapid7 LLC. Metasploit - Penetration Testing Resources. Website. http://www.metasploit.com/, aufgerufen am 9. März 2011.

[257] Canonical Ltd. Ubuntu. Website. http://www.ubuntu.com, last seen 21. Juli 2011.

[258] Computer Network Defence Ltd (CND Ltd). Talisker Computer Network Defence Operational Picture. Website. http://www.securitywizardry.com/radar.htm, aufgerufen am 18. April 2011.

[259] Virus Bulletin Ltd. Virus Bulletin VB100 award. Website. http://www.virusbtn.com/vb100/latest_comparative/index, aufgerufen am 23. August 2011.

[260] VirusBlokAda Ltd. VBA32 Anti-Virus. Website. http://www.anti-virus.by/en/index.shtml, aufgerufen am 18. April 2011.

[261] Gordon (Fyodor) Lyon. Downloading Nmap. Website. http://nmap.org/download.html.

[262] Gordon (Fyodor) Lyon. insecure.org. Website. http://insecure.org/, aufgerufen am 8. März 2011.

[263] M86 Security Labs. Security Labs Report July - December 2009 Recap. Technical report, M86 Security Labs, 2009. www.m86security.com/labs.

[264] M. Mahoney and P.K. Chan. PHAD: Packet header anomaly detection for identifying hostile network traffic. *Florida Institute of Technology technical report CS-2001-04*, 2001. Citesser.

[265] Matthew Mahoney and Philip Chan. An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection. In Giovanni Vigna, Christopher Kruegel, and Erland Jonsson, editors, *Recent Advances in Intrusion Detection*, volume 2820 of *Lecture Notes in Computer Science*, pages 220–237. Springer Berlin / Heidelberg, 2003. DOI: 10.1007/978-3-540-45248-5_13.

[266] Mo Chun Man and Victor K. Wei. A Taxonomy for Attacks on Mobile Agent. In *Trends in Communications, EUROCON 2001*, volume 2, pages 385–388, 2001.

[267] ManageEngine. Analyze Bandwidth: NetFlow Analyzer. Website. http://www.manageengine.com/products/netflow/download.html, aufgerufen am 16. April 2011.

[268] Thomas Marill and Lawrence G. Roberts. Toward a Cooperativ Network of Time-Shared Computers. In *AFIPS 66 (Fall) Proceedings of the November 7-10, 1966, fall joint computer conference*, pages 425–431, 1966.

[269] E.P. Markatos, S. Antonatos, M. Polychronakis, and K.G. Anagnostakis. Exclusion-based signature matching for intrusion detection. In *Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN)*, pages 146–152. Citesser, 2002.

[270] Andreas Marx. Die Geschichte der Computerviren. Website, 1997. http://www.virushelpmunich.de/konferenz/1997/history.htm, aufgerufen am 7. März 2011.

[271] Roy A. Maxion and M.C. Tan Kymie. Benchmarking Anomaly-Based Detection Systems. In *DSN*, DSN 00. IEEE, IEEE, 2000. 0-7695-0707-7/00.

[272] M. McCormick. Data Theft: A Prototypical Insider Threat. *Advances in Information Security*, 39(1):53–68, April 2008. ISBN-10:0-387-77321-5.

[273] J. McHugh, A. Christie, and J. Allen. Defending yourself: the role of intrusion detection systems. *Software, IEEE*, 17(5):42–52, Sep/Oct 2000. DOI: 10.1109/52.877859.

[274] John McHugh. Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *ACM Trans. Inf. Syst. Secur.*, 3(4):262–294, November 2000. DOI: http://doi.acm.org/10.1145/382912.382923.

[275] Robert McMillian. After Hack, RSA Offers to Replace SecureID Tokens. Website, Juni 2011. http://www.pcworld.com/businesscenter/article/229553/after_hack_rsa_offers_to_replace_secureid_tokens.html, aufgerufen am 23. August 2011.

[276] Peter Mell, Vincent Hu, Richard Lippmann, Josh Haines, and Marc Zissmann. An Overview of Issues in Testing Intrusion Detection Systems. Technical report, National Institute of Standards and Technology ITL and MIT Lincoln Laboratory, 2003. http://csrc.nist.gov/publications/nistir/nistir-7007.pdf, aufgerufen am 4. April 2011.

[277] Nikolay Melnikov and Jürgen Schönwälder. Cybermetrics: User Identification through Network Flow Analysis. In Burkhard Stiller and Filip De Turck, editors, *Mechanisms for Autonomous Management of Networks and Services*, volume 6155 of *Lecture Notes in Computer Science*, pages 167–170. Springer Berlin / Heidelberg, 2010. DOI: 10.1007/978-3-642-13986-4_24.

[278] Trend Micro. Data Loss Prevention. Website. http://de.trendmicro.com/imperia/md/content/de/products/data-loss-prevention/datasheet_data-loss-prevention_101013_de.pdf.

[279] Kevin D. Mitnick and William L. Simon. *The Art of Deception: Controlling the Human Element of Security*. John Wiley & Sons, Inc., New York, NY, USA, 2003.

[280] Abhishek Mitra, Walid Najjar, and Laxmi Bhuyan. Compiling PCRE to FPGA for Accelerating SNORT IDS. In *Architectures for Networking and Communications Systems*, ANCS 07. ACM/IEEE, ACM, Dezember 2007. ACM 978-1-59593-945-6/07/0012.

[281] Mitre. CVE-2006-2341. Website, Mai 2006. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2341, aufgerufen am 13. März 2011.

[282] Mitre. CVE-2008-0971. Website, Dezember 2008. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0971, aufgerufen am 13. März 2011.

[283] Mitre. CVE-2011-0394. Website, 2011. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-0394, aufgerufen am 13. März 2011.

[284] Massimiliano Montoro. Cain & Abel. Website. http://www.oxid.it/cain.html, aufgerufen am 10. März 2011.

[285] Andrew Moore and Konstantina Papagiannaki. Toward the Accurate Identification of Network Applications. In Constantinos Dovrolis, editor, *Passive and Active Network Measurement*, volume 3431 of *Lecture Notes in Computer Science*, pages 41–54. Springer Berlin / Heidelberg, 2005.

[286] Andrew P. Moore, Robert J. Ellison, and Richard C. Linger. Attack Modeling for Information Security and Survivability. Technical report, Carnegie Mellon University, März 2001. Technical Note CMU/SEI-2001-TN-001.

[287] Benjamin Morin and Ludovic Mé. Intrusion detection and virology: an analysis of differences, similarities and complementariness. *Journal in Computer Virology*, 3(1):39–49, 2007. http://dx.doi.org/10.1007/s11416-007-0036-2.

[288] msdn. Data Execution Prevention. Website. http://msdn.microsoft.com/en-us/library/aa366553%28v=vs.85%29.aspx, aufgerufen am 11. März 2011.

[289] NASK. ARAKIS. Website, 2007. http://www.arakis.pl/en/index.html, aufgerufen am 4. April 2011.

[290] Jeff Nathan. Nemesis. Website. http://nemesis.sourceforge.net/, aufgerufen am 16. April 2011.

[291] Netfilter. patch-o-matic external repository. Website. http://www.netfilter.org/projects/patch-o-matic/pom-external.html, aufgerufen am 15. August 2011.

[292] NetOptics. Netzwerk Taps / Ethernet Taps / Fiber Taps / Copper Taps. Website. http://www.network-taps.de/products/products_networktaps.php, aufgerufen am 15. August 2011.

[293] Arbor Networks. ATLAS Dashboard Global. Website. http://atlas.arbor.net/, aufgerufen am 2. Februar 2011.

[294] THETA Networks. Shallow Packet Inspection. Website. http://www.thetanetworks.com/resources/shallow_packet_inspection.html, aufgerufen am 18. April 2011.

[295] James Newsome, Elaine Shi, Dawn Song, and Adrian Perring. The Sybil Attack in Sensor Networks: Analysis & Defenses. In *Proceedings to the ISPN*. ACM, 2004.

[296] Landon Curt Noll. Fowler / Noll / Vo (FNV) Hash. Website. http://isthe.com/chongo/tech/comp/fnv/, aufgerufen am 25. Mai 2011.

[297] Inc. NSS Labs. NSS Labs Tests 13 Leading Intrusion Prevention Systems. Website, Januar 2011. http://www.nsslabs.com/assets/other/PR_2010_IPS_GTR_FINAL.pdf.

[298] Philippe Oechslin. Making a Faster Cryptanalytic Time-Memory Trade-Off. In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 617–630. Springer Berlin / Heidelberg, 2003. http://lasecwww.epfl.ch/pub/lasec/doc/Oech03.pdf.

[299] Renate Oettinger. Haftungsfragen rund um die IT-Sicherheit. Website, Februar 2010. http://www.computerwoche.de/management/compliance-recht/1926396/index7.html, aufgerufen am 22. Februar 2011.

[300] MoMe Cluster of European Projects aimed at Monitoring and Measurement. MOME Database. Website, 2009. http://www.ist-mome.org/database/, aufgerufen am 11. August 2011.

[301] University of Southern California. RFC791 - Internet Protocol. Protocol specification, DARPA Defense Advanced Research Projects Agency, September 1981. http://www.faqs.org/rfcs/rfc791.html.

[302] University of Southern California. RFC793 - Transmission Control Protocol. Protocol specification, DARPA Defense Advanced Research Projects Agency, September 1981. http://www.faqs.org/rfcs/rfc793.html.

[303] Tim O'Neill. SPAN Port or TAP? Website, August 2007. http://www.lovemytool.com/blog/2007/08/span-ports-or-t.html, aufgerufen am 13. April 2011.

[304] Spiegel Online. Einkaufen im Web. Website, Juli 2009. http://www.spiegel.de/wirtschaft/0,1518,638779,00.html, aufgerufen am 3. Februar 2011.

[305] Spiegel Online. Milliardenumsatz. Website, November 2009. http://www.spiegel.de/wirtschaft/unternehmen/0,1518,661789,00.html.

[306] M. Otey, S. Parthasarathy, A. Ghoting, G. Li, S. Narravula, and D. Panda. Towards NIC-based intrusion detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 723–728, New York, NY, USA, 2003. ACM. http://doi.acm.org/10.1145/956750.956847.

[307] Hewlett Packard. ProCurve Networking by HP - Application notes. Website. , aufgerufen am 16. März 2011.

[308] Kostas Pagiamtzis. Content-Addressable Memory Introduction. Website, Juni 2007. http://www.pagiamtzis.com/cam/camintro.html, aufgerufen am 15. April 2011.

[309] Pandalabs. Annual Report Pandalabs 2009. Technical report, Pandalabs, 2010. www.pandasecurity.com, aufgerufen am 4. Februar 2011.

[310] plixer International. Scrutinizer NetFlow & sFlow Analyzer. Website. http://www.plixer.com/products/netflow-sflow/free-netflow-scrutinizer.php, aufgerufen am 16. April 2011.

[311] Klaus Pommerening. Kryptographische Basisfunktionen und ihre Äquivalenz, 2. Hashfunktionen. Website, 2000. http://www.staff.uni-mainz.de/pommeren/Kryptologie/Asymmetrisch/6_Einweg/.

[312] Openwall Project. John the Ripper password cracker. Website. http://www.openwall.com/john/, aufgerufen am 10. März 2011.

[313] The Tor Project and Electronic Frontier Foundation. HTTPS Everywhere. Website, 2010. http://www.eff.org/https-everywhere, aufgerufen am 16. Februar 2011.

[314] D.V. Pryor, M.R. Thistle, and N. Shirazi. Text searching on splash 2. In *FPGAs for Custom Computing Machines, 1993. Proceedings. IEEE Workshop on*, pages 172–177. IEEE, 1993. ISBN 0818638907.

[315] Thomas H. Ptacek and Timothy N. Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical report, SECURE NETWORKS INC CALGARY ALBERTA, Januar 1998.

[316] N.J. Puketza, K. Zhang, M. Chung, B. Mukherjee, and R.A. Olsson. A methodology for testing intrusion detection systems. In *Software Engineering, IEEE Transactions on*, volume 22, pages 719–729. IEEE, 1996. ISSN 0098-5589.

[317] Jun Qian, Chao Xu, and Meilin Shi. Redesign and Implementation of Evaluation Dataset for Intrusion Detection System. In G. Müller, editor, *LNCS 3995*, ETRICS 2006, pages 451–465. Springer-Verlag Berlin Heidelberg, 2006.

[318] Yan Qiao and Xie Weixin. A Network IDS with low false positive rate. In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, volume 2, pages 1121 –1126, 2002. DOI: 10.1109/CEC.2002.1004400.

[319] Serge Radovcic. European Internet Exchange Association – 2010 Report on European IXPs. Technical report, Euro-IX, 2011. http://www.euro-ix.net, aufgerufen am 18. April 2011.

[320] Daniel Ramsbrock, Robin Berthier, and Michel Cukier. Profiling Attacker Behavior Following SSH Compromises. In *Dependable Systems and Networks*, volume 37 of *IEEE/IFIP International Conference DSN07*. IEEE, IEEE Computer Society, 2007.

[321] Marcus J. Ranum. Experiences Benchmarking Intrusion Detection Systems. Technical report, NFR Security, Inc., Dezember 2001.

[322] Marcus J. Ranum. Experiences Benchmarking Intrusion Detection Systems. Technical report, NFR Security, Dezember 2001.

[323] Frank Rieger. Trojaner Stuxnet: Der digitale Erstschlag ist erfolgt. Website, September 2010. FAZ.NET, http://www.faz.net/-01r3qd, aufgerufen am 18. April 2011.

[324] A. Riezler. Definition des Begriffs Mittelstand. Website, 2010. http://www.mittelstand-optimierung.de/definition_mittelstand.shtml.

[325] Lawrence G. Roberts. Multiple Computer Networks and Intercomputer Communication. In *SOSP 67 Proceedings of the first ACM symposium on Operating System Principles*, New York, 1967. ACM.

[326] Grigore Roşu. Equality of streams is a π_2^0 -complete problem. *SIGPLAN Not.*, 41(9):184–191, September 2006. DOI: http://doi.acm.org/10.1145/1160074.1159827.

[327] Jeanine Rother. Die Geschichte der Computerviren. Website, 2005. http://www.securitymanager.de/magazin/artikel_742_die_geschichte_der_computerviren.html, as seem on 7. März 2011.

[328] G. Roualland and J.M. Saffroy. IP Personality. Website, 2001. http://ippersonality.sourceforge.net/, aufgerufen am 7. März 2011.

[329] Paul Rouget. Firefox 4: HTTP Strict Transport Security (force HTTPS). Website, August 2010. http://hacks.mozilla.org/2010/08/firefox-4-http-strict-transport-security-force-https/, aufgerufen am 16. Februar 2011.

[330] Joanna Rutkowska. Red pill... or how to detect VMM using (almost) one CPU instruction. Website, November 2004. http://invisiblethings.org/papers/redpill.html, aufgerufen am 11. März 2011.

[331] Joanna Rutkowska and Rafal Wojtczuk. Qubes OS Architecture. Technical report, Invisible Things Lab, Januar 2010. http://qubes-os.org/files/doc/arch-spec-0.3.pdf, aufgerufen am 13. März 2011.

[332] M. Rybnik, P. Panasiuk, and K. Saeed. User Authentication with Keystroke Dynamics Using Fixed Text. In *Biometrics and Kansei Engineering, 2009. ICBAKE 2009. International Conference on*, pages 70–75. IEEE Computer Society, Juni 2009. DOI: 10.1109/ICBAKE.2009.42.

[333] F. Sabahi and A. Movaghar. Intrusion Detection: A Survey. In *Systems and Networks Communications, 2008. ICSNC '08. 3rd International Conference on*, pages 23–26. IEEE Computer Society, Oktober 2008. DOI 10.1109/ICSNC.2008.44.

[334] Scott C. Sanchez. IDS Zone Theory Diagram, Juli 2000. http://infosec.gungadin.com.

[335] SANS. Survival Time. Website. http://isc.sans.edu/survivaltime.html aufgerufen am 1. Februar 2011.

[336] Gregor Schaffrath. Network Intrusion Detection Systems & Encryption: Friends or Foes? Presentation, Aug 2008. Communication Systems Group, University of Zürich.

[337] Jürgen Schmidt. JIT-Spraying: Exploits trotz DEP und ASLR. Website, Januar 2011. http://www.heise.de/security/artikel/Die-Rueckkehr-des-Sprayers-Exploits-trotz-DEP-und-ASLR-1169279.html, aufgerufen am 19. Februar 2011.

[338] Bruce Schneier. Attack Trees. Presentation, Oktober 1999. SANS Network Security 99.

[339] Bruce Schneier. Real-World Passwords. Website, Dezember 2006. http://www.schneier.com/blog/archives/2006/12/realworld_passw.html, aufgerufen am 11. August 2011.

[340] Erick Schonfeld. Google Processing 20,000 Terabytes A Day, And Growing. Website, Januar 2008. http://techcrunch.com/2008/01/09/google-processing-20000-terabytes-a-day-and-growing/, aufgerufen am 31. Januar 2011.

[341] Tenable Network Security. the Network Vulnerability Scanner. Website. http://www.nessus.org/nessus/, aufgerufen am 9. März 2011.

[342] The H Security. Secure deletion: a single overwrite will do it. Website, Januar 2009. Aufgerufen am 12. März 2011.

[343] The H Security. Microsoft issues warning about critical IE hole. Website, Dezember 2010. http://www.h-online.com/security/news/item/Microsoft-issues-warning-about-critical-IE-hole-1158684.html?view=print, aufgerufen am 15. April 2011.

[344] The H Security. Microsoft warns of thumbnail hole in Windows. Website, Januar 2011. http://www.h-online.com/security/news/item/Microsoft-warns-of-thumbnail-hole-in-Windows-1163562.html?view=print, aufgerufen am 15. April 2011.

[345] SecurityFocus. BugTraq. Website. http://www.securityfocus.com/archive/1, aufgerufen am 9. März 2011.

[346] Christian Seifert. Analyzing Malicious SSH Login Attempts. Website, September 2009.

[347] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou. Specification-based anomaly detection: a new approach for detecting network intrusions. In *Proceedings of the 9th ACM conference on Computer and communications security*, CCS '02, pages 265–274, Washington, DC, USA, 2002. ACM. ISBN: 1-58113-612-9.

[348] M. Sharif, T. Faiz, and M. Raza. Time signatures - an implementation of Keystroke and click patterns for practical and secure authentication. In *Digital Information Management, 2008. ICDIM 2008. Third International Conference on*, pages 559 – 562. IEEE Computer Society, November 2008. DOI: 10.1109/ICDIM.2008.4746782.

[349] ACM SIGCOMM. Internet Traffic Archive. Website. http://www.sigcomm.org/ITA/, aufgerufen am 11. August 2011.

[350] Mike Simons. Ministry of Defence in new data loss scandal. Website, Oktober 2008. http://www.cio.co.uk/news/3225/ministry-of-defence-in-new-data-loss-scandal/, aufgerufen am 7. März 2011.

[351] Stephen E. Smaha. Haystack: An Intrusion Detection System. In *Proceedings to the Fourth Aerospace Computer Security Applications Conference*, pages 37–44. IEEE, 1988. DOI: 10.1109/ACSAC.1988.113412.

[352] Craig Smith and Peter Grundl. Passive Fingerprinting. Website, Honeynet Project, März 2002. http://old.honeynet.org/papers/finger/.

[353] Jon A. Solworth. Ethos: an operating system which creates a culture of security. Website, 2007. http://rites.uic.edu/~solworth/ethos.html, aufgerufen am 13. März 2011.

[354] SOURCEfire. Network Awareness. Website. http://www.sourcefire.com/security-technologies/cyber-security-products/3d-system/network-awareness, aufgerufen am 7. April 2011.

[355] SOURCEfire. Snort. Website. http://www.snort.org/snort, aufgerufen am 4. April 2011.

[356] I. Sourdis, D.N. Pnevmatikatos, and S. Vassiliadis. Scalable multigigabit pattern matching for packet inspection. In *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, volume 16, pages 156–166. IEEE, 2008. 1063-8210.

[357] Ioannis Sourdis and Dionisios Pnevmatikatos. Fast, Large-Scale String Match for a 10Gbps FPGA-Based Network Intrusion Detection System. In *Field-Programmable Logic and Applications*, volume 2778 of *Lecture Notes in Computer Science*, pages 880–889. Springer Berlin / Heidelberg, 2003.

[358] Horst Speichert. *Praxis des IT-Rechts*. Friedr. Vieweg & Sohn Verlag, GWV Fachverlage GmbH, Wiesbaden, Mai 2007. ISBN: 978-3-8348-0112-8.

[359] Anna Sperotto. *Flow-Based Intrusion Detection*. PhD thesis, University of Twente, The Netherlands, 2010.

[360] Anna Sperotto, Ramin Sadre, Frank van Vliet, and Aiko Pras. A Labeled Dataset for Flow-Based Intrusion Detection. In G. Nunzi, C. Scoglio, and X. Li, editors, *LNCS 5843*, IPOM 2009, pages 39–50. Springer-Verlag Berlin Heidelberg, 2009.

[361] Anna Sperotto and Remco van de Meent. A Survey of the High-Speed Self-learning Intrusion Detection Research Area. In A.K. Bandara and M. Burgess, editors, *LNCS 4543*, AIMS 2007, pages 196–199. Springer-Verlag Berlin Heidelberg, 2007.

[362] D. Spinellis. Reliable identification of bounded-length viruses is NP-complete. *Information Theory, IEEE Transactions on*, 49(1):280–284, Januar 2003. DOI: 10.1109/TIT.2002.806137.

[363] Sid Stamm. Force-TLS. Website, November 2009. https://addons.mozilla.org/en-US/firefox/addon/force-tls/, aufgerufen am 16. Februar 2011.

[364] B. Stelte, R. Koch, and G. Dreo Rodosek. Vorstudie HeviSens. Technical report, Universität der Bundeswehr, 2009.

[365] B. Stelte, R. Koch, and M. Ullmann. Towards integrity measurement in virtualized environments - A hypervisor based sensory integrity measurement architecture (SIMA). In *Technologies for Homeland Security (HST), 2010 IEEE International Conference on*, pages 106–112. IEEE, November 2010. DOI 10.1109/THS.2010.5655084.

[366] Packet Storm. adore, adore-ng rootkit. Website, 2000. http://packetstormsecurity.org/search/files/?q=adore-ng%20rootkit, aufgerufen am 11. März 2011.

[367] Chris Sullo and David Lodge. Nikto2. Website. http://cirt.net/nikto2, aufgerufen am 9. März 2011.

[368] Aurobindo Sundaram. An introduction to intrusion detection. *Crossroads*, 2(4):3–7, April 1996. DOI: http://doi.acm.org/10.1145/332159.332161.

[369] symweb. Zero-Day-Exploit. Website. http://www.symweb.de/glossar/zero-day-exploit__973.htm, aufgerufen am 8. März 2011.

[370] 16 Systems. The Great Zero Challenge. Website. http://16s.us/zero/, aufgerufen am 12. März 2011.

[371] T. Taleb, Z.M. Fadlullah, K. Hashimoto, Y. Nemoto, and N. Kato. Tracing back attacks against encrypted protocols. In *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, pages 121–126. ACM, 2007.

[372] Gaurav Tandon, Philip Chan, and Debasis Mitra. MORPHEUS: motif oriented representations to purge hostile events from unlabeled sequences. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, VizSEC/DMSEC '04, pages 16–25, New York, NY, USA, 2004. ACM. ISBN 1-58113-974-8.

[373] Core Security Technologies. CORE IMPACT pro overview. Website. http://www.coresecurity.com/content/core-impact-overview, aufgerufen am 9. März 2011.

[374] Telecom ABC. RIR - Regional Internet Registry. Website, 2005. http://www.telecomabc.com/r/rir.html, aufgerufen am 13. August 2011.

[375] TeleGeography. TeleGeography Report. Website. http://www.telegeography.com/product-info/map_traffic/images/global-traffic-map-large.png, aufgerufen am 16. Februar 2011.

[376] The MITRE Corporation. MITRE – Applying Systems Engineering and Advanced Technology to Critical National Problems. Website. http://www.mitre.org/, aufgerufen am 23. August 2011.

[377] Marina Thottan and Chuanyi Ji. Anomaly Detection in IP Networks. In *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, volume 51, No. 8, August 2003. Digital Object Identifier 10.1 109/TSP.2003.814797.

[378] Rob Turner. Swiss tax dodge probe could net twice the amount of cash expected. Website, Februar 2010. http://www.dw-world.de/dw/article/0,,5201374,00.html, aufgerufen am 7. März 2011.

[379] Banken und Börsenlexikon. Definition Insider. Website. http://www.rba.ch/webmodule/tools/glossar/glossar/sites/doc.php?border=&hfont=&bspace=&revor=false&width=&site=result2.html&signfrom=I, aufgerufen am 17. Februar 2011.

[380] International Telecommunication Union. Data Networks and Open System Communications, Open Systems Interconnection – Model and Notation. ITU-T Recommencation X.200, Juli 1994.

[381] Vladimir N. Vapnik. An overview of statistical learning theory. In *Neural Networks, IEEE Transactions on*, volume 10, pages 988–999. IEEE, 1999.

[382] viprinet. Multichannel VPN Router. Website. http://www.viprinet.com/de/products/multichannel-vpn-router, aufgerufen am 8. Februar 2011.

[383] Ke Wang and Salvatore J. Stolfo. Anomalous Payload-Based Network Intrusion Detection. In Erland Jonsson, Alfonso Valdes, and Magnus Almgren, editors, *Recent Advances in Intrusion Detection*, volume 3224 of *Lecture Notes in Computer Science*, pages 203–222. Springer Berlin / Heidelberg, 2004. http://dx.doi.org/10.1007/978-3-540-30143-1_11.

[384] Erlang Web. Erlang Programming Language. Website. http://www.erlang.org/, aufgerufen am 11. August 2011.

[385] Ollie Whitehouse. An Analysis of Address Space Layout Randomization on Windows Vista. Technical report, Symantec Corporation, 2007.

[386] Thorsten Wichmann. VoIP, Messaging, Mobile Mail & Co. Technical report, Berlecon Research GmbH, Februar 2006.

[387] IT Law Wiki. Zero day exploit. Website. http://itlaw.wikia.com/wiki/Zero_day_exploit, aufgerufen am 8. März 2011.

[388] Philipp Winter, Eckehard Hermann, and Markus Zeilinger. Inductive Intrusion Detection in Flow-Based Network Data Using One-Class Support Vector Machines. In *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*. IEEE, 2011. DOI 10.1109/NTMS.2011.5720582.

[389] Gabler Wirtschaftslexikon. Globalisierung. Website. http://wirtschaftslexikon.gabler.de/Definition/globalisierung.html, aufgerufen am 15. Februar 2011.

[390] Wirtschaftsschutz Niedersächsisches Ministerium für Inneres und Sport. Sicherheitslücke Mensch.

[391] Richard Wray. Internet data heads for 500bn gigabytes, Mai 2009. http://www.guardian.co.uk/business/2009/may/18/digital-content-expansion/print.

[392] Charles V. Wright, Fabian Monrose, and Gerald M. Masson. On Inferring Application Protocol Behaviors in Encrypted Network Traffic. *J. Mach. Learn. Res.*, 7:2745–2769, Dezember 2006. http://portal.acm.org/citation.cfm?id=1248547.1248647.

[393] Craig Wright, Dave Kleiman, and Shyaam Sundhar R.S. Overwriting Hard Drive Data: The Great Wiping Controversy. In R. Sekar and Arun Pujari, editors, *Information Systems Security*, volume 5352 of *Lecture Notes in Computer Science*, pages 243–257. Springer Berlin / Heidelberg, 2008. http://dx.doi.org/10.1007/978-3-540-89862-7_21.

[394] Candid Wüest. The Risk of Social Networking. Technical report, Symantec Corporation, 2010.

[395] www.uni-protokolle.de. Echtzeit. Website. http://www.uni-protokolle.de/Lexikon/Echtzeit.html, aufgerufen am 15. Februar 2011.

[396] A. Yamada, Y. Miyake, K. Takemori, A. Studer, and A. Perrig. Intrusion detection for encrypted web accesses. Technical report, Department of Computer Science, Florida State University, 2007. IEEE Computer Society.

[397] Roman V. Yampolskiy and Venu Govindaraju. Behavioral biometrics: a survey and classification. *International Journal of Biometrics*, 1(1):81–113, 2008. Inderscience.

[398] Alec Yasinsac. An environment for security protocol intrusion detection. *Journal of Computer Security*, 10(1-2):177–188, 2002.

[399] Alec Yasinsac and Sachin Goregaoker. An Intrusion Detection System for Security Protocol Traffic. Citeseer.

[400] Adam Young and Moti Yung. Cryptovirology: Extortion-Based Security Threats and Countermeasures. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*. IEEE, Mai 1996.

[401] Yan Yu, Shanqing Guo, Shaohua Lan, and Tao Ban. Anomaly Intrusion Detection for Evolving Data Stream Based on Semi-supervised Learning. In Mario Köppen, Nikola Kasabov, and George Coghill, editors, *Advances in Neuro-Information Processing*, volume 5506 of *Lecture Notes in Computer Science*, pages 571–578. Springer Berlin / Heidelberg, 2009.

[402] Michael Zalewski. p0f - SYN fingerprints. Source Code, 2006.

[403] Michal Zalewski. the new p0f: 2.0.8 (2006-09-06). Website, September 2006. http://lcamtuf.coredump.cx/p0f.shtml, aufgerufen am 8. März 2011.

[404] Erik Zettel. SUID-, SGID- und Sticky-Bit. Website. http://www.zettel-it.de/docs/SUID-SGID-und-Sticky-Bit.pdf, aufgerufen am 10. März 2011.

[405] Huanguo Zhang. Research and Development of Trusted Computing in China. Presentation. School of Computer, Wuhan University.

[406] X. Zhang, C. Li, and W. Zheng. Intrusion prevention system design. In *Computer and Information Technology, 2004. CIT '04. The Fourth International Conference on*, pages 386–390. IEEE Computer Society, September 2004.

[407] Maria Zheng. CCP Sees Every Chinese as a Potential Spy. Internet, September 2007. aufgerufen am 14. November 2010.

B Abbildungsverzeichnis

C Tabellenverzeichnis

D Glossar

ARPANET
Von ARPA im Auftrag des US-Verteidigungsministeriums im Jahre 1969 entwickeltes, dezentrales und ausfallsicheres Netz.

Authentizität
Mit dem Begriff Authentizität wird die Eigenschaft bezeichnet, die gewährleistet, dass ein Kommunikationspartner tatsächlich derjenige ist, der er vorgibt zu sein. Bei authentischen Informationen ist sichergestellt, dass sie von der angegebenen Quelle erstellt wurden.

Blackbox
Gerät, dessen Funktionsweise von Außen nicht erkennbar ist und das keinen Zugriff auf die darin verarbeiteten Daten ermöglicht.

Bytecode
Assemblercode zum Einschleusen in einen Buffer im Rahmen eines Buffer-Overflows. Der Code muss eine geschlossene Einheit bilden und es dürfen bestimmte Sonderzeichen nicht genutzt werden.

Firesheep
Ein Addon für den Mozilla Firefox-Browser, zur Demonstration von HTTP session hijacking Angriffen.

Firewall
Gerät oder Software zur Filterung des Netzverkehrs anhand definierter Regeln.

Gateway
Protokollumsetzer. Mit der Verbreitung von IP-Netzen ist heutzutage nur noch selten eine Umsetzung notwendig, insb. entsprechen die sog. Default-Gatways mittlerweile meist Default-Routern.

https-Scanning
Technisches Verfahren zur Untersuchung einer verschlüsselten Verbindung auf Viren. Hierbei werden die Daten entschlüsselt, auf Schadsoftware gescannt und anschließend wieder verschlüsselt.

Industriespionage
Industriespionage muss in die beiden Felder Wirtschaftsspionage und Konkurrenzausspähung unterteilt werden. Ersteres bezeichnet die staatlich gelenkte Ausforschung von Wirtschaftsunternehmen durch fremde Nachrichtendienste, während Konkurrenzausspähung die Aktivitäten eines konkurrierenden Unternehmes gegen ein anderes Unternehmen meint.

Integrität
Integrität bezeichnet die Sicherstellung der Korrektheit (Unversehrtheit) von Daten und der korrekten Funktionsweise von Systemen. Wenn der Begriff Integrität auf Daten angewendet wird, drückt er aus, dass die Daten vollständig und unverändert sind.

IX
Netzinfrastruktur zum Datenaustausch zwischen ISPs.

Spam
Unsolicited bulk Email, unverlangte Massen-/ kommerzielle Email.

Sybil
Bei einem sogenannten Sybil Angriff versucht ein einzelner Angreifer in einem Peer-to-Peer Netzwerk [*sic.*] mit der Hilfe von massenhaft erzeugten, nicht wirklich existierenden Pseudo- Clients, einen großen Einfluß auf das Netz zu nehmen [159].

Vertraulichkeit
Vertraulichkeit ist der Schutz vor unbefugter Preisgabe von Informationen. Vertrauliche Daten und Informationen dürfen ausschließlich Befugten in der zulässigen Weise zugänglich sein.

Wirtschaftskriminalität
Zur Wirtschaftskriminalität existiert in Deutschland keine eindeutige Legaldefinition. Die Polizei bedient sich daher bei der Zuordnung von Straftaten zur Wirtschaftskriminalität des Katalogs von §74c Abs. 1 Nr. 1 bis 6b des GVG. Hierzu zählen u.a. die Bereiche Kreditbetrug, Insolvenzdelikte, Subventionsbetrug und Wucher. Obwohl allgemein dem Bereich der Wirtschaftskriminalität zugerechnet, fällt Computerbetrug wegen der Dominanz der Automatenmanipulationen gemäß Abstimmung mit der Kommission Wirtschaftskriminalität nicht immer darunter. Der Bereich Computerkriminalität wird daher in den PKSen des BKA extra geführt.

E Abkürzungsverzeichnis

ACK	Acknowledge
AES128-CBC	Advanced Encryption Standard 128 bit Cipher Block Chaining
AFRINIC	African Network Information Center
AIX	Amsterdam Internet Exchange
APNIC	Asia Pacific Network Information Center
ARIN	American Registry for Internet Numbers
ARP	Address Resolution Protocol
ARPA	Advanced Research Projects Agency
ASLR	Address Space Layout Randomization
ATLAS	Active Threat Level Analysis System
ATM	Asynchronous Transfer Mode
AUNIC	Australian Network Information Center
bash	bourne again shell
BDSG	Bundesdatenschutzgesetz
BetrVG	Betriebsverfassungsgesetz
BGH	Bundesgerichtshof
BIOS	Basic Input Output System
BKA	Bundeskriminalamt
BMI	Bundesministerium des Inneren
BPersVG	Personalvertretungsgesetz des Bundes
bps	bits per second
BRAS	Broadband Remote Access Server
BSI	Bundesamt für Sicherheit in der Informationstechnik
BSS	Block Started by Symbol
C&C	Command and Control, auch C2
CERT	Computer Emergency Response Team
CIDS	Central IDS
CNAME	Canonical Name
CRC	Cyclic Redundancy Check
CSRF	Cross-Site Request Forgery
CVE	Common Vulnerabilities and Exposures
DARPA	Defense Advanced Research Projects Agency

DDoS	Distributed Denial of Service
DEP	Data Execution Prevention
DES	Data Encryption Standard
DF	Don't Fragment
DLP	Data Leakage Prevention
DMI	Desktop Management Interface
DMZ	Demilitarized Zone
DNS	Domain Name System
DoS	Denial of Service
DPI	Deep Packet Inspection
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
EB	Exabyte
ECB	Electronic Code Book
EDV	Elektronische Datenverarbeitung
EMERALD	Monitoring Enabling Responses to Anomalous Live Disturbances
ESP	Encapsulating Security Payload
EWS	Early Warning System
FAR	False Alert Rate
FBI	Federal Bureau of Investigation
FCS	Frame Check Sequence
FIN	No more data from sender
FN	False Negative
FNV	Fowler, Noll und Vo
FP	False Positive
FPGA	Field Programmable Gate Array
FSH	File System Hierarchy
FTP	File Transfer Protocol
GB	Gigabyte
Gbps	Gigabit per second
GHz	Gigahertz
HMM	Hidden Markov Model
HSTS	HTTP Strict Transport Security
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol

IDS	Intrusion Detection System
IDT	Interrupt Description Table
IDTR	Interrupt Description Table Register
IETF	Internet Engineering Task Force
IHL	Internet Header Length
IMAP	Internet Message Access Protocol
IMAPS	IMAP over SSL
IOS	Internetwork Operating System
IP	Internet Protocol
IPFIX	Internet Protocol Flow Information Export
IPS	Intrusion Prevention System
IPsec	Internet Protocol Security
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IPX	Internetwork Packet eXchange
ISC	Internet Storm Center
ISO	International Organization for Standardization
ISP	Internet Service Provider
IT	Informationstechnologie
ITL	Information Technology Laboratory
IuK	Informations- und Kommunikationstechnik
IX	Internet Exchange
IXP	Internet Exchange Provider
JIT	Just-in-Time
JPNIC	Japan Network Information Center
k-NN	k-Nearest Neighbor
Kbps	Kilobits per second
KDD	Knowledge Discovery and Data Mining
KMU	Kleine und mittlere Unternehmen
LACNIC	Latin and Caribbean Internet Addresses Registry
LAN	Local Area Network
LIDS	Learning Intrusion Detection System
LoC	Lines of Code
LTS	Long Term Support
LWL	Lichtwellenleiter
MAC	Media Access Control
MAWI	Measurement and Analysis on the WIDE Internet
MB	Megabyte
Mbps	Megabit per second
MFM	Magnetic Force Microscopy

MIB	Management Information Base
MITM	Man-in-the-Middle
mp3	MPEG-1 und MPEG-2 Audio Layer 3
MPLS	Multi-Protocol Label Switching
ms	Millisekunden
MSS	Maximum Segment Size
NAT	Network Adress Translation
NIC	Network Interface Card
NIDS	Network-based Intrusion Detection System
NIST	National Institute of Standards and Technology
NISTIR	NIST Interagency or Internal Report
nmap	Network Mapper
OS	Operating System
OSI	Open System Interconnection
OTP	Open Transaction Platform
OWASP	The Open Web Application Security Project
P2P	Peer-to-Peer
P4	Pentium 4
pcap	Packet Capture Library
PHAD	Packet Header Anomaly Detector
PKI	Public Key Infrastructure
PKS	Polizeiliche Kriminalstatistik
PLC	Programmable Logic Controller
POP	Post Office Protocol
POPS	POP over SSL
R2L	Remote-to-Local
RCP	Remote Copy
RDP	Remote Desktop Protocol
RFC	Request for Comments
RFID	Radio Frequency Identification
RIPE	Réseaux IP Européens
RIPE NCC	RIPE Network Coordination Centre
Rlogin	Remote login
RSH	remote shell
RST	Reset
RTP	Realtime Transport Protocol
RTT	Round Trip Time
S2E2	Security System for Encrypted Environments
SANS	SysAdmin, Audit, Network, Security

SCADA	Supervisory Control and Data Aquisition
SCP	Secure Copy
SCTP	Stream Control Transmission Protocol
SELinux	Security-Enhanced Linux
SFTP	Secure File Transfer Protocol
SFV	Standardfestverbindung
SGID	Set-Group-ID
SIDT	Store IDT Register
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SMTP	Simple Mail Transfer Protocol
SMTPS	SMTP over SSL
SNMP	Simple Network Management Protocol
SOA	Start of Authority
SOM	Self-Organizing Feature Maps
SQL	Structured Query Language
SRC	Supercomputing Research Center, Bowie, MD, USA
SSH	Secure Shell
SSL	Secure Sockets Layer
StGB	Strafgesetzbuch
SUID	Set-User-ID
SVM	Support Vector Machine
SYN	Synchronize
TAB	Tabulator
TB	Terabyte
TCAM	Ternary Content-Addressable Memories
TCO	Total Cost of Ownership
TCP	Trusted Computing Platform
TCP	Transmission Control Protocol
TDDSG	Teledienstedatenschutzgesetz
TK	Telekommunikation
TKG	Telekommunikationsgesetz
TKÜV	Telekommunikations-Überwachungsverordnung
TLS	Transport Layer Security
TN	True Negative
TOS	Type of Service
TOTEM	TOolbox for Traffic Engineering Methods
TP	True Positive
TPM	Trusted Platform Module
TTL	Time-to-Live
TWNIC	Taiwan Network Information Center
U2R	User-to-Root

UDP	User Datagram Protocol
URL	Uniform Resource Locator
USB	Universal Serial Bus
USSS	United States Secret Service
VLAN	Virtual LAN
VM	Virtual Machine
VNC	Virtual Network Computing
VNI	Visual Networking Index
VoD	Video on Demand
VoIP	Voice over IP
VPN	Virtual Private Network
WAN	Wide Area Network
WEP	Wired Equivalent Privacy
wmv	Windows Media Video
WOMBAT	Worldwide Observatory of Malicious Behaviors and Attack Threats
WPA	Wi-Fi Protected Access
WWW	World Wide Web
XML	Extensible Markup Language
XOR	Exclusive OR
XSS	Cross-Site Scripting

F Anhang

F.1 Ergänzungen zu Kapitel 2

Nachfolgend finden sich Ergänzungen und detaillierte Ausführungen zu den in Kapitel 2 behandelten Themen. Hierzu gehören insbesondere die Analyse des Szenarios sowie der Angriffsdurchführung auf IT-Systeme.

F.1.1 Gefährdungen gem. Grundschutzkataloge

Nachfolgend sind die für das in Kapitel 2 vorgestellte Szenario gem. IT-Grundschutzkatalogen identifizierten relevanten Gefährdungen aufgelistet. Die Gefährdungslage ist gem. den Katalogen in mehrere Bereiche aufgeteilt, wobei im vorliegenden Kontext insbesondere *Menschliche Fehlhandlungen*, *Technisches Versagen* sowie *Vorsätzliche Handlungen* berücksichtigt werden müssen. Weiterhin sind die Referenzen zu den jeweiligen Kapiteln der Grundschutzkataloge der aufgeführten Gefährdungen sowie der zugehörigen Bausteine angegeben; die Relevanz für das vorliegende Szenario wird durch den entsprechenden Hinweise *Szenario* gezeigt.

Von den zu *Menschlichen Fehlhandlungen* zählenden Gefährdungen müssen im Szenario maßgeblich folgende Punkte berücksichtigt werden:

- Vertraulichkeits- oder Integritätsverlust von Daten durch Fehlverhalten G 3.1, B 1.7, B 1.9
- Fehlerhafte Nutzung von IT-Systemen G 3.8, B 1.9, B 4.1
- Weitergabe falscher oder interner Informationen G 3.13, *Szenario*
- Ungewollte Freigabe des Dateisystems G 3.26, *Szenario*
- Ungeeignete Konfiguration der aktiven Netzkomponenten G 3.28, B 4.1, B 4.2
- Unerlaubte private Nutzung des dienstlichen Telearbeitsrechners G 3.30, *Szenario*
- Sorglosigkeit im Umgang mit Informationen G 3.44, B 1.9, B 4.4, *Szenario*
- Fehlerhafte Konfiguration von Routern und Switchen G 3.64, *Szenario*

Die im Teilgebiet *Technisches Versagen* für das Szenario relevanten Gefährdungen sind:

- Bekanntwerden von Softwareschwachstellen G 4.8, B 1.9

- Komplexität der Zugangsmöglichkeiten zu vernetzten IT-Systemen G 4.10, B 1.9, B 4.1
- Software-Schwachstellen oder -Fehler G 4.22, B 1.6, B 1.7, B 1.9
- Ausfall oder Störung von Netzkomponenten G 4.31, B 1.9, B 4.1, B 4.2
- Unsichere Default-Einstellungen auf Routern und Switches G 4.49, *Szenario*

Weiterhin umfasst die Kategorie der *Vorsätzlichen Handlungen*:

- Manipulation an Informationen oder Software G 5.2, B 1.6, B 1.9, B 4.1, B 4.2
- Abhören von Leitungen G 5.7, B 4.1
- Unberechtigte IT-Nutzung G 5.9, B 1.9, B 4.1, B 4.2, *Szenario*
- Missbrauch von Fernwartungszugängen 5.10, *Szenario*
- Systematisches Ausprobieren von Passwörtern G 5.18, B 4.1, B 4.2, *Szenario*
- Missbrauch von Benutzerrechten G 5.19, *Szenario*
- Missbrauch von Administratorrechten G 5.20, B 4.1, *Szenario*
- Trojanische Pferde G 5.21, B 1.9, *Szenario*
- Schadprogramme G 5.23, B 1.6, B 1.9, *Szenario*
- Wiedereinspielen von Nachrichten G 5.24, *Szenario*
- Maskerade G 5.25, *Szenario*
- Verhinderung von Diensten G 5.28, B 1.6, B 4.1, B 4.2
- Missbrauch des ICMP-Protokolls G 5.50, *Szenario*
- Missbrauch der Routing-Protokolle G 5.51, *Szenario*
- Missbrauch von Remote-Zugängen für Managementfunktionen von Routern G 5.61, *Szenario*
- Verhinderung der Dienste eines Datenbanksystems G 5.65, *Szenario*
- Unberechtigter Anschluss von IT-Systemen an ein Netz G 5.66, B 4.1, B 4.2
- Vertraulichkeitsverlust schützenswerter Informationen G 5.71, B 1.6, B 1.7, B 1.9, B 4.4
- Überlastung durch eingehende E-Mails G 5.75, *Szenario*

- DNS-Spoofing und Pharming G 5.78, *Szenario*
- Gefälschte Zertifikate G 5.84, B 1.7, B 1.9
- Integritätsverlust schützenswerter Informationen G 5.85, B 1.6, B 1.7
- Manipulation von Managementparametern G 5.86, B 4.2
- Web-Spoofing G 5.87, B 1.9
- Missbrauch aktiver Inhalte G 5.88, *Szenario*
- Hijacking von Netz-Verbindungen G 5.89, *Szenario*
- Nutzung des VPN-Clients als VPN-Server G 5.92, B 4.4
- Erlauben von Fremdnutzung von VPN-Komponenten G 5.93, B 4.4
- Hacking Lotus Notes G 5.101, *Szenario*
- Missbrauch von Webmail G 5.103, *Szenario*
- Ausspähen von Informationen G 5.104, *Szenario*
- Manipulation von ARP-Tabellen G 5.112, *Szenario*
- MAC-Spoofing G 5.113, *Szenario*
- Missbrauch von Spanning Tree G 5.114, *Szenario*
- Überwindung der Grenzen zwischen VLANs G 5.115, *Szenario*
- Angriffe über TCP/IP auf z/OS-Systeme G 5.121, *Szenario*
- Spyware G 5.127, *Szenario*
- Unberechtigter Zugriff auf Daten durch Einbringen von Code in ein SAP System G 5.128, *Szenario*
- Manipulation von Daten über das Speichersystem G 5.129, *Szenario*
- Manipulation der Konfiguration des Speichersystems G 5.130, *Szenario*
- SQL-Injection G 5.131, *Szenario*
- Kompromittierung einer RDP-Benutzersitzung unter Windows Server 2003 G 5.132, *Szenario*
- Unautorisierte Benutzung webbasierter Administrationswerkzeuge G 5.133, *Szenario*
- Verbreitung von Schadprogrammen über mobile Datenträger G 5.142, B 1.6

- Man-in-the-Middle-Angriff G 5.143, *Szenario*
- Kompromittierung von Verzeichnisdiensten durch unbefugten Zugriff G 5.144, *Szenario*
- Manipulation von Daten und Werkzeugen beim Patch- und Änderungsmanagement G 1.145, *Szenario*

Tabelle F.1 gibt eine Zusammenfassung der im Szenario vorkommenden Gefährdungen wieder und gruppiert diese in verschiedene Kategorien, namentlich:

- Angriffe bis zur Schicht 4 des International Organization for Standardization (ISO) / OSI-Referenzmodells, also maßgeblich Angriffe auf der Netzebene, gegen Übertragungsprotokolle, IP-Stacks, etc.
- Angriffe auf höheren Schichten, insbesondere Schicht 7, d.h. der Anwendungsschicht. Entsprechende Angriffe richten sich direkt gegen Applikationen, bspw. den Webbrowser.
- DoS dient der Verhinderung von Diensten. Dies kann bspw. durch massive, parallele Anfragen an einen Webserver erfolgen, so dass der Server ausgelastet ist und sich ggf. aufhängt und gewollte, gutartige Anfragen nicht mehr verarbeitet werden können. Auch das Abstürzen lassen eines Dienstes mittels bspw. einem Exploit, speziell formulierter Anfragen (Logikbomben), etc. ist in diesem Bereich zu sehen.
- Privilegien Missbrauch umfasst alle Gefährdungen, welche durch einen vorsätzlichen Missbrauch von Rechten, die einem Nutzer zugeteilt wurden, entstehen. Hierunter können allerdings auch indirekte Abläufe fallen, wenn der Inhaber der Privilegien bspw. durch Social Engineering Techniken dazu gebracht wird, unbewusst einen Rechtemissbrauch zu begehen.
- Manipulation bezeichnet alle Aktionen, welche nicht autorisierte Änderungen an Hard- oder Software durchführen.
- Datenabfluss beinhaltet sowohl absichtlich herbeigeführten Datenabfluss, bspw. das Kopieren von internen Informationen durch einen Innentäter, als auch unbewusst herbeigeführten Datenverlust, bspw. die Herausgabe im Rahmen eines Social Engineering Angriffs.

Entsprechend den Gefährdungen werden durch die Grundschutzkataloge Massnahmen zum Schutz aufgeführt. Tabelle F.2 zeigt die Zuordnung der jeweiligen Massnahmen zu den Gefährdungskategorien. Hierbei handelt es sich im Einzelnen um:

- Audit und Protokollierung der Aktivitäten im Netz, M 4.81
- Sichere Konfiguration der aktiven Netzkomponenten, M 4.82
- Kommunikation durch Paketfilter auf Minimum beschränken, M 4.98

Tabelle F.1: Gruppierung der Gefährdungen gem. den für das Szenario identifizierten Aspekten laut IT-Grundschutzkatalogen zu übergeordneten Kategorien.

Gefährdung	bis Layer 4	Layer 7	DoS	Privilegien Missbrauch	Manipulation	Datenabfluss	Gefährdung	bis Layer 4	Layer 7	DoS	Privilegien Missbrauch	Manipulation	Datenabfluss
G 3.1						✓	G 5.71						✓
G 3.8				✓	✓	✓	G 5.75			✓			
G 3.13						✓	G 5.78	✓					
G 3.26						✓	G 5.84					✓	
G 3.28						✓	G 5.85						✓
G 3.30						✓	G 5.86					✓	
G 3.44						✓	G 5.87		✓				
G 3.64						✓	G 5.88		✓				
G 4.8		✓	✓				G 5.89	✓					
G 4.10						✓	G 5.92						✓
G 4.22		✓	✓	✓	✓		G 5.93						✓
G 4.31						✓	G 5.101		✓				
G 4.49						✓	G 5.103		✓				
G 5.2					✓	✓	G 5.104						✓
G 5.7						✓	G 5.112	✓					
G 5.9				✓			G 5.113	✓					
G 5.10				✓			G 5.114	✓					
G 5.18		✓					G 5.115	✓					
G 5.19				✓			G 5.121	✓					
G 5.20				✓			G 5.127						✓
G 5.21					✓	✓	G 5.128		✓				
G 5.23					✓	✓	G 5.129					✓	
G 5.24					✓		G 5.130					✓	
G 5.25					✓		G 5.131		✓				
G 5.28			✓				G 5.132	✓					
G 5.50					✓		G 5.133				✓		
G 5.51					✓		G 5.142					✓	
G 5.61					✓		G 5.143	✓					
G 5.65			✓				G 5.144					✓	
G 5.66					✓		G 5.145					✓	

Tabelle F.2: Massnahmen zur Handhabung der Gefährdungen gem. den aufgestellten Kategorien.

Massnahme	bis Layer 4	Layer 7	DoS	Privilegien Missbrauch	Manipulation	Datenabfluss	Gefährdung	bis Layer 4	Layer 7	DoS	Privilegien Missbrauch	Manipulation	Datenabfluss
M 4.81	✓		✓				M 4.345						✓
M 4.82					✓	✓	M 5.8					✓	
M 4.98						✓	M 5.68						✓
M 4.206					✓	✓	M 5.71	✓	✓	✓	✓	✓	✓

- Sicherung von Switch-Ports, M 4.206
- Schutz vor unerwünschten Informationsabflüssen, M 4.345
- Regelmäßiger Sicherheitscheck des Netzes, M 5.8
- Einsatz von Verschlüsselungsverfahren zur Netzkommunikation, M 5.68
- Intrusion Detection und Intrusion Response Systeme, M 5.71

F.1.2 Spezialisierung des eCrime-Marktes

Die Attraktivität, Angriffe mit wirtschaftlichen Interessen im Internet durchzuführen, steigt in den letzten Jahren stark an. Zum einen liegt dort ein hoher Umsatz vor, andererseits lassen sich Angriffe über das Netz aus relativ sicherer Entfernung durchführen. Dies hat den Bedarf an wirksamer Schadsoftware gesteigert, die insbesondere auch von technisch nicht versierten Personen genutzt werden kann. Die Folge ist eine zunehmende Professionalisierung des Untergrundmarktes im Internet, der zahlreiche Spezialisierungen und Ausprägungen hat. Abbildung F.1 zeigt einige der im eCrime-Markt im Bereich Spamming auftretenden Spezialisierungen. Der eigentliche Versender der unerwünschten Nachrichten, der Spammer, bezieht hierbei Aufträge von der Industrie oder einzelnen Verkäufern. Zum Versand der Spam-Mails werden heutzutage insbesondere mit Bots infizierte Heim-PCs eingesetzt, da offene Relays nur noch in den wenigsten Fällen zur Verfügung stehen bzw. leicht identifizierbar und durch die Mailprovider einfacher filterbar sind. Die Bots werden im Rahmen von sog. Botnetzen durch deren Betreiber zur Verfügung gestellt, hierbei sind z.B. Anmietungen eines Netzes auf stündlicher oder täglicher Basis möglich; die Kosten hierfür sind insbesondere von der Größe des Botnetzes abhängig und beginnen bereits bei acht US-Dollar pro Stunde[1]. Für den Aufbau der Botnetze wird wiederum Schadsoftware benötigt, welche auf möglichst vielen infizierbaren Rechnern im Internet installiert wird. Die reine Durchführung dieser Infektionen

[1] Stand Ende 2010 / Anfang 2011.

und Bereitstellung von Bots wird als extra Dienstleistung angeboten. Sehr erfolgreiche Akteure sind hierbei in der Lage, mehrere tausend Rechner pro Woche zu infizieren und können damit bis zu fünfstellige Erträge im Monat erwirtschaften. Die Schadsoftware wird hierbei im Rahmen entsprechender Programme und Kampagnen zur Verfügung gestellt. Oftmals erhält man zu solchen Netzen nur durch persönliche Einladung Zugang, welche auf Basis der Bekanntheit bzw. der Erfolge der jeweiligen Angreifer verschickt werden. Neben dem Versand der Spam-Mails werden die Bots jedoch auch genutzt, Daten über den Besitzer des infizierten PCs zu eruieren, bspw. durch die Integration von Keyloggern und dem Ausspähen von vertraulichen Informationen. Insbesondere Kreditkarteninformationen sind hier ein umfassend gehandeltes Gut und werden im Internet abhängig des Kartentyps und des Herkunftslandes zu unterschiedlichen Preisen gehandelt. Um die gestohlenen Kreditkarteninformationen zu nutzen, werden die Leistungen von eShops und Abwurfdiensten benutzt bzw. missbraucht. Hier erfolgt bspw. der Kauf von Artikeln, die wiederum auf Auktionsseiten, etc. weiterverkauft werden.

F.1.3 Zweidimensionale Angriffsmatrix

Tabelle F.3 zeigt die zweidimensionale Angriffsmatrix nach Perry und Wallich. Hierbei wird die Tätergruppe dem Resultat der unerwünschten Aktion gegenüber gestellt. Aufgrund des Alters der Angriffsmatrix sind nicht mehr zeitgemäße Einträge wie bspw. *Per Modem* in geeigneter Art zu ersetzen, bspw. durch *Per Fernzugriff*. Der Aufbau der Matrix zeigt jedoch weiterhin eine Möglichkeit einer zweidimensionalen Einteilung von Angriffen.

F.1.4 Hilfsprogramme zur Informationssammlung

Nachfolgend werden die mittels einfacher Hilfsprogrammen sammelbaren Informationen beispielhaft dargestellt, mit deren Hilfe die Angriffsvorbereitungen auf eine Zielumgebung getroffen werden können. Die hieraus beschafften Daten können eine umfassende Grundlage für die weiteren Angriffsschritte bilden.

nslookup Eine Datenbankabfrage des Domänennamens mittels `nslookup` liefert die IP-Adressen des zuständigen Nameservers sowie des abgefragten Webservers und deren kanonische Namen.

```
$ nslookup www.unibw.de
Server:    137.193.10.21
Address:   137.193.10.21#53


www.unibw.de   canonical name = websrv.RZ.UniBw-Muenchen.de.
Name: websrv.RZ.UniBw-Muenchen.de
Address: 137.193.6.24
```

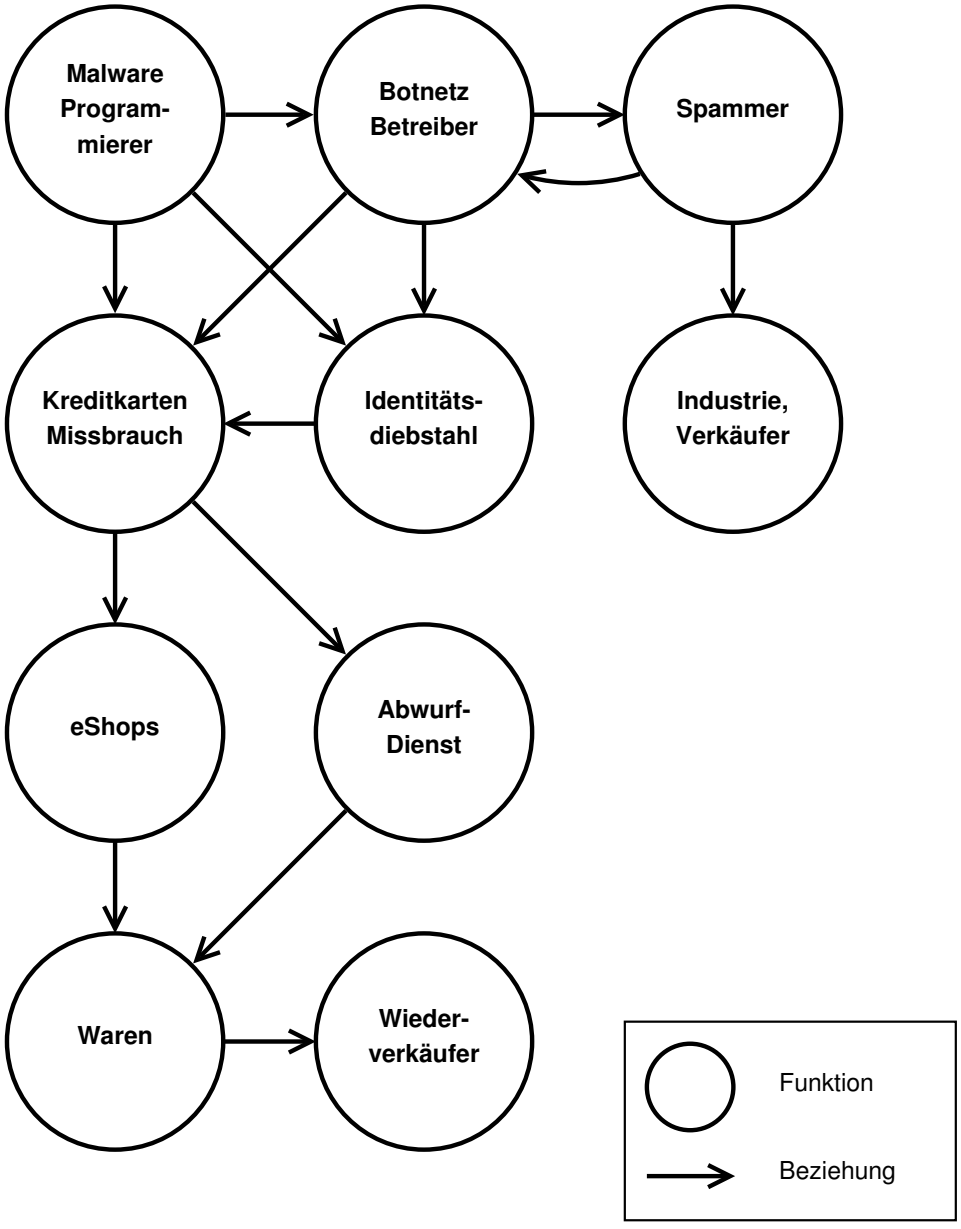


Abbildung F.1: Spezialisierung des eCrime-Marktes am Beispiel Spamversandt und der dabei involvierten Prozesse nach [52].

Tabelle F.3: Zweidimensionale Angriffsmatrix nach Perry und Wallich (vgl. [199]). Die aufgrund des Entstehungszeitpunktes nicht mehr zeitgemäß Eintragung *Per Modem* ist sinnvollerweise durch *Per Fernzugriff* zu ersetzen.

	Adminstratoren	Programmierer	Dateneingebender	Interne Nutzer	Externe Nutzer	Eindringlinge
Physikalische Zerstörung	Kurzschlüsse					
Daten Zerstörung	Löschen von Festspeichern	Schadsoftware			Schadsoftware	Per Modem
Daten Verfälschung		Schadsoftware	Falsche Eingabe			
Diebstahl von Daten				Nichtautorisierter Zugriff	Per Modem	
Diebstahl von Diensten		Diebstahl Nutzer		Nichtautorisierte Aktion	Per Modem	
Browsing	Diebstahl von Medien			Nichtautorisierter Zugriff	Per Modem	

host Informationssammlung mittels des DNS-Hilfsprogrammes `host`. Das Programm dient dazu, IP-Adressen in Namen zu konvertieren und umgekehrt; macht man eine Abfrage anhand des 2nd-Level-Label und der Top-Level-Domain, erhält man bspw. folgende Information zurückgeliefert:

```
$ host unibw.de
unibw.de mail is handled by 11 gold2srv.RZ.UniBw-Muenchen.de.
```

Hiermit ist der Mailserver der Domäne bekannt. Mittels der entsprechenden Schalter lassen sich weitere, detaillierte Informationen abrufen.

```
$ host -l -v -t any unibw.de
Trying "unibw.de"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10274
;; flags: qr aa ra; QUERY: 1, ANSWER: 79, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;unibw.de.      IN  AXFR

;; ANSWER SECTION:
unibw.de.   3600  IN  SOA dns01.rz.unibw-muenchen.de. hostmaster.unibw.
   de. 2011022200 1200 180 1209600 3600
unibw.de.   3600  IN  NS  dns01.rz.unibw-muenchen.de.
unibw.de.   3600  IN  NS  gatesrv.rz.unibw-muenchen.de.
unibw.de.   3600  IN  NS  ws-kar1.win-ip.dfn.de.
unibw.de.   3600  IN  MX  11 gold1srv.RZ.UniBw-Muenchen.de.
unibw.de.   3600  IN  MX  11 gold2srv.RZ.UniBw-Muenchen.de.
unibw.de.   3600  IN  TXT "v=spf1 mx -all"
WWW.agis.unibw.de.  3600  IN  CNAME bartok.BAUV.UniBw-Muenchen.de.
FTP.bauv.unibw.de.  3600  IN  CNAME bauv110.BAUV.UniBw-Muenchen.de.
  [...]
wts.unibw.de.   3600  IN  CNAME wts.RZ.UniBw-Muenchen.de.
www.unibw.de.   3600  IN  CNAME websrv.RZ.UniBw-Muenchen.de.
wwwsrv.unibw.de.  3600  IN  CNAME wwwsrv.RZ.UniBw-Muenchen.de.
zpm.unibw.de.   3600  IN  CNAME zpm-proxy.RZ.UniBw-Muenchen.de.
unibw.de.   3600  IN  SOA dns01.rz.unibw-muenchen.de. hostmaster.unibw.
   de. 2011022200 1200 180 1209600 3600

Received 2141 bytes from 137.193.10.21#53 in 5 ms
```

Hierdurch erhält man bereits die in der Domäne eingetragenen Mail- und Nameserver der Domäne sowie ggf. weiterer, eingetragener Server. Maßgeblich kommen hier die Einträge `MX` (Mail Exchanger, Mailserver für die Domäne), `NS` (Nameserver der Domäne) sowie `Start of Authority (SOA)` mit den Verwaltungseinträgen der Domäne. Ebenfalls sind `Canonical Name (CNAME)`-Einträge weiterer Systeme vorhanden, so dass z.B. Web- und FTP-Server zu finden sind.

whois Mittels des Hilfsprogrammes `whois` können Informationen über den Registrar einer IP-Adresse bzw. eines Domain-Namens abgefragt werden. Hierfür können die Datenbanken der jeweiligen regionalen Registrierungsorganisationen befragt werden (vgl. Abbildung F.2), für Europa ist das Réseaux IP Européens (RIPE) zuständig. Insgesamt

Abbildung F.2: Aufteilung der regionalen Internet Registries [374].

existieren fünf regionale Organisationen, wobei das APNIC nochmals unterteilt werden kann:

- African Network Information Center (AFRINIC)
- Asia Pacific Network Information Center (APNIC) mit
  - Japan Network Information Center (JPNIC)
  - Taiwan Network Information Center (TWNIC)
  - Australian Network Information Center (AUNIC)
- American Registry for Internet Numbers (ARIN)
- Latin and Caribbean Internet Addresses Registry (LACNIC)
- RIPE Network Coordination Centre (RIPE NCC)

Die Abfragen erfolgen gem. der Protokollspezifikation der RFC 3912.

Abbildung F.3 zeigt die Zuständigkeit der Registrierungsorganisationen zu den jeweiligen Adressbereichen. Einige Netze stehen auch komplett unter der Kontrolle von Firmen und anderen Organisationen.

Nachfolgend ist ein Beispiel einer Abfrage gezeigt.

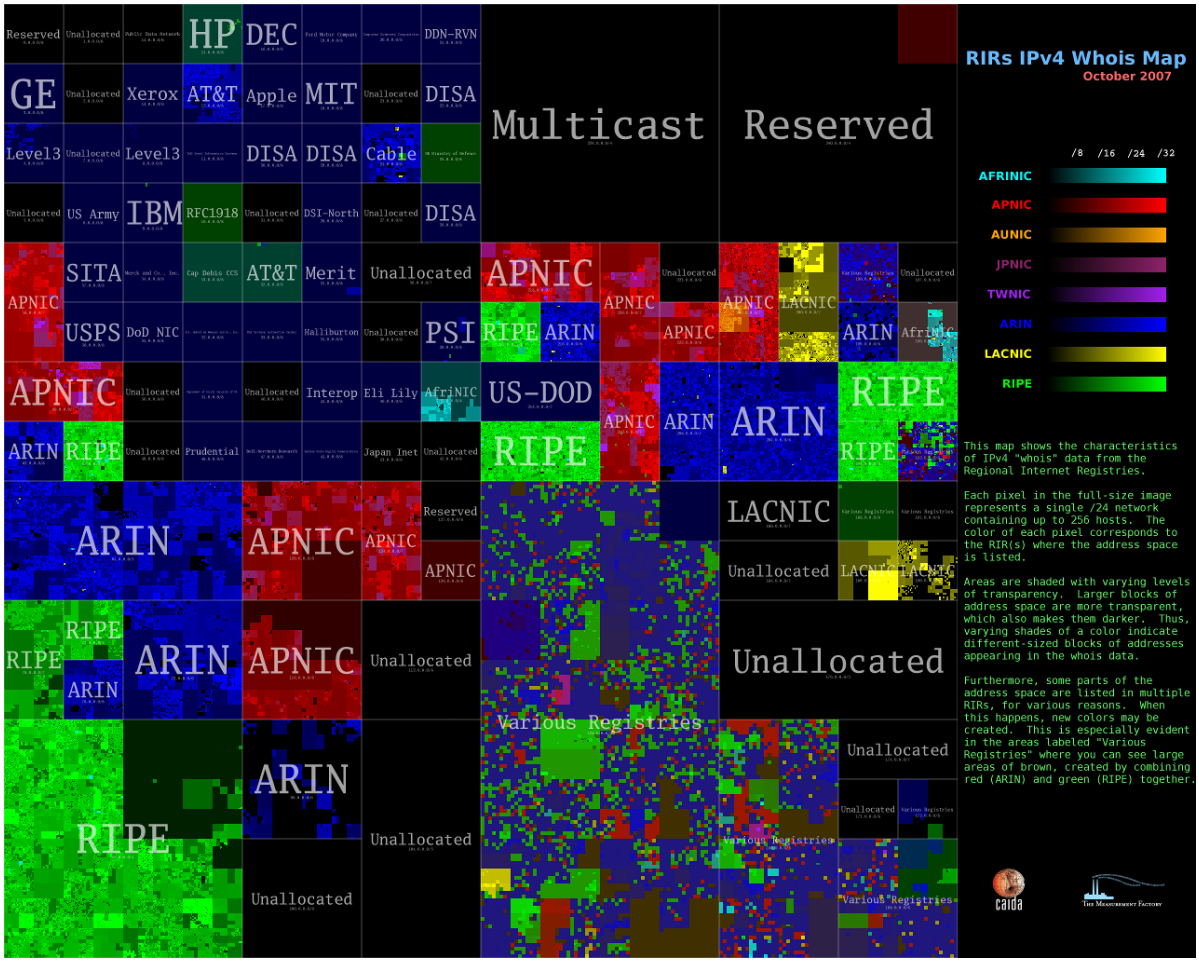


Abbildung F.3: Zuständigkeiten der Regional Internet Registries zu den IPv4-Adressbereichen (Stand 2007) [103].

```
$ whois unibw.de
% Copyright (c) 2010 by DENIC
% Version: 2.0
  [...]

Domain: unibw.de
Nserver: dns01.rz.unibw-muenchen.de
Nserver: gatesrv.rz.unibw-muenchen.de
Nserver: ws-kar1.win-ip.dfn.de
Status: connect
Changed: 2010-12-16T11:59:07+01:00
  [...]

[Zone-C]
Type: PERSON
Name: Stefan Schwarz
Address: Universitaet der Bundeswehr
Address: Rechenzentrum
/*Address: Werner-Heisenberg-Weg 39
PostalCode: 85579
City: Neubiberg
CountryCode: DE
Phone: +49 89 6004 3200
Fax: +49 89 6004 3254
Email: stefan.schwarz@unibw-muenchen.de*/
Changed: 2011-02-01T12:48:05+01:00
```

Sind die Einträge korrekt geführt, können hier Informationen über den Registrator und insbesondere auch ein für die Adressen verantwortlicher Ansprechpartner gefunden werden. Dies kann wiederum als Grundlage für den Start eines Social Engineering Angriffs genutzt werden. Detaillierte Informationen können von den zuständigen, regionalen Registrierungsorganisationen abgerufen werden.

```
$ whois -h whois.arin.net -V -a 137.193.6.24
  [...]

NetRange:       137.193.0.0 - 137.195.255.255
CIDR:           137.194.0.0/15, 137.193.0.0/16
OriginAS:
NetName:        RIPE-ERX-137-193-0-0
NetHandle:      NET-137-193-0-0-1
Parent:         NET-137-0-0-0-0
NetType:        Early Registrations, Transferred to RIPE NCC
Comment:        These addresses have been further assigned to users in
Comment:        the RIPE NCC region.  Contact information can be found
   in
Comment:        the RIPE database at http://www.ripe.net/whois
RegDate:        2004-02-18
Updated:        2004-02-18
Ref:            http://whois.arin.net/rest/net/NET-137-193-0-0-1

OrgName:        RIPE Network Coordination Centre
```

```
OrgId:          RIPE
Address:        P.O. Box 10096
City:           Amsterdam
StateProv:
PostalCode:     1001EB
Country:        NL
RegDate:
Updated:        2011-03-01
Ref:            http://whois.arin.net/rest/org/RIPE

ReferralServer: whois://whois.ripe.net:43

OrgTechHandle: RNO29-ARIN
OrgTechName:   RIPE NCC Operations
OrgTechPhone:  +31 20 535 4444
OrgTechEmail:  do_not_email@ripe.invalid
OrgTechRef:    http://whois.arin.net/rest/poc/RNO29-ARIN
  [...]

% Information related to '137.193.0.0 - 137.193.255.255'

inetnum:       137.193.0.0 - 137.193.255.255
netname:       UNIBWMNET
descr:         Universitaet der Bundeswehr Muenchen; Rechenzentrum
descr:         Werner-Heisenberg-Weg 39, D-85579 Neubiberg
country:       DE
admin-c:       LB4-RIPE
tech-c:        LB4-RIPE
status:        ASSIGNED PI
mnt-by:        DFN-LIR-MNT
mnt-lower:     DFN-LIR-MNT
mnt-routes:    DFN-MNT
mnt-irt:       IRT-DFN-CERT
source:        RIPE # Filtered

irt:             IRT-DFN-CERT
address:         DFN-CERT Services GmbH
address:         Sachsenstrasse 5
address:         20097 Hamburg
address:         Germany
phone:           +49 40 808077 555
fax-no:          +49 40 808077 556
abuse-mailbox:   dfncert@dfn-cert.de
signature:       PGPKEY-80FFBF15
encryption:      PGPKEY-80FFBF15
admin-c:         TI123-RIPE
tech-c:          TI123-RIPE
auth:            PGPKEY-80FFBF15
remarks:         emergency phone number +49 40 808077 555
remarks:         timezone GMT+1 (GMT+2 with DST)
remarks:         https://www.trusted-introducer.org/teams/dfn-cert.html
remarks:         This is a TI accredited CSIRT/CERT
```

```
irt-nfy:        dfncert@dfn-cert.de
mnt-by:         TRUSTED-INTRODUCER-MNT
source:         RIPE # Filtered

person:         Ludwig Bayer
address:        Universitaet der Bundeswehr Muenchen
address:        Rechenzentrum
address:        Werner-Heisenberg-Weg 39
address:        85579 Neubiberg
address:        Germany
phone:          +49 89 6004 3219
fax-no:         +49 89 6004 3254
nic-hdl:        LB4-RIPE
mnt-by:         DFN-NTFY
source:         RIPE # Filtered

% Information related to '137.193.0.0/16AS1275'

route:          137.193.0.0/16
descr:          UNIBWMNET
origin:         AS1275
mnt-by:         DFN-MNT
source:         RIPE # Filtered

% Information related to '137.193.0.0/16AS680'

route:          137.193.0.0/16
descr:          UNIBWMNET
origin:         AS680
mnt-by:         DFN-MNT
source:         RIPE # Filtered
```

traceroute Das Hilfsprogramm `traceroute` dient der Feststellung, über welche Router ein Datenpaket auf dem Weg zu seinem Ziel transportiert wird. Dies kann genutzt werden, um weitere Adressen von Systemen im Zielnetz ausfindig zu machen. In der Praxis ist es häufig jedoch der Fall[2], dass die hierbei involvierten ICMP-Pakete an einer Firewall verworfen werden und somit keine zusätzlichen Informationen über das interne Netz gewonnen werden können. Nachfolgend ist die Ausgabe eines kurzen Trace gezeigt.

```
$ traceroute www.unibw.de
traceroute to www.unibw.de (137.193.6.24), 30 hops max, 60 byte packets
 1  137.193.63.1 (137.193.63.1)  0.640 ms  0.620 ms  0.603 ms
 2  juliett.RZ.UniBw-Muenchen.de (137.193.69.217)  0.586 ms  0.869 ms
    0.857 ms
 3  websrv.RZ.UniBw-Muenchen.de (137.193.6.24)  1.475 ms  1.508 ms
    1.481 ms
```

[2]Bei einem angemessen geschützten Netz.

Delivery Failure Eine weitere Möglichkeit, Informationen über die Zielumgebung zu sammeln, ist bspw. durch das Versenden einer Mail an einen nicht vorhandenen Empfänger gegeben. Hierbei erzeugt der Mailserver, zu dem die Adresse zugehörig sein müsste, typischerweise einen entsprechenden Fehler (die sog. Delivery Status Notification, Non Delivery Notifications bzw. Bounce Message), aus dem Informationen bzgl. eingesetzter Software ersichtlich sind. Aufgrund der vorherrschenden Spam-Problematik erzeugen jedoch zahlreiche Mailserver keine entsprechenden Meldungen mehr, da diese für die Suche nach tatsächlichen Konten genutzt werden können oder bspw. auch die Gefahr eines Blacklistings eröffnen.

```
X-Envelope-Sender:  as seen by kommk3.rz.unibw-muenchen.de
Return-Path: <MAILER-DAEMON@kommk3.rz.unibw-muenchen.de>
Received: from gold2srv.rz.unibw-muenchen.de (gold2srv.rz.unibw-
    muenchen.de [137.193.6.46])
  by kommk3.rz.unibw-muenchen.de (8.14.3/8.14.3/Debian-5+lenny1) with
     ESMTP id p27Fwh4W031321
  (version=TLSv1/SSLv3 cipher=RC4-SHA bits=128 verify=FAIL)
  for <Robert.Koch@unibw.de>; Mon, 7 Mar 2011 16:58:49 +0100
Message-Id: <8efa38$aqsib@gold2srv.rz.unibw-muenchen.de>
Received: from localhost by gold2srv.rz.unibw-muenchen.de;
  07 Mar 2011 16:58:49 +0100
Date: 07 Mar 2011 16:58:49 +0100
To: Robert.Koch@unibw.de
From: "Mail Delivery System" <MAILER-DAEMON@gold2srv.rz.unibw-muenchen.
    de>
Subject: Delivery Status Notification (Failure)
MIME-Version: 1.0
Content-Type: multipart/report; report-type=delivery-status; boundary="
    hN9A.4dUQIJAvL.1IAVxb.66jJpN8"

--hN9A.4dUQIJAvL.1IAVxb.66jJpN8
content-type: text/plain;
    charset="iso-8859-15"
Content-Transfer-Encoding: quoted-printable

The following message to <unknownuser@unibw.de> was undeliverable.
The reason for the problem:
5.1.0 - Unknown address error 550-'5.1.1 <unknownuser@unibw.de>...
    Recipien=
t unknown'

--hN9A.4dUQIJAvL.1IAVxb.66jJpN8
content-type: message/delivery-status

Reporting-MTA: dns; gold2srv.rz.unibw-muenchen.de

Final-Recipient: rfc822;unknownuser@unibw.de
Action: failed
Status: 5.0.0 (permanent failure)
Remote-MTA: dns; [137.193.6.23]
Diagnostic-Code: smtp; 5.1.0 - Unknown address error 550-'5.1.1 <
```

```
    unknownuser@unibw.de>... Recipient unknown' (delivery attempts: 0)

--hN9A.4dUQIJAvL.1IAVxb.66jJpN8
content-type: message/rfc822

X-IronPort-Anti-Spam-Filtered: true
X-IronPort-Anti-Spam-Result: ArEGAJKPdE2JwT+0/2dsb2JhbACYTI58vkSFYgSMMA
X-IronPort-AV: E=McAfee;i="5400,1158,6277"; a="11366986"
X-IronPort-AV: E=Sophos;i="4.62,277,1297033200";
   d="scan'208";a="11366986"
Received: from ibm01.informatik.unibw-muenchen.de (HELO ibm-x3550.
    informatik.unibw-muenchen.de) ([137.193.63.180])
  by gold2srv.rz.unibw-muenchen.de with ESMTP; 07 Mar 2011 16:58:49
     +0100
Received: from [IPv6:::1] (localhost [127.0.0.1])
  by ibm-x3550.informatik.unibw-muenchen.de (Postfix) with ESMTP id 768
     BD1FE81
  for <unknownuser@unibw.de>; Mon,  7 Mar 2011 16:58:48 +0100 (CET)
Message-ID: <4D7500B8.3050500@UniBw.de>
Date: Mon, 07 Mar 2011 16:58:48 +0100
From: Robert Koch <Robert.Koch@UniBw.de>
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.13)
    Gecko/20101208 Thunderbird/3.1.7
MIME-Version: 1.0
To: unknownuser@unibw.de
Subject: Test
X-Enigmail-Version: 1.1.2
Content-Type: text/plain; charset=ISO-8859-15
Content-Transfer-Encoding: 7bit

Test-Mail

--hN9A.4dUQIJAvL.1IAVxb.66jJpN8--
```

Zu erkennen ist sowohl das genutzte System aus der Serie von Cisco IronPort, als auch die eingesetzten Antivirenprogramme von McAfee und Sophos. Diese Informationen können bspw. auch genutzt werden, um Schadsoftware unerkannt einzuschleusen, da natürlich auch Virenscanner regelmäßig fehlerhafte Routinen aufweisen und hierdurch Fehler in der Detektion bzw. Programmausführen haben können, oder bspw. bestimmter Schadecode nicht erkannt wird (vgl. bspw. [259]).

nmap `nmap` ist ein mächtiger, aktiver Scanner, der umfangreiche Optionen zur Identifikation von Rechnern und Diensten in einem Netz bereitstellt [261]. Er unterstützt zahlreiche verschiedene Scan-Technologien, um in Netzen mit IP-Filtern, Firewalls, etc. Auswertungen durchführen zu können. Hierbei können bspw. Erkennungen von Betriebssystemen und Diensten durchgeführt werden. Nachfolgend ist ein Auszug eines Scanlaufs dargestellt:

```
$ nmap -v -PU 137.193.63.0/24

Starting Nmap 5.00 ( http://nmap.org ) at 2011-03-08 11:56 CET
```

```
NSE: Loaded 0 scripts for scanning.
Initiating ARP Ping Scan at 11:56
Scanning 218 hosts [1 port/host]
Completed ARP Ping Scan at 11:56, 3.25s elapsed (218 total hosts)
Initiating Parallel DNS resolution of 218 hosts. at 11:56
Completed Parallel DNS resolution of 218 hosts. at 11:56, 0.01s elapsed
Initiating SYN Stealth Scan at 11:56
Scanning 64 hosts [1000 ports/host]
Discovered open port 1025/tcp on 137.193.63.141
Discovered open port 25/tcp on 137.193.63.99
Discovered open port 111/tcp on 137.193.63.11
Discovered open port 111/tcp on 137.193.63.7
Discovered open port 111/tcp on 137.193.63.121
Discovered open port 111/tcp on 137.193.63.185
        [...]
Discovered open port 88/tcp on 137.193.63.81
Completed SYN Stealth Scan against 137.193.63.39 in 43.44s (62 hosts
   left)
Discovered open port 548/tcp on 137.193.63.43
        [...]
```

Zunächst erfolgt eine Festellung, unter welchen Adressen Hosts online sind, mittels der Durchführung eines `Address Resolution Protocol (ARP) Ping Scans`, anschließend werden die jeweils verfügbaren Dienste bzw. offenen Ports durch einen `SYN Stealth Scan` ermittelt.

Ist aufgrund von zu wenig vorhandener Informationen nur eine Schätzung der laufenden Systeme bzw. Dienste und Versionen möglich, wird eine Einschätzung bzgl. deren Qualität mit angegeben.

```
$ nmap -v -sSV -O www.unibw.de
        [...]
Scanning 2 services on websrv.RZ.UniBw-Muenchen.de (137.193.6.24)
Completed Service scan at 14:21, 12.10s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against websrv.RZ.UniBw-Muenchen.de
   (137.193.6.24)
        [...]
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd 2
443/tcp open  ssl/http Apache httpd 2
Warning: OSScan results may be unreliable because we could not find at
   least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.15 - 2.6.26
Uptime guess: 151.654 days (since Thu Oct  7 23:40:13 2010)
TCP Sequence Prediction: Difficulty=203 (Good luck!)
        [...]
```

F.1.5 Änderung von Kernelparametern

Kernelparameter lassen sich in GNU/Linux einfach während der Laufzeit beeinflussen. Je nach Art und Umfang des konfigurierten und installierten Kernels stehen zahlreiche Parameter zur Verfügung.

Die zur Laufzeit manipulierbaren Parameter eines Kernelmoduls können mittels des Befehls `modinfo` abgefragt werden:

```
$ modinfo -p ${modulename}
```

Die Änderung eines entsprechenden Wertes kann ebenfalls auf einfache Weise mit dem Kommando `cat` erfolgen, hier anhand des Beispiels der initialen TTL-Wertes gezeigt:

```
$ cat /proc/sys/net/ipv4/ip_default_ttl
64
$ echo 68 > /proc/sys/net/ipv4/ip_default_ttl
$ cat /proc/sys/net/ipv4/ip_default_ttl
68
```

Der veränderte Wert wird hierbei unmittelbar übernommen und verwendet. In Windows-Systemen ist eine Änderung des zuständigen Registry-Keys in `HKEY_LOCAL_MACHINE` erforderlich, um den gleichen Effekt zu erzeugen. Eine Überprüfung ist unmittelbar z.B. mittels der Analyse des Netzverkehrs mit dem Tool Wireshark [145] möglich. Fängt man ein beliebiges Paket auf, das den eben veränderten Rechner verlässt, erkennt man den nun höheren, initialen TTL-Wert.

```
Internet Protocol, Src: 137.193.63.218 (137.193.63.218), Dst:
   137.193.61.65 (137.193.61.65)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 60
    Identification: 0x0f43 (3907)
    Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 68
    Protocol: TCP (0x06)
    Header checksum: 0x96db [correct]
    Source: 137.193.63.218 (137.193.63.218)
    Destination: 137.193.61.65 (137.193.61.65)
```

F.1.6 TTL-Werte

Der TTL-Wert verhindert, dass Pakete endlos im Netz kreisen. Hierfür wird ein initialer Wert gesetzt, der gem. RFC 791 von jedem Router, der das Paket weitervermittelt, um die Anzahl der Sekunden der Alterung des Pakets reduziert werden muss, mindestens jedoch um den Wert 1. Tabelle F.4 gibt eine Übersicht von initialen TTL-Werten für TCP- sowie UDP-Pakete verschiedener Betriebssysteme an. Leicht zu erkennen ist, dass die genutzten Werte sehr unterschiedlich gesetzt sind. Diese Informationen können ebenfalls für die Identifikation eines Systems herangezogen werden.

Tabelle F.4: TTL-Werte für verschiedene Betriebssysteme [233], [63], [113].

Betriebssystem	TCP-TTL	UDP-TTL
AIX	60	30
FreeBSD 2.1R	64	65
HP/UX 10.01	64	64
Irix 6.x	60	60
Linux	64	64
MacOS/macTCP 2.0.x	60	60
OS/2 TCP/IP 3.0	64	64
OSF/1 V3.2A	60	30
Solaris 2.x	255	255
SunOS 4.1.3/4.1.4	60	60
MS Windows 2000	128	128
MS Windows XP	128	128
MS Windows Vista	64	64

F.1.7 Reduzierung von ICMP-Nachrichten

Der GNU/Linux Kernel berücksichtigt die in der RFC 1812 gegebene Empfehlung, ICMP-destination unreachable Nachrichten (Typ 3) zu reduzieren. Ein entsprechender Hinweis findet sich in der Datei `net/ipv4/icmp.c` [14] der Kernelquellen:

```
/*
 *      Check transmit rate limitation for given message.
 *      The rate information is held in the destination cache now.
 *      This function is generic and could be used for other purposes
 *      too. It uses a Token bucket filter as suggested by Alexey
    Kuznetsov.
 *
 *      Note that the same dst_entry fields are modified by functions
    in
 *      route.c too, but these work for packet destinations while
    xrlim_allow
 *      works for icmp destinations. This means the rate limiting
    information
 *      for one "ip object" is shared - and these ICMPs are twice
    limited:
 *      by source and by destination.
 *
 *      RFC 1812: 4.3.2.8 SHOULD be able to limit error message rate
 *                        SHOULD allow setting of rate limits
 *
 *      Shared between ICMPv4 and ICMPv6.
 */
```

F.1.8 Paketanalyse

Die unterschiedliche Handhabung von Paketen und Flags, einerseits durch eine ungenaue Spezifikation in den jeweiligen RFCs, aber auch durch fehlerhafte oder bewusst geänderte Implementierungen verschiedener Systeme ermöglicht eine Identifikation des Systems mit entsprechenden Wahrscheinlichkeiten. Nachfolgend werden die Flags eines Netzpaketes dargestellt und deren Verwendung für die Erstellung eines Fingerabdrucks für den Scanner `p0f` gezeigt.

```
Internet Protocol, Src: 137.193.63.218 (137.193.63.218), Dst:
   137.193.68.205 (137.193.68.205)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 60
    Identification: 0xee3e (60990)
    Flags: 0x02 (Don't Fragment)
        0.. = Reserved bit: Not Set
        .1. = Don't fragment: Set
        ..0 = More fragments: Not Set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0xb453 [correct]
    Source: 137.193.63.218 (137.193.63.218)
    Destination: 137.193.68.205 (137.193.68.205)
Transmission Control Protocol, Src Port: 52866 (52866), Dst Port: eyetv
    (2170), Seq: 0, Len: 0
    Source port: 52866 (52866)
    Destination port: eyetv (2170)
    [Stream index: 6]
    Sequence number: 0    (relative sequence number)
    Header length: 40 bytes
    Flags: 0x02 (SYN)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...0 .... = Acknowledgement: Not set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..1. = Syn: Set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0x2a6d [validation disabled]
    Options: (20 bytes)
        Maximum segment size: 1460 bytes
        SACK permitted
        Timestamps: TSval 120639691, TSecr 0
        NOP
        Window scale: 7 (multiply by 128)
```

Aus den vorliegenden Werten kann der zugehörige Fingerabdruck wie folgt erzeugt werden: Die Fenstergröße als Vielfaches der *Maximum Segment Size (MSS)* ($\frac{5840}{1460} \widehat{=} S4$), der initiale *TTL*-Wert (64), das gesetzte *Don't Fragment (DF)*-Bit, die Gesamtgröße des *SYN*-Pakets (60), sowie die Optionswerte in ihrer Reihenfolge (M*, S, T, N, W7).

Durchsucht man die bei `p0f` vorhandenen Fingerprints mittels des erzeugten Wertes, erhält man den Eintrag

```
S4:64:1:60:M*,S,T,N,W7:.:Linux:2.6 (newer, 3)
```

also eine Klassifizierung als GNU/Linux System mit einem Kernel der Serie 2.6; dies entspricht einer korrekten Evaluation des Rechners, der den Verbindungsaufbau initiierte.

F.1.9 Schwachstelleninformationen

Die Kenntnis über vorhandene Schwachstellen ist für Angreifer genau wie für Verteidiger von besonderer Bedeutung. Entsprechend finden sich zahlreiche Informationsquellen, die zur Bekanntgabe von sicherheitsrelevanten Informationen wie Schwachstellen und Sicherheitslücken genutzt werden können, bspw. die CVE-Datenbank [105] der MITRE Corporation [376]. Hier finden sich neben detaillierten Informationen zur jeweiligen Schwachstelle auch die betroffenen Programmversionen, Links zu externen Seiten, etc.

```
CVE-2011-0037

    Summary: Microsoft Malware Protection Engine before 1.1.6603.0, as
       used in Microsoft Malicious Software Removal Tool (MSRT),
       Windows Defender, Security Essentials, Forefront Client
       Security, Forefront Endpoint Protection 2010, and Windows Live
       OneCare, allows local users to gain privileges via a crafted
       value of an unspecified user registry key.
    Published: 02/25/2011
    CVSS Severity: 7.2 (HIGH)
```

Weitere Informationsquellen, wie bspw. Mailinglisten, stellen ebenfalls ausführliche Informationen bzgl. den Schwachstellen von Software zur Verfügung. Nachfolgend ist eine bei SecurityFocus veröffentlichte Schwachstelle vorgestellt:

```
Vulnerability ID: HTB22869
Reference: http://www.htbridge.ch/advisory/
   sql_injection_in_1_flash_gallery_wordpre
ss_plugin.html
Product: 1 Flash Gallery wordpress plugin
Vendor: 1plugin.com ( http://1plugin.com/ )
Vulnerable Version: 0.2.5
Vendor Notification: 22 February 2011
Vulnerability Type: SQL Injection
Risk level: High
Credit: High-Tech Bridge SA - Ethical Hacking & Penetration Testing (
   http://www.htbridge.ch/)
```

```
Vulnerability Details:
The vulnerability exists due to failure in the "/wp-content/plugins/1-
   flash-gallery/massedit_album.php" script to properly sanitize user-
   supplied input in "gall_id" variable.
Attacker can alter queries to the application SQL database, execute
   arbitrary queries to the database, compromise the application,
   access or modify sensitive data, or exploit various vulnerabilities
    in the underlying SQL database.

The following PoC is available:

<form action="http://[host]/wp-content/plugins/1-flash-gallery/
   massedit_album.
php" method="post" name="main" >
<input type="hidden" name="album_id" value="1" />
<input type="hidden" name="images" value="1" />
<input type="hidden" name="gall_id" value="SQL_CODE_HERE" />
<input type="submit" value="submit" name="submit" />
</form>
```

Detektion von Virtualisierungen Für die Durchführung eines Angriffs ist die Kenntnis, ob man sich innerhalb einer Virtualisierung oder in einem direkt auf der Hardware ausgeführten System befindet, von besonderer Bedeutung. Zum einen kann eine Virtualisierung eine Erhöhung der Sicherheit bedeuten, da zunächst nur die angegriffene Maschine selbst kompromittiert werden kann. Allerdings erlauben Fehler in der Virtualisierung, bspw. durch Softwarefehler, Hintertüren der VM oder andere Schwachstellen, einen Ausbruch aus der Virtualisierung. Dies kann insbesondere zur Folge haben, dass sämtliche VM-Instanzen auf dem angegriffenen System kompromittiert werden können. Eine Detektion, ob man sich innerhalb einer Virtualisierung befindet oder nicht, ist prinzipiell immer möglich [229, 365]. Ein besonders einfaches Beispiel ist nachfolgend gezeigt, welches das Vorhandensein einer Backdoor unter VMware ausnutzt, um diese Virtualisierung zu erkennen.

```
uint32 verMajor, verMinor, magic, dout;
__asm__  __volatile__ ("
   mov $0x564D5868, %%eax;
   mov $0x3c6cf712, %%ebx;
   mov $0x0000000A, %%ecx;
   mov $0x5658, %%edx;
   in %%dx, %%eax;
   mov %%eax, %0;mov %%ebx, %1;
   mov %%ecx, %2;mov %%edx, %3;
": "=r"(verMajor), "=r"(magic), "=r"(verMinor), "=r"(dout));
if (magic == 0x564D5868)
        printf("Running inside VMware.");
```

Sind keine derartigen (trivialen) Möglichkeiten vorhanden, kann eine Detektion anhand logischer oder zeitlicher Abweichungen erfolgen. Hierzu kann bspw. die nachfolgen-

de Überprüfung der Register genutzt werden:

```
unsigned long idt, idts, gdt, gdts, ldt, eax, ebx, ecx, edx;
__asm__ volatile ("sidt %0" : "=m" (idtr) : );
idts = *(unsigned short *) &idtr[0];
idt = *(unsigned long *) &idtr[2];-/
printf("idtr = { 0x%lx, 0x%lx }\n", idts, idt);

__asm__ volatile ("sgdt %0" : "=m" (idtr) : );
gdts = *(unsigned short *) &idtr[0];
gdt = *(unsigned long *) &idtr[2];-/
printf("gdtr = { 0x%lx, 0x%lx }\n", gdts, gdt);

ldt = 0;
__asm__ volatile ("sldt %0" : "=m" (ldt) : );
printf("ldt = 0x%lx\n", ldt);

if (idts != 0x7ff)
   printf("sidt returned an unusual result\n");
if ((gdts != 0x7f) && (gdts<=0xff))
   printf("sgdt returned an unusual result\n");
if ((gdts > 0xff) || (ldt>0xff) || ( ((idt>>24) == 0xff) && (((gdt
   >>24) == 0xff))))
   printf("possible VMM detected\n");
```

Rutkowska schlägt eine einfache Detektionsmöglichkeit auf Basis des *Store IDT Register (SIDT)*-Befehls vor [330]:

```
int swallow_redpill() {
   unsigned char m[2+4], rpill[] = "\x0f\x01\x0d\x00\x00\x00\x00\xc3";

   *((unsigned*)&rpill[3]) = (unsigned)m;
   ((void(*)())&rpill)();

   return (m[5]>0xd0)?1:0;
}
```

Das Programmstück macht sich zunutze, dass der *SIDT*-Befehl im nicht-privilegierten Modus ausgeführt werden kann, jedoch einen Registerwert zurückliefert, welcher intern durch das Betriebssystem genutzt wird (Ausführungen nach [330]). Hierbei wird der Inhalt des *Interrupt Description Table Register (IDTR)* zurückgegeben, der einmalig in einem System ist. Dies erfordert eine Dislozierung des Registers an eine andere, sichere Adresse, die nicht mit dem Hostsystem in Konflikt steht. Da durch die virtuellen Maschinen jederzeit ein Zugriff auf den Inhalt aus dem nicht-privilegierten Modus erfolgen kann und somit auch keine Exception erzeugt wird, muss hierbei die *ausgelagerte* Adresse auf die Tabelle zurückgegeben werden. Beispiele hierfür sind `0xffXXXXXX` bei VMWare Workstation 4 bzw. `0xe8XXXXXX` bei Virtual PC 2004, jeweils auf Windows XP als Hostsystem.

Informationen des Dateisystems Bei der Arbeit im Dateisystem werden verschiedene Informationen festgehalten, bspw. der letzte Zugriff auf eine Datei. Dies ist sowohl

im Rahmen der Spurensuche nach einem Angriff, aber auch beim Angriff selbst zur Verwischung möglicher Spuren von Bedeutung. Die genauen Informationen, welche festgehalten werden, hängen stark vom verwendeten Dateisystem ab. Nachfolgend ist ein Auszug der Definition eines Inode des Dateisystems *Extended 4* gezeigt:

```
struct ext4_inode {
  __le16  i_mode;   /* File mode */
  __le16  i_uid;    /* Low 16 bits of Owner Uid */
  __le32  i_size_lo;  /* Size in bytes */
  __le32  i_atime;  /* Access time */
  __le32  i_ctime;  /* Inode Change time */
  __le32  i_mtime;  /* Modification time */
  __le32  i_dtime;  /* Deletion Time */
  __le16  i_gid;    /* Low 16 bits of Group Id */
  __le16  i_links_count;  /* Links count */
  __le32  i_blocks_lo;  /* Blocks count */
  __le32  i_flags;  /* File flags */
  [...]
  __le16  i_extra_isize;
  __le16  i_pad1;
  __le32  i_ctime_extra;  /* extra Change time      (nsec << 2 | epoch)
    */
  __le32  i_mtime_extra;  /* extra Modification time(nsec << 2 | epoch)
     */
  __le32  i_atime_extra;  /* extra Access time      (nsec << 2 | epoch)
     */
  __le32  i_crtime;       /* File Creation time */
  __le32  i_crtime_extra; /* extra FileCreationtime (nsec << 2 | epoch)
     */
  __le32  i_version_hi; /* high 32 bits for 64-bit version */
};
```

F.1.10 netfilter-Firewall

`netfilter` stellt die Firewall von GNU/Linux seit der Kernelversion 2.4 dar. Die zugehörigen Userspace-Programme zur Konfiguration und Verwaltung der Firewall sind unter dem Namen `iptables` verfügbar, der oftmals auch für die Bezeichnung der Firewall selbst genutzt wird. Abbildung F.4 stellt den Paketfluss durch die verschiedenen Instanzen der Firewall graphisch dar (vgl. [386][3]).

[3] Angemerkt sei, dass die Darstellung in der Quelle einen Fehler enthält: Bei den beiden letzten Stufen rechts unten im Bild handelt es sich nicht, wie im Original eingezeichnet, um die Tabellen *mangle PREROUTING* und *nat PREROUTING*, sondern wie in Abbildung F.4 dargestellt, um *mangle POSTROUTING* und *nat POSTROUTING*.

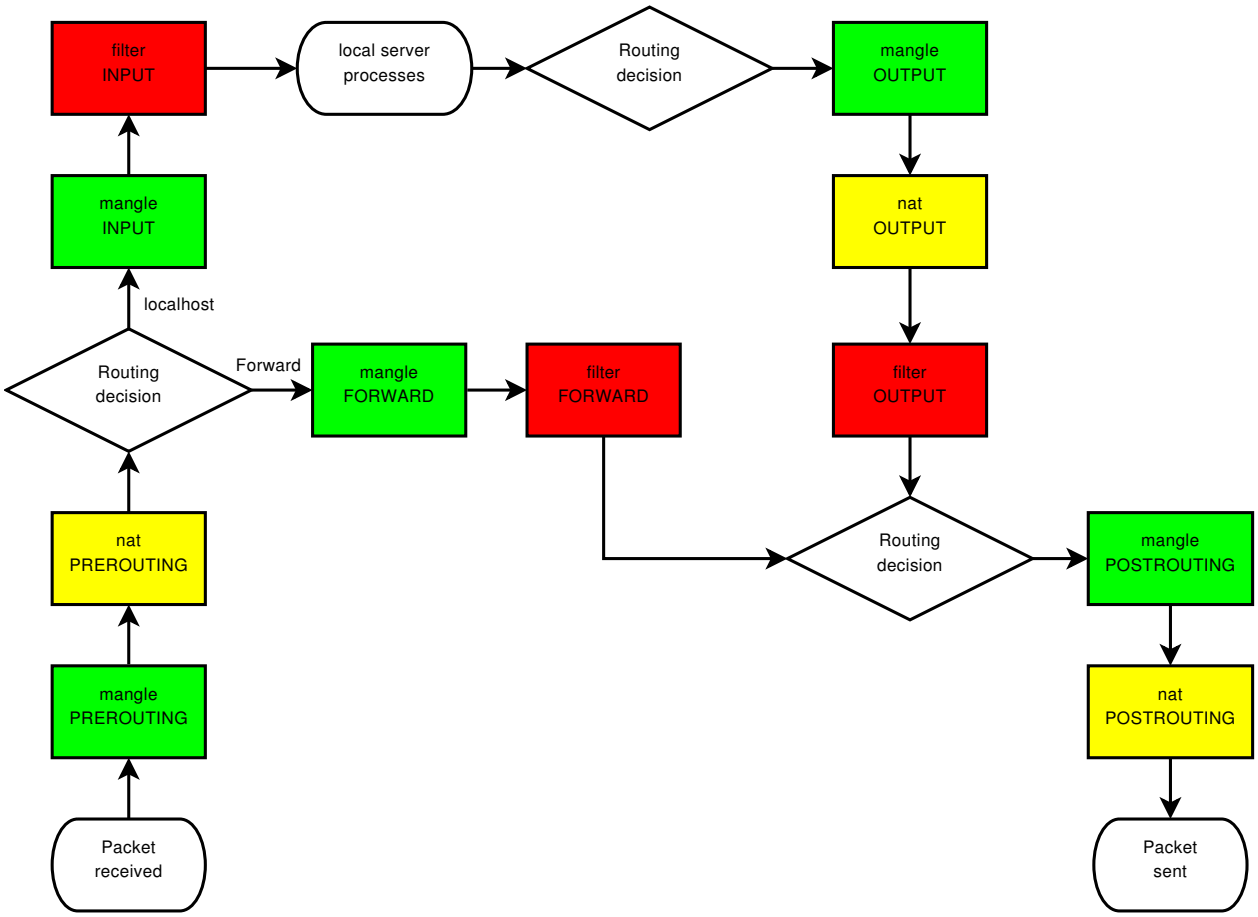


Abbildung F.4: Paketfluß durch die netfilter-Firewall. In der Standardtabelle *filter* lassen sich alle Filterregeln zum Sperren von Paketen ablegen, *nat* dient der Adressumsetzung (Network Address Translation) sowie speziellen Weiterleitungen, die Einträge der *mangle*-Tabelle dienen der Manipulation von Paketen. Jede Tabelle besteht weiterhin aus mehreren *Chains*, die festlegen, wann ein Paket jeweils geprüft werden soll. Hierbei gibt es die Alternativen *INPUT*, *OUTPUT*, *FORWARD*, *PREROUTING* und *POSTROUTING*, wobei nicht jede Tabelle über jede Chain verfügt. *INPUT* und *OUTPUT* betreffen Pakete, welche für den Rechner selbst bestimmt sind bzw. von diesem erzeugt wurden, *PREROUTING* behandelt Pakete vor dem Treffen der Routing-Entscheidung, *POSTROUTING* danach. *FORWARD* betrifft Pakete, die vom Rechner im Sinne eines Gateways weitergegeben werden.

F.2 Ergänzungen zu Kapitel 4

Hier finden sich ergänzende Hinweise und Ausführungen zu den in Kapitel 4 vertieften Themen. Insbesondere werden hier Details zu den verschiedenen Detektionsparadigmen, Informationsquellen sowie deren Manipulationsimöglichkeiten vorgestellt. Frühwarnsysteme und die Entwicklung des IP-Adressraumes finden sich ebenfalls in diesem Kapitel wieder.

F.2.1 Entwicklung der Einbruchserkennung

Der Ursprung des Internets lässt sich bis 1957 zurückverfolgen; nachdem die damalige UdSSR mit *Sputnik* den ersten Satelliten der Geschichte auf eine Umlaufbahn um die Erde schoss, mussten die Vereinigten Staaten reagieren. Dadurch kam es zur Gründung der Advanced Research Projects Agency (ARPA), welche Kommunikationstechnologien und Datenübertragungsverfahren erforschen sollte, um einen technischen Vorsprung gegenüber der UdSSR zu erhalten.

Im Zeitraum von 1960 bis 1962 [4] erarbeitete Paul Baran, ein Ingenieur der RAND Corporation, die 11-teilige Serie „On Distributed Communications“ [50], die im Auftrag der US Air Force durchgeführt wurde und das Ziel hatte, Kommunikationsverfahren zu entwickeln, die auch nach einem atomaren Erstschlag noch funktionsfähig sind. Zentrales Ergebnis der Studie war das Konzept des *Packet Switchings*: Die Daten werden in kleine Pakete, sog. Datagramme, aufgeteilt und einzeln versendet. Elementar dabei ist, dass die Pakete solange gesendet werden, bis sie beim Empfänger angekommen sind.

Das erste experimentelle Netz wurde 1966 auf Basis der ARPA-Studie *Toward a Cooperativ Network of Time-Shared Computers* [268] mit einem TX-2 und einem AN/FSQ-32 System über eine 1200 bits per second (bps)-Leitung aufgebaut. 1968 wurden mehrere weiterführende Studien vergeben, im Jahre 1969 wurde die erste Arbeit zum Design des ARPANET veröffentlicht [325], und die ersten vier Knoten des ARPANET vernetzt. Das Netz wurde schnell ausgebaut, 1973 folgten die ersten internationalen Anschlüsse. Am 25. Dezember 1975 kam es zum ersten Ausfall des Netzes, nachdem durch einen Fehler sämtlicher Verkehr über den Harvard-Rechner geleitet wurde und dieser dadurch abstürzte. Eine Anomalie, die auf Basis einer Fehlfunktion von Hardware initiiert wurde und die Erzeugung einer Serie von fehlerhaften Netzpaketen zur Folge hatte, legte das ARPANET am 27.10.1980 lahm [24].

Im Jahre 1987 überstieg die Anzahl der angeschlossenen Hosts bereits 10000, ein Jahr später infizierte der sog. *Internet Worm* 6000 von mittlerweile 60000 Rechnern. Aufgrund der explosionsartigen Ausdehnung des Netzes, die so während der Design- und Planungsphase nicht erwartet wurde, öffneten sich zahlreiche sicherheitskritische Schwachstellen, wie schon die ersten Viren und Würmer zeigten.

Die Gefährdungen des Netzes und die daraus resultierende Notwendigkeit einer Überwachung wurde erstmalig in der Studie *Computer Security Technology Planning* im Jahre 1972 untersucht [34]. In der Studie wurden Sicherheitsprobleme der US Airforce analy-

[4] Der gesammelte Band erschien 1964.

siert, die maßgeblich durch den steigenden Bedarf der gemeinsamen Nutzung von Systemen mit unterschiedlich klassifizierten Informationen, andererseits durch die zunehmende Vernetzung und komplexeren Netzstrukturen entstehen. 1980 wurden in einer weiteren Studie, *Computer Security Threat Monitoring and Surveillance* [35], Konzepte zur Verbesserung der Sicherheit und zur Systemüberwachung diskutiert. Der Ursprung der Idee einer automatisierten Einbruchserkennung wird James P. Anderson mit dieser Studie zugeschrieben. Die Notwendigkeit einer Einbruchserkennung wird in [368] insbesondere auf folgenden Punkte begründet:

- Die Realisierung eines vollkommen sicheren Systems ist in der Praxis kaum möglich. Auch wenn es entsprechende Projekte und Forschungen gibt, steht der hierbei erforderliche Aufwand den Markterfordernissen hinsichtlich Kosten, Funktionalität, etc. gegenüber. Die meisten Projekte setzen daher wiederum auf bereits vorhandenen Komponenten auf, bspw. einem minimalen und gehärteten GNU/Linux und der Nutzung von Virtualisierungen oder dem Trusted Platform Module (TPM) (vgl. z.B. Trusted Computing Platform (TCP) [405, 132], Qubes OS [331], Ethos OS [353]). Weiterhin geht eine hohe Gefahr nicht durch das Betriebssystem selbst, sondern insbesondere durch die zahlreichen und technisch immer komplexeren Anwendungen aus (vgl. Kapitel 3).
- Selbst wenn ein sicheres System entwickelt wird, ist eine weltweite Einführung hinsichtlich der Zahl und Art der weltweit im Einsatz befindlichen Systeme extrem aufwändig.
- Die Nutzung kryptographischer Verfahren eröffnet systemimmanent neue Probleme, bspw. beim Verlust eines Zugangs-Tokens (vgl. z.B. auch [275, 64]).
- Auch ein sicheres System ist durch einen Missbrauch von Innentätern gefährdet.
- Restriktive Sicherheitsmechanismen verringern die Nutzereffizienz.

F.2.2 Techniken wissensbasierter Systeme

Zur Realisierung wissensbasierter Systeme lassen sich maßgeblich vier Techniken einsetzen:

- Signaturbasierte Analyse: Hierbei werden Muster (Signaturen, Pattern) von bekannten Angriffen in einer Datenbank gespeichert und der Datenstrom des Netzes auf das Vorhandensein entsprechender Pattern hin geprüft. Werden entsprechende Muster erkannt, zum Beispiel die Abfolge bestimmter Pakete, Sequenzen, etc. erfolgt eine Alarmierung.

  Bei diesem Verfahren sind die Notwendigkeit einer aktuellen und konsistenten Datenbasis sowie das Defizit, nur bekannte Pattern erkennen zu können, von besonderem Nachteil. Eine Detektion neuer und noch unbekannter Angriffe ist nicht möglich, manche neueren Systeme sind jedoch in der Lage, neuere Formen alter

Angriffe auf Basis des bereits vorhandenen Wissens zu beschreiben. Die Nutzung solcher heuristischer Verfahren bedingt jedoch wiederum eine Erhöhung der Fehlalarmraten.

Der Aufwand der Durchführung einer Signaturanalyse durch einen netzgestützten Sensor muss ebenfalls insbesondere mit Hinblick auf Speicher- und Rechenleistungsbedarf betrachtet werden. Um eine entsprechende Untersuchung des Datenstroms durchführen zu können, müssen die Datenpakete entsprechend ihrer eigentlichen Reihenfolge betrachtet werden, die durch das Übertragungsverfahren mittels mehrerer Teilpakete nicht notwendigerweise eingehalten wird. Entsprechend muss ein Sicherheitssystem die Daten lange genug und im notwendigen Umfang zwischenspeichern, um mittels einer Rekonstruktion der jeweiligen Verbindungen einen Rückschluss auf einen Angriff zu erlauben, während Angreifer versuchen können, durch entsprechende Manipulation des Datenstromes einer Entdeckung zu entgehen (vgl. z.B. [315]). Insbesondere muss daher die Reihenfolge durch temporäre Speicherung rekonstruiert werden und Paket-Fragmentierung sowie Strom-Segmentierung berücksichtigt werden. Auch der Verschleierung von URLs oder der Hypertext Markup Language (HTML)-Codierung muss bspw. Rechnung getragen werden.

- Expertensysteme: Das Expertensystem enthält ein Regelwerk zur Beschreibung von Angriffen, während die aufgezeichneten und auszuwertenden Ereignisse in entsprechende Fakten mit semantischer Bedeutung umgesetzt werden. Anschließend können diese korreliert werden. Die größten Probleme bei diesem Verfahren sind die Schwierigkeit, das Wissen über Angriffe in geeigneter Weise in ein Expertensystem umzusetzen, weiterhin ist das Vorgehen vergleichsweise langsam und erlaubt typischerweise keine Detektion in Echtzeit. Entsprechende Systeme wurden daher lediglich in einigen Prototypen eingesetzt.

- Petrinetze: Mittels Petrinetzen kann ebenfalls eine einfache Repräsentation von Signaturen erfolgen. Wissenschaftler der Purdue Universität haben ein entsprechendes IDS, basierend auf farbigen Petrinetzen, entwickelt (vgl. [236], [237]). Der Administrator kann hierbei eigene Signaturen unterstützt durch das System eingeben. Ein einfaches Beispiel ist in Abbildung F.5 gezeigt; hier werden fehlerhafte Loginversuche ausgewertet, nach einer Maximalzahl erfolgloser Versuche innerhalb einer festgelegten Zeit wird ein Alarm ausgelöst. Ein Nachteil des Verfahrens ist, dass die Evaluation komplexer Signaturen sehr rechenintensiv werden kann.

- Zustandsübergangsdiagramme: Bei diesem Verfahren werden Angriffe durch Ziele und Transitionen beschrieben und in einem Zustandsübergangsdiagramm repräsentiert. USTAT ist ein Beispiel eines entsprechenden Systems [207], Abbildung F.6 zeigt die Definition einer nicht-autorisierten Referenzierung einer Datei.

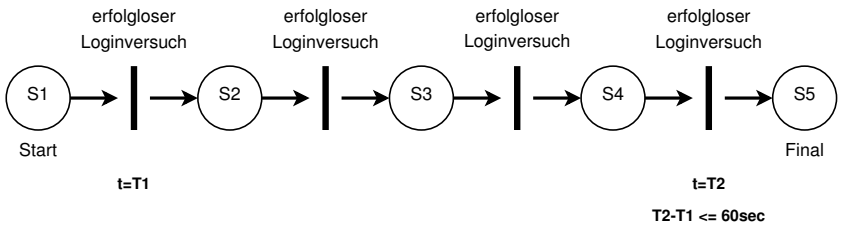


Abbildung F.5: Beispiel eines Petrinetzes zur Detektion fehlerhafter Loginversuche [115].

F.2.3 Fehlalarmraten signaturbasierter Verfahren

Obwohl signaturbasierte Detektionsverfahren grundsätzlich geringere Fehlalarmraten aufweisen, als verhaltensbasierte Systeme, müssen diese detailliert an die Einsatzumgebung angepasst werden. Erfolgt dies nicht, entstehen auch bei signaturbasierter Analyse hohe Zahlen von Fehlalarmen. Eine Testinstallation im Fakultätsnetz ergab bspw. für eine Installation von Snort 2.9.0.4 und dem aktuellen Regelwerk der verfügbaren *VRT Certified Rules* mit 4438 Regeln die in Abbildung F.7 aufgeführten Alarme in einem Beobachtungszeitraum von 9 Tagen. Im Schnitt wurden 22490 Alarme pro Tag erzeugt, was bedeutet, dass ca. alle 3.8 Sekunden ein neuer Alarm erscheint. Diese Datenflut ist von einem Bediener nach kurzer Zeit nicht mehr zu überblicken.

F.2.4 Techniken verhaltensbasierter Systeme

Verhaltensbasierte Systeme benötigen oftmals eine Lernphase, um das Normalverhalten der Umgebung zu analysieren und anschließend mittels eines Modells umzusetzen. Bei den Lernverfahren kommen maßgeblich folgende Techniken zum Einsatz:

- Statistisch: Bei statistischen Verfahren wird das Nutzer- oder Systemverhalten anhand verschiedener Parameter wie bspw. Login- und Logauszeit, Ressourcennutzung oder Datenvolumen eines Dienstes, in Abhängigkeit der Zeit gemessen. Die Zeitspanne kann hier von wenigen Minuten bis zu Monaten betragen, um z.B. tageszeitabhängige Schwankungen oder spezifische Ausprägungen zu Arbeitstagen und Wochenenden, etc. einzubeziehen. Das unterliegende Modell nutzt die

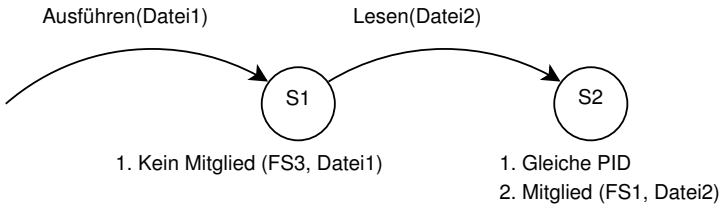


Abbildung F.6: Beispiel eines Zustandsübergangs zur Detektion einer nichtautorisierten Referenzierung einer Datei im IDS nach Ilgun [207].

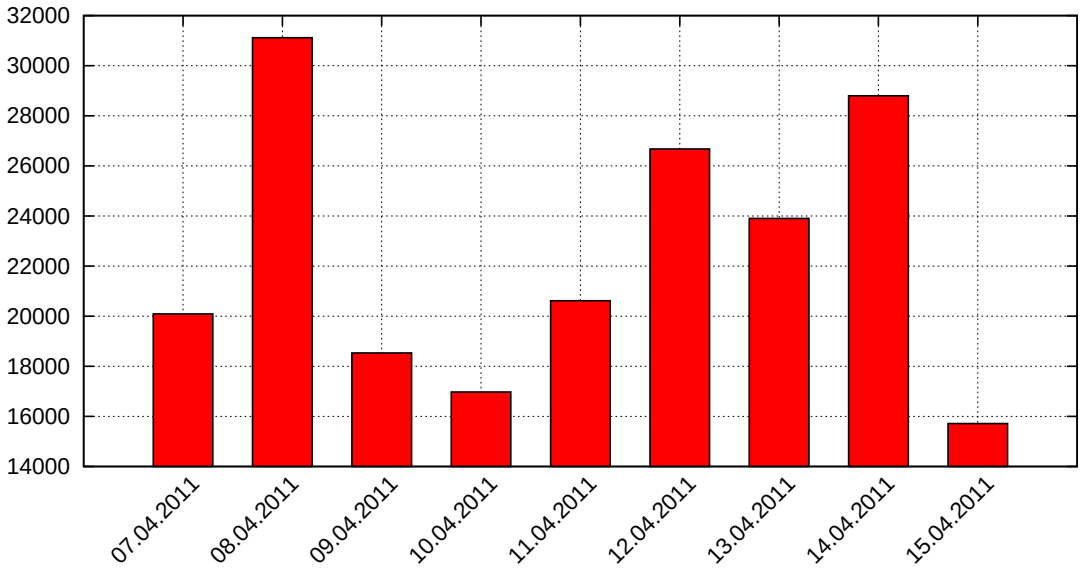


Abbildung F.7: Beobachtung der Anzahl von täglichen Alarmen einer Snort-Standardinstallation.

Durchschnittswerte der entsprechenden Variablen um zu Prüfen, ob Schwellwertüberschreitungen außerhalb der Standardabweichung erfolgen; zur Verbesserung der Ergebnisse müssen hier bspw. genauere Differenzierungen zwischen lang- und kurzzeitigen Nutzeraktionen durchgeführt werden.

- Expertensystem: Auch im Bereich der Verhaltensanalyse werden Expertensysteme eingesetzt. Hierbei werden Regeln des Nutzer- oder Systemverhaltens auf Basis der aufgezeichneten, statistischen Daten erzeugt. Die gegenwärtigen Aktivitäten werden nachfolgend im Betrieb mit den Regeln abgeglichen, bei inkonsistentem Verhalten erfolgt eine Alarmierung.

- Data Mining: Bei Data Mining Verfahren werden Algorithmen und Techniken eingesetzt, um Muster und Regeln in großen Datenmengen erkennen zu können. Spätere Abweichungen zu den identifizierten Mustern können entsprechend detektiert werden; das Hauptproblem hierbei ist, dass eine entsprechende Evaluation regelmäßig nicht in Echtzeit, sondern typischerweise nur *offline* erfolgen kann.

- Ansätze mit (nicht-) beaufsichtigtem Lernen, z.B. Neuronale Netze: Neuronale Netze verfolgen das Ziel, die Strukturen des Gehirns nachzubilden und so ein Lernverhalten zu ermöglichen. Das prinzipielle Vorgehen ist hierbei, dass zunächst die Beziehungen zwischen zwei Datenmengen erlernt werden und anschließend eine Generalisierung durchgeführt wird, um zu neuen Eingabewerten entsprechende Ausgabewerte zu erzeugen. Vorteilhaft bei der Nutzung Neuronaler Netze insbesondere im Vergleich mit statistischen Verfahren ist die einfache Möglichkeit, nicht-lineare Zusammenhänge zwischen Variablen darzustellen und die Möglichkeit, die Netze

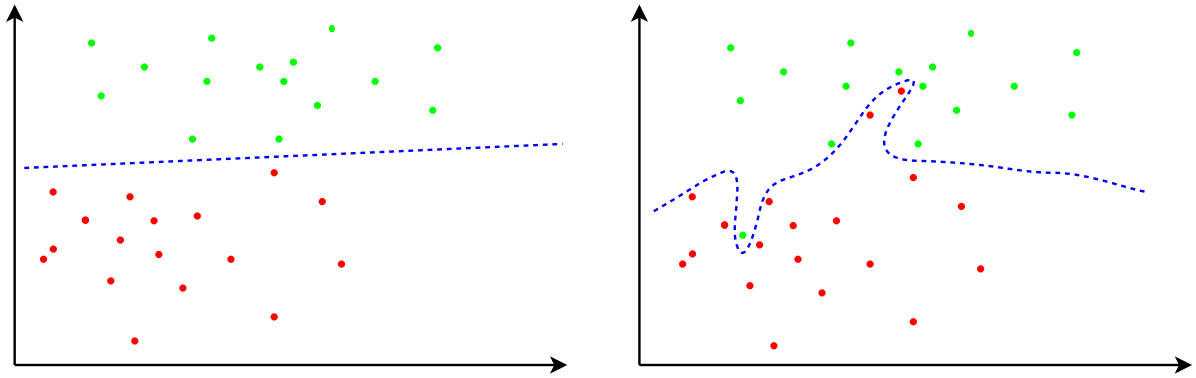


Abbildung F.8: Linear separierbar und linear nicht separierbarer Raum.

automatisch zu trainieren. Wird nur ein einzelnes Neuron dieser Bauart verwendet, können damit lediglich linear separierbare Probleme gelöst werden (vgl. Abbildung F.8). Beispielsweise ist bereits die Umsetzung der Funktion `XOR` auf diese Weise nicht mehr möglich. Dieses Problem kann jedoch durch die Einführung mehrerer Schichten von Neuronen und deren Verknüpfung untereinander gelöst werden (vgl. Abbildung F.9).

Von Nachteil erweist sich, dass das Netz keine Ursache bzw. Begründung für eine Entscheidung liefern kann, d.h. bei einem evaluierten Angriff kann keine Aussage getroffen werden, *warum* genau dieser ausgelöst wurde.

Der Lernvorgang innerhalb eines Neuronalen Netzes erfolgt durch Selbstmodifikation und kann anhand des Lernparadigmas klassifiziert werden. Die prinzipiellen Typen der Lernparadigmen sind:

- Überwachtes Lernen: Die Modifikation des Netzes erfolgt durch die externe Bereitstellung der zur Eingabe gehörigen korrekten Ausgabe bzw. der Differenz zwischen tatsächlicher und richtiger Ausgabe, die zur Modifikation des Netzes genutzt wird.
- Bestärkendes Lernen: Es wird lediglich mitgeteilt, ob die Ausgabe korrekt oder falsch war.
- Unbeaufsichtigtes/nicht-überwachtes Lernen: Es erfolgt keine Beeinflussung von Außen, das Netz versucht selbstständig die Daten in Ähnlichkeitsklassen aufzuteilen. Beispiele hierfür sind die Verfahren *k-means* [190], bei dem aus einer Menge von Objekten eine geforderte Anzahl von k Gruppen gebildet wird, oder SOM, einem Verfahren nach Kohonen zur Reduzierung multidimensionaler Daten (siehe z.B. [230]).

- Identifikation der Nutzerabsicht: Ziel dieser Technik ist es, das normale Nutzerverhalten anhand von Aufgaben, die sie auf dem System ausführen müssen, zu modellieren. Diese werden widerum in einzelne Aktionen verfeinert, die in Bezug

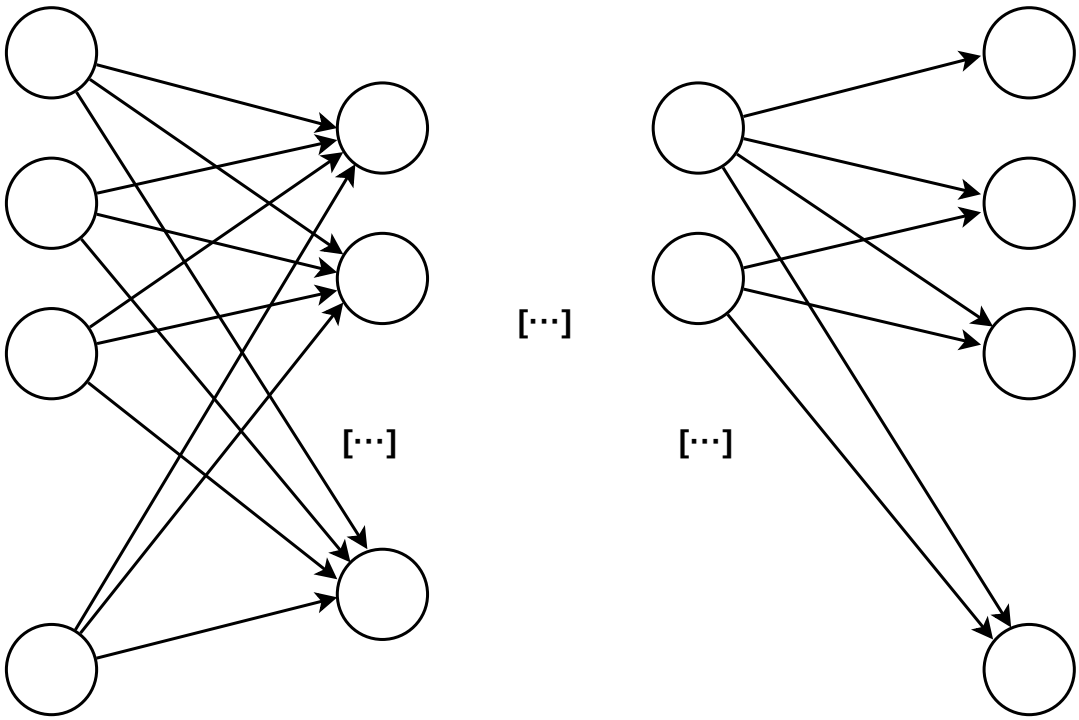


Abbildung F.9: Beispielhafter Aufbau eines Neuronalen Netzes

zu den beobachtbaren Prüfdaten gestellt werden können. Die auswertende Einheit erhält eine Liste von Aktionen, die jeder Nutzer durchführen darf; kann eine Aktion nicht in das Nutzerprofil abgebildet werden, wird ein Alarm ausgelöst.

- Computer Immunologie: Im Gegensatz zum vorherigen Verfahren zur Detektion des Nutzerverhaltens, wird hier ein Modell des Normalverhaltens der (UNIX-)[5] Netz-Dienste erstellt. Hierbei enthält das Modell kurze Sequenzen von *System Calls* der jeweils beteiligten Prozesse und eine Detektion von Angriffen erfolgt aufgrund der Feststellung, dass die Nutzung von Exploits typischerweise in Ausführungswegen resultiert, die nicht dem normalen Verhalten entsprechen [135]. Zur Erstellung der notwendigen Modellreferenzen müssen zunächst korrekte Systemaufrufe analysiert werden, von denen die zugehörigen Sequenzen extrahiert werden. Die Methode kann gute Detektionsraten mit einer niederigen Fehlalarmrate liefern, wenn das zugrunde liegende Modell hinreichend genau ist, jedoch können bspw. Angriffe, die aufgrund einer fehlerhaften Konfiguration möglich sind, nicht erkannt werden.

[5] Die originäre Forschungsarbeit basiert auf einem UNIX-System, eine äquivalente Umsetzung bzgl. anderer Systeme ist entsprechend möglich.

F.2.5 Informationsquellen hostbasierter Systeme

Hostbasierte Detektoren können direkt in das zu schützende System integriert werden und haben somit maßgeblich Zugang zu wichtigen Systeminformationen. Nachfolgende Informationsquellen können bei hostbasierten Systemen herangezogen werden:

- Systeminformationen: Betriebssysteme bieten eine Reihe von Kommandos an, mit deren Hilfe Momentaufnahmen des Systems erstellt werden können. Unter GNU/-Linux existieren z.B. Hilfsprogramme wie `ps` und `top`, um Informationen über Prozesse zu erhalten, in den Verzeichnissen von `/proc` sind zu allen Bereichen des Systems Informationen abrufbar, bspw. Interrupts, Ports, geladene Module und detaillierte Informationen zu jedem laufenden Prozess. Eine beispielhafte Auswertung der verfügbaren Informationen über eine laufende Bash-Konsole sind in Kapitel F.2.7 dargestellt. Hieraus ist erkenntlich, dass auf Host-Ebene umfangreiche Informationen zur Verfügung stehen; die strukturierte Erfassung und Auswertung kann jedoch im Rahmen eines IDS sehr aufwändig sein.

- Accounting: Zur Analyse des Systemverhaltens ist Accounting eine weit verbreitete Technik. Hierbei wird die Nutzung und Auslastung von geteilten Ressourcen durch Nutzer bzw. deren Prozesse überwacht; hierzu zählen Rechenzeit, Speichernutzung, Zugriffe auf das Netz, etc. Eine Evaluation der Ressourcennutzung eines Systems nach der benötigten Systemzeit sowie der Abruf der durch einen Nutzer eingegebenen Befehle ist aus Kapitel F.2.8 ersichtlich.

- Syslog: Eine weitere Informationsquelle ist der Syslog-Dienst. An diesen werden von verschiedenen Diensten und Applikationen Meldungen und Warnungen gesendet, die mit Zeitstempel zentral verfügbar sind, bspw. in der Datei `/var/log/messages` oder durch Auslesen des Kernel-Ringbuffers durch den Befehl `dmesg`. Unter Windows-Systemen sind entsprechende Informationen in der Ereignisanzeige zu finden.

F.2.6 Informationsquellen netzbasierter Systeme

Aufgrund der entsprechenden Positionierung im Netz selbst, verfügen netzbasierte Systeme nicht über Zugriffsmöglichkeiten auf Prozess- oder Nutzerevaluationen. Andererseits lassen sich durch den Zugriff auf den gesamten Netzverkehr bspw. verteilte Angriffe erkennen. Die wichtigsten hierfür nutzbaren Informationsquellen sind:

- Netz-Pakete: Hauptinformationsquelle sind die Daten selbst, die in Form von Paketen und mit zusätzlichen Informationen der involvierten Protokolle versehen über das Verbindungsnetz übertragen werden. Für ein IDS sind hierbei insbesondere die Schichten 3 und 4 des OSI-Referenzmodells [380], die Vermittlungs- und die Transportschicht, entscheidend. Die höchste Bedeutung hat hier IP, andere Protokolle dieser Schicht sind bspw. Internetwork Packet eXchange (IPX) oder X.25. Abhängig des genutzten Protokolls befinden sich zahlreiche Informationen in den zugehörigen Headern, die für Angriffe missbraucht (vgl. Kapitel 2.3) und andererseits durch IDS ausgewertet werden können. Abbildung 2.14 auf Seite 52 zeigt

den Header-Aufbau des derzeit dominierenden IP-Protokolls der Version 4. Auch wenn das Nachfolgeprotokoll IPv6 bereits seit 1998 ausgearbeitet ist und mittlerweile eine breite Unterstützung durch Betriebssysteme und Hardware erfährt, ist die Nutzung bisher äußerst gering (vgl. jährliche Verteilung des IP-Datenverkehrs, Kapitel F.2.17). Dadurch, dass die letzten Adressblöcke von IPv4 im Februar 2011 an die regionalen Registraturen (vgl. Anhang F.1.4) vergeben wurden [192], ist aufgrund des schnellen Wachstums des Netzes und insbesondere der Zahlen von Mobilgeräten und neuen Diensten und Geräten davon auszugehen, dass dies die Einführung von IPv6 künftig beschleunigen sollte.

Durch ein IDS kann nun zum einen lediglich der Header-Bereich ausgewertet werden; dies wird auch als *Shallow Packet Inspection*[6] bezeichnet. Hierdurch können Angriffsversuche auf den Schichten 3 und 4 erkannt werden. Um auch Angriffe auf höheren Schichten zu erkennen, also bspw. den Einsatz eines Exploits gegen einen Dienst oder eine Applikation, muss eine Analyse der Nutzdaten des Pakets erfolgen. Diese Art der Evaluation wird als DPI bezeichnet.

- SNMP-Daten: In der Management Information Base (MIB) (vgl. RFC 3418 [180]) des SNMP (vgl. u.a. RFC 3410) werden die Objekte gespeichert, die zur Verwaltung von Netzgeräten wie Switches, Drucker, Server oder Router ausgelesen werden können. Hierbei sind sowohl Konfigurationsdaten, als auch Accounting- und Daten zur Leistungsmessung enthalten, die als Ressource herangezogen werden können, um Aussagen über das Netz bzw. die einzelnen Komponenten zu liefern. Während die früheren Varianten von SNMP (v1, v2, v2c) unsicher waren und daher nur in einzelnen Projekten genutzt wurden (vgl. z.B. [222]), bietet die aktuelle Version, SNMPv3 (siehe RFC 2576 [178]) Möglichkeiten zur Verschlüsselung, Authentifizierung und Zugangskontrolle und wird somit als Datenquelle für IDS wieder attraktiver (vgl. z.B. [406]).

- Flow-Daten: Flow-Daten bieten eine weitere Möglichkeit, ein Netz bzw. den darin ablaufenden Verkehr zu analysieren. Für den Begriff gibt es mehrere Definitionen, bspw. gem. der RFC 3954:

  Definition (IP Flow / Flow)**.** *An IP Flow, also called a Flow, is defined as a set of IP packets passing an Observation Point in the network during a certain time interval. All packets that belong to a particular Flow have a set of common properties derived from the data contained in the packet and from the packet treatment at the Observation Point [181].*

  Die gemeinsamen Eigenschaften, welche die Pakete dabei haben können, sind in der durch Internet Protocol Flow Information Export (IPFIX) gegebene Definition genauer spezifiziert und beinhalten die nachfolgenden Eigenschaften:

[6] Die Abkürzung SPI ist hier jedoch weniger gebräuchlich. Diese bezieht sich üblicherweise auf *Stateful Packet Inspection* im Rahmen einer Firewall.

- Eines oder mehrere Felder des Paket-Headers, z.B. die Ziel-IP Adresse, des Transport-Headers, bspw. die Port-Nummer oder des Applikations-Headers, z.B. ein Realtime Transport Protocol (RTP)-Feld.
- Eine oder mehrere Eigenschaften des Pakets selbst, z.B. die Anzahl von MPLS-Labeln.
- Eines oder mehrere Felder der Paketbehandlung, z.B. die IP Adresse des nächsten HOP.

Eine verbreitete Bezeichnung eines Flows besteht oft aus `Source IP`, `Destination IP`, `Source Port`, `Destination Port` sowie `Protokoll` (vgl. z.B. [359]).

Die wichtigsten Standards sind *NetFlow* von Cisco und *sFlow*, das maßgeblich auf den Arbeiten von Hewlett Packard und der Universität von Genf beruht [307]. Die erste Version von NetFlow wurde 1996 von Cisco entwickelt, die entsprechenden Funktionalitäten gingen in Ciscos Internetwork Operating System (IOS) ein. Seither wurde die ursprüngliche Variante mehrmals erweitert, aktuell ist Version 9 aus dem Jahre 2004 (RFC 3954 [181]), jedoch ist Version 5 noch am weitesten verbreitet (vgl. z.B. [243]). NetFlow Version 9 ist weiterhin die Basis für IPFIX (RFC 3917 [182], RFC 5101 [183]), das von der Internet Engineering Task Force (IETF) als offener Standard entwickelt wird. Die erste Version von sFlow wurde 2001 verabschiedet (RFC 3176 [179]), seit 2002 ist Hardware mit entsprechender Unterstützung verfügbar, z.B. Switche von Hewlett Packard, D-Link und Allied Telesyn.

Ein wichtiger Unterschied zwischen NetFlow und sFlow liegt darin, dass NetFlow als Software im IOS implementiert ist und jedes Paket, das durch den Router läuft, entsprechend evaluiert wird. sFlow ist in Form von dedizierten Chips hardwareseitig implementiert und *sampled* die Pakete lediglich, d.h. dass bspw. bei einer Sampling-Rate von 100 nur jedes 100ste Paket analysiert wird. Entsprechend müssen hier Algorithmen genutzt werden, um eine korrekte, statistische Repräsentation des Datenverkehrs zu erhalten.

F.2.7 Prozessinformationen

Mittels einfacher Zugriffe und weniger Befehle können umfangreiche Informationen zu laufenden Prozessen abgerufen werden. Nachfolgend sind die Daten, welche mittels des Hilfsprogrammes `ps` sowie einem Zugriff auf das `proc`-Dateisystem erlangt werden können, aufgeführt.

```
$ ps auxfg | grep bash
roko      10854  0.0  0.0  19468  1492 pts/42   Ss   Mar13   0:00  \_ /
    bin/bash
$ cat /proc/10854/
attr/             cwd/              limits            net/
auxv              environ           loginuid          numa_maps
cgroup            exe               maps              oom_adj
clear_refs        fd/               mem               oom_score
```

```
cmdline           fdinfo/           mountinfo         pagemap
coredump_filter   io                mounts            personality
cpuset            latency           mountstats        root/
$ cat /proc/10854/status
Name: bash
State:  S (sleeping)
Tgid: 10854
Pid:  10854
PPid: 25838
TracerPid:  0
Uid:  1000  1000  1000  1000
Gid:  1000  1000  1000  1000
FDSize: 256
Groups: 4 20 24 46 105 119 122 124 1000
VmPeak:    19532 kB
VmSize:    19468 kB
VmLck:         0 kB
VmHWM:      2264 kB
VmRSS:      1492 kB
VmData:      488 kB
VmStk:        88 kB
VmExe:       876 kB
VmLib:      2108 kB
VmPTE:        56 kB
Threads:  1
SigQ: 0/16382
SigPnd: 0000000000000000
ShdPnd: 0000000000000000
SigBlk: 0000000000010000
SigIgn: 0000000000384004
SigCgt: 000000004b813efb
CapInh: 0000000000000000
CapPrm: 0000000000000000
CapEff: 0000000000000000
CapBnd: ffffffffffffffff
Cpus_allowed: f
Cpus_allowed_list:  0-3
Mems_allowed: 00000000,00000001
Mems_allowed_list:  0
voluntary_ctxt_switches:  122
nonvoluntary_ctxt_switches: 3
```

F.2.8 Auswertung von Accounting-Daten

Nachfolgend ist ein Zusammenfassung der Ressourcennutzung nach Zeit gelistet, weitere Darstellungsformen, bspw. nach Prozessen, sind möglich:

```
$ sa -cjpm
              1157 100.00%  0.91re 100.00% 0.00cp 100.00% 402889min 172
                 maj 0swp
root          1052  90.92%  0.89re  88.79% 0.00cp  66.67% 354767min 101
   maj 0swp
```

```
roko            92    7.95%   1.25re   10.94% 0.01cp   31.40%   42174min   59
    maj 0swp
sshd             9    0.78%   0.12re    0.10% 0.00cp    0.97%    2932min    0
    maj 0swp
Debian-exim      4    0.35%   0.44re    0.17% 0.01cp    0.97%    3016min   12
    maj 0swp
```

Mittels des Befehls `lastcomm` können weiterhin alle Befehle, die ein Nutzer ausgeführt hat, ausgegeben werden:

```
$ lastcomm roko
lastcomm           roko     stderr      0.02 secs Wed Mar 16 08:52
man                roko     stderr      0.03 secs Wed Mar 16 08:52
pager              roko     stderr      0.00 secs Wed Mar 16 08:52
nroff              roko     stderr      0.00 secs Wed Mar 16 08:52
groff              roko     stderr      0.00 secs Wed Mar 16 08:52
grotty             roko     stderr      0.01 secs Wed Mar 16 08:52
  [...]
```

Hierbei wird die Zeit, zu welcher der jeweilige Prozess beendet wurde, angegeben. Da die Genauigkeit hierbei nur auf Sekundenbasis arbeitet, kann dies zu Problemen beim Nachvollziehen eines genauen Befehlsablaufes führen.

F.2.9 Beeinflussung von Netzverhalten*

Verhaltensbasierte Systeme benötigen für die Bewertung des Netzzustandes oftmals entsprechende Messdaten des Netzes, welche gegen die Erwartungen des aufgebauten Modells geprüft werden. Hierfür werden oftmals Flow-Daten eingesetzt, die eine effiziente Möglichkeit der Netzüberwachung und -analyse darstellen. Insbesondere mit den durch steigende Datenraten entstehenden Problemen im Bereich der Einbruchserkennung wird der Einsatz von Flow-Daten zur Analyse immer wichtiger. Hierbei wird jedoch typischerweise kein getrenntes Netz zur Datenübermittlung verwendet, so dass die generierten Flow-Daten einer möglichen Manipulation ausgesetzt sind, wenn sie nicht ausreichend abgesichert und geprüft werden. Auch wenn ein isolierter Übertragungsweg vorhanden ist, kann dieser durch die Gefahr von Insidern ebenfalls der Gefährdung einer Manipulation ausgesetzt sein. Um die daraus resultierende Verwundbarkeit verhaltensbasierter Analyseverfahren zu untersuchen und zu bewerten, wurde die Testumgebung gem. Abbildung F.10 aufgesetzt.

Für den Versuchsaufbau wurden ein Cisco 7200 Router zur Erzeugung von NetFlow-Daten, sowie eine Hewlett Packard ProCurve 5304xl Switch zur Erzeugung der benötigten sFlow-Daten eingesetzt. Zur Auswertung der NetFlow-Daten wurde das Produkt *Scrutinizer v6* der Firma plixer International eingesetzt [310]. Mehrere Angriffe zur Manipulation der Flow-Daten wurden durchgeführt, angefangen mit der Injektion von Flow-Paketen auf Basis der bereits vorhandenen, durch den Router erzeugten Pakete.

*Dieser Abschnitt ist eine Zusammenfassung des Artikels „Changing Network Behavior“, Proceedings of the 3rd IEEE International Conference on Network and System Security (NSS 2009), Seiten 60–66, IEEE Computer Society, 2009.

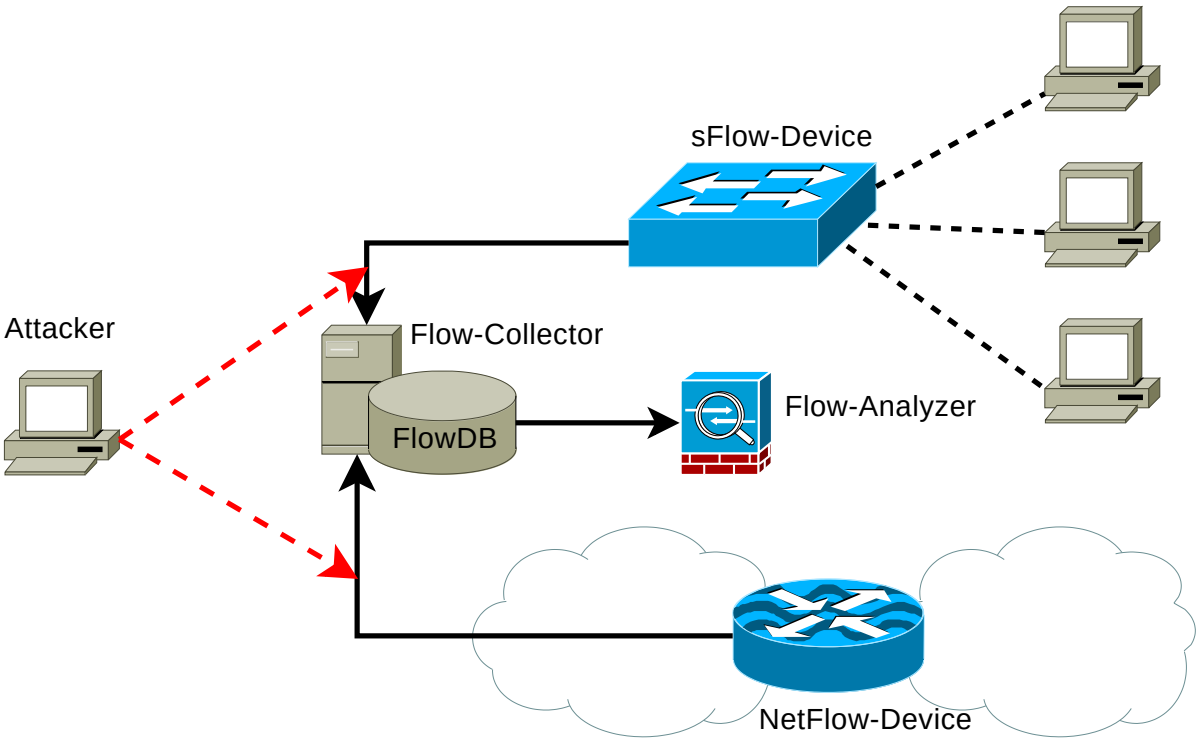


Abbildung F.10: Analyse-Umgebung für Flow-Angriffe. Eine zentrale Switch des Firmennetzes erzeugt sFlow-Daten, der Borderrouter NetFlow-Pakete. Die Pakete werden zu einer zentralen Datenbank gesendet und anschließend von einem Analysetool ausgewertet.

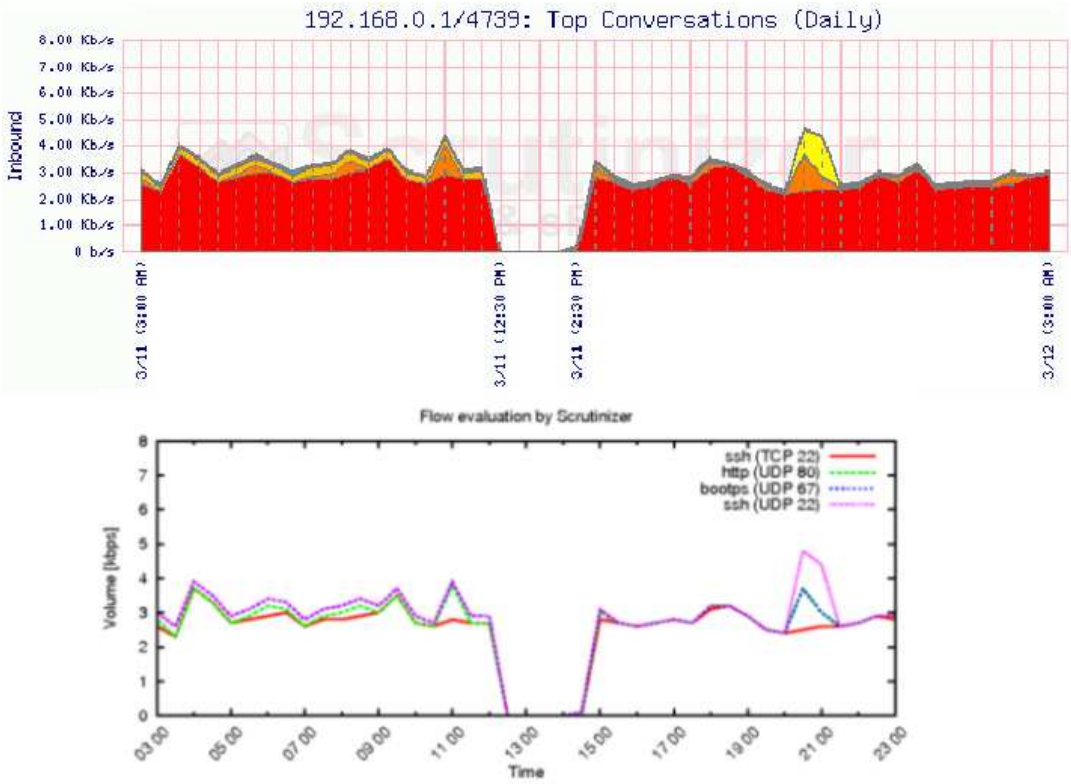


Abbildung F.11: Vergleich der Originalausgabe von Scrutinizer und der entsprechenden Darstellung mittels gnuplot. Durch die Skalierung des Graphen auf die Originalgröße und die Positionierung anhand dieser, ist der Graph leicht unscharf. Diese Phänomen tritt bei den nachfolgenden Graphen entsprechend nicht mehr auf.

Diese wurden auf dem Weg vom Router zum Analyse-System abgehört und zur Konstruktion des Payloads manipulierter Flow-Pakete herangezogen. Hierbei wurden die MAC-Adresse der Flow-Quelle, die Sequenz-Nummer der Pakete, die Zeiten, sowie die IP-Adressen genutzt und mittels des Paket-Injektors *nemesis* [290] versendet. Nach der Injektion zusätzlicher Pakete wurden zwei weitere Angriffe durchgeführt, das Verwerfen von Flow-Paketen sowie das Manipulieren und Weitersenden der originären Pakete; diese Angriffe wurden mittels einer transparenten Brücke, die in das Netz eingefügt wurde, durchgeführt.

Abbildung F.11 zeigt einen Vergleich der Originalausgabe von Scrutinizer und der in *gnuplot* [4] erzeugten Graphen, die aufgrund der in diesem Rahmen ungenügenden Qualität der Ausgabe des Analyse-Tools erzeugt wurden.

Das Ergebnis der Manipulationen ist in Abbildung F.12 dargestellt. Die Injektion von NetFlow-Paketen zur Vortäuschung von Datenverkehr ergibt sich in der Analyse durch *Scrutinizer* gem. Abbildung F.12a. Zunächst wurde ein geringes Datenvolumen durch entsprechende Flow-Pakete simuliert, um 19:00 Uhr wurde das simulierte Datenvolumen in den Flow-Paketen erhöht. In einem weiteren Angriff wurde das Volumen des

Tabelle F.5: Angriffsmöglichkeiten gegen NetFlow.

Art	TCP	UDP	Verschlüsselt	Auswirkung
Paket-Injektion		X		Legitimierung
Paket-Verwerfung		X	X	Täuschung
Paket-Manipulation	X	X		Verstecken, Legitimieren
Ressourcen-Erschöpfung	X	X	X	Täuschen
Erzeugung von Datenverkehr	X	X	X	Täuschen, Legitimierung

reale Datenverkehrs, welches durch den Router gemeldet wurde, manipuliert. Durch ein schrittweises Absenken der Volumen-Informationen in den Flow-Daten wurden mehrere Verkehrsarten ausgeblendet (vgl. Abbildung F.12c). Hierbei wurde zunächst ein HTTP-Datenstrom simuliert und in den nächsten Schritten die Datenvolumina der realen Daten exakt denen des simulierten HTTP-Datenstromes angepasst, um sämtliche Spuren in der graphischen Ausgabe des Analysetools zu verstecken. Ab 17:30 Uhr ist dadurch nur noch der Datenstrom der vorgetäuschten Verbindung ersichtlich.

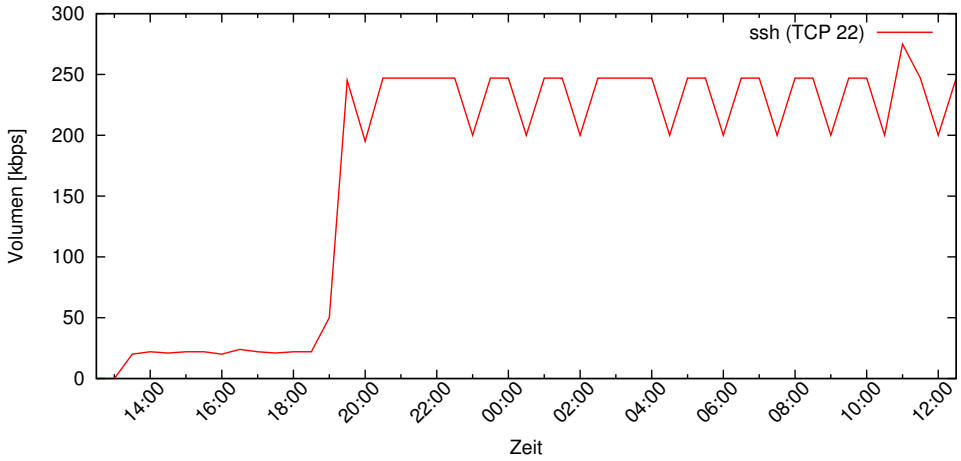
Wie ersichtlich ist, ist eine Manipulation der Flow-Daten leicht möglich. Während der Versuchsdurchführung wurden zu keiner Zeit Alarme oder Anomalien durch das Analysetool gemeldet. Auch mit der Nutzung von kryptographischen Verfahren bei der Übermittlung der Flow-Daten bleiben Angriffsmöglichkeiten bestehen; Tabelle F.5 zeigt einen Überblick über die Angriffsverfahren bei NetFlow.

Durch eine Verschlüsselung der Flow-Daten wird eine direkte Manipulation der Payload-Informationen verhindert, jedoch lassen sich weiterhin Pakete verwerfen, Angriffe zur Erschöpfung der Ressourcen fahren oder, durch die Integration von transparenten Brücken, vorgetäuschter Datenverkehr erzeugen.

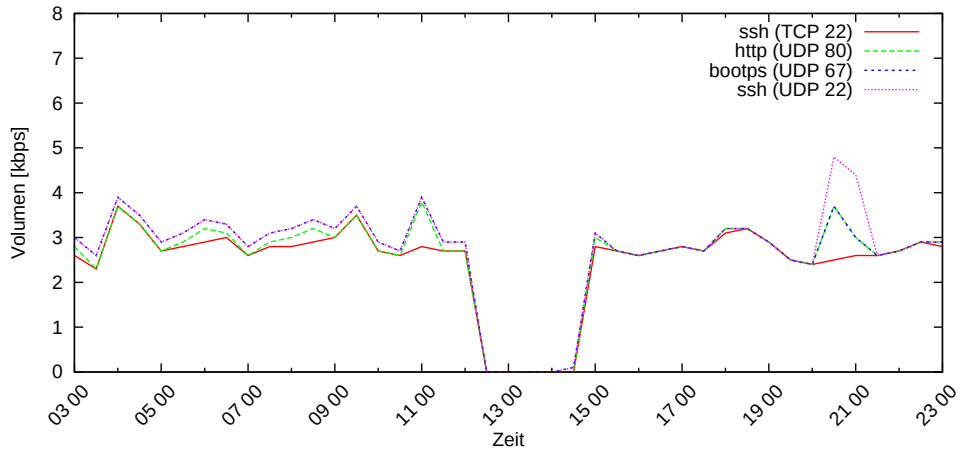
In einer weiteren Versuchsreihe wurde die Einflussmöglichkeit auf durch *sFlow*-fähige Komponenten erzeugte Daten untersucht. Zur Auswertung der durch HP-Switche erzeugten Daten wurde das Programm *NetFlow Analyzer 7 Professional Plus* von ManageEngine verwendet [267]. Wie im vorausgegangenen Falle der Manipulation von NetFlow-Daten wurden die entsprechenden Felder der sFlow-Pakete angegriffen. Tabelle F.6 zeigt die relevanten Strukturen der sFlow-Datenpakete.

sFlow-Daten werden als verbindungslose UDP-Pakete übertragen; da sFlow auf Sampling basierend arbeitet, ist der Verlust einiger Daten-Pakete nicht kritisch und wird daher typischerweise *nicht* von den Analyse-Tools gemeldet, solange der Datenstrom nicht zu lange abreißt. Jedes sFlow-Datagramm kann mehrere Samples beinhalten, deren Anzahl durch den Wert an der Adresse `0x42` angezeigt wird. Ein Sample kann wiederum vom Typ *Flow Sample*, *Counter Sample*, *Extended Flow Sample* und *Extended Counter Sample* sein. Für die Manipulationen wird zunächst lediglich der Typ Flow Sample mit dem Wert 1 benötigt (Adresse `0x46`).

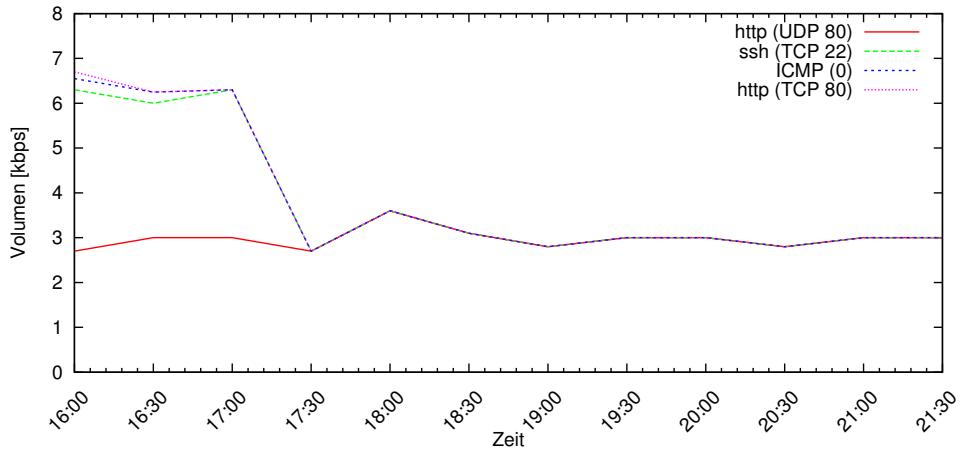
Als erster Angriff wurden Pakete eingespielt, um zusätzliche Verbindungen im Netz zu fingieren. Um die Manipulation leicht im Analyse-Tool nachvollziehen zu können, wurden für die Absender- bzw. Zieladresse der simulierten Datenverbindung die gut erkennbaren



(a) Injektion von NetFlow-Paketen zur Vortäuschung von Datenverkehr.



(b) Verwerfen von NetFlow-Paketen: Ausblenden von Datenverkehr.



(c) Manipulation von NetFlow-Paketen: Verstecken von Datenverkehr.

Abbildung F.12: Durchführung von NetFlow-Angriffen.

Tabelle F.6: Wichtige Elemente eines sFlow-Paketes.

Adresse	Wert	Bedeutung
002A	00 00 00 05	sFlow V5
0032	ac 10 13 0a	Agent Address
003A	00 00 c6 17	Sequence Number
003E	0b 5f 0c b6	System Uptime
0042	00 00 00 01	Number of Samples
0046	00 00 00 01	Sample Type
0062	00 00 00 00	SNMP Input Interface
0066	00 00 00 00	SNMP Output Interface
006A	00 00 00 01	Number of Flow Records
006E	00 00 00 01	Flow Format
0076	00 00 00 01	Header Protocol
007A	00 00 05 ee	Length before Sampling
007E	00 00 00 04	Stripped Bytes
0086	ab ab ab ab ab ab	Destination MAC
008C	ba ba ba ba ba ba	Source MAC
00A0	ed ed ed ed	Source IP
00A4	de de de de	Destination IP
00A8	00 16	Source Port
00AA	e5 26	Destination Port

IP-Adressen `237.237.237.237` respektive `222.222.222.222` verwendet. Für die Durchführung wurde ein sFlow-Payload erzeugt, der entsprechenden SSH-Datenverkehr auf dem Standard-Port `22` simulierte und mittels des Tools *nemesis* auf Basis eines Intervalls von einer Sekunde versendet wurde:

```
$ nemesis udp -H 00:0b:45:11:22:33 -M 00:0f:1f:33:22:11 -S 172.16.18.10
    -D 192.168.0.30 -x 1024 -y 6343 -P inject.data
```

Das resultierende Ergebnis der Evaluation ist in Abbildung F.13 ersichtlich. Durch das hohe, vorgetäuschte Volumen an Datenverkehr sind sämtliche reale Verbindungen anteilig so gering, dass sie in der graphischen Darstellung des Analyse-Tools nicht mehr zu erkennen sind.

Weiterhin sind in Abbildung F.14 die graphischen Repräsentationen abgebildet, mittels derer das Analyse-Tool die ausgewerteten Daten darstellt. Wie zu erkennen ist, wird die Einfärbung der Segmente der Kreise abhängig des jeweiligen Anteils am gesamten Datenvolumen durchgeführt. Insbesondere wird das höchste Datenvolumen immer mit der selben Farbe dargestellt. Abbildung F.14a zeigt die Anzeige vor Beginn des sFlow-Angriffes, während Abbildung F.14b die Auswirkungen des sFlow-Angriffes darstellt.

Die vorgetäuschte Verbindung entspricht einer SSH Sitzung zwischen den Adressen `237.237.237.237` und `222.222.222.222`, welche 36 Prozent des gesamten, durch das

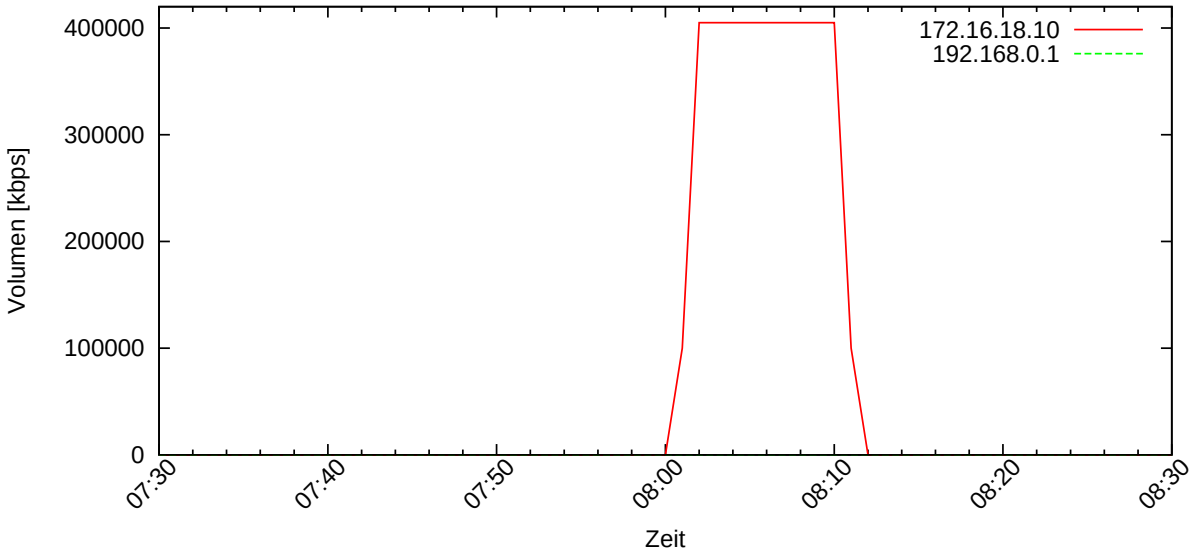
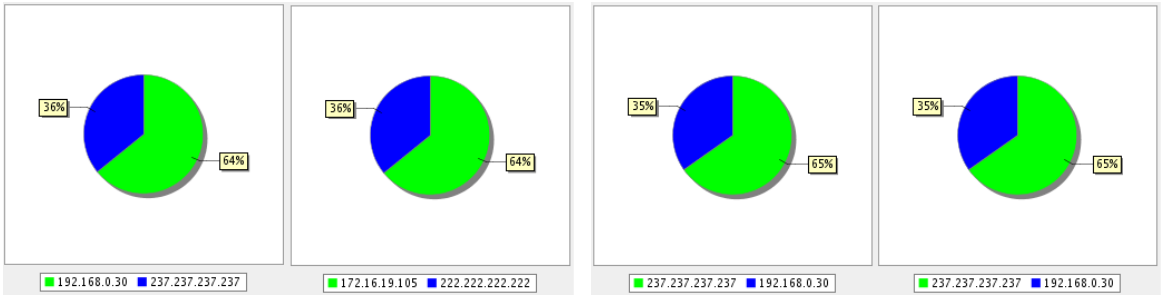


Abbildung F.13: Injektion von sFlow-Paketen zur Vortäuschung von Datenverkehr. Das vorgetäuschte Datenvolumen ist so hoch, dass der reale Datenverkehr in der Darstellung des Analyse-Tools nicht mehr erkennbar ist.



(a) Verbindungsstatistik des Device 172.16.18.10 zu Beginn der Paket-Injektion.

(b) Verbindungsstatistik nach Abschluss der Injektion. Die vorgetäuschte Verbindung zeigt die Charakteristik der realen Verbindung.

Abbildung F.14: Durchführung eines sFlow-Injektionsangriffs, Sichtweise des Analyse-Tools *NetFlow Analyzer 7 Professional Plus*.

Tabelle F.7: Möglichkeiten der sFlow-Manipulation.

Art	Auswirkung
Paket-Injektion	Legitimieren
Paket-Dropping	Täuschen
Paket-Manipulation	Verstecken, Legitimieren
Erzeugung von Datenverkehr	Täuschen, Legitimieren

Interface der Switch transportierten Daten entspricht. Die weiteren 64 Prozent stammen von einer realen Verbindung zwischen den Adressen `192.168.0.30` und `172.16.19.105`. Nach der Durchführung des Angriffes entstammt 65 Prozent der analysierten Daten aus der vorgetäuschten Verbindung, während der reale Datenverkehr lediglich noch mit 35 Prozent eingeht. In der optischen Darstellung des Analyse-Tools führt dies dazu, dass sich die Proportionen der Datenströme umdrehen, was nur schwer erkennbar ist. In diesem Fall kann ein Administrator leicht den Wechsel der IP-Adresse übersehen, was eine Täuschung der Lernphase eines verhaltensbasierten Systems unterstützt.

Für weitere Untersuchungen wurde wiederum eine transparente Brücke in den Pfad zwischen Switch und Analyse-System eingefügt. Dies ermöglicht auch hier die Durchführung von zwei weiteren Angriffen, dem Verwerfen von Paketen sowie der Manipulation der Payload-Informationen. Aufgrund der verschiedenen Möglichkeiten und zahlreichen Optionen, ein sFlow-Paket aufzubauen, ist es für eine Paket-Manipulation einfacher, lediglich die benötigten Daten des abgehörten sFlow-Paketes wie bspw. MAC- und IP-Adressen sowie die SNMP-Schnittstellen für Ein- und Ausgabe, Sequenznummer und Uptime für die Generierung eines neuen Paketes zu nutzen und das originäre Paket zu verwerfen. Eine entsprechende Änderung des Payloads wird von den Analyse-Tools typischerweise nicht erkannt, was sich im zugrunde liegenden Sampling-Verfahren begründet. Wichtig bei der Manipulation ist jedoch, dass die relevanten Adressen, die ID des Flow-Agenten und die genutzten SNMP-Schnittstellen korrekt erzeugt werden, da sonst die Gefahr besteht, dass im Auswertewerkzeug ein neues Flow-Gerät erscheint, bzw. eine entsprechende Meldung generiert wird. Zur Beeinflussung des Datenvolumens wurde das Feld *Frame length before sampling* (vgl. Adresse `0x7A`, Tabelle F.6) geändert.

Das Ergebnis ist in Abbildung F.15 ersichtlich. Nach einer schrittweisen Absenkung einer vorhandenen Verbindung wurde das Datenvolumen für eine kurze Zeit komplett ausgeblendet, anschließend wieder auf ein hohes, konstantes Niveau eingeblendet. Während keiner der Schritte wurde eine Alarmierung durch das Analyse-System ausgelöst.

Tabelle F.7 fasst die Manipulationsmöglichkeiten von sFlow-Datenströmen zusammen.

Werden die Flow-Datenpakete verschlüsselt, schränkt dies insbesondere die Manipulationsmöglichkeiten der übertragenen Daten aus. Eine Lernphase eines verhaltensbasierten Systems ist jedoch trotzdem weiterhin angreifbar: Abbildung F.16 zeigt einen einfachen Aufbau mittels zwei transparenter Brücken zur Beeinflussung der Lernphase bei verschlüsselter Kommunikation der Flow-Instanzen.

Eine andere Möglichkeit, einen entsprechenden Angriff ohne Veränderungen der Hard-

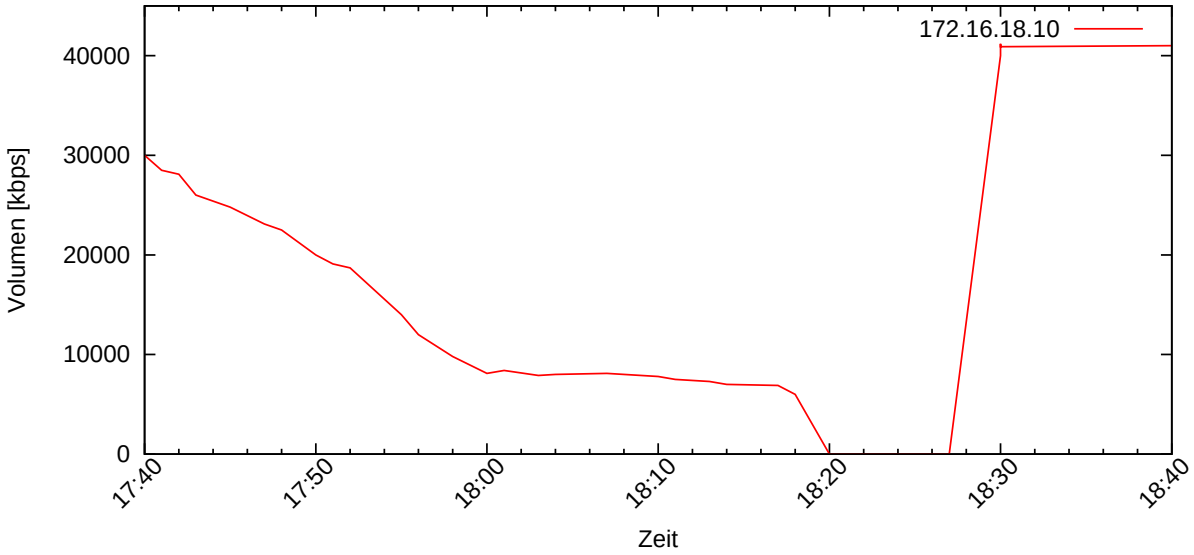


Abbildung F.15: Manipulation von sFlow-Paketen: Ausblenden von Datenverkehr. Durch Änderung der gemeldeten Werte wird die tatsächlich übertragene Datenmenge verschleiert und zunehmend ausgeblendet. Nachdem der Verkehr für einen Zeitraum komplett ausgeblendet wird, erfolgt abschließend das Vorspielen einer konstanten, hohen Datenmenge.

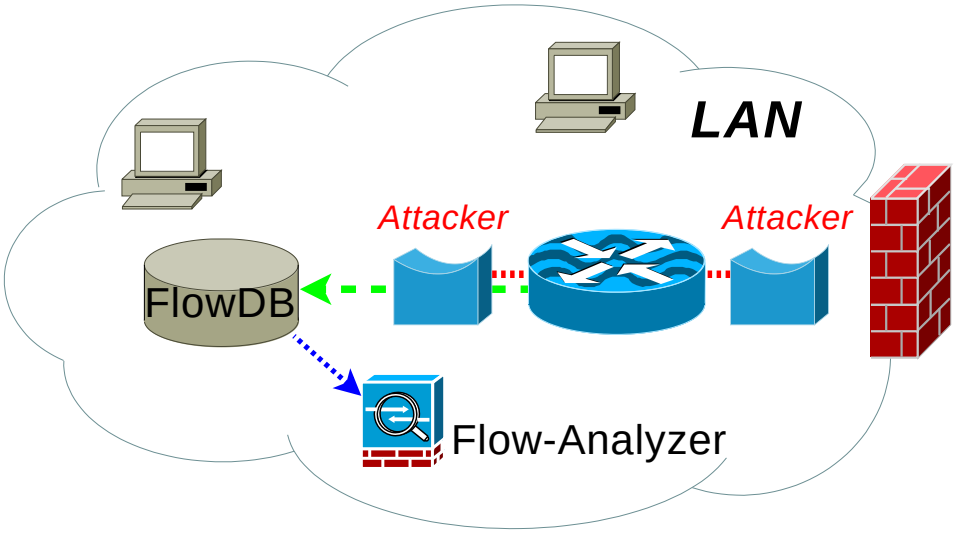


Abbildung F.16: Flow-Angriff bei verschlüsselter Kommunikation der Instanzen. Über transparente Brücken wird Datenverkehr erzeugt, der im restlichen Netz nicht sichtbar ist.

ware vorzunehmen, ist die Durchführung eines ARP-Spoofing-Angriffes: Hierbei werden von einem Angreifer gezielt falsche Adressinformationen in einem lokalen Netz verschickt, um die ARP-Tabellen der Systeme zu manipulieren.

Neben diesen bewusst herbeigeführten Manipulationen der Lernphase besteht jedoch auch weiterhin die Gefahr, dass bereits im Netz befindliche Schadsoftware mit in den Lernvorgang einbezogen wird. Dieses dann fälschlicherweise als konform interpretierte Verhalten wird später als entsprechend legitim ausgewertet, so dass keine Alarmierung erfolgt.

Die Untersuchungen der Manipulationsmöglichkeiten von Net- und sFlow-basierten Analysesystemen hat gezeigt, dass die Lernphase eines verhaltensbasierten Systems sowie die ungesicherte Nutzung von Flowpaketen eine bedeutende Schwachstelle darstellt.

F.2.10 Professionalisierung von Werkzeugen

Seit den 1980er Jahren hat eine stete Professionalisierung der Werkzeuge und Angriffstools stattgefunden, so dass das für die Durchführung von Angriffen erforderliche Wissen entsprechend kontinuierlich abgenommen hat. Die heute verfügbaren Angrifftoolkits erlauben somit auch fachlich nicht versierten Personen, gefährliche Schadsoftware zu generieren und effektive Angriffe durchzuführen. Abbildung F.17 zeigt die Entwicklung der Werkzeuge respektive des erforderlichen Angreiferwissens.

F.2.11 Entwicklung der europäischen IXPs

Die Datenmengen, welche täglich über das Internet transportiert werden, steigen kontinuierlich an. Dies stellt besondere Herausforderungen im Bereich der IDSs dar, da diese kaum noch in der Lage sind, den steigenden Datenmengen mit gleichzeitig komplexer werdenden Analysen Schritt zu halten. Abbildung F.18a zeigt die Entwicklung der Anzahl von Internet Exchange Providers (IXPs) in Europa im Zeitraum von 1992 bis 2010.

Tabelle F.8 zeigt die von den IXPs in den Jahren 2009 und 2010 übertragenen Datenvolumina (Spitzenwerte). Ausgenommen Belgien und Griechenland, die einen negativen Trend aufweisen, sind fast ausnahmslos zweistellige Zuwächse zu verzeichnen; die gesamte Zunahme beträgt 62.69 Prozent.

Dass diese starken Zuwächse keine Ausnahme, sondern die typische Entwicklung seit mehreren Jahren sind, zeigt Abbildung F.18b.

F.2.12 IPv6-Datenvolumen

Obwohl das Nachfolgeprotokoll von IPv4 schon weit über ein Jahrzehnt verfügbar ist und gängige Betriebssysteme sowie Hardwarekomponenten dieses voll unterstützen, konnte es sich in der praktischen Nutzung bisher nicht durchsetzen. Abbildung F.19 zeigt den Anteil von IPv6-Netzverkehr am übertragenen Gesamtverkehr am Beispiel des Amsterdam Internet Exchange (AIX); von Anfang 2010 bis Anfang 2011 konnte dieser 0.5 Prozent nicht überschreiten.

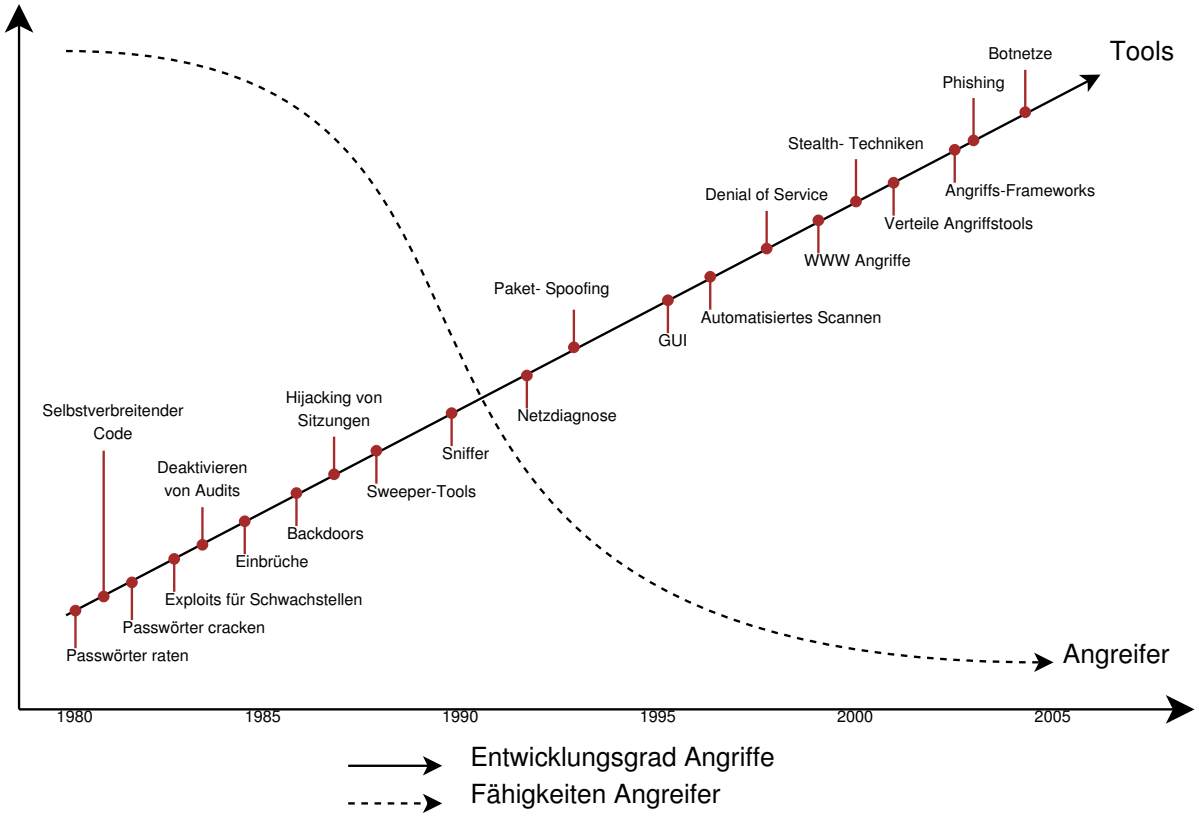
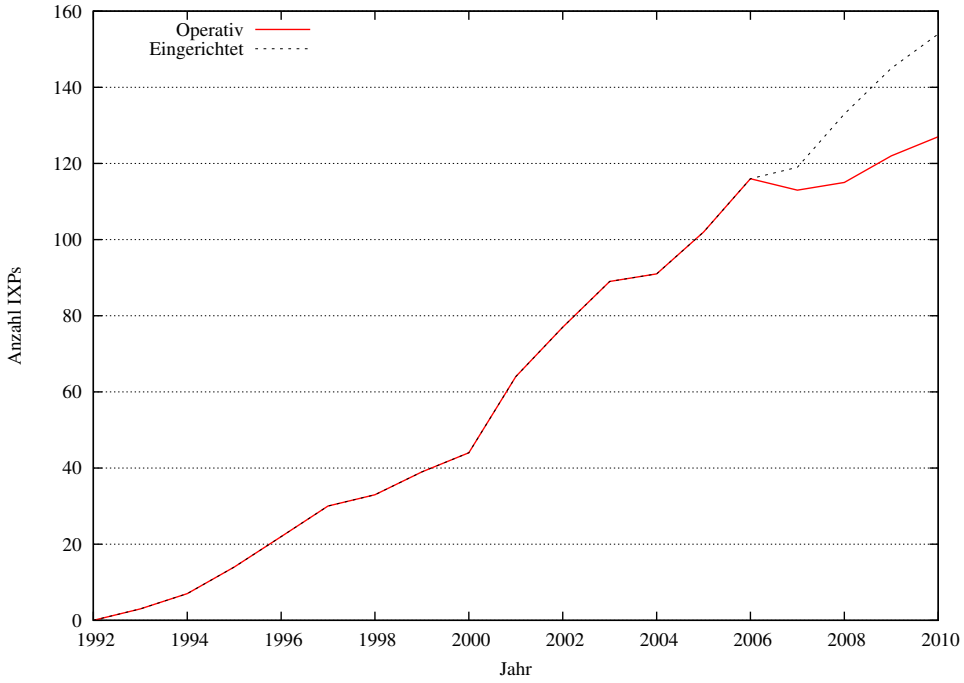
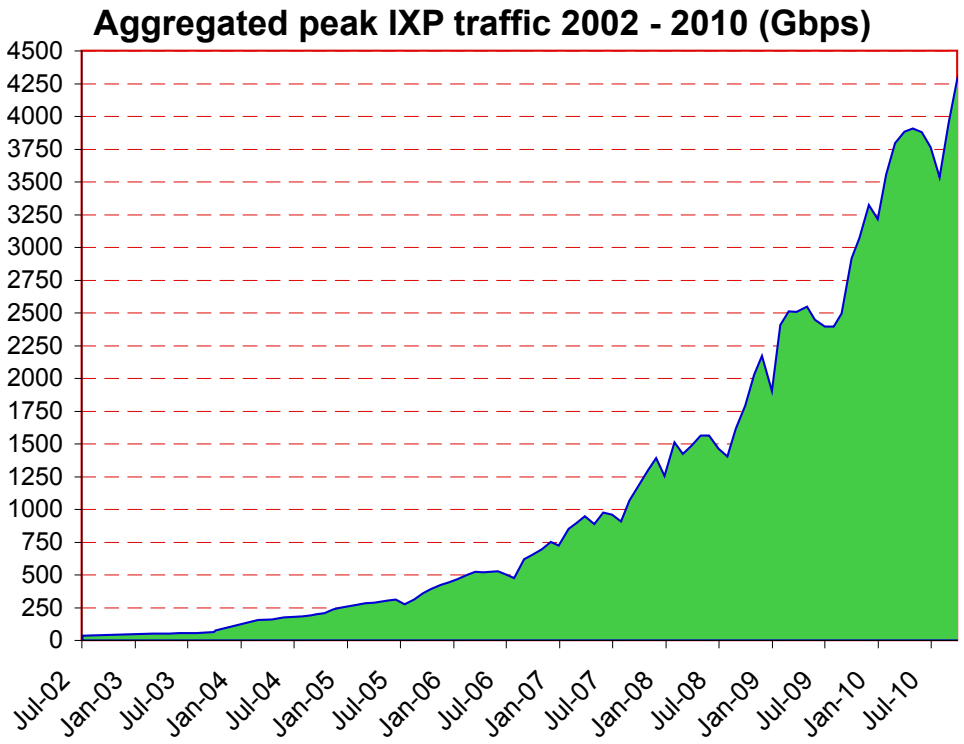


Abbildung F.17: Entwicklung von Angriffen und Tools im Vergleich zum für den Einsatz erforderlichen Fachwissen (Ergänzte Darstellung nach [273]).



(a) Entwicklung der europäischen IXP [319]. Seit 2007 habe 27 Provider den Dienst eingestellt, jedoch wurden im gleichen Zeitraum 38 neu gegründet und in Betrieb genommen. Lediglich 2007 lag eine negative Bilanz vor, hier wurden 6 IXPs geschlossen und nur 3 neu gegründet.



(b) Entwicklung der Verkehrsspitzen der Datenübertragung der europäischen IXP (Quelle: [319]).

Abbildung F.18: Europäische IXP und Datenraten.

Tabelle F.8: Datenverkehr der europäischen IXP nach Ländern in den Jahren 2009 und 2010.

Land	Gbps 2010	Anteil	Gbps 2009	Veränderung zu 2009
Belgien	15.910	0.36	17.600	-9.60
Bulgarien	37.230	0.84	0.100	37130.00
Dänemark	10.700	0.24	10.400	2.88
Deutschland	1069.850	24.23	624.720	71.25
Estland	1.379	0.03	1.179	16.96
Finnland	30.315	0.69	22.093	37.22
Frankreich	124.395	2.82	93.340	33.27
Griechenland	13.540	0.31	14.516	-6.72
Großbritannien	650.240	14.73	460.115	41.32
Island	1.390	0.03	0.508	173.62
Irland	5.506	0.12	4.198	31.16
Italien	78.334	1.77	48.602	61.17
Kroatien	0.961	0.02	0.345	178.55
Lettland	9.430	0.21	2.544	270.68
Luxemburg	60.950	0.02	0.197	382.23
Niederlande	1053.670	23.86	674.980	56.10
Norwegen	29.542	0.67	18.850	56.72
Österreich	36.090	0.82	21.430	68.41
Polen	101.810	2.31	64.170	58.66
Portugal	6.330	0.14	6.213	1.88
Rumänien	47.820	1.08	19.993	139.18
Rußland	303.946	6.88	125.753	141.70
Schweden	183.910	4.16	121.297	51.62
Schweiz	27.600	0.63	10.730	157.22
Slovakai	29.930	0.68	18.528	61.54
Slovenien	26.000	0.59	20.000	30.00
Spanien	110.812	2.51	91.053	21.70
Tschechische Republik	128.500	2.91	71.800	78.97
Ukraine	159.920	3.62	52.511	204.55
Ungarn	119.600	2.71	96.300	24.20
Zypern	0.120	0.00	0.100	20.00
Gesamt	**4.415.730**		**2.714.165**	**62.69**

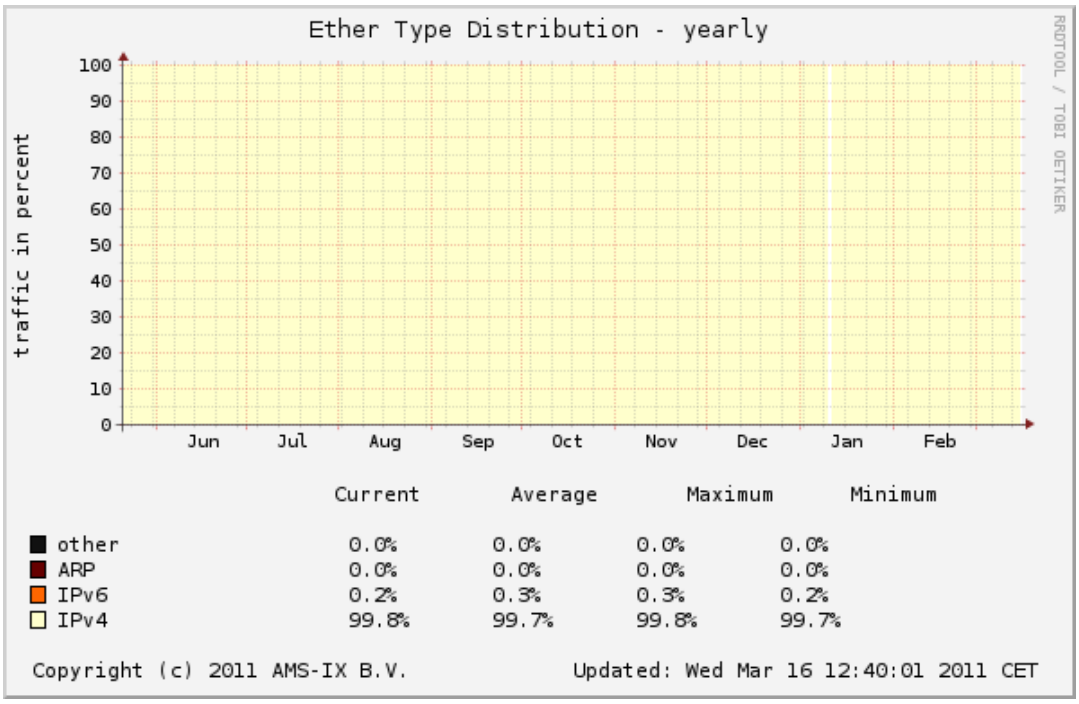


Abbildung F.19: Anteil des IPv6-Datenverkehrs bis Anfang 2011 am Beispiel Amsterdam Internet Exchange [129]. Gut zu erkennen ist, dass der Großteil des Datenverkehrs immer noch auf IPv4 basiert, lediglich 0.3 Prozent des Verkehrsaufkommens wird über IPv6 abgewickelt.

Nachdem zu Beginn des Jahres 2011 jedoch die letzten Adressblöcke von IPv4 an die regionalen Organisationen zur weiteren Verteilung vergeben wurden, werden schätzungsweise bis 2012 sämtliche Adressen des alten Protokolls vergeben sein. Auch wenn mittels Verfahren wie NAT oder der besseren Ausnutzung der bereits vergebenen, jedoch wenig intensiv belegten Adressbereiche eine Umstellung noch hinausgezögert werden kann, kann erwartet werden, dass der Mangel an Adressen die Einführung von IPv6 künftig nachhaltig beschleunigen wird.

F.2.13 Zonenmodell nach S. Sanchez

Die Fehlalarmraten sind ein bedeutendes Problem beim Einsatz von IDSs. Dies betrifft sowohl wissensbasierte, als auch insbesondere verhaltensbasierte Systeme. Um eine Reduzierung der Fehlalarmraten zu erreichen, schlägt Sanchez ein Zonenmodell für die Integration von IDSs vor, um mittels verschieden eingestellter Sensitivitäten in den jeweils bzgl. den Sicherheitsanforderungen verschiedenen Netzbereichen der Systeme die Anzahl der Fehlalarme zu reduzieren (vgl. Abbildung F.20).

F.2.14 Phänomen der *Base Rate Fallacy*

Nachfolgend ist das Phänomen des Trugschluss der Eintrittswahrscheinlichkeit *(Base Rate Fallacy)*[7], welches bei der Nutzung von *Bayesschen Wahrscheinlichkeiten* als Metrik zum Vergleich der IDSs beobachtet werden kann, kurz vorgestellt.

Axelsson hat dieses für den Bereich von IDSs untersucht und kam zu dem Schluss, dass die Wahrscheinlichkeiten für das tatsächliche Vorliegen eines Angriffs bei einer Alarmierung auch bei hohen Detektions- und geringen Fehlalarmwahrscheinlichkeiten des Systems sehr viel geringer sind, als man dies gefühlsmäßig erwarten würde. Demnach gilt für die Wahrscheinlichkeit, dass ein ausgelöster Alarm aufgrund eines tatsächlichen Angriffs erfolgt,

$$P(I|A) = \frac{P(E)P(A|E)}{P(E)P(A|E) + P(\neg E)P(A|\neg E)} \tag{F.1}$$

wobei E bzw. $\neg E$ für Angriffs- bzw. normales Verhalten stehen und A bzw. $\neg A$ für das Auslösen bzw. Ausbleiben eines Alarms. $P(A|E)$ entspricht somit der Detektionsrate (TP), $P(A|\neg E)$ der Fehlalarmrate (FP). Da Angriffe in einer realen Netzumgebung im Vergleich zum gesamten Datenaufkommen relativ gering sein werden, ergibt sich eine entsprechend geringe Wahrscheinlichkeit, bspw. $P(E) = \frac{2 \cdot 10}{1 \cdot 10^6} = 2 \cdot 10^{-5}$, falls mit zwei Angriffen pro Tag gerechnet wird, eine Million Audit-Daten (z.B. Netzpakete) aufgezeichnet werden und jeder Angriff im Schnitt in 10 Audit-Daten erkannt werden kann. Aus diesem Grund dominiert der Faktor der Fehlalarmrate den Faktor der Detektionsrate deutlich:

$$P(I|A) = \frac{2 \cdot 10^{-5} \cdot P(A|E)}{2 \cdot 10^{-5} \cdot P(A|E) + 0.99998 \cdot P(A|\neg E)} \tag{F.2}$$

[7] Ausführungen gem. [39].

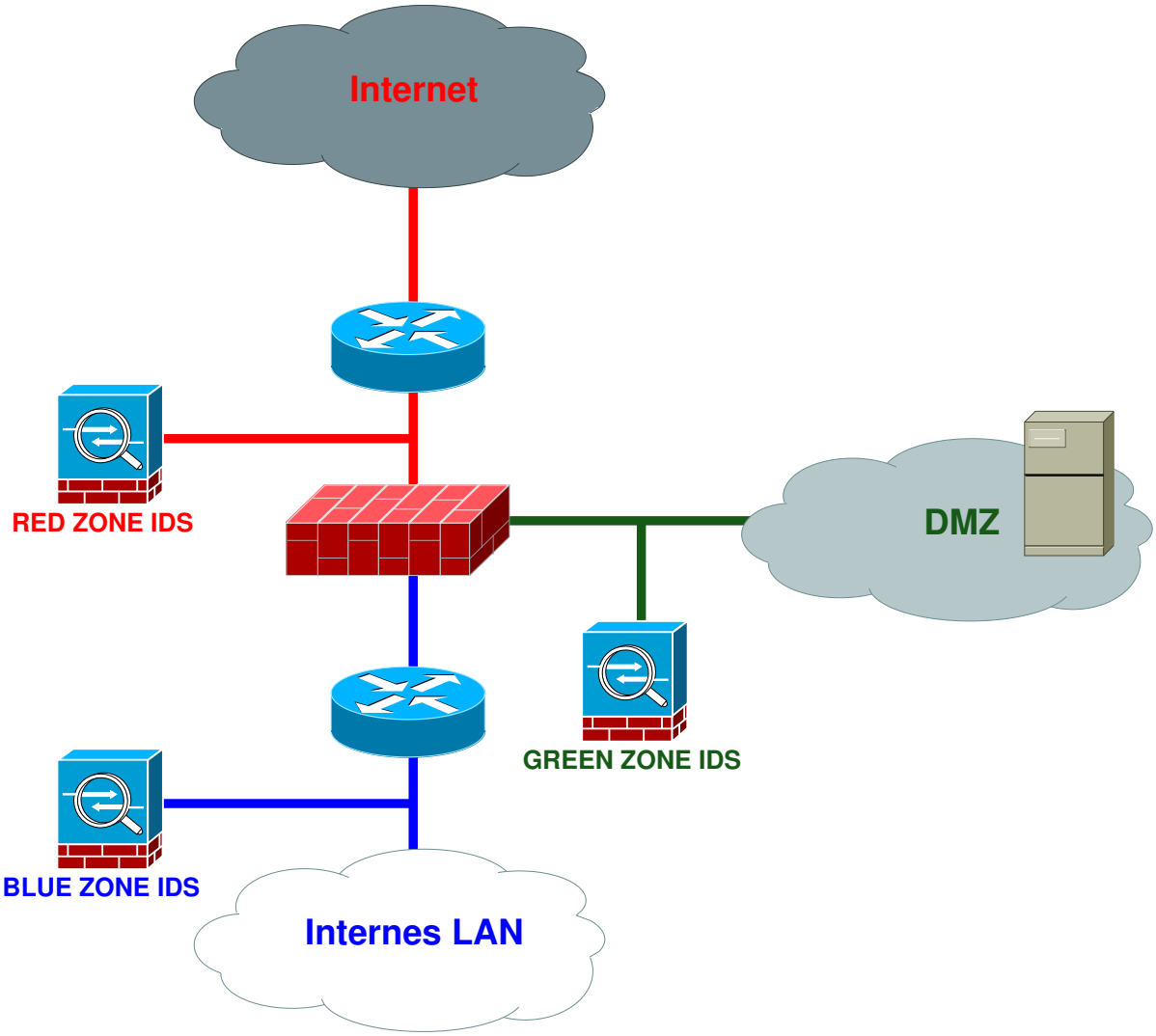


Abbildung F.20: IDS-Zonenmodell nach S. Sanchez [334]. Durch verschiedene Sensitivitäten der IDS in den unterschiedlich eingestuften Zonen lassen sich die Fehlalarmraten reduzieren.

Geht man nun von einer (unrealistisch hohen) Detektionsrate von 1.0 und einer (ebenfalls unrealistisch geringen) Fehlalarmrate von $1 \cdot 10^{-5}$ aus, ergibt sich für die Bayesische Detektionsrate ein Wert von 66 Prozent. Dies bedeutet, dass in zwei Drittel aller Fälle eines ausgelösten Alarmes auch tatsächlich ein Alarm vorliegt.

Betrachtet man jedoch die durchschnittlichen Angriffszahlen auf populäre Webseiten, kann mit 27 Angriffen pro Minute gerechnet werden [214]. Geht man von 40 Millionen Audit-Daten am Tag aus, ergibt sich für $P(I|A)$ ein Wert von 0.9990, falls man von den gleichen Detektions- und Fehlalarmraten wie zuvor ausgeht. Dies entspricht einem sehr guten Ergebnis, da in diesem Falle fast alle Alarme aufgrund eines tatsächlichen Angriffes erfolgen; nimmt man jedoch geringere Detektionsraten, bspw. $P(A|E) = 0.95$ und $P(A|\neg E) = 0.1$, ergibt sich für $P(I|A)$ ein Wert von 0.085. In diesem Falle sind also nur knapp 9 Prozent aller Alarme aufgrund tatsächlicher Angriffe ausgelöst.

Mit realistischeren Werten für die Detektions- und Fehlalarmraten wird dieses Verhältnis schlechter. Im Kontext einer Interaktion des IDS mit dem Nutzer ist dies von Bedeutung, da bei einer hohen Zahl von Fehlalarmen dem System nach kurzer Zeit keine Beachtung mehr geschenkt werden wird.

F.2.15 Aufbau von Frühwarnsystemen

Frühwarnsysteme dienen der frühzeitigen Detektion von Systemanomalien oder Angriffen in Teilbereichen eines Netzes, um mittels des gefundenen Wissens andere, noch nicht angegriffe Netzbereiche zu schützen. Abbildung F.21 zeigt den typischen Aufbau eines EWS. Anhand verschiedener Quellen wie bspw. Firewall- und IDS- Logdateien werden Informationen gesammelt, weiterhin werden nicht genutzte, aber öffentlich routbare IP- Adressen (diese werden als sog. *Darknet*-Bereich bezeichnet) herangezogen, um Honeypot-Systeme zu betreiben. Dies ermöglicht eine detaillierte Untersuchung von Angriffen. Die Daten werden zentral gesammelt und automatisch verarbeitet sowie durch Personal analysiert, typischerweise wird auf dieser Basis eine Sicherheitsstufe der derzeitigen Lage, bspw. durch eine Farbcodierung der Farben Grün, Gelb und Rot, bekannt gegeben.

F.2.16 Datenvolumen mobiler Geräte

Durch die Leistungssteigerung mobiler Geräte und die stetig wachsenden Kapazitäten im Bereich der mobilen Datenübertragung bekommt dieser Bereich eine immer wichtigere Bedeutung. Im Rahmen der Umsetzung eines Sicherheitssystems für entsprechende Systeme entstehen hier besondere Problematiken durch die Besonderheiten und Anforderungen mobiler Systeme wie bspw. relativ geringe Rechen- und Speicherkapazitäten, die Erfordernis einer besonders hohen Energieeffizienz bzgl. der Akkuleistung für eine lange Betriebszeit, etc. Tabelle F.9 zeigt einen Vergleich der Entwicklung der Datenraten in Bezug auf ein traditionelles Mobiltelefon. Ein Sicherheitssystem muss hier entsprechend hocheffizient umgesetzt werden, um den kontroversen Anforderungen genüge zu tun.

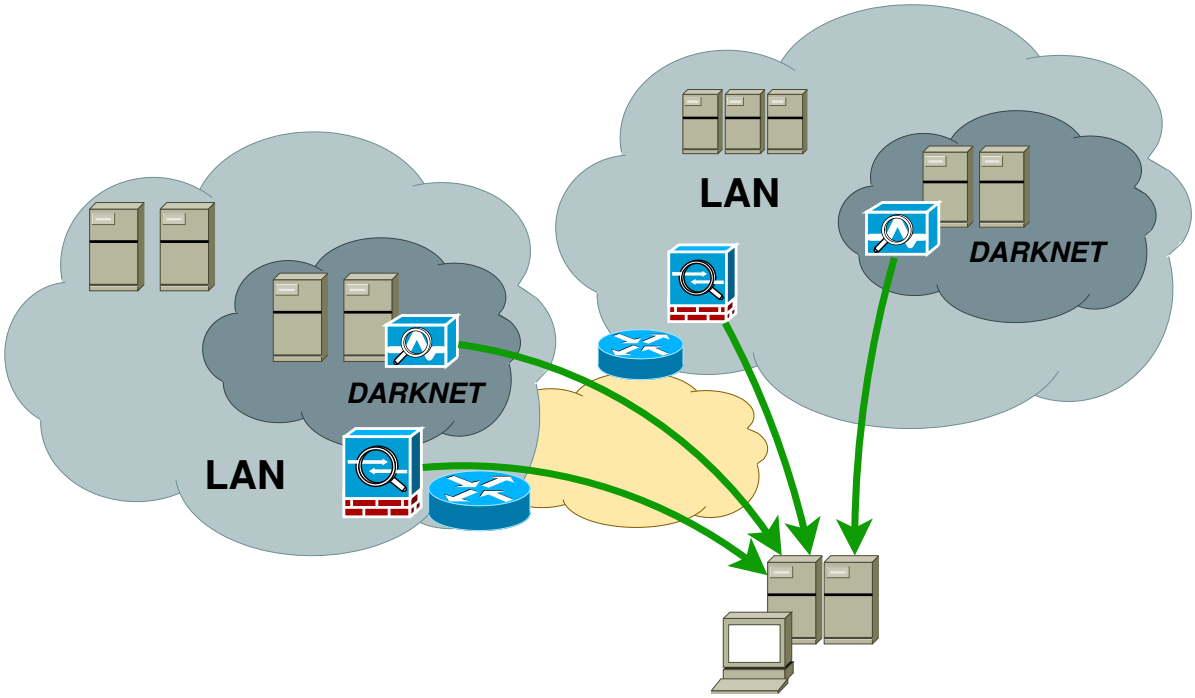


Abbildung F.21: Aufbau eines EWS. Von verschiedenen Netzen werden Firewall- und IDS-Logdateien gesammelt und Flow-Daten des Netzverkehrs evaluiert. Weiterhin sind im Darknet-Bereich (routbare aber nicht genutzten Adressen) Honeypots angelegt, die Angriffe auf sich ziehen und zusätzliche Informationen bereit stellen. Anhand der zentralen Auswertung erfolgt die Lagebeurteilung.

Tabelle F.9: Datenvolumina verschiedener Mobilgeräte in Bezug auf das monatliche Volumen eines Handys [210].

Gerät	Datenvolumen
Handy	1.0
Smartphone	10.0
Digitaler Bilderrahmen	10.0
Videokamera	100.0
Projektor-Handy	300.0
Laptop	1300.0

Tabelle F.10: Vergleich der Aufteilung des IPv4- und des IPv6-Adressraumes, prozentuale Anteile nach [202].

	Südamerika	Nordamerika	Afrika	Europa	Ozeanien	Asien
Adressen IPv4	1	65	< 0.5	16	1	13
Prefix IPv4	2	63	< 0.5	12	5	14
Autonome Systeme	2	57	< 0.5	21	2	13
Adressen IPv6	4	38	< 0.1	35	< 1	28
Prefix IPv6	< 1	19	< 0.2	41	< 1	34
Autonome Systeme	< 1	18	< 0.2	49	< 1	27

F.2.17 Aufteilung des IPv4- und IPv6-Adressraumes

Tabelle F.10 zeigt die prozentuale Aufteilung des IPv4- bzw. IPv6-Adressraumes, bezogen auf geographische Regionen. Gut zu erkennen ist, dass die Aufteilung in den Industrieländern sehr viel homogener erfolgt, als dies bei IPv4 der Fall war. Im Gegensatz dazu ist die prozentuale Zuteilung des Adressraumes von IPv6 an Entwicklungsländer jedoch noch geringer als beim Vorgängerprotokoll. Während der prozentuale Anteil der für Nordamerika reservierten Adressen, Prefixe und Systeme sinkt und insbesondere Europa und Asien höhere Anteile erhalten, liegt der Anteil der Reservierungen für Afrika geringer als bei IPv4. Aufgrund des enormen Adressraumes von IPv6 sollte dies jedoch zu keinen Problemen führen.

F.2.18 IPv4 Census-Map

Abbildung F.22 zeigt die Census-Karte, welche die Belegung des IPv4-Adressraumes wiederspiegelt. Die Reihe der Census-Karten wurde in den Jahren 2003 bis 2006 angelegt; entsprechend sollte heute eine insgesamt höhere Auslastung des Adressraumes gegeben sein, jedoch kann davon ausgegangen werden, dass insbesondere in den Bereichen großer allokierter Adressblöcke einzelner Firmen weiterhin umfangreiche Adressbereiche nicht genutzt werden. Beispielsweise haben Unternehmen und Einrichtungen wie IBM, AT&T, Apple, Xerox, HP oder das MIT jeweils eigene Klasse-A Netze zugewiesen bekommen.

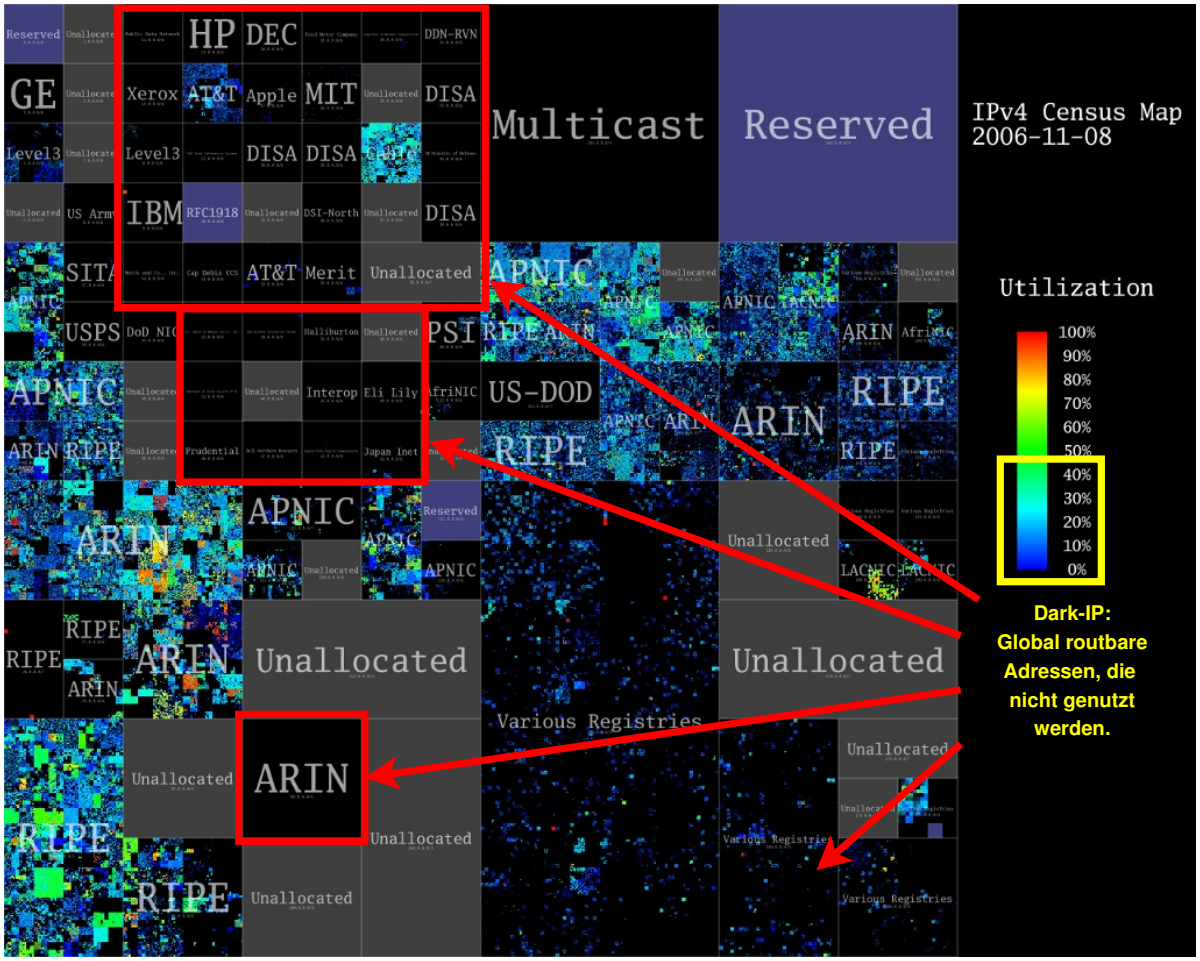


Abbildung F.22: Die IPv4 Census Map zeigt die Vergabe und Belegung des IPv4-Adressraumes [130].

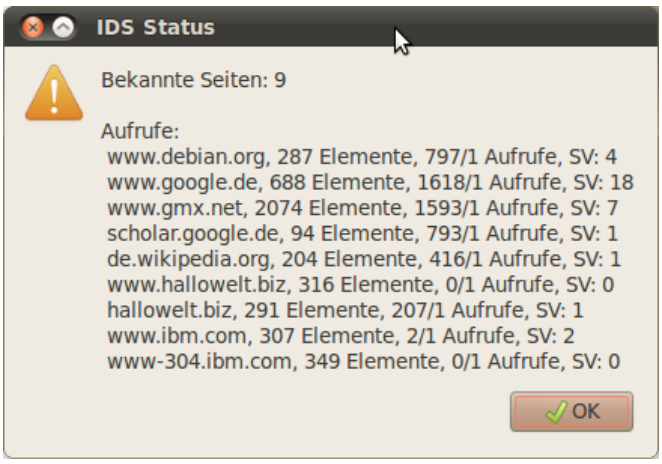


Abbildung F.23: Infobox der zusammengefassten Statistik eines Layer 7- Agenten für den Mozilla Firefox.

F.3 Ergänzungen zu Kapitel 5

In den Ergänzungen zu Kapitel 5 finden sich Details zu Sensorik, welche auf Applikationsebene eingesetzt werden kann bzw. Sensorik für Netzabschlüsse von DSL-Verbindungen. Weiterhin werden zusätzliche Informationen zur Umsetzung und Implementierung der Architektur des vorgestellten Sicherheitssystems für verschlüsselte Umgebungen gegeben.

F.3.1 Software-Sensoren auf Schicht 7

Die Untersuchung von verschlüsselten Verbindungen auf das Vorhandensein von Angriffen auf der Anwendungsebene kann mit den in der vorliegenden Arbeit entwickelten Verfahren erfolgen. Da das entsprechende, statistische Vorgehen immer eine gewisse Fehlerrate aufweist[8], kann eine Verbesserung der Detektionsqualität erreicht werden, indem zusätzliche Informationen der Hosts ausgewertet werden. Hier ist bspw. eine verhaltensbasierte Evaluation auf Layer 7, in Form von in die jeweiligen Applikationen integrierten Sensoren möglich. Zur Untersuchung entsprechender Sensorik wurde ein IDS-Modul für den *Mozilla Firefox* Browser entwickelt, welches Aspekte des Seitenaufbaus überwacht und eine Auswertung des Verhaltens einer Webseite zur Angriffs- und Manipulationsdetektion vornimmt. Das Modul ist als Ergänzung für den Browser zu installieren und arbeitet vollständig transparent im Hintergrund. Werden Manipulations- oder Angriffsanzeichen detektiert, erfolgt eine Alarmierung des Nutzers durch ein Popup-Fenster. Weiterhin können statistische Informationen zum jeweiligen Bewertungsstand der bekannten Webseiten abgerufen werden (vgl. Abbildung F.23).

Besondere Bedeutung können entsprechende Verfahren insbesondere im Bereich der Frühwarnung haben (vgl. Anhang F.3.10); hierbei können die gewonnenen Informationen

[8] Dies gilt jedoch auch für andere Verfahren wie bspw. signaturbasierter Detektion, vgl. Kapitel 4.4.

anstelle oder zusätzlich zu einer Nutzerbenachrichtigung an ein zentrales Auswertesystem im Netz gesendet werden. Im Kontext des hier entwickelten Sicherheitssystems werden entsprechende Sensoren jedoch im Rahmen der identifizierten Forderungen an das System nicht weiter betrachtet (vgl. Kapitel 3). Eine detaillierte Untersuchung von Anwendungs- und Integrationsmöglichkeiten solcher Sensorik in das Sicherheitssystem bietet sich jedoch für künftige Arbeiten an.

F.3.2 Sensoren für Netzabschlüsse*

Eine weitere Möglichkeit, eine Angriffserkennung auf Basis von verteilten Daten bzw. intrinsisch vorhandenem, dislozierten Wissen durchzuführen, ist die Nutzung von Netzabschlüssen wie bspw. dem Modem einer DSL-Verbindung. Diese Art der Anschlüsse ist in Privathaushalten mittlerweile weit verbreitet und findet sich auch häufig bei Firmen als Zugangsverfahren zum Internet. Der Vorteil der Einbeziehung entsprechender Modems als Sensoren in ein IDS ist u.a. die für den Nutzer transparente Analyse des Datenverkehrs sowie die breite Verteilung der Sensorik, um lokale Anomalien im Datenverkehr detektieren zu können. Auch hier können die in der Arbeit vorgestellten Verfahren (vgl. Kapitel 5) genutzt werden, um eine Einbruchs- oder Ausbruchserkennung durchzuführen, ohne den Anwender mit komplexen Konfigurationen oder Installationen zu fordern. Insbesondere können hierdurch auch Ausbruchsaktivitäten, wie bspw. von Bots durchgeführte Aktionen, erkannt werden. Diese sind regelmäßig erschwert zu erkennen, da sie von Innen heraus, aus dem vertrauenswürdigen Netz, erfolgen.

Abbildung F.24 zeigt einen Überblick der technischen Zugangsarchitektur eines IPS zum Internet. Die einzelnen Kundenanschlüsse laufen im Konzentratornetz des ISP auf die Digital Subscriber Line Access Multiplexers (DSLAMs) auf, von dort erfolgt der Transport typischerweise mittels Asynchronous Transfer Mode (ATM) oder IP Encapsulated zu den Broadband Remote Access Servers (BRASs), welche den Datentransport durch das ISP-Netz bewerkstelligen. Mittels der Edge-Router erfolgt der Datenaustausch zwischen den ISPs bzw. den Teilnetzen des Internets. Typischerweise ist im Konzentratornetz eine Installation von Sensorik nicht ohne erheblichen Aufwand möglich, basierend auf der genutzten Transporttechnologie. Ebenso lässt sich hier keine akzeptable Reaktion bei Erkennung von unerwünschtem Datenverkehr durchführen, da bspw. der komplette HTTP-Verkehr gesperrt werden müsste, was aus Nutzersicht nicht akzeptabel ist. Auch im Kernnetz des Providers bzw. an den Edge-Routern ist es nur unter erheblichem Ressourcen-Einsatz möglich, eine Analyse des gesamten Datenverkehrs der Kunden durchzuführen, was gerade aus wirtschaftlicher Sicht nicht im Interesse des ISP liegt, der insbesondere nur für die Bereitstellung der Transportleistung, jedoch nicht für deren Absicherung zuständig ist. Die Nutzung der DSL-Hardware stellt somit eine interessante Alternative dar, neue Sensorik in bereits bestehende Komponenten ohne zusätzliche Hardware zu integrieren und diese umfassend zu verteilen.

*Dieser Abschnitt enthält eine Zusammenfassung von Teilen des Artikels „Bot-Netz ohne Fritz – Ein Frühwarn- und Abwehrsystem für ISPs basierend auf in DSL-Routern platzierten Sensoren“, Sicherheit in vernetzten Systemen, 18. DFN Workshop, 2011.

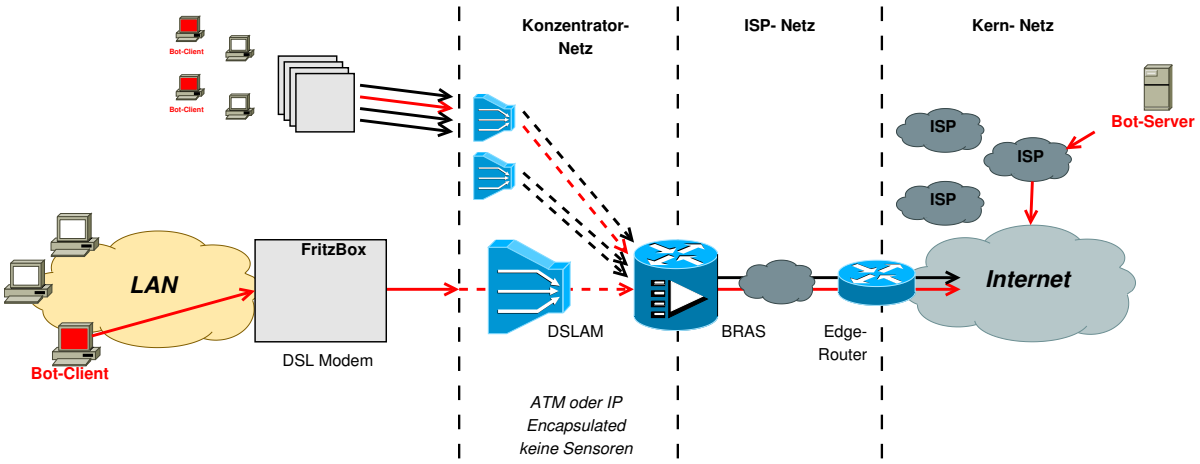


Abbildung F.24: ISP-Architektur für den Zugang zum Internet. In den DSLAMs laufen die Teilnehmeranschlußleitungen zusammen. Die Weiterleitung der Daten erfolgt gebündelt zum BRAS. Der Bot-Client kann ungehindert eine Kommunikation mit dem C2-Server aufnehmen, da Kommunikationsrichtung und -protokoll typischerweise nicht blockiert werden.

Die Architektur eines auf der Nutzung entsprechender, in DSL-Routern installierter Sensorik basierenden Sicherheitssystems ist in Abbildung F.25 ersichtlich. Diese besteht wesentlich aus den folgenden zwei Hauptkomponenten:

- Verteilte Sensorik *vor* dem Konzentratornetz sowie
- fehlertolerante, korrelierte Auswertung der Sensordaten im Bereich des ISP

Mittels der Evaluation von im Datenverkehr erkennbaren Anomalien lassen sich bspw. Botnetz-Aktivitäten, Wurm-Aktivitäten oder ungewollte Verbindungen (bspw. eines Innentäters) erkennen (vgl. Abbildungen F.26 und F.27). Weiterhin können die im Rahmen dieser Arbeit vorgestellten Verfahren genutzt werden, um bspw. eine Analyse ausgehender, verschlüsselter Verbindungen durchzuführen, um Aktivitäten von Bots welche per verschlüsselten Kanälen kommunizieren, zu detektieren und zu unterbinden.

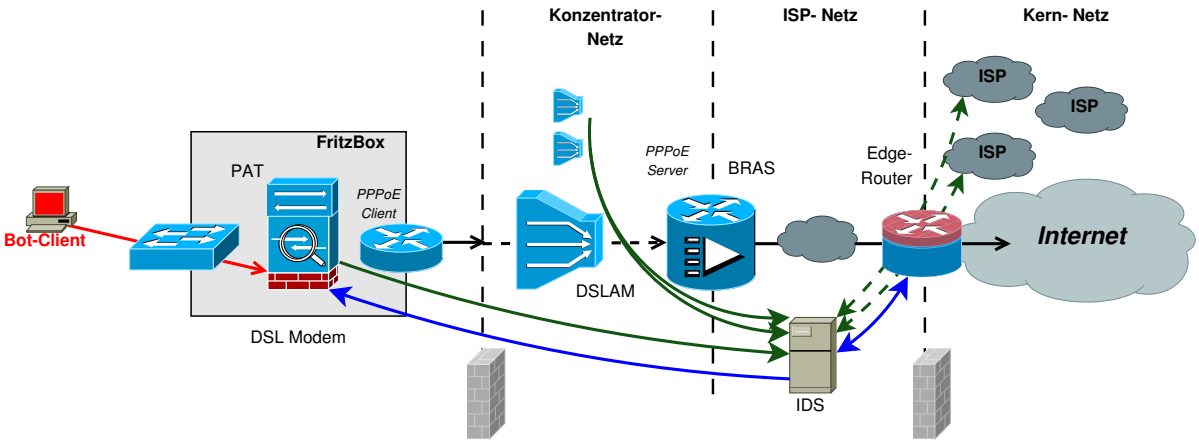


Abbildung F.25: Architektur für ein IDS basierend auf in DSL- Routern verbauter Sensorik. Während die Sensoren *vor* dem Konzentratornetz auf Kundenseite installiert sind, erfolgt eine zentrale Auswertung und Korrelationen der verteilt gewonnenen Daten im ISP-Netz.

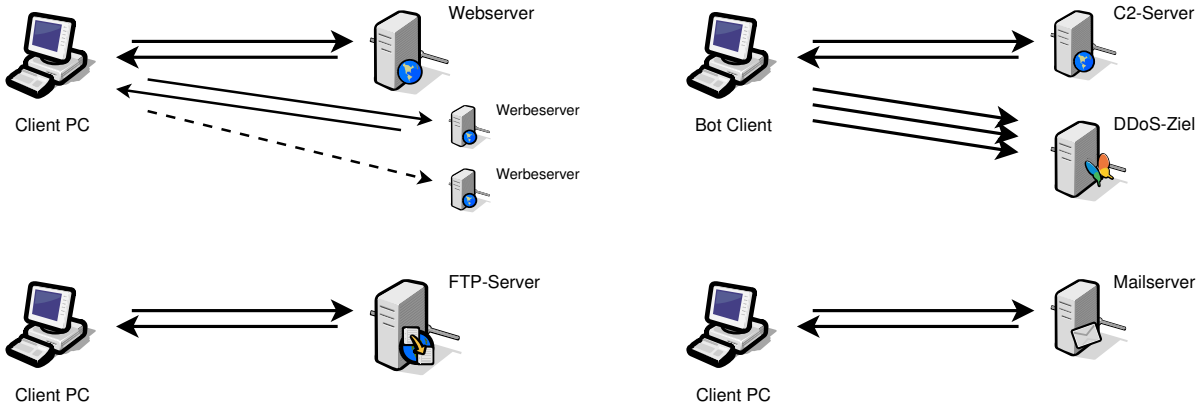


Abbildung F.26: Schematischer Ablauf der Kommunikation zwischen einem Client-PC und verschiedenen Diensten. Die Anzahl von aufgebauten Verbindungen sowie die Datenmengen und Transportrichtungen können Aufschluss über das Vorhandensein unerwünschter Kommunikation geben.

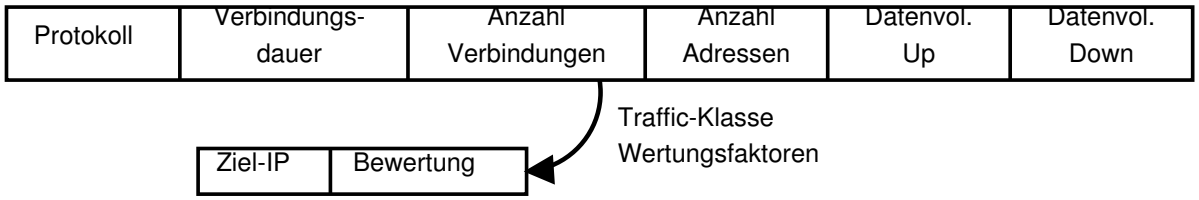


Abbildung F.27: Auswertung des Datenverkehrs auf einer FritzBox. Für jede Verbindung wird ein Datensatz mit Ziel-Adresse und einem Bewertungsfaktor erzeugt. Je höher der Wert, desto wahrscheinlicher handelt es sich um eine unerwünschte Verbindung.

F.3.3 Abhören mittels pcap

Zum Abhören des Datenverkehrs auf der transparenten Brücke können die Funktionen der pcap-Bibliothek verwendet werden. Hierfür muss zunächst die Ethernet-Schnittstelle, auf welcher abgehört werden soll, definiert werden. Anschließend kann der gewünschte Filter-Ausdruck kompiliert werden, bspw. um jeglichen Datenverkehr abzuhören, oder diesen auf bestimmte IP-Adressen oder Ports zu beschränken. Die möglichen Primitive zur Erzeugung der Filter sind ausführlich in den Manpages von pcap aufgeführt (`pcap-filter(7)`). Für die Aufgaben der Sonde wird ein Filter zum Abhören jeglichen TCP-Datenverkehrs erzeugt. Anschließend kann dieser initialisiert und gestartet werden; hierbei ist die Angabe der Anzahl der abzuhörenden Pakete möglich, alternativ kann `NULL` angegeben werden, um jeglichen Verkehr in einer Endlosschleife abzuhören. Nachfolgend sind die erforderlichen Aufrufe aufgeführt:

```
char filter_exp[] = "tcp";
descr = pcap_open_live(dev, SNAP_LEN, 1, 1000, errbuf);
pcap_compile(descr, &filter, filter_exp, 0, net);
pcap_setfilter(descr, &filter);
pcap_loop(descr, num_packets, got_packet, NULL);
```

F.3.4 FNV-1a Hash

FNV-Hashes sind hinsichtlich Geschwindigkeit unter gleichzeitig niedrigen Kollisionsraten konzipiert. Die hohe Geschwindigkeit ermöglicht es, viele Daten unter Wahrung einer geringen Kollisionsrate zu hashen. Durch die hohe Streuung der FNV-Hashes sind diese andererseits auch für das Hashing fast identischer Strings wie bspw. URLs, Hostnamen, Dateinamen, Text, IP-Adressen, etc. geeignet[9].

```
    hash = offset_basis
    for each octet_of_data to be hashed
      hash = hash xor octet_of_data
      hash = hash * FNV_prime
    return hash
```

F.3.5 Hashtable

Für die Umsetzung der Hashes wird die Hashtable-Implementierung von Christopher Clark [97][10] herangezogen. Die relevanten Operationen sind nachfolgend aufgeführt; als Hashfunktion wird hierbei die FNV1-Implementierung eingesetzt (vgl. Definition der Funktion `hashfromkey()`).

```
static unsigned int hashfromkey(void *ky) {
    struct conn_key *k = (struct conn_key *)ky;
```

[9] Übersetzt von [296].

[10] Die Seite stand zum Zeitpunkt der letzten Sichtung im Juli 2011 *nicht* mehr zur Verfügung, allerdings finden sich unter den Schlagwörtern *Christopher Clark hashtable* noch zahlreiche andere Seiten im Web, auf denen die Implementierung verfügbar ist.

```
    return fnv_32_str(k->ip_string, FNV1_32_INIT);
}

struct hashtable * create_hashtable(unsigned int minsize,
                 unsigned int (*hashfunction) (void*),
                 int (*key_eq_fn) (void*,void*));
int hashtable_insert(struct hashtable *h, void *k, void *v);
void * hashtable_search(struct hashtable *h, void *k);
void * hashtable_remove(struct hashtable *h, void *k);
unsigned int hashtable_count(struct hashtable *h);
void hashtable_destroy(struct hashtable *h, int free_values);

int hashtable_iterator_advance(struct hashtable_itr *itr);
int hashtable_iterator_remove(struct hashtable_itr *itr);
int hashtable_iterator_search(struct hashtable_itr *itr,
                           struct hashtable *h, void *k);
```

F.3.6 Identifizierung der Kommunikationsrichtung

Da die Kommunikationsrichtung nicht in den für die Hashtabellen genutzten Schlüsseln enthalten ist und eine dortige Speicherung den Verwaltungs- und Zugriffsaufwand erhöhen würde, wird die Transportrichtung durch die Erzeugung zweier Schlüssel, welche beide Richtungen abdecken, jedoch nur in der ursprünglichen Richtung in der Tabelle erscheinen können, gemäß nachfolgendem Schema identifiziert.

Algorithmus 1 Bestimmung der Transportrichtung.

function DIRECTION($Packet(tcp)$)
  $dport \leftarrow (tcp \rightarrow th\_dport)$
  $sport \leftarrow (tcp \rightarrow th\_sport)$
  $ipString \leftarrow ip \rightarrow ip\_src + IP\_TOK + ip \rightarrow ip\_dst + IP\_TOK + sport$
  $revipString \leftarrow ip \rightarrow ip\_dst + IP\_TOK + ip \rightarrow ip\_src + IP\_TOK + dport$
  $\leftarrow$
  if $hashtable\_search(ipString) \neq NULL$ **then** ▷ Paket Client → Server
  else if $hashtable\_search(revipString) \neq NULL$ **then** ▷ Paket Server → Client
  else ▷ Neue Verbindung, Paket Client → Server
  end if
end function

Hierdurch wird es ermöglicht, die Rolle einer IP, Client bzw. Server, zu identifizieren.

F.3.7 Datenstruktur der Verbindungen

Nachfolgend ist das Datenkonstrukt aufgezeigt, welches die benötigten, statistischen Informationen der verschiedenen Verbindungen speichert. Einige Variablen sind bzgl. der in der Struktur gespeicherten Informationen redundant, werden jedoch aus Effizienzgründen vorgehalten (bspw. die Summen der Payload-Größen der jeweiligen Transportrichtung, welche auch durch die einzelnen, gespeicherten Paketgrößen ersichtlich sind).

```
struct connection {
        unsigned int            c_payload[MAX_SERIES][MAX_PACKETS];
        unsigned int            s_payload[MAX_SERIES][MAX_PACKETS];
        unsigned int            c_pos[MAX_SERIES], s_pos[MAX_SERIES];
        struct timeval          c_seen[MAX_SERIES][MAX_PACKETS];
        struct timeval          s_seen[MAX_SERIES][MAX_PACKETS];
        double                  correlation_valv[MAX_SERIES][
           MAX_SERIES_CORRS];
        struct conn_key         corr_partner[MAX_SERIES][
           MAX_SERIES_CORRS];
        unsigned long long      sum_up[MAX_SERIES];
        unsigned long long      sum_down[MAX_SERIES];

        unsigned int            num_series;
        unsigned int            used_for_corr; //[MAX_SERIES];
        unsigned int            conn_status;
        struct timeval          creation_time;
        struct timeval          last_seen;
        unsigned long long      sum_up_all;
        unsigned long long      sum_down_all;
        struct conn_key         key;
};
```

F.3.8 Padding

Bei der Verschlüsselung müssen grundsätzlich zwei Verfahren unterschieden werden, Strom-Chiffren und Block-Chiffren. Während erstere die einzelnen Zeichen eines Datenstroms mit den Zeichen des Schlüsselstroms mittels Exclusive OR (XOR) verknüpfen, werden bei Block-Chiffren immer Datenblöcke fester Größe verschlüsselt. Ist die Länge der zu verschlüsselnden Informationen kürzer als die Blockgröße bzw. ein Vielfaches davon, muss dieser mit zusätzlichen Bytes aufgefüllt werden (*Padding*). Hierfür kommen folgende Varianten zum Einsatz [249]:

- Auffüllen aller Stellen mit dem Wert, der der Anzahl der zu füllenden Bytes entspricht.
- Der Wert `0x80` gefolgt von Null-Bytes.
- Eine Serie von Null-Bytes und als letztes Byte die Anzahl der Null-Bytes.
- Eine Serie von Null-Zeichen.
- Eine Serie von Leerzeichen.

In der Praxis kommt es ebenfalls zum Einsatz von *Random Padding*, welches in zwei Formen auftreten kann: Wird ein Block nicht vollständig durch die Nachricht gefüllt und mit Nullen aufgefüllt, ergibt dies bei Nutzung von Electronic Code Book (ECB) zur Verschlüsselung immer den selben Chiffretext. Daher kann das Padding auch mit

Zufallswerten gefüllt werden, wodurch sich auch der verschlüsselte Text ändert. Ein weiterer Einsatz von Random Padding kann vorkommen, falls durch die Nachrichtenlänge des Ciphertextes auf Aspekte der Originalnachricht geschlossen werden kann. In diesem Falle kann eine zufällige Anzahl zufälliger Bytes als Padding angehängt werden, wobei typischerweise das letzte Byte die Länge des Pads angibt, wodurch dieser auf maximal 255 Byte beschränkt ist. Hierdurch wird die genaue Länge des Originaltextes verschleiert. Für die Analyse durch das Sicherheitssystem ist dies entscheidend, da in diesem Falle die genaue Größe der ursprünglichen Nachricht nicht mehr zu Evaluation zur Verfügung steht.

Bekannte Beispiele für Block-Chiffren sind Triple Data Encryption Standard (DES), Blowfish oder AES. Die hierbei verwendeten Blockgrößen betragen 8 Byte für (Triple) DES und Blowfish, 16 Byte für AES.

F.3.9 Manipulation der Einbruchserkennung

Um einer Detektion durch die Einbruchserkennung zu entgehen, kann der Angreifer versuchen, auf den Paketstrom der Datenverbindung Einfluss zu nehmen. Bspw. kann das Einstreuen zusätzlicher Tastendrücke genutzt werden, um eine Identifizierung durch Extraktion der biometrischen Daten zu entgehen. Angemerkt werden muss jedoch, dass eine solche Verfälschung als Anomalie wahrnehmbar ist und somit auch in diesem Falle eine Alarmierung erfolgen kann. Ein erster, trivialer Start zur Manipulation der Nutzererkennung kann daher bspw. wie folgt aussehen:

```
#!/bin/bash
for (( ;; ))
do
echo -e "1\x8\c"
sleep 1
done
```

Wird das Skript im Hintergrund ausgeführt (Parameter `&`), werden hierdurch in einem Takt von einer Sekunde Zeichen auf die Konsole geschrieben, die sofort wieder gelöscht werden. Somit wird zusätzlicher Datenverkehr erzeugt, ohne dass dies die Nutzung der Konsole einschränkt. Das gezeigte Beispiel erzeugt lediglich Datenverkehr in eine Richtung, kann jedoch als Startpunkt entsprechender Manipulationsversuche genutzt werden.

F.3.10 Integration von EWS-Informationen in ein IDS der nächsten Generation*

Heute haben sich mehrere Frühwarnsysteme im Internet etabliert, weitere Systeme befinden sich in der Forschung und im Aufbau. Beispiele entsprechender Systeme sind das Internet Storm Center von SANS [89], ATLAS von Arbor Networks [293], AlertCon von IBM [205], Deepsight von Symantec [106] oder ARAKIS des polnischen CERT [289].

*Dieser Abschnitt enthält eine Zusammenfassung von Teilen des Artikels „Towards Next-Generation Intrusion Detection“, Proceedings of the 3rd International Conference on Cyber Conflict (ICCC), IEEE, 2011.

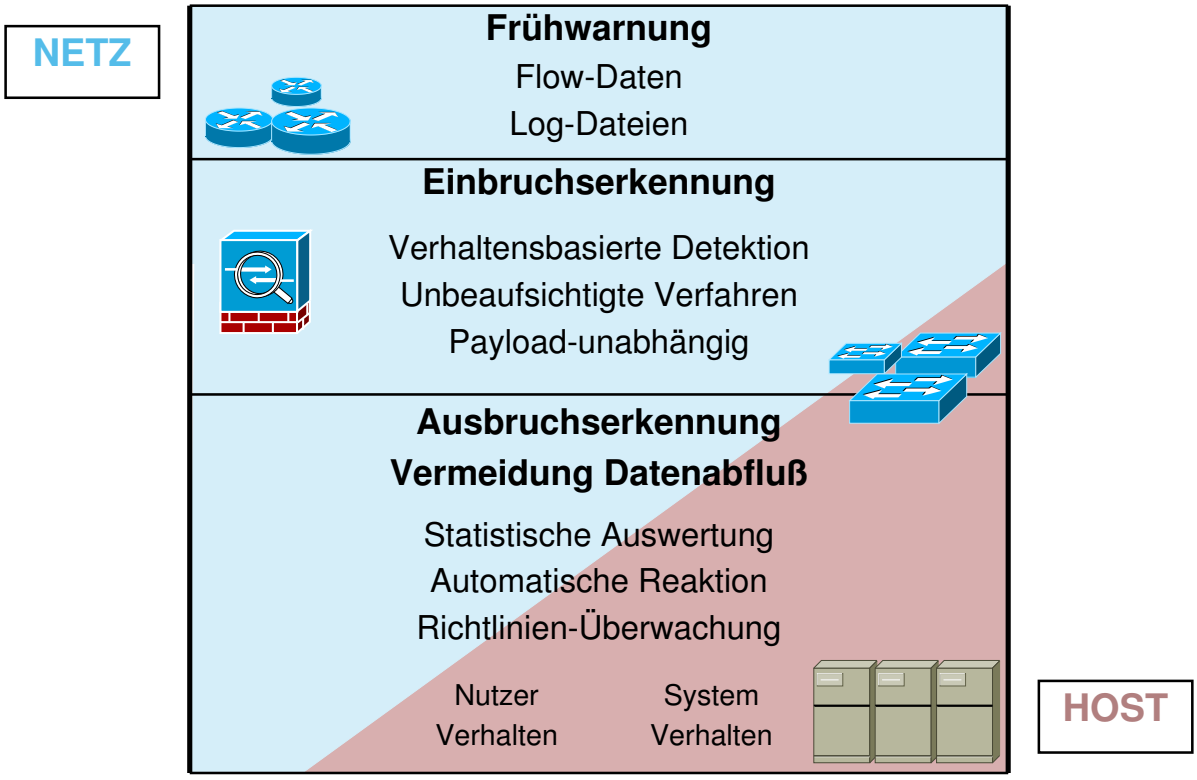


Abbildung F.28: Architektur eines IDS der nächsten Generation. Für eine umfassende Detektion sind verschiedenen Verfahren und Ebenen notwendig.

Frühwarnsysteme ermöglichen es, anhand von Anomalien und Angriffen in Teilnetzen des überwachten Bereichs, noch nicht betroffene Bereiche zu informieren und ermöglichen es somit, entsprechende Schutzmaßnahmen einzuleiten. Die Informationen dieser Systeme können in ein System zur Ein- und Ausbruchserkennung integriert werden.

Aufbau

Abbildung F.28 zeigt die Architektur für ein Sicherheitssystem der nächsten Generation, welches ebenfalls EWS-Daten in die Evaluation einbezieht.

Hierbei können drei Ebenen unterschieden werden, welche die verschiedenen Systembereiche wiederspiegeln. Die unterste Ebene (Ausbruchserkennung, Hostebene) repräsentiert die Server und Hosts des zu überwachenden Netzes während die mittlere Ebene (Einbruchserkennung, Netzebene) den Kern der Überwachung darstellt und im lokalen Netz sowie dessen Infrastrukturkomponenten angesiedelt ist. Netz- und Hostebene nehmen eine besondere Stellung ein, da die dort vorhandenen und involvierten Komponenten in beiden Schichten zur Auswertung herangezogen werden. Die obere Ebene (Frühwarnung, Weitverkehrsebene) wird durch die Evaluation der Gesamtheit der jeweiligen Teilnetze erzeugt.

Diskussion der Komponenten

Nachfolgend werden die einzelnen Komponenten der EWS-Integration kurz betrachtet.

Weitverkehrsebene Hier laufen die aufbereiteten Evaluationsergebnisse der Analysen von Flow-Informationen und Log-Informationen, insbesondere von Firewalls und IDSs von verschiedenen Netzen (autonomen Systemen) zusammen. Auf Basis dieser weit verteilt gesammelten Informationen lassen sich ungewöhnliche Entwicklungen, bspw. durch Schadsoftware verursachte Netzanomalien, erkennen. Diese können wiederum genutzt werden, um die jeweiligen, insbesondere noch nicht betroffenen Teilnetze zu schützen, bspw. durch die Sperrung von Firewall-Ports oder das Blockieren von IP-Bereichen, welche für Angriffe genutzt oder missbraucht werden. Hierfür werden die entsprechenden Informationen von der Ebene der Frühwarnung an die Sicherheitssysteme der Ebene der Einbruchserkennung weitergegeben.

Netzebene Die Einbruchserkennung wird maßgeblich in die Netz-Infrastruktur der jeweiligen lokalen Netze integriert. Durch die Auswahl entsprechender Komponenten, wie bspw. Gateways oder Routern, ist ein umfassender Zugriff auf den Datenaustausch mit anderen Netzen möglich, so dass verteilte Angriffe detektiert werden können und eine umfassende, statistische Analyse erfolgen kann. Da hierbei eine verhaltensbasierter Evaluation erforderlich ist, um zielgerichtete oder neue Angriffe erkennen zu können, muss auf eine Lernphase verzichtet werden oder diese muss in einer gesicherten oder unbeaufsichtigten Weise erfolgen.

Durch die Nutzung von in den Hosts installierten Agenten können detaillierte Auswertungen bzgl. des Systemverhaltens erfolgen und an die zentrale Auswerteinstanz der örtlichen Netzebene weitergegeben werden. Dies ist jedoch optional und für die Evaluation nur von nachgeordneter Bedeutung.

Die Ausbruchserkennung erfolgt ebenfalls auf der Netzebene, da insbesondere Innentäter neben der Autorisierung auch typischerweise physikalischen Zugriff auf ihren Host haben und somit komplexe Manipulationen des Systems möglich werden. Da hierzu insbesondere auch die Beeinflussung eines lokal installierten Agenten oder Sensors zählt, muss die Detektion von Insidern ebenfalls durch statistische Verfahren auf Netzebene erfolgen.

Hostebene Die Hostebene bietet umfangreiche Untersuchungsmöglichkeiten und kann somit die Auswertung über das Vorliegen von Angriffen oder Manipulationen auf Netzebene unterstützen. Da hier jedoch auch die größten Manipulationsmöglichkeiten bestehen und die Pflege und Wartung einer hohen Zahl von Sensoren bzw. Agenten in einer größeren Netzumgebung insbesondere auch eine administrative Herausforderung darstellt, sind entsprechende Hostkomponenten nur optional und lediglich unterstützend einzusetzen.

Abbildung F.29 zeigt die Kommunikationsbeziehungen der verschiedenen Ebenen.

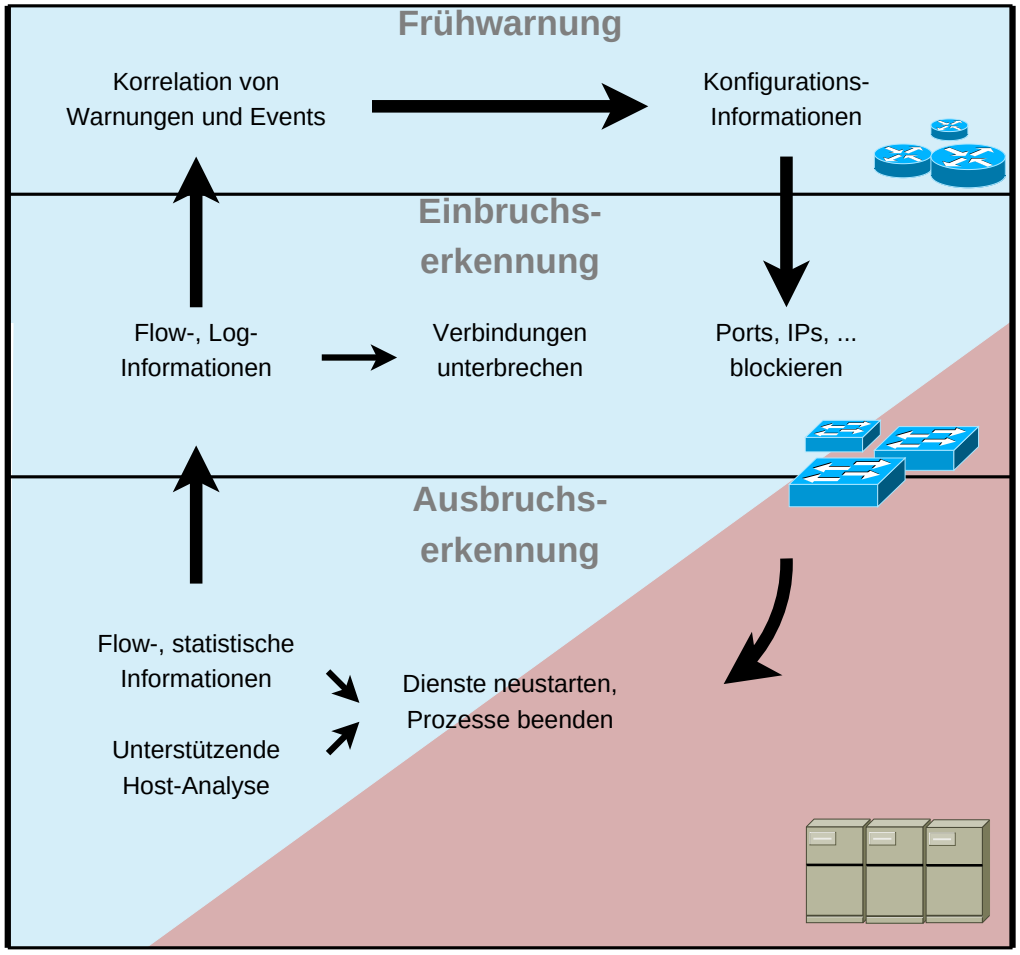


Abbildung F.29: Informationsfluss für ein IDS der nächsten Generation.

F.3.11 Algorithmen

Nachfolgend sind einige der in der Architektur umgesetzten Kernalgorithmen dargestellt. Zur Verbesserung der Lesbarkeit und Verständlichkeit werden nicht alle Stellen detailliert dargestellt, sondern nach Möglichkeit sprechend zusammengefasst.

Suche nach Clustergrenzen

Algorithmus 2 zeigt die Umsetzung der Suche nach den Clustergrenzen innerhalb des Datenstromes gem. den in Kapitel 5.1.3 aufgeführten Funktionsprinzipien.

Schnelle Brute Force-Erkennung

Zur Umsetzung der schnellen, netzbasierten Brute Force-Erkennung sind die verschiedenen, in Kapitel 5.1.5 aufgeführten Fälle zu berücksichtigen. In der Umsetzung als Modul im Rahmen des Sicherheitssystems S2E2 werden hierfür in der Funktion `cluster_type()` drei grundlegende Analysen implementiert (vgl. Algorithmen 3, 4 und 5).

Teilzieldetektion der Angriffsbäume

Nach der Identifizierung der bei einer Nutzersitzung eingegebenen Befehle erfolgt die Auswertung der Befehlssequenzen, um anhand dieser die mögliche Erfüllung von Teilzielen der Angriffsbäume zu überprüfen. Das hierfür in Kapitel 5.1.5 vorgeschlagene Verfahren wird gem. Algorithmus 6 umgesetzt.

Algorithmus 2 Suche der Clustergrenzen.

```
function CLUSTERING(Packet_i)
    while nächstes Paket do
        if size(Packet_i) == 0 then                          ▷ ACK, etc., wird verworfen
            return
        end if
        if cluster_start == NULL then
            if source(Packet_i) == Client&&size(Packet_i) == size(keystroke)
then
                cluster_start ← Packet_i
            end if
        else
            if source(Packet_i) == Client then
                if minAnswerTime&&minSize then               ▷ Beginn neuer Befehl
                    minAnswerTime ← false
                    minSize ← false
                    length_in ← 1
                    length_out ← 1
                else                                         ▷ Fortsetzung Eingabe
                    inputSeries[] ← payload(Packet_i)
                    inputTiming[] ← timestamp(Packet_i)
                    lenght_in ← length_in + 1
                end if
            else                                             ▷ Server-Paket
                if size(Packet_i) == size(keystroke)&&(echokeys + 1 == length_in)
then
                    echokeys ← echokeys + 1
                else                                         ▷ Server-Antwort
                    outputSeries[] ← payload(Packet_i)
                    outputTiming[] ← timestamp(Packet_i)
                    minSize ← true
                end if
            end if
        end if
    end while
end function
```

Algorithmus 3 Schnelle Brute Force-Erkennung, ein Port.

```
1: function CORRELATION_TYPE(connectionconn)
2:     if !connected&&!FIN&&num(packets_server[]) ≥ packets_min&&(serverSeries_continue + n · size(win) + serverValues_used ≤ serverSeries_position) then ▷ Ein Port, kein FIN
3:         for i ← 0; i < shortwins; ++i do
4:             crosscorrelate(serverValues[serverSeries_continue + serverValues_used + i · size(win)], serverValues[serverSeries_continue + serverValues_used + (i + 1) · size(win)], result[i], delay)
5:         end for
6:         for i ← 0; i < shortwins; ++i do
7:             correlation_sum ← correlation_sum + result[i]
8:         end for
9:         correaltion_sum ← correlation_sum/num(shortwins)
10:        if correlation_sum ≥ MIN_CORR_DELTA then
11:            check_alert()
12:        else
13:            ++conn → unsuspicious[idx]
14:            if conn → unsuspicious[idx] ≥ LOW_CROSS_NUM_THRES then
15:                conn → connected[idx] ← 1
16:                ++conn → numConnected
17:            end if
18:        end if
19:    end if
```

Algorithmus 4 Schnelle Brute Force-Erkennung, mehrere Ports mit Schwellwert.

```
20:     if !connected&&num(FIN) > FIN_thres then                ▷ Thres., Ein- oder mehrere Ports
21:         maxValv ← MAX_CORR_VALVS
22:         for i ← 0; i > connection → activePorts; ++i do
23:             if (conn → FIN[i] > 0)&&(conn → serverSeries_position[i] ≥
    MIN_CORR_VALVS) then
24:                 if firstIdx == −1 then
25:                     firstIdx ← i
26:                 end if
27:                 ++num(valvs)
28:                 if conn → serverSeries_position[i] < maxValv then
29:                     maxValv ← conn → serverSeries_position[i]
30:                 end if
31:             end if
32:             if num(valvs) < MAX_CONNS_SGL_THRES then
33:                 break
34:             end if
35:             for i ← conn → activePorts − 1; i ≥ 0; −−i do
36:                 if (conn → FIN[i] > 0)&&(conn → serverSeries_position[i] ≥ maxValv) then
37:                     break
38:                 end if
39:             end for
40:             for j ← i − 1; j ≥ 0; −− j do
41:                 if (conn → FIN[j] > 0)&&(conn → serverSeries_position[j] ≥ maxValv) then
42:                     crosscorrelate(conn → serverSeries[i][0], conn →
    serverSeries[j][0], result[k], delay)
43:                     i ← j
44:                     ++k
45:                     if k ≥ MAX_USED_PORTS then
46:                         break
47:                     end if
48:                 end if
49:             end for
50:             for i ← 0; i < k; ++i do
51:                 correlation_sum ← correlation_sum + result[i]
52:             end for
53:             correlation_sum ← correlation_sum / k
54:         end for
55:         if correlation_sum ≥ MIN_CORR_DELTA then                ▷ Alarmbedingung
56:             check_alert()                                       ▷ Blockieren nach 3x
57:         else
58:             −−conn → FIN[firstIdx]
59:             num(conn → FIN)
60:             −−conn → activePorts
61:         end if
```

Algorithmus 5 Schnelle Brute Force-Erkennung, mehrere Ports ohne Schwellwert.

```
62:     else                                                    ▷ Kein Thres., mehrere Ports
63:         for i ← (conn → activePorts − 1); i ≥ 0; − − i do
64:             if (conn → connected[i] == 0)&&(conn → serverSeries_position[i] ≥ serverValues_used)
    then
65:                 correlation_idx[richPay] ← i
66:                 + + richPay
67:             end if
68:         end for
69:         if richPay < MAX_CONNS_SGL_THRES then
70:             break
71:         end if
72:         j ← richPay − 1
73:         if richPay > MAX_USED_PORTS then
74:             richPay ← MAX_USED_PORTS
75:         end if
76:         for i ← 0; i < (richPay − 1); + + i do
77:             crosscorrelate(conn → serverSeries[correlation_idx[j − i][0]], conn →
    serverSeries[correlation_idx[j − i − 1][0]], result[countEvals], delay)
78:             + + countEvals
79:         end for
80:         if countEvals then
81:             for i ← 0; i < countEvals; + + i do
82:                 correlation_sum ← correlation_sum + result[i]
83:             end for
84:             correlation_sum ← correlation_sum / countEvals
85:             if correlation_sum ≥ MIN_CORR_DELTA then
86:                 check_alert()
87:             else
88:                 + + conn → unsuspicious[idx]]
89:                 if conn → continueCheck[idx] == 0 then
90:                     conn → continueCheck[idx] ← conn → continueCheck[idx] +
    SHORT_CORR_WINS ∗ (NUM_SHORT_WINS − 1)
91:                 else
92:                     conn → continueCheck[idx] ← conn → continueCheck[idx] +
    SHORT_CORR_WINS ∗ NUM_SHORT_WINS
93:                 end if
94:                 if conn → unsuspicious[idx] ≥ LOW_CROSS_NUM_THRES then
95:                     conn → connected[idx] ← 1
96:                     + + conn → numConnected
97:                 end if
98:             end if
99:         end if
100:    end if
101: end function
```

Algorithmus 6 Überprüfung der Erfüllbarkeit von Teilzielen im Rahmen der Befehlsevaluation.

```
1: function TreeCheck(cl_i)
2:     num_subtree_t,k ← 0
3:     for all elements cl of clusters do
4:         for all zutreffende Angriffsbäume tree_t do        ▷ Angriffsabäume abh. des Angriffsschrittes
5:             wähle cmd_i,j ∈ cl_i mit j ∈ [0, .., 2]
6:             for j = 0; j ≤ 3; + + j do
7:                 if cmd_i,j ∈ subtree_t,k then
8:                     num_subtree_t,k ← num_subtree_t,k + 1
9:                     cmds_t,k ← cmds_t,k ∪ cmd_i,j
10:                    continue
11:                end if
12:            end for
13:        end for
14:    end for
15:    for all zutreffende Angriffsbäume tree_t do
16:        if num_subtree_t,k > 2 then
17:            for all elements cl of clusters, cmds_t,k ∉ cl do
18:                for all cmd ∈ cmd_i,j, cmd ∉ cmds_t,k do
19:                    testvalv[cmd] = similarity(cmd) · probability(subtree_t,k, cmd)
20:                end for
21:                sort(testvalv)
22:                for all j = 0; j ≤ 3; + + j do
23:                    if cmd_i,j ∈ subtree_t,k then
24:                        num_subtree_t,k ← num_subtree_t,k + 1
25:                        cmds_t,k ← cmds_t,k ∪ cmd_i,j
26:                        continue
27:                    end if
28:                end for
29:            end for
30:        end if
31:    end for
32:    for all zutreffende Angriffsbäume tree_t do
33:        if num_subtree_t,k > 4 || attacksum_t,k ≥ 3.0 then
34:            block_rule()
35:        end if
36:    end for
37: end function
```

F.4 Ergänzungen zu Kapitel 6

Die Ergänzungen zu Kapitel 6 beinhalten die Vorstellung der Test- und Evaluationsumgebung, welche im Rahmen der Arbeit konzipiert und genutzt wurde, sowie ergänzende Messungen und Ausgaben der prototypischen Implementierungen.

F.4.1 Test- und Evaluationssystem

Eine Evaluation von IDSs stellt mehrere Herausforderungen dar, welche in Kapitel 4.4 dargelegt wurden. Die dort festgestellte Notwendigkeit, insbesondere verhaltensbasierte Systeme unter den realen Bedingungen des Produktivnetzes erproben zu müssen, steht im Konflikt mit der Sicherheit des produktiven Netzbetriebes. Weiterhin ist zur Feststellung der Leistungsfähigkeit in Bezug auf Erkennungs- und Fehlerraten eine genaue Kenntnis der analysierten Daten notwendig. Insbesondere muss die Klasse der jeweiligen Daten, d.h. gutartig oder bösartig, bekannt sein, um die Korrektheit der Analyse zu verifizieren. Dies ist bei Produktivdaten jedoch regelmäßig nicht möglich. Weiter kann ein System in der Entwicklungsphase regelmäßig nicht in einem produktiven Netz eingesetzt werden, insbesondere nicht, wenn es sich um ein IPS handelt, welches auch aktiv in den Netzverkehr eingreift. Daher wird eine *transparente, korrelationsfreie Evaluation in der produktiven Netzumgebung* benötigt.

Um den produktiven Betrieb nicht zu stören, jedoch die entsprechenden Daten uneingeschränkt zur Verfügung zu haben, wurde eine spezielle Evaluationsumgebung aufgesetzt. Neben einer abgeschotteten Untersuchung der Daten ermöglicht die Umgebung weiterhin, das IDS in einfacher Weise zu parallelisieren.

Es existieren drei typische Varianten, mittels deren der Datenverkehr eines Netzes kopiert werden kann:

- **SPAN-Port:** Insbesondere bei der Nutzung von SPAN-Ports kann es zum Verlust von Paketen oder Verfälschungen bzgl. des Timings, bspw. durch das Hinzufügen eines zusätzlichen Delays, kommen [303]. Vor allem bei hoher Last an Datenverkehr kann ein mehr oder weniger zufälliger Verlust an Paketen erfolgen. Betrachtet man bspw. vier Ports `A, B, C` und `D`, wovon jeweils `A` und `B` bzw. `C` und `D` im Full Duplex Modus und mit den maximalen Datenraten der Ports kommunizieren, muss die Backplane der Switch die doppelte Datenrate eines Ports transportieren. Wird hierbei also ein Port der Switch als Span-Port genutzt, können nicht alle Pakete weitergegeben werden.

- **TAP-Device:** Hierbei handelt es sich um spezielle Hardwarekomponenten, die in einen Netzlink zwischengeschaltet werden können und sämtlichen Datenverkehr, welcher übertragen wird, an entsprechende Ports zu weiteren Nutzung kopieren. Hierbei gilt, dass Taps den kompletten Datenverkehr inklusive aller Layer 1 und 2 Fehler duplizieren, ohne dadurch Engpässe oder Fehlerpunkte zu generieren [292]. Von der Nutzung von SPAN-Ports muss daher im Rahmen einer Netzüberwachung zugunsten von Tap-Devices abgeraten werden (vgl. z.B. [303]).

Tabelle F.11: Verfahren zum Kopieren von Netzverkehr und deren Vor- und Nachteile.

	SPAN-Port	TAP-Device	Firewall
Vollständigkeit	✗	✓	✓
Bestehendes Equipment	(✓)	✗	✓
Manipulationsmögl.	✗	✗	✓

- **Firewall-basiert:** Während Tap-Devices den gesamten Datenverkehr als Kopie zur Verfügung stellen können, wirkt sich nachteilig aus, dass keine Einflussnahme auf den Datenstrom möglich ist, sowohl hinsichtlich der Auswahl an Daten als auch der aktiven Einflussnahme auf die gesehenen Verbindungen. Dies lässt sich umgehen, indem man als dritte Variante eine Firewall-basierte Umsetzung für das Kopieren des Datenverkehrs nutzt. Hierfür wird nachfolgend insbesondere das sog. `TEE-Target` verwendet.

Tabelle F.11 stellt die Vor- und Nachteile der einzelnen Konzepte gegenüber.

Im Folgenden werden die genutzten Konzepte im Rahmen von Firewalls, welche für die Evaluation zum Einsatz kamen, vorgestellt.

Evaluationsumgebung mittels *ROUTE*-Patch

Der *ROUTE*-Patch ist eine Erweiterung der *netfilter*-Firewall von Cédric de Launois, der das zusätzliche Target `ROUTE` hinzufügt. Dieses kann genutzt werden, um ungewöhnliche Paketrouten umzusetzen und wird mittels der *Patch-o-Matic* integriert [291]. Der Code erweist sich jedoch als schlecht dokumentiert und wird auch seit längerem nicht mehr gepflegt.

Die korrekte Versionsauswahl der einzelnen Komponenten, namentlich Kernel, ROUTE-Patch sowie *iptables* ist von besonderer Bedeutung, da der Kompiliervorgang des Patches in vielen Fällen aufgrund von Abhängigkeiten und geänderten Datenstrukturen abbricht. Folgende Kombination hat sich als lauffähig und stabil erwiesen:

- `GNU/Linux 2.6.29.6`
- `iptables-1.4.1.1`
- `patch-o-matic-ng-20091109`

Mittels dieser Konstellation wurden mehrere Versuchsreihen und Experimente, welche in die vorliegende Arbeit eingegangen sind, durchgeführt. Im Laufe der Arbeit wurden jedoch sämtliche Sonden auf das *TEE*-Target umgestellt, welches seit Kernelversion `2.6.35` im GNU/Linux-Kernel aufgenommen ist.

Evaluations- und Parallelisierungsumgebung mittels TEE-Target

xtables-addons stellt Erweiterungen zur Verfügung, die aufgrund des experimentellen Status noch keinen Eingang in *iptables* bzw. den Kernel gefunden haben. Es stellt den Nachfolger der *Patch-o-Matic* dar, in welchem auch das *TEE*-Target des *ROUTE*-Patchs Einzug gefunden hat. Seit Kernel Version `2.6.35` ist das Target auch in den offiziellen Kernelzweig mit aufgenommen worden. Entsprechend kann es bei der Kernelkonfiguration wie folgt als Modul (M) aktiviert und anschließend übersetzt werden:

```
Networking support -- Networking Options -- Network packet filtering
   framework (Netfilter) -- Core Netfilter Configuration -- ''TEE``-
   packet cloning to alternate destination\\
```

Anschließend steht das neue Target `TEE` nach Laden des Kernelmoduls `xt_TEE` in der Firewallkonfiguration zur Verfügung.

Bei der Einstellung des Gateway-Zieles ist es wichtig, dass dieses im *selben* Netzsegment erreichbar ist, also innerhalb von *einem Hop*. Wird eine andere Weiterleitung benötigt, muss dies durch mehrere Stufen erfolgen. Da durch ein zusätzliches Netzinterface ein separater, üblicherweise privater Netzbereich eingerichtet werden kann, ist dies im vorliegenden Szenario jedoch unproblematisch. Eine entsprechende Regel kann wie folgt aussehen:

```
iptables -t mangle -A PREROUTING -i br0 -j TEE --gateway 192.168.10.190
```

Detaillierte Informationen zum Umgang mit dem `TEE`-Target sind in der *manpage* verfügbar.

Abbildung F.33 zeigt die Evaluationsumgebung für produktive Netze. Hierbei werden transparente Brücken in die Netzverbindung integriert, auf welchen *netfilter*-Firewalls installiert sind. Mittels des `TEE`-Targets werden die Daten kopiert und an Instanzen in den Testumgebungen weitergegeben, adressiertes Ziel ist hierbei das `TEE`-Target selbst, wobei verschiedene Auswerteinstanzen dazwischen integriert werden können. Die Datenübertragung zu den jeweiligen Instanzen der Testumgebung erfolgt mittels Datendioden (vgl. z.B. [172]), so dass keine Beeinflussung des produktiven Netzes durch Aktivitäten in den Testumgebungen erfolgen kann. Andererseits ist es einfach möglich, einen zusätzlichen Kanal von einer Testumgebung zu den Brücken bzw. der Firewall zu schalten, so dass Manipulationsmöglichkeiten für das produktive Netz zur Verfügung stehen, falls diese benötigt werden.

Die transparenten Brücken und die Firewall lassen sich auch komplett auf einem System integrieren, indem eine Firewall auf einer transparenten Brücke installiert wird und sämtliche Regeln zum Kopieren des Datenverkehrs vor und nach der Abarbeitung der übrigen Firewallregeln gesetzt werden.

Der Aufbau und die Integrationsweise ermöglichen eine umfangreiche Weitergabe bzw. Verarbeitung der Daten, sowohl auf dem System welches als transparente Brücke fungiert, als auch in Form von dislozierten Komponenten, welche bspw. mittels der unidirektionalen Datenverbindungen versorgt werden. Abbildung F.31 zeigt entsprechende Anbindungsmöglichkeiten.

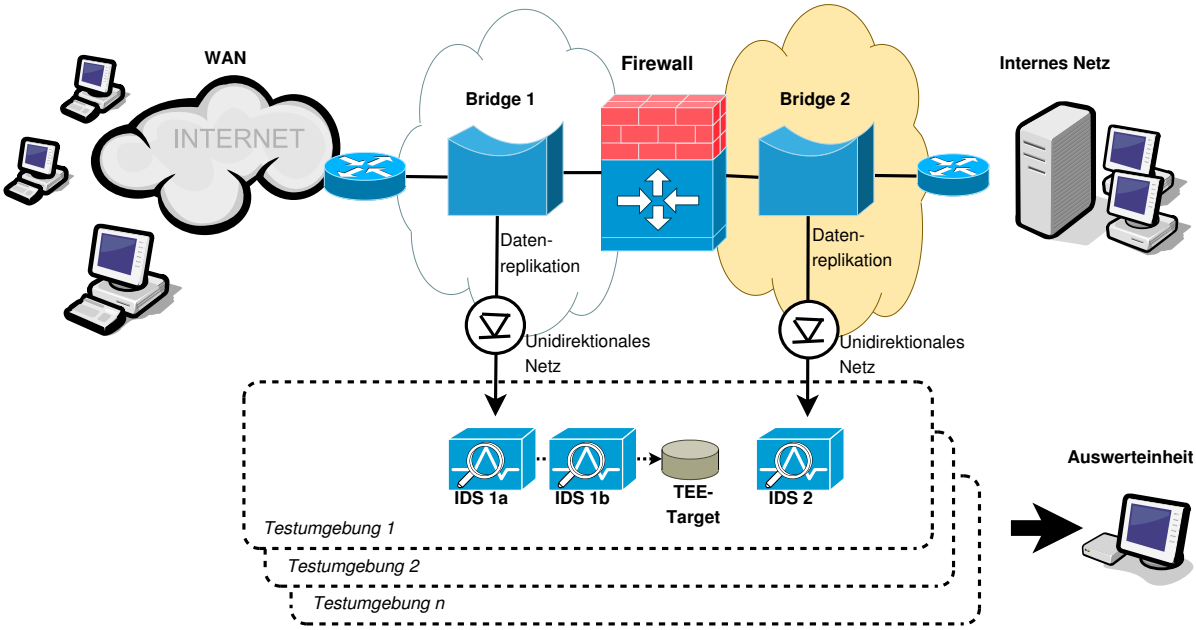


Abbildung F.30: Evaluationsumgebung für produktive Netze. Der Datenverkehr wird mittels transparenter Brücken kopiert und per unidirektionaler Netzverbindungen an die Testumgebungen weitergegeben, in der sich verschiedene IDSs befinden können. Dies ermöglicht einen einfachen Vergleich sowohl bzgl. den Leistungsparameter verschiedener Systeme, als auch der Auswirkung von Änderungen von Konfigurationen eines Systems. Dargestellt ist die Nutzung zweier transparenter Brücken, um somit zum einen den gesamten Datenverkehr, zum anderen nur denjenigen, der die Firewall passiert und im internen Netz transportiert wird, zu analysieren.

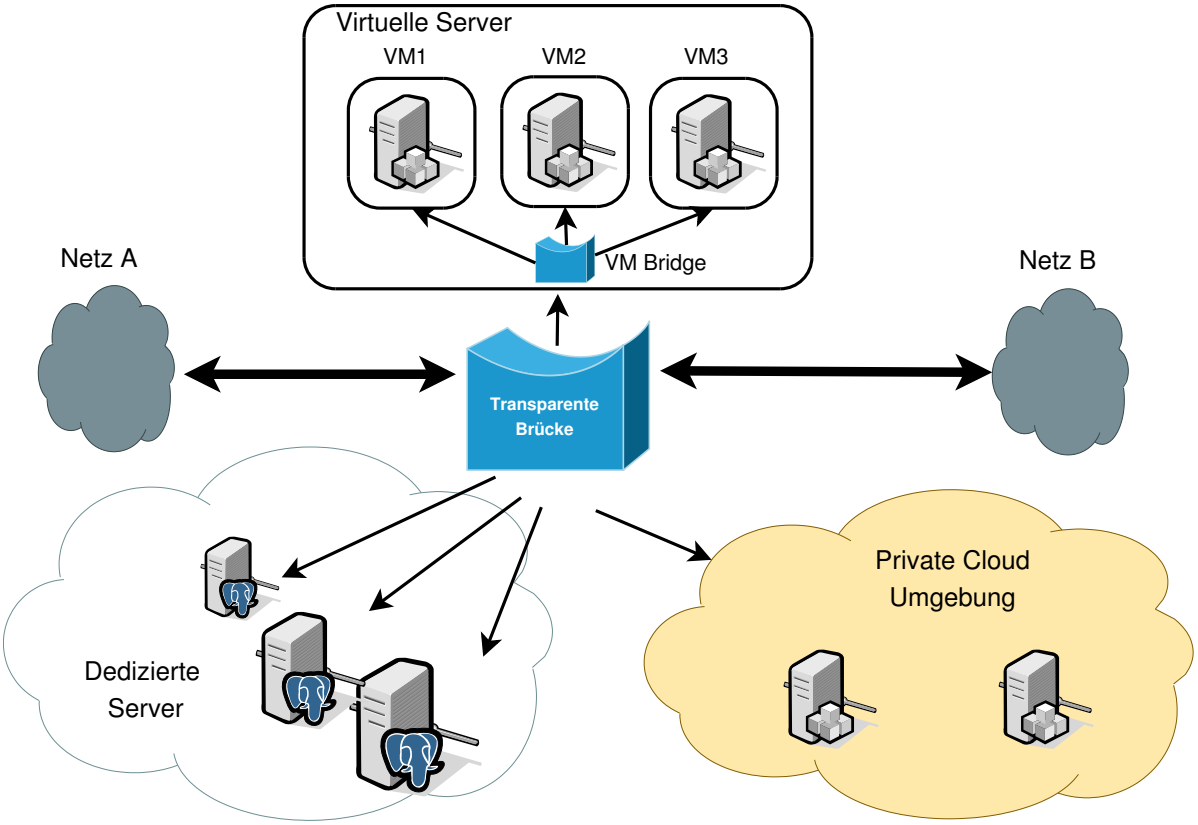


Abbildung F.31: Schnittstellen der Evaluationsumgebung.

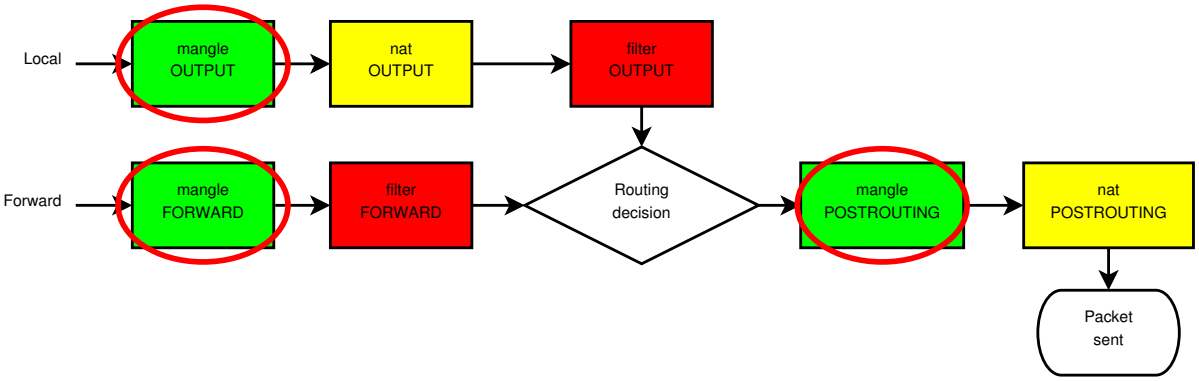


Abbildung F.32: Paketduplizierung mittels der *netfilter*-Firewall. Die Kopien für das `TEE`-Target werden lokal erzeugt und somit in der *OUTPUT*-Queue gezählt. In *POSTROUTING* finde sich sowohl die lokal erzeugten, als auch die weitergeleiteten, ursprünglichen Pakete.

Für die Auswertung ist von besonderer Bedeutung, dass keine Pakete verloren gehen. Um dies zu überprüfen, wurden die Paketzähler der *netfilter*-Firewall über einen längeren Zeitraum aufgezeichnet und anschließend ausgewertet. Abbildung F.33 zeigt eine graphische Darstellung der Entwicklung der Paketzahlen über einen ausgewählten Zeitbereich. Ausgewertet werden hier die *mangle*-Tabellen, wobei *FORWARD*, *OUTPUT* und *POSTROUTING* betrachtet werden müssen. Hierbei gilt, dass die durch die Brücke weitergeleiteten Pakete von *FORWARD* behandelt werden und entsprechend hier gezählt werden, die lokal erzeugten Kopien durch *OUTPUT*. Beim Verlassen der Firewall werden beide dieser, nachdem die Routing-Entscheidung getroffen wurde, durch *POSTROUTING* weiterverarbeitet (vgl. Abbildung F.32). Entsprechend muss der zugehörige Zählwert doppelt so hoch sein, wobei *OUTPUT* und *FORWARD* gleiche Werte aufweisen müssen.

Hierbei sei angemerkt, dass die Zähler regelmäßig keine *exakte* Übereinstimmung haben werden: Insbesondere laufen über den *OUTPUT*-Zähler gerade solche Pakete, welche von dem betreffenden System selbst erzeugt werden; dies sind neben den duplizierten Paketen auch zusätzliche wie bspw. ARP-Anfragen, etc. Der Datenstrom wurde daher sowohl bei der Weiterleitung, als auch beim Ausgang zur Evaluationsumgebung aufgezeichnet und verglichen. Nach Entfernung der genannten, für den Vergleich nicht erwünschten Pakete, konnte die korrekte Duplizierung bestätigt werden.

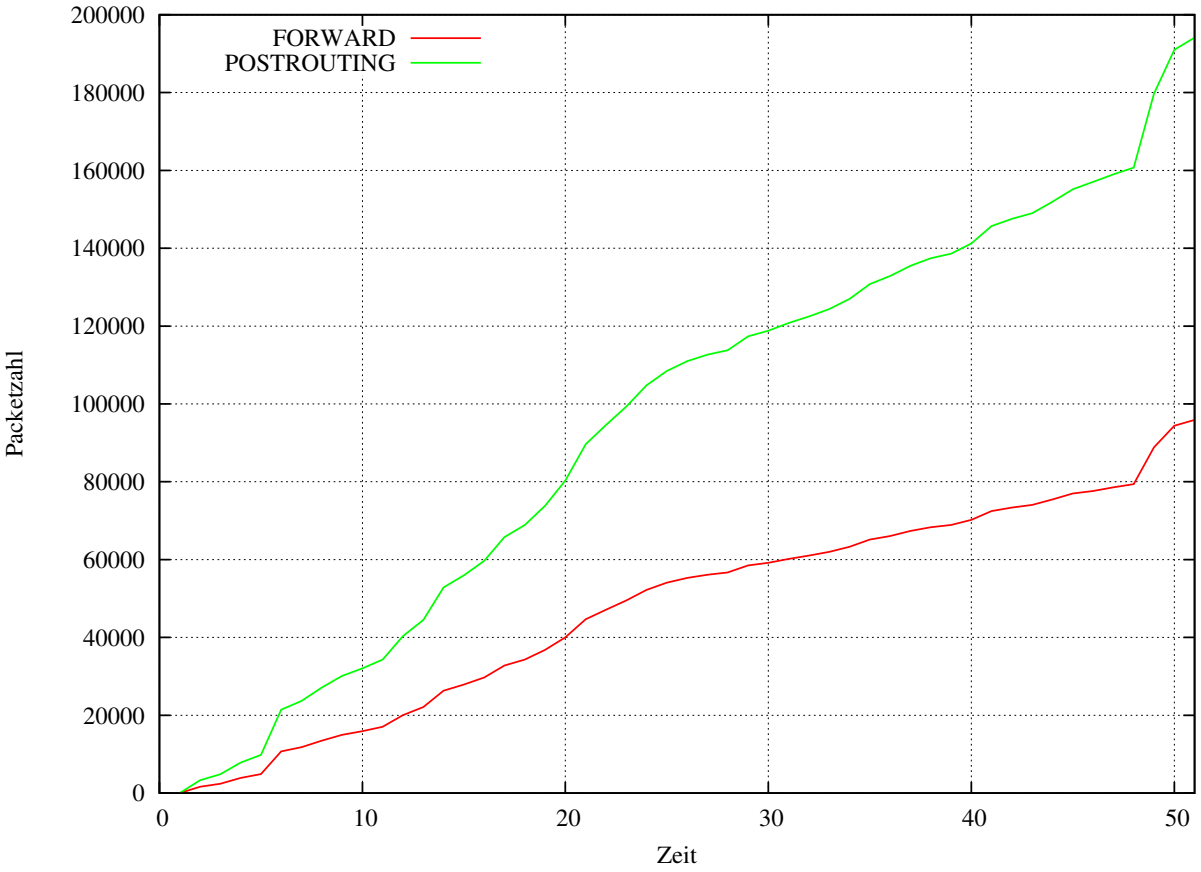


Abbildung F.33: Paketduplizierung mittels TEE-Target. Während eines mehrwöchigen Beobachtungszeitraumes konnte kein Paketverlust festgestellt werden.

F.4.2 Evaluation mittels TEE-Target

Nachfolgend sind einige Ergebnisse aufgeführt, welche während des Vergleichs und der Evaluation von unterschiedlichen IDSs bzw. verschiedenen Konfigurationen gleicher Systeme mittels der Evaluationsumgebung ermittelt wurden. Die transparente Brücke wurde hierbei basierend auf *Ubuntu 10.04.1 Long Term Support (LTS) Server Edition* mit *iptables 1.4.10*, *bridge-utils 1.4-5* und dem *GNU/Linux-Kernel 2.6.37* betrieben.

Zunächst wurden verschiedene Snort-Installationen und Regelsätze über einen Zeitraum von 19 Stunden im Produktivnetz des Institutes verglichen. Hierbei wurden zum einen eine Standardinstallation von *Snort 2.8* sowie *Snort 2.9* genutzt, weiterhin wurde ein *Snort 2.9* mit dem *Emerging Threat* Regelsatz betrieben. Als Grundsystem wurden drei identische Systeme gleicher Hardware und mit gleichem Betriebssystem genutzt, hierbei kamen Intel P4 2.8 GHz mit 2 GB Arbeitsspeicher unter *Ubuntu 10.04.2 LTS* mit *Kernel 2.6.32-30* zum Einsatz. Im einzelnen wurden folgende Systeme genutzt:

- Snort-Standardinstallation aus den Ubuntu-Repositories, *Snort Version 2.8.5.2, Build 121*, sowie Standardregelwerk *2.8.5.2-2build1* mit 3382 Regeln.
- Snort-Installation aus den aktuellen Quellen, *Version 2.9.0.4*, mit den *Sourcefire VRT Certified Rules*, 4438 Regeln.
- Snort-Installation aus den aktuellen Quellen, *Version 2.9.0.4*, mit dem *Emerging Threat Rules* [2], 9692 Regeln.

Abbildung F.34 zeigt den Vergleich der gemeldeten Alarme der jeweiligen Systeme. Während in den Nachtstunden ein relativ gleichmäßiger Verlauf der Alarme bei allen Instanzen erkennbar ist, steigt die Anzahl der Meldungen in den Morgenstunden an.

Insbesondere die Installation mit dem erweiterten *Emerging*-Regelsatz meldet hier eine hohe Anzahl von Vorfällen, während die anderen Systeme einen deutlich geringeren Anstieg verzeichnen. Für eine detaillierte Betrachtung der durch die verschiedenen Systeme gemeldeten Alarme wurde ein Zeitfenster von einer Stunde näher untersucht (vgl. Abbildung F.35).

Die jeweiligen Alarme sind hierbei nach den aufgetretenen Typen zusammengefasst. Gut zu erkennen sind die teils sehr unterschiedlichen Meldungen der Systeme. Während beide Snort 2.9- Installationen gleichermaßen ICMP-Aktivitäten melden, wird ein *Session Initiation Protocol (SIP) message flooding* lediglich von der Snort 2.8 Standardinstallation gemeldet. Andererseits wird eine umfassende *SSH Scan*- Aktivität, welche im untersuchten Zeitfenster vorliegt, nur von der Installation von Snort 2.9 mit *Emerging*-Regeln gemeldet; dies gilt auch für die im selben Kontext durchgeführten *LibSSH Based SSH Connection* Verbindungsversuche. Der Angriff auf SSH-Dienste, welcher im Zeitfenster zu finden ist, wurde entsprechend nur von einem der drei verglichenen Installationen gemeldet. Wie gut erkennbar ist, hängt die Erkennungsleistung eines Systems maßgeblich von dessen Konfiguration und natürlich vom genutzten Regelsatz ab. Die Evaluationsumgebung bietet hier eine einfache Möglichkeit, Systeme zu vergleichen und

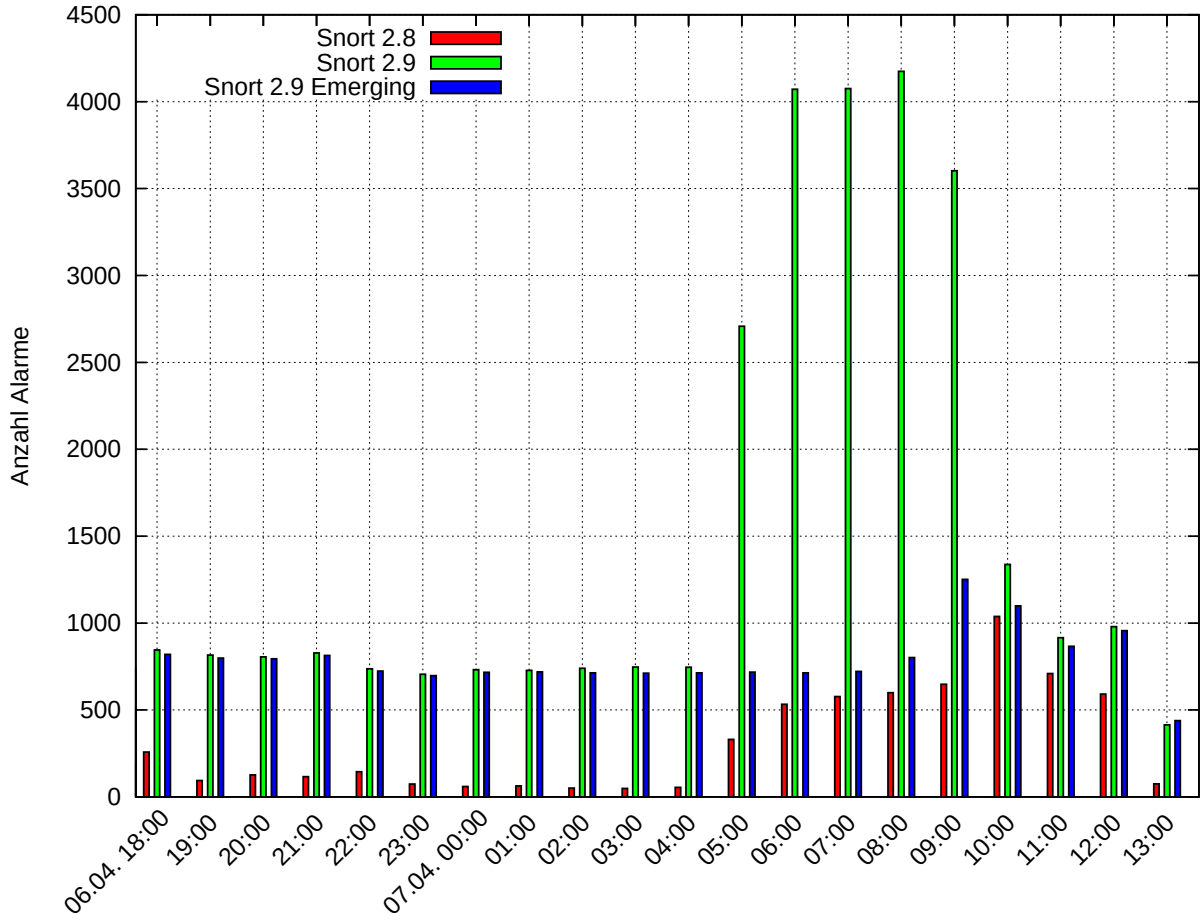


Abbildung F.34: Anzahl der erkannten Angriffe bei der Auswertung durch verschiedene IDSs auf Basis des selben Datensatzes.

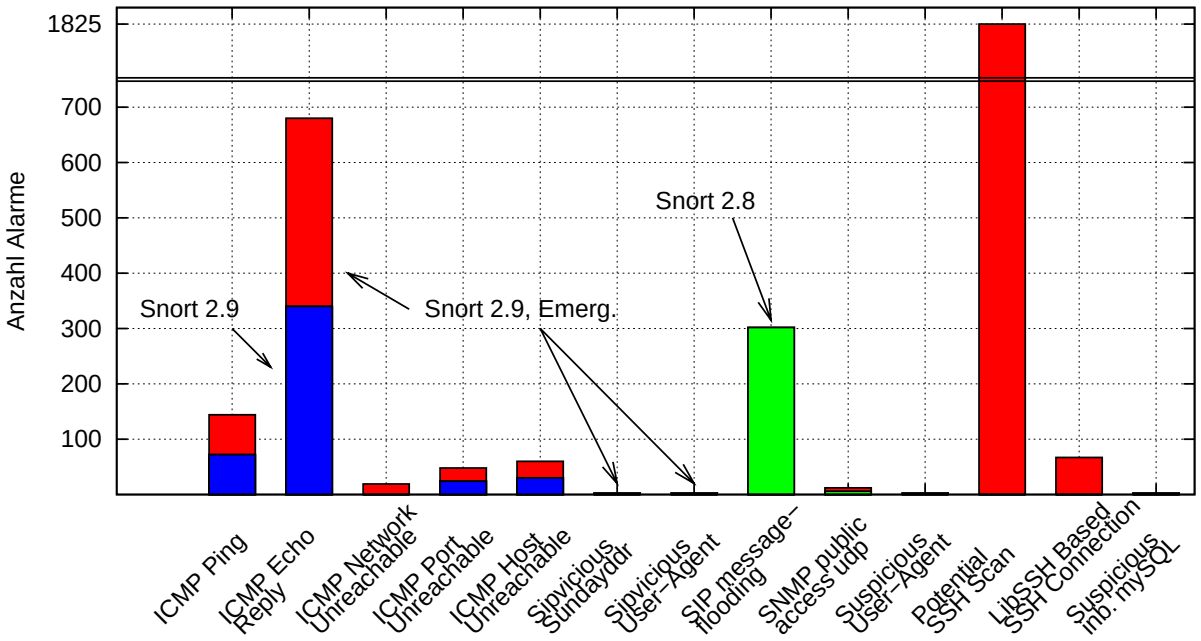


Abbildung F.35: Vergleich der erkannten Angriffstypen bei der Auswertung durch verschiedene IDSs auf Basis des selben Datensatzes.

Konfigurationen zu optimieren, ohne negativen Einfluss auf das Produktivnetz zu riskieren. Gerade die Optimierung von IDSs muss in der produktiven Umgebung erfolgen (vgl. z.B. [297]).

Ein weiterer Aspekt, der mittels der Evaluationsumgebung einfach geprüft werden kann, ist der Ressourcenbedarf der jeweiligen Installationen. Abbildung F.36 zeigt einen Ausschnitt der Ressourcennutzung durch die bereits vorgestellten Installationen. Das 30 Sekunden darstellende Zeitfenster ist repräsentativ für den typischen Ressourcenbedarf der Systeme während durchschnittlichem Lastverhalten auf dem untersuchten 100 Mbps- Link. Gut zu erkennen ist, dass die Installation von Snort 2.9 mit den *Emerging*-Regeln mit Abstand die höchste CPU-Nutzung aufweist, wobei diese auch den größten Regelsatz umfasst. Alle Systeme zeigen eine geringe, durchschnittliche Last auf und auch Spitzenwerte bei den *Emerging*-Regeln gehen lediglich bis 25 Prozent Auslastung auf der genutzten (älteren) P4-Plattform. Somit sind die entsprechenden Systeme ohne Gefahr von unzureichenden Kapazitäten in der geprüften Produktivumgebung einsetzbar.

F.4.3 Modul zur Befehlsevaluation

Nachfolgend ist beispielhaft eine Ausgabe der prototypischen Implementierung der Befehlsevaluation dargestellt.

```
Position 029 to 031, searching clusters...
Next cluster is 022 - 029 (delta is 039, sizes 48, 872), sorting
   packets...
Looking for std.-cmds, cmd 00, length out 07, length in 01 (found 08)
```

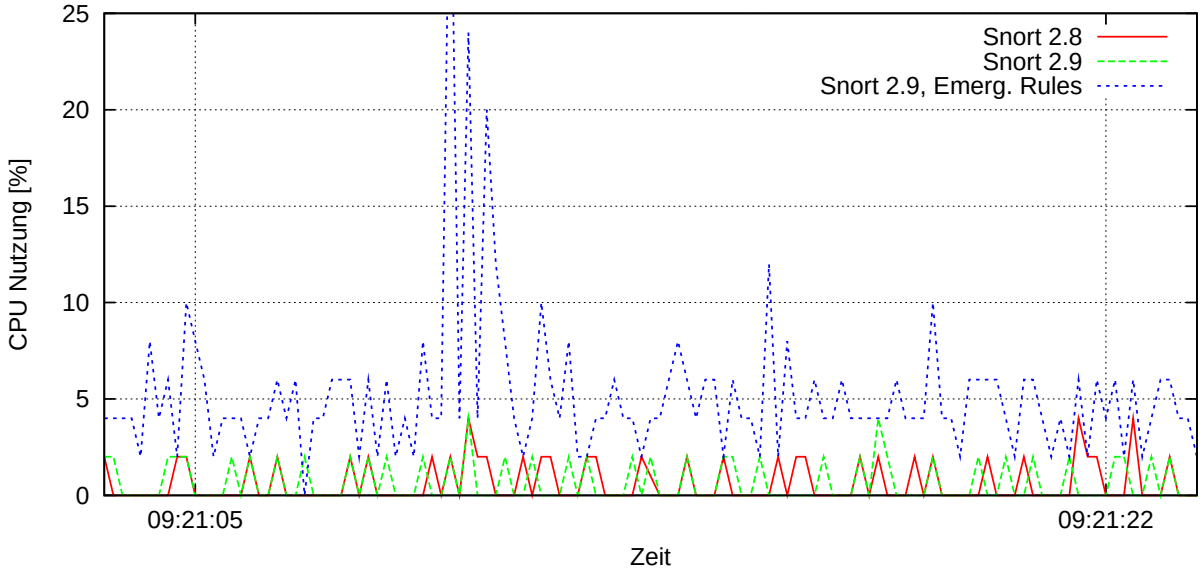


Abbildung F.36: Ressourcen-Nutzung verschiedener IDSs auf Basis des selben Datensatzes.

```
Reference: 48 48 48 48 48 48 48 96
Looking for std.-cmds, cmd 01, length out 05, length in 02 (found 08)
Reference: 48 48 48 48 48 1448 08
Looking for std.-cmds, cmd 02, length out 03, length in 01 (found 08)
Reference: 48 48 48 176
Looking for std.-cmds, cmd 03, length out 11, length in 10 (found 08)
Reference: 48 48 48 48 48 48 48 48 48 48 48 64 1448 1448 1248 1448 1448
    112 1448 1448 1248
Looking for std.-cmds, cmd 04, length out 08, length in 04 (found 08)
Reference: 48 48 48 48 48 48 48 48 1448 1448 1248 1376
Looking for std.-cmds, cmd 05, length out 07, length in 01 (found 08)
Reference: 48 48 48 48 48 48 48 160
Looking for std.-cmds, cmd 06, length out 06, length in 01 (found 08)
Reference: 48 48 48 48 48 48 96
Client ok, continuing...
Server-Answer to short, cutting to 7. Forces penalty.
Record: 48 48 48 48 48 48 1448
Correlation (06): 0.693
Looking for std.-cmds, cmd 07, length out 09, length in 01 (found 08)
Reference: 48 48 48 48 48 48 48 48 48 192
Looking for std.-cmds, cmd 08, length out 06, length in 08 (found 08)
Reference: 48 48 48 48 48 48 80 1152 1448 376 1448 472 464 80
Client ok, continuing...
Record: 48 48 48 48 48 48 1448 872
Correlation (08): 0.577
Looking for std.-cmds, cmd 09, length out 03, length in 02 (found 08)
Reference: 48 48 48 576 80
Looking for std.-cmds, cmd 10, length out 06, length in 01 (found 08)
Reference: 48 48 48 48 48 48 768
```

```
Client ok, continuing...
Server-Answer to short, cutting to 7. Forces penalty.
Record: 48 48 48 48 48 48 1448
Correlation (10): 0.997
Checking extended commands...
Checking answer 00...
Reference: 48 48 48 48 48 48 1448 216
Client ok, continuing...
Record: 48 48 48 48 48 48 1448 872
Correlation (11/00): 0.924
Checking answer 01...
Reference: 48 48 48 48 48 48 1448 1448 1248 1448 1448 1448 568 1448
   1448 1448 1448 1448 1448 848
Client ok, continuing...
Record: 48 48 48 48 48 48 1448 872
Correlation (11/01): 0.971
Position 030 to 031, searching clusters...
```

G Lebenslauf des Autors

Robert Koch wurde am 20. Juni 1979 in Rothenburg ob der Tauber geboren. Von 1999 bis 2002 studierte er Informatik an der Universität der Bundeswehr München mit Vertiefung im Bereich der Technischen Informatik. Im Rahmen seiner Tätigkeit als Führungsmitteltechnikoffizier war er für die Verfügbarkeit und Einsatzbereitschaft sämtlicher operativer IT, Funk- und Kommunikationsanlagen sowie Navigations- und Radarsysteme verantwortlich. Hierbei erhielt er auch die Ausbildung zum IT-Sicherheitsbeauftragten der Bundeswehr.

2008 ging Robert Koch als wissenschaftlicher Mitarbeiter an den Lehrstuhl für Kommunikationssysteme und Internet-Dienste von Prof. Dr. Gabi Dreo Rodosek im Institut für Technische Informatik der Universität der Bundeswehr München. Seine Hauptinteressen liegen in den Forschungsgebieten der Netzsicherheit mit den Schwerpunkten in der Ein- und Ausbruchserkennung, Visualisierung und der Anwendung von Verfahren der künstlichen Intelligenz. Robert Koch ist Mitglied in der ENISA Expertengruppe für proaktive Erkennung von Netzsicherheitsvorfällen (Expert group on "Proactive Detection of Network Security Incidents“).

H Veröffentlichungen im Rahmen der Dissertation

Auflistung der im Rahmen der Dissertation entstandenen Veröffentlichungen:

- Koch, R., Dreo, G., Fast Learning Neural Network Intrusion Detection System. Proceedings of the 3rd International Conference on Autonomous Infrastructure, Management and Security: Scalability of Networks and Services, Springer-Verlag, 2009
- Koch, R., Changing Network Behavior. Proceedings of the 3rd International Conference on Network and System Security (NSS), IEEE, 2009
- Stelte, B., Koch, R., Absicherung von Xen-basierten Virtualisierungen – Selbstschutz durch den Einsatz von Sensoragenten. 17. DFN CERT Workshop Sicherheit in vernetzten Systemen, C. Paulsen (Ed.), 2010
- Koch, R., Dreo, G., Command Evaluation in Encrypted Remote Sessions. Proceedings of the 4th International Conference on Network and System Security (NSS), IEEE, 2010
- Kretzschmar, M., Stelte, B., Koch, R., Cyber Defence in Future Communication Networks – A Multilayer Security Architecture. Poster at 6th Security Research Conference (Future Security 2010), 2010
- Koch, R., Dreo, G., Security System for Encrypted Environments (S2E2). Proceedings of the 13th International Conference on Recent Advances in Intrusion Detection (RAID), Springer Verlag, 2010
- Koch, R., Dreo, G., User Identification in Encrypted Network Communications. International Conference on Network and Service Management (CNSM), IEEE, 2010
- Stelte, B., Koch, R., Ullmann, M., Towards integrity measurement in virtualized environments - A hypervisor based sensory integrity measurement architecture (SIMA). International Conference on Technologies for Homeland Security (HST), IEEE, 2010
- Stelte, B., Koch, R., Bot-Netz ohne Fritz - Ein Frühwarn- und Abwehrsystem für ISPs basierend auf in DSL-Routern platzierten Sensoren. Sicherheit in vernetzten Systemen, 18. DFN Workshop, 2011

- Koch, R., Towards Next-Generation Intrusion Detection. Proceedings of the 3rd International Conference on Cyber Conflict (ICCC), IEEE, 2011
- Koch, R., Holzapfel, D., Dreo, G., Data Control in Social Networks. Proceedings of the 5th International Conference on Network and System Security (NSS), IEEE, 2011