

Ausgewählte Themen der IT-Sicherheit

Prof. Dr. Gabi Dreo
Mario Golling
Björn Stelte
(Hrsg.)

Institut für Technische Informatik

Bericht 2012-02
September 2012

Informations- und Kommunikationstechnologie (IKT) ist heute in allen Bereichen von zentraler Bedeutung: Gesundheit, Mobilität, Bildung, Unterhaltung, Produktion, Logistik, Handel, Finanzen oder auch Versorgung (z. B. Energie, Wasser) und der öffentlichen Verwaltung. IKT ist der Innovationstreiber für eine Zukunft der digitalen Gesellschaft (z.B. Smart Grids, Smart Meter, Internet of Things). Bereits heute entsteht ein großes Wertschöpfungspotenzial durch die Verlagerung von Geschäftsprozessen und Dienstleistungen in das Internet; dies wird in Zukunft unser soziales, gesellschaftliches und berufliches Leben gravierend beeinflussen.

Der Grad der Abhängigkeit moderner Industriestaaten, im öffentlichen wie privaten Sektor, von IKT hat eine Dimension erreicht, welche vor einigen Jahren nicht denkbar war. Gerade Informationssicherheit und Verlässlichkeit als Basis dieses Katalysators wirtschaftlicher Entwicklung sind in der vernetzten Welt von heute eine kritische Herausforderung für alle Unternehmen, öffentlichen Dienstleister und Privatpersonen, ja sogar ganzer Staaten. Ihre Gewährleistung ist in Zeiten zunehmender Wirtschaftsspionage und Sabotage von vernetzten Komponenten mit milliardenschwerem Business-Impact zu einem entscheidenden wirtschaftlichen Erfolgsfaktor geworden. Gleichzeitig entwickelt sich die Bedrohung kritischer Infra- und Kommunikationsstrukturen für Staaten von einer abstrakten zu einer konkreten Gefahr (Cyber War), wie die KRITIS-Studie des Bundesministerium des Innern (BMI)¹ veranschaulicht und STUXNET² in der Praxis eindrucksvoll demonstriert hat.

Während in der Vergangenheit Massenwürmer wie SASSER oder CONFICKER die IT-Systeme bedroht haben, zeigen die Lehren von STUXNET, dass der Trend zu immer gezielteren, skalpellartigen Angriffen mit hohem Entwicklungsaufwand und überaus großem Schadenspotential geht. Cyber Angriffe sind in diesem Umfeld immer schwerer zu erkennen und abzuwehren, da die inhärent hohe Komplexität der modernen IT-Landschaft, die Anzahl der Systeme und die dynamischen wechselseitigen Abhängigkeiten die natürlichen "Gegner" der IT-Sicherheit sind. Cyber Defence, die Verhinderung und Abwehr von Angriffen in der virtuellen Welt, ist ein Schlagwort unter dem verschiedenste Ansätze zu verstehen sind, um Bedrohungen mit Mitteln der Informationstechnik entgegen zu wirken.

Vor diesem Hintergrund fand im Frühjahr 2012 an der Universität der Bundeswehr ein Seminar über ausgewählte Kapitel der IT-Sicherheit statt. Das Ziel war es, einen fundierten Überblick über speziell ausgewählte Aspekte als Basis für weitere Forschungsarbeiten in diesem Bereich zu schaffen und gleichzeitig den Studenten der Vertiefungsrichtung Cyber Defense weitere Hintergrundinformationen zu den einzelnen Kapiteln der Vorlesungen an die Hand zu geben.

Wir wünschen eine interessante und aufschlußreiche Lektüre.

¹Bundesministerium des Inneren, "Nationale Strategie zum Schutz Kritischer Infrastrukturen (KRITIS-Strategie)", 2009, <http://www.bmi.bund.de/cae/servlet/contentblob/598730/publicationFile/34416/kritis.pdf>.

²Falliere, N., Murchu, L.O. and Chien, E., "W32. stuxnet dossier", White paper, Symantec Corp., Security Response, 2011, http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf.

Inhaltsverzeichnis

1	Sicherheit mobiler Geräte - Schutzmaßnahmen, Angriffsarten & Angriffserkennung auf Android	7
	<i>Julian Scheid</i>	
2	IPv6-Sicherheit	33
	<i>Stefan Schiller</i>	
3	IT-Sicherheitsmanagement	61
	<i>Nico Hamann</i>	
4	Einblick in den Aufbau und Analysemöglichkeiten von Speichermedien und Standarddateisystemen vor einem forensischen Hintergrund	83
	<i>Sandy-Dorothea Hein</i>	
5	Sicherheit Sozialer Netzwerke	105
	<i>Julian Petery</i>	
6	Sicherheit von Smart Grids	117
	<i>Johann Dreßler</i>	
7	Analyse von Botnetzen	131
	<i>Dominik Holzapfel</i>	
8	Multicast-Encryption	147
	<i>Christian Marciniak</i>	

9 Social VPN: Vergleich verschiedener VPN-P2P Ansätze 165*Marcel Bassüner***Acronyms 181**

Kapitel 1

Sicherheit mobiler Geräte - Schutzmaßnahmen, Angriffsarten & Angriffserkennung auf Android

Julian Scheid

Dieses Kapitel widmet sich der Sicherheit mobiler Geräte. Dabei wird im Speziellen ein Blick auf die Schutzmaßnahmen des führenden Betriebssystems Android geworfen. Des Weiteren werden Angriffsmöglichkeiten auf Android erläutert. Abschließend werden verschiedene Möglichkeiten der Erkennung von Angriffen dargestellt.

Inhaltsverzeichnis

1.1	Einleitung	9
1.2	Android Schutzmaßnahmen	9
1.2.1	Android Plattform & Updates	9
1.2.2	Application Stores	12
1.2.3	Application Signing	14
1.2.4	Application Sandbox	14
1.2.5	Permission Model	16
1.2.6	Dalvik Virtual Machine	17
1.3	Angriffsmöglichkeiten	18
1.3.1	Android Plattform & Updates	18
1.3.2	Application Stores	19
1.3.3	Permission Model	19
1.3.4	Rooting	21
1.4	Angriffserkennung	22
1.4.1	Intrusion Detection Systeme	22
1.4.2	Intrusion Response	24
1.4.3	Wissensbasierte Erkennung	24
1.4.4	Verhaltensbasierte Erkennung	25
1.4.5	Honeypots	26
1.4.6	Cloudbasierte Erkennung	27
1.5	Fazit	27

1.1 Einleitung

Das Mooresche Gesetz besagt, dass sich die Transistordichte auf integrierten Schaltkreisen ungefähr alle 2 Jahre verdoppelt [1]. Dadurch entstehen im Rahmen der technischen Weiterentwicklung immer mehr Möglichkeiten. Der Trend geht zu mobilen Geräten - zum Beispiel Smartphones, Tablets und Notebooks. Laut einer Pressemitteilung von Gartner wurden 2011 weltweit 1,8 Milliarden mobile Geräte verkauft - darunter 472 Millionen Smartphones [2]. Aufgrund der Ressourcenlimitierung¹ von mobilen Geräten werden spezielle beziehungsweise speziell angepasste Betriebssysteme benötigt. In einem Pressebericht von IDC heißt es, dass im ersten Quartal 2012 weltweit insgesamt 152,3 Millionen Smartphones verkauft wurden, wovon 59,0% mit Android und 23,0% mit Apple iOS betrieben werden [3]. Allein in den Jahren 2011 bis 2012 ist die Anzahl der Mobilgeräte-Malware-Varianten gemäß McAfee von unter 2000 auf über 8000 angestiegen [4].

Diese Arbeit beinhaltet Schutzmaßnahmen die Android implementiert, auf welche in Kapitel 1.2 näher eingegangen wird, von Angriffsmöglichkeiten auf Android, die in Kapitel 1.3 erläutert werden und zuletzt Möglichkeiten der Angriffserkennung in Kapitel 1.4.

1.2 Android Schutzmaßnahmen

In diesem Kapitel werden die wichtigsten Schutzmaßnahmen von Android näher erläutert. Im Kapitel 1.2.1 wird erläutert auf welches Betriebssystem sich Android abstützt und wie Android-Updates für den Nutzer bereitgestellt werden. Danach werden dem Leser im Kapitel 1.2.2 Informationen über die verschiedenen Application Stores vermittelt und erläutert welche Sicherheitsmaßnahmen im Play Store² von Google implementiert sind, um den Nutzer besser vor Schadsoftware schützen zu können. Damit eine Anwendung auf Android installiert werden kann, muss sie signiert werden. Näheres dazu im Kapitel 1.2.3. Bei der Installation einer Anwendung wird diese in der Sandbox platziert. Wie das vom Prinzip her funktioniert und wie der Schutz an dieser Stelle erhöht werden kann, wird im Kapitel 1.2.4 näher erläutert. In Android muss der Entwickler Zugriffsrechte für seine Anwendung definieren, welche seiner Anwendung gewährt werden sollen und bietet dem Nutzer somit eine gewisse Kontrollmöglichkeit. Wie das im Detail umgesetzt wird, ist im Kapitel 1.2.5 erläutert. Die fertige Anwendung eines Entwicklers wird dann letztlich größtenteils auf der Dalvik Virtual Machine ausgeführt, welche im Kapitel 1.2.6 erklärt wird.

1.2.1 Android Plattform & Updates

Die Android Plattform wird von der Open Handset Alliance unter Führung von Google seit Ende 2007 entwickelt und ist seit Ende 2008 Open Source, was bedeutet, dass jeder

¹dies sind unter anderem geringe CPU-Leistung, niedrige Batterieleistung, geringe Speicherkapazität, etc. sein

²<https://play.google.com/store> (Aufruf Juni 2012)

sich den Code herunterladen und diesen laufen lassen kann, wodurch ein mobiles Gerät betrieben werden kann[5]. Ziel der Open Handset Alliance - einer Organisation aus 84 Firmen - ist es eine offene und leicht verständliche Plattform für mobile Geräte zu entwickeln, damit neue Produkte schneller und kostengünstiger auf den Markt gebracht werden können [6].



Abbildung 1.1: Android Architecture [7]

In Abbildung 1.1 ist die System Architektur von Android dargestellt. Sie ist untergliedert in verschiedene Schichten - Linux Kernel, Libraries, Android Runtime, das Application Framework und Applications.

Die Android Plattform basiert gegenwärtig auf dem *Linux Kernel* in Version 2.6 und nutzt als Basis dessen Möglichkeiten wie zum Beispiel verschiedene Schutzmechanismen, Speichermanagement, Prozessmanagement und das Treibermodell. Ein weiterer Vorteil des Linux Kernels ist, dass er als Abstraktion zwischen Hardware und Software dient. Dadurch kann Android auf einer Vielzahl von Geräten eingesetzt werden, sofern das System entsprechend angepasst wird. Android unterstützt eine Reihe von Prozessoren und nutzt deren Vorteile bezüglich Schutz - zum Beispiel die des ARM (Advanced Risk Machines). Dabei wird beispielsweise das XN-Bit (eXecute-Never-Bit) im Prozessor benutzt, um zu deklarieren, dass ein Speicherbereich nicht ausführbaren Code enthält. Soll also der Code aus diesem Speicherbereich ausgeführt werden und ist das XN-Bit gesetzt, so wird ein Fehler ausgegeben [8]. Auf Geräte-Ressourcen wie Kamera, GPS, Bluetooth, usw. kann über

das Betriebssystem zugegriffen werden [9]. Da Linux in sehr vielen sicherheitsrelevanten Systemen verwendet wird und in der Vergangenheit ständig untersucht und angegriffen wurde, ist Linux mittlerweile sehr stabil und viele Sicherheitsexperten vertrauen dem Linux Kernel [10]. Deswegen greift Android auf viele Schutzmechanismen des Linux Kernels zurück beziehungsweise nutzt diesen als Basis. Dabei geht jede Komponente davon aus, dass alle Komponenten, die sie benutzt, ausreichend gesichert sind. Nur ein kleiner Teil des Systems wird mit Root-Rechten betrieben. Bezüglich der Architektur, welche in Abbildung 1.1 dargestellt ist, läuft der Code, der auf dem Linux Kernel aufsetzt innerhalb der Application Sandbox (siehe Kapitel 1.2.4) und wird durch diese eingeschränkt [9].

Android stellt von Haus aus gewisse *Bibliotheken* (engl. Libraries) zur Verfügung. Unter anderem Bibliotheken für Medien (Audio-, Video- und Bildformate), eine Webbrowser-Engine (WebKit³), 3D Bibliotheken, eine Bibliothek für Relationale Datenbanken (SQLite⁴) und noch einige andere.

Die *Android Runtime* besteht aus den Kernbibliotheken, welche den größten Teil der Funktionalität der Programmiersprache Java bereitstellen, und der Dalvik Virtual Machine (im Detail erklärt im Kapitel 1.2.6)

Das *Application Framework* gibt dem Entwickler die Möglichkeit sehr innovative Applikationen zu entwickeln, da der Entwickler die Möglichkeit hat auf die gleichen Framework APIs zuzugreifen, wie die Kernapplikationen, allerdings wird die Applikation dabei gewissen Schutzmaßnahmen unterworfen (siehe Kapitel 1.2.5).

Android liefert von Haus aus viele *Applikationen* - zum Beispiel ein E-Mail und SMS Programm, einen Kalender und viele andere. Alle diese Applikationen sind in Java geschrieben und werden in der Dalvik Virtual Machine betrieben (weiter erklärt im Kapitel 1.2.6).

Im Folgenden wird noch auf die Update-Problematik eingegangen.

In Abbildung 1.2 wird verdeutlicht, wie die verschiedenen Android Versionen im Moment⁵ verteilt sind. Android 2.x hat in der Summe 89,3%.

In Abbildung 1.3 ist zu sehen, wie die einzelnen Android Versionen die jeweils vorherigen verdrängen. Hierbei sind die älteren Versionen weiter oben, die Neueren weiter unten angesiedelt. Es wird deutlich, dass einige Hersteller im Moment dabei sind auf die neueren Versionen von Android umzustellen. Angesichts der Tatsache, dass die Version 4.x von Android schon seit Ende 2011 [12] auf dem Markt verfügbar ist, ist die Verbreitung mit nur 7,1% doch extrem niedrig. Doch woran liegt dies?

Wird von Google eine neue Version veröffentlicht, so kann diese noch nicht auf jedem mobilen Gerät installiert werden, da Android nicht von Haus aus auf jedem Gerät lauffähig ist. Die Handyhersteller überlegen sich welche Geräte das neue Update erhalten und wann dies erfolgen soll. Das kostet natürlich Zeit. Die Plattform (Linux Kernel) muss nun angepasst werden. Anpassungen müssen für alle Geräte-Ressourcen des Gerät durchgeführt

³<http://www.webkit.org/> (Aufruf Juni 2012)

⁴<http://www.sqlite.org/> (Aufruf Juni 2012)

⁵Juni 2012 - die Daten wurden 14 Tage lang, beginnend mit 1. Juni 2012, aufgenommen

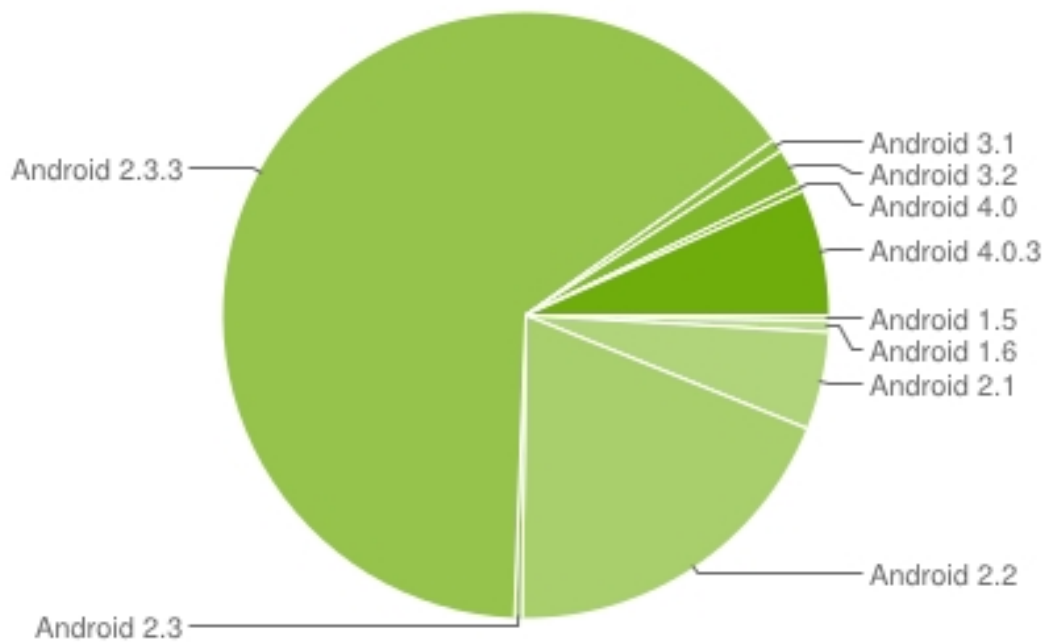


Abbildung 1.2: Android Plattform Versionen [11]

beziehungsweise überprüft werden, da die Treiber für diese Geräte-Ressourcen abhängig von der jeweiligen Hardware des Smartphones sind. Dies umfasst zum Beispiel Anpassungen für Bluetooth, WLAN, Kamera, GPS, usw. [7]. Sind die Anpassungen geschrieben, so müssen diese erst getestet werden, um deren korrekte Funktion zu gewährleisten. Dieses Testen kann einige Monate dauern - natürlich nur weil die Hersteller die Qualität für ihre Kunden möglichst hoch halten möchten. Doch oftmals ist es das primäre Ziel der Hersteller neue Geräte zu verkaufen und weniger die Alten noch lange aktuell zu halten. Ein Nutzer, der auf dem Stand der Technik bleiben möchte, ist gezwungen sich ein neues Gerät zu kaufen. Hinzu kommt, dass es einen Gerätehersteller viel Geld kostet, eine neue Version bereitzustellen. Sind die Anpassungen für die ausgewählten Geräte geschrieben und ausreichend getestet, so erhalten die Mobilfunkbetreiber die angepasste Version. Diese schreiben nun noch Anpassungen und gesonderte Anwendungen, die für ihre spezielle Version des Gerätes zur Verfügung stehen sollen (Branding) und testen das Gerät für ihr Netzwerk. Sind alle Tests abgeschlossen, so wird das Update für die ausgewählten Geräte bereitgestellt [13].

1.2.2 Application Stores

Im Gegensatz zu iOS (Apple), bei dem der Nutzer an den hauseigenen Apple App Store gebunden ist, stehen bei Android mehrere Application Stores zur Auswahl. Bei Android existieren viele Stores. Häufig stellen Mobilfunk Anbieter auf ihren gebrandeten Geräten auch einen eigenen Application Store bereit. Der meist genutzte Application Store, mit einer Milliarde App Downloads pro Monat, dürfte jedoch Google Play⁶ sein [14]. Ein

⁶<https://play.google.com/store> (Aufruf Juni 2012)

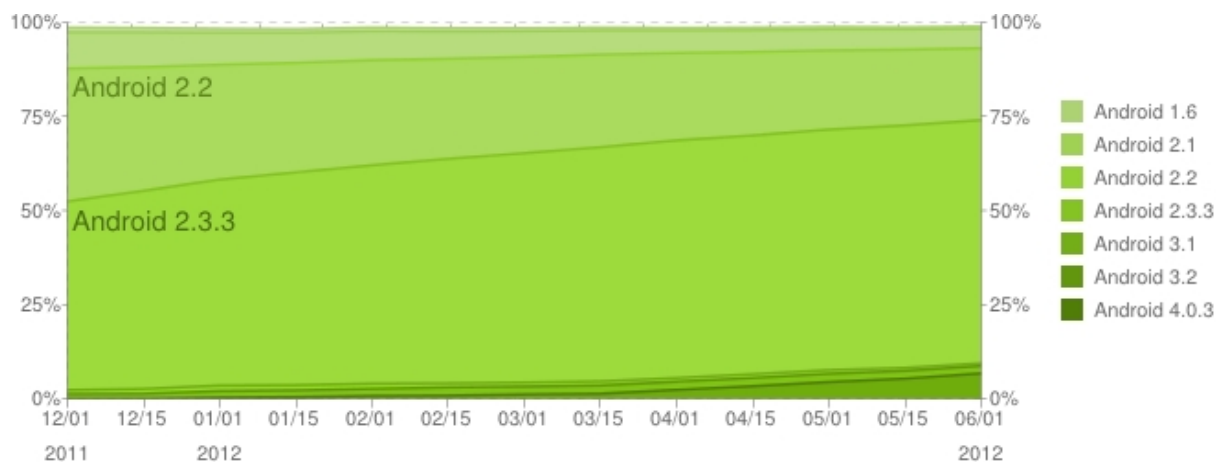


Abbildung 1.3: Android Plattform Versionen - Entwicklung [11]

Application Store dient einerseits dem Nutzer als Hauptquelle für den Bezug von Applikationen und andererseits dem Entwickler dazu seine Applikationen zu vertreiben. Um Applikationen auf Google Play veröffentlichen zu können, bedarf es einer vorherigen Registrierung (inklusive der Erstellung eines Profils, der Zahlung einer Registrierungsgebühr sowie der Akzeptanz der Lizenz von Google) [15].

2011 wurden mehr als 11 Milliarden Applikationen von dem Google Play heruntergeladen. Da die Plattform immer weiter wächst, möchte Google die besten und neuesten Innovationen auf den Markt bringen, welche unter anderem auch die Sicherheit erhöhen sollen. Eine davon ist ein Dienst namens Bouncer. Dieser soll Applikationen automatisiert scannen, um zu prüfen, ob diese schädlich sind. Dazu werden verschiedene Analysen gemacht, die sich auf die Applikation und den Entwickler-Account beziehen. Der Dienst startet, wenn eine neue Applikation hochgeladen werden soll. Diese wird analysiert, um herauszufinden, ob es sich eventuell um bekannte Malware handelt. Prinzipiell funktioniert dies wie bei einem Virens scanner (siehe Kapitel 1.4.3). Zusätzlich wird das Verhalten der Anwendung analysiert, indem diese in Googles Infrastruktur gestartet und simuliert wird. Bei der Simulation wird versucht herauszufinden, wie sich die Applikation auf einem Android System verhält, um eventuelles Fehlverhalten zu erkennen. Zusätzlich werden Vergleiche zu bereits analysierten Applikationen gezogen. Der Account des Entwicklers wird untersucht, um festzustellen, ob dieser Entwickler schon einmal unter einem anderen Account Schadsoftware vertrieben hat. Der Dienst ist durchaus erfolgreich. Zwischen der ersten und zweiten Hälfte 2011 ist die Anzahl potenziell gefährlicher Downloads um 40% zurückgegangen [16].

Android bietet die Möglichkeit, Applikationen über eine Remote-Verbindung zu löschen und zu installieren. Dies geschieht über eine Applikation namens GTalkService. Android erstellt dazu eine TCP / SSL / XMPP⁷ Verbindung zu Googles GTalk Server. Über diese Verbindung werden sogenannte "heartbeat messages" gesendet und Google ist in der Lage, Nachrichten an das Gerät zu schicken. Wenn eine Applikation vom Market heruntergeladen werden soll, so sendet Googles Server eine INSTALL_ASSET Nachricht an das Gerät

⁷XMPP heißt Extensible Messaging and Presence Protocol - siehe RFC 6120 <http://tools.ietf.org/html/rfc6120>, März 2011 (Aufruf Juni 2012)

und veranlasst das Gerät dadurch, eine Applikation herunterzuladen. REMOVE_ASSET bewirkt das Gegenteil. Es bringt das Gerät dazu eine Applikation zu deinstallieren. Mit Hilfe dieses Features hat Google die Möglichkeit, Schadsoftware zu löschen, was die Sicherheit erhöhen soll (siehe Kapitel 1.3.2) [17].

1.2.3 Application Signing

Unter Android werden Applikationen vom Entwickler signiert. Sie müssen nicht durch eine Zertifizierungsstelle (Certificate Authority) signiert werden. Das ist legitim, da das Zertifikat in Android dazu dient, die verschiedenen Entwickler voneinander abzugrenzen und nicht dazu, die Identität des Entwicklers zu offenbaren [18]. Deswegen muss auch jede Applikation ein Zertifikat besitzen. Wird versucht eine Applikation zu installieren, die kein Zertifikat besitzt, so wird diese vom Package Installer verworfen. Besitzt sie ein Zertifikat, so wird dieses benutzt, um die Applikation in der Application Sandbox (siehe Kapitel 1.2.4) zu platzieren, indem der Applikation anhand des Zertifikats eine User ID zugeordnet wird [9]. Setzt ein Entwickler, im Manifest seiner Android Applikationen, den gleichen Wert für die "sharedUserId", so wird diesen eine User ID vom System zugeordnet, sofern die Anwendungen das gleiche Zertifikat besitzen. Applikationen, die die gleiche User ID besitzen, können auf die gleichen Daten zugreifen und sie teilen ihre Berechtigungen (siehe Kapitel 1.2.5). Sofern der Entwickler es wünscht, können diese Anwendungen sogar im gleichen Prozess ausgeführt werden [19] [20].

1.2.4 Application Sandbox

Linux ist ein Mehrbenutzer-Betriebssystem. Aufgabe eines solchen Systems ist es unter anderem Ressourcen eines Nutzers vor einem anderen zu schützen. Damit bietet es Android eine gute Möglichkeit, das Sandboxing auf Basis des Nutzermanagements, also auf Ebene des Linux Kernels, zu implementieren. So kann garantiert werden, dass selbst nativer Code in der Sandbox ausgeführt und durch diese eingeschränkt wird. Dadurch soll verhindert werden, dass eine Anwendung einer anderen Anwendung, dem Android System oder dem Gerät Schaden zufügen kann.

Android vergibt für jede Applikation eine Linux User ID (UID) (siehe auch Kapitel 1.2.3) und führt diese in einem separaten Prozess unter der User ID aus. Dadurch läuft jede Anwendung mit eigenen Nutzer-Rechten, wodurch der Schutz zwischen Anwendungen und dem System auf Prozess-Ebene beziehungsweise Kernel-Ebene geregelt wird. Somit kann eine Anwendung standardmäßig nicht auf die Ressourcen einer anderen Anwendung zugreifen, da die Daten über Dateisystemrechte auf Basis des Nutzermanagements von Linux geschützt werden. Solange der Entwickler die Daten nicht frei gibt, können seine Daten nicht von einer anderen Applikation gelesen oder geändert werden. Ausnahme bilden hier Daten, die auf einem externen Speicher (SD-Karte) liegen. Auf diesen wird normalerweise das Dateisystem FAT (File Allocation Table) verwendet, welches keine Nutzerrechte unterstützt. Somit sind alle Daten auf dem externen Speicher für alle Anwendungen vollständig zugänglich. Abhilfe könnte hier die Verschlüsselung der Daten schaffen. Dadurch, dass sich Android auf Schutzmechanismen des Kernels abstützt, läuft nativer wie auch

interpretierter Code innerhalb der Sandbox. Android ist so aufgebaut, dass die Software oberhalb des Linux Kernels innerhalb der Sandbox läuft (vgl. Abbildung 1.1). Um aus der Sandbox auszubrechen, müssen also die Schutzmaßnahmen des Linux Kernels ausgehebelt werden. Weitere Einschränkungen werden durch das Permission Model (siehe Kapitel 1.2.5) geregelt [9] [21].

Es existieren Projekte, die gewisse Zugriffskontrollstrategien verwenden, die die Schutzmechanismen des Systems forcieren. Als Beispiele dürfen hier AppArmor⁸, SE Linux (Security-Enhanced)⁹ und TOMOYO Linux¹⁰ genannt werden, welche Mandatory Access Control als Zugriffskontrollstrategie benutzen, und grsecurity¹¹, welches Role-Based Access Control als Zugriffskontrollstrategie benutzt.

Bei der *Mandatory Access Control (MAC)* Strategie werden durch regelbasierte Festlegungen globale Eigenschaften spezifiziert, die durch das System umgesetzt werden. Durch den Benutzer festgelegte Rechte haben dabei geringere Priorität. Erlaubt zum Beispiel ein vom Benutzer vergebenes Recht den Zugriff auf eine Ressource, so wird dieser verweigert, sofern eine vom System spezifizierte Regelung existiert, die den Zugriff untersagt. Umgekehrt kann auch der Benutzer Rechte einschränken, die das System gewähren würde. Eine andere Strategie wird mit *Role-Based Access Control (RBAC)* verfolgt. Hierbei wird dem Nutzer eine gewisse Rolle zugeschrieben. Welche Aufgaben ein Nutzer durchführen darf, wird dann anhand seiner Rolle bestimmt. Anhand dessen werden auch die Zugriffsrechte bestimmt, die der Nutzer zur Umsetzung seiner Aufgabe benötigt. Die meisten Betriebssysteme verwenden ein Gruppenkonzept zur Vergabe von Rechten. Diese Art der Umsetzung kann allerdings nicht als Umsetzung der Role-Based Access Control Strategie gesehen werden [22].

AppArmor reglementiert den Zugriff auf Ressourcen, der durch ein Programm erfolgt, mit Hilfe von Attributen, die an das Programm gekoppelt werden. AppArmor fokussiert sich also auf die Möglichkeiten, die ein Programm hat. Es wird nicht betrachtet, wer dieses Programm ausführt. AppArmor bietet zwei Modi - "enforcement" und "complain". Im "enforcement" Modus werden die spezifizierten Einschränkungen für ein Programm strikt eingehalten und zusätzlich geloggt. Im "complain" Modus wird der Zugriff zu einer eingeschränkten Ressource auch zugelassen, wenn dies eigentlich nicht erlaubt ist. Der unerlaubte Zugriff auf die Ressource wird geloggt. AppArmor ist mit dem Kernel ab Version 2.6.36 aufwärts verfügbar, was ein Vorteil von AppArmor sein dürfte [23].

TOMOYO Linux setzt die Mandatory Access Control Strategie ähnlich um. Dabei wird ein Blick auf das Verhalten des Systems geworfen. Die Idee ist, dass ein Prozess einen gewissen Zweck erfüllt, wofür dieser Ressourcen benötigt und der gewisse Verhaltensweisen an den Tag legt. Normalerweise werden Anwendungen in einem Betriebssystem nicht überwacht und es ist schwer festzustellen, welche Operationen eine Anwendung durchführt. TOMOYO Linux überwacht die Operationen, die ein Prozess ausführt. Dabei lässt es nur Verhaltensweisen zu, beziehungsweise gewährt nur Zugriff auf Ressourcen, die durch den

⁸<https://wiki.ubuntu.com/AppArmor>, 26.04.2012 (Aufruf Juni 2012)

⁹<http://www.nsa.gov/research/selinux/>, 15.01.2009 (Aufruf Juni 2012)

¹⁰<http://tomoyo.sourceforge.jp/index.html.en>, 18.06.2012 (Aufruf Juni 2012)

¹¹<http://grsecurity.net/>, 2011 (Aufruf Juni 2012)

Administrator gewährt wurden. Eine Konfiguration kann automatisch erstellt werden, indem die Operationen, die eine Anwendung ausführt, zu einer sogenannten "Access Control List" (ACL) hinzugefügt werden. Diese kann eingesehen werden und dadurch helfen zu verstehen, wie sich eine Anwendung im System verhält [24]. An diesem Punkt sei auf ein Projekt namens "TrustDroid"¹² von der TU Darmstadt verwiesen, welches unter anderem TOMOYO Linux benutzt, um den Schutz den Android bietet, zu forcieren. Vorteil dieser Lösung soll sein, dass kaum Overhead entsteht und der Akku des mobilen Gerätes dadurch kaum in Anspruch genommen wird.

1.2.5 Permission Model

Android besitzt ein Rechtesystem, welches dazu dient, Kontrolle über die Rechte einer Applikation auf dem System auszuüben. Standardmäßig kann eine Applikation nur auf einen kleinen Teil der Systemressourcen zugreifen. Wie die Ressourcen vor Zugriff geschützt sind, ist unterschiedlich implementiert. Generell wird aber versucht, so viel wie möglich auf einem möglichst niedrigen Level - bezüglich der Architektur (siehe Abbildung 1.1) - zu implementieren. Prinzipiell sind die Ressourcen nur über das Android-System verfügbar. Zu den geschützten Ressourcen gehören zum Beispiel Kamera, GPS, Bluetooth, Funktionen zum Telefonieren und für SMS / MMS, sowie für Netz- und Datenverbindungen. Um auf diese Ressourcen zugreifen zu dürfen, muss eine Applikation im Android-Manifest deklarieren, auf welche APIs sie zugreifen möchte. Es existieren standardmäßig über 100 solcher Permissions¹³, wobei weitere von jedem Entwickler selbst definiert werden können. Soll nun eine Applikation installiert werden, so wird dem Nutzer je nach Schutzstufe (siehe weiter unten) angezeigt, welche Rechte sie erfordert. Dabei kann dieser der Rechteanforderung nur zustimmen oder sie verwerfen. Stimmt er nicht zu, wird die Anwendung nicht installiert. Einzelne Rechte können standardmäßig nicht eingeschränkt werden. Wird eine Anwendung deinstalliert, so werden die Rechte, die der Anwendung gewährt wurden, vom System verworfen. Standard-Anwendungen, die mit dem mobilen Gerät ausgeliefert werden, erfragen keine Rechte. Diese wurden ihnen bereits durch den Hersteller gewährt. Einige Ressourcen beziehungsweise Geräte können systemweit ausgeschaltet werden - zum Beispiel GPS, Bluetooth, WLAN oder mobile Netze. Sind diese ausgeschaltet, kann keine Applikation mehr darauf zugreifen. Sollte eine Applikation dennoch auf APIs zugreifen, für die diese keine Zugriffsrechte angefordert hat, so löst dies eine "Security Exception" aus [9].

Permissions sind in vier verschiedene Schutzstufen unterteilt - "normal", "dangerous", "signature" und "signatureOrSystem". Dabei gibt die jeweilige Schutzstufe an, wie gefährlich eine Funktion sein kann und wie das System vorgeht, um zu prüfen, ob ein Recht gewährt werden soll oder nicht. Im weiteren werden die einzelnen Schutzstufen näher erläutert. Die Schutzstufe "normal" definiert einen Standardwert und gewährt Zugriff auf Funktionen,

¹²"Practical and Lightweight Domain Isolation on Android" http://www.trust.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_TRUST/PubsPDF/spsm18-bugiel.pdf, 17.10.2011 (Aufruf Juni 2012)

¹³Diese können unter <https://developer.android.com/reference/android/Manifest.permission.html>, 28.06.2012 (Aufruf Juni 2012) eingesehen werden

die ein niedriges Gefahrenpotential für andere Anwendungen, das System, beziehungsweise den Nutzer darstellen. Rechte mit dieser Schutzstufe werden direkt durch das System gewährt und müssen nicht durch den Nutzer akzeptiert werden, können aber von ihm bei der Installation eingesehen werden. Die Schutzstufe "dangerous" ist eine Kategorie für Funktionen, die potentiell ein hohes Risiko darstellen. Applikationen mit solchen Rechten erhalten eventuell Zugriff auf private Daten des Nutzers oder Kontrolle über die Anwendung. Rechte aus dieser Kategorie werden nicht automatisch durch das System gewährt. Sie bedürfen der Zustimmung durch den Nutzer. Die Funktionen, die in die Schutzstufe "signature" fallen, werden durch das System nur gewährt, wenn die Anwendung, die die Erlaubnis erteilt bekommen möchte, die gleiche Signatur (siehe Kapitel 1.2.3) aufweist, wie die Anwendung, die dieses Recht deklariert hat. Ist dies der Fall, so gewährt das System automatisch dieses Recht, ohne den Nutzer darüber zu informieren beziehungsweise um Erlaubnis zu fragen. Dies könnte zum Beispiel der Zugriff auf das Internet sein. Die letzte Schutzstufe heißt "signatureOrSystem" und wird vom System nur gewährt, wenn die Applikation, die die Erlaubnis für die Funktion erfragt, das gleiche Zertifikat besitzt, wie eine Applikation im Systemabbild oder die Applikation selbst im Systemabbild ist. Von dieser Schutzstufe wird abgeraten und auf die - in den meisten Fällen - ausreichende Schutzstufe "signature" verwiesen [25].

1.2.6 Dalvik Virtual Machine

Die Dalvik Virtual Machine ist Bestandteil der Android Runtime (siehe Abbildung 1.1) und basiert auf Registern. Sie wurde von Dan Bornstein entworfen und geschrieben. Sie wurde optimiert, um auf Geräten mit geringer Arbeitsspeichergröße und geringer Prozessorgeschwindigkeit ausgeführt werden zu können, da sie schließlich auch auf Geräten ausgeführt werden soll, bei denen es darauf ankommt die Akkulaufzeit zu schonen. Sie erlaubt die Verwendung mehrerer Instanzen der virtuellen Maschine, basierend auf der Prozessisolation, dem Speichermanagement und der Threadunterstützung des Betriebssystems - im Falle von Android ist dies Linux als Basis. Die Dalvik Virtual Machine basiert auf der Java Virtual Machine, ist aber streng genommen keine Java Virtual Machine, da der ausgeführte Bytecode im Dex-Format vorliegt und nicht als Java Bytecode. Auf der Dalvik Virtual Machine werden prinzipiell Programme ausgeführt, die in Java geschrieben sind. Dabei wird der Java-Quellcode in Java-Bytecode (Class-Files) übersetzt. Diese wiederum werden mit einem Tool namens dx in .dex (Dalvik Executables) übersetzt, welcher dann auf der Dalvik Virtual Machine ausgeführt wird. Die .dex Dateien werden nach dem Übersetzen in eine .apk (Android Package) Datei gepackt [26].

Android bietet also die Möglichkeit reguläre Java Programme zu schreiben, wobei die meisten Standardbibliotheken verfügbar sind. Da die Dalvik Virtual Machine ihrerseits mehrere Instanzen unterstützt, erhält unter Android jede Anwendung ihre eigene Instanz der Dalvik Virtual Machine [7].

Das bedeutet, dass eine Android Anwendung innerhalb der Sandbox ausgeführt wird, also mit eigener User ID. Mit Hilfe der "sharedUserId" können noch andere Anwendungen in der "gleichen" Sandbox ausgeführt werden, was bedeutet, dass die Prozesse die gleiche User ID besitzen. Jede Anwendung wird in einem eigenen Prozess ausgeführt, in der dann

eine Dalvik Virtual Machine gestartet wird, die für die Interpretation des dex-Codes verantwortlich ist.

Es existieren auch virtuelle Maschinen, die Dienste des unterliegenden Betriebssystems vermitteln und den Zugriff auf diese reglementieren, wodurch der Schutz erhöht werden soll. Dies ist nicht der Sinn und Zweck der Dalvik Virtual Machine. Java bietet die Möglichkeit nativen Code auszuführen, was auch auf der Dalvik Virtual Machine möglich ist. Da diese in der Sandbox läuft (siehe Kapitel 1.2.4), ist es ohne weitere Einschränkungen möglich, nativen Code auszuführen, welcher dann in der Sandbox ausgeführt wird. Somit dient die Dalvik Virtual Machine nicht wirklich der Erhöhung des Schutzes, als viel mehr der Möglichkeit, kleine speicher- und prozessoroptimierte Programme zu schreiben [27].

1.3 Angriffsmöglichkeiten

Im Kapitel 1.2 wurden die Schutzmaßnahmen von Android erläutert, in dem vielleicht auch schon ein paar Schwächen von Android klar geworden sind. In diesem Kapitel sollen nun ein paar Angriffsmöglichkeiten auf Android erläutert werden. Dazu wird zunächst auf die Android Plattform und die Updates für diese im Kapitel 1.3.1 eingegangen. Im Kapitel 1.3.2 wird erläutert, warum Googles Play Store ein mögliches Angriffsziel sein könnte, worauf einige Kritik zum Permission Model in Kapitel 1.3.3 folgt. Abschließend wird im Kapitel 1.3.4 erklärt, welche Folgen das Rooting hat und wie es ausgenutzt werden kann.

1.3.1 Android Plattform & Updates

In Version Android 3.0 wurde der Kernel von Android auf die Version 2.6.36 aktualisiert. Das heißt die Versionen von Android 3.0 aufwärts laufen immerhin mit der Kernel Version 2.6.36 oder höher. Dies betrifft aber nur knapp 10% der Geräte mit Android (siehe Abbildung 1.2). Alle Geräte davor laufen mit einer kleineren Kernel Version [28]. Letzte stabile Kernel Version ist 3.4.2 (Aufruf Juni 2012)¹⁴. Es empfiehlt sich die aktuellste Version des Kernels zu verwenden, da in neueren Versionen unter Umständen Sicherheitslücken behoben wurden. Android nutzt einige Dienste von Linux, die mit Root-Rechten ausgeführt werden und in denen sich teilweise Bugs befinden, die einem Angreifer ermöglichen, sich Root-Rechte zu verschaffen. Auch im Kernel existieren manchmal Sicherheitslücken, die in späteren Versionen gefixt wurden, die sich in Android aber unter Umständen weiterhin ausnutzen lassen, da eine veraltete Kernel Version verwendet wird [20]. Aufgrund der Update Problematik, die im Kapitel 1.2.1 beschrieben wird und auf die hier auch nochmal kurz eingegangen wurde, wird klar, dass Android Updates stark verzögert durchgeführt werden, was dazu führt, dass bekannte Lücken unter Umständen sehr lange nutzbar sind und somit zum Beispiel Rootkits, die unter Linux lauffähig sind, auch unter Android laufen können (siehe Kapitel 1.3.4) [20].

¹⁴<http://www.kernel.org/> (Aufruf Juni 2012)

Die Schutzmaßnahmen von Android basieren im Hauptbestandteil auf den Schutzmaßnahmen des Kernels - unter anderem dessen Benutzerverwaltung. Dabei setzen viele dieser Schutzmaßnahmen darauf, dass die Anwendungen eingeschränkte Nutzerrechte und eben keine Root-Rechte besitzen. Hat eine Anwendung Root-Rechte, so hat sie Vollzugriff auf das ganze System - unter anderem auf das Dateisystem, alle Prozesse sowie die Nutzer. Im Sinne von Android sind dies die Applikationen, welche beliebig geändert werden dürfen. Dadurch werden die Schutzmaßnahmen von Android vollständig untergraben. Android-Geräte werden gern "gerootet", um ein neues Betriebssystem zu installieren und eine neuere Version von Android aufzuspielen. Dazu muss der Bootloader freigeschaltet werden, wobei alle Benutzerdaten gelöscht werden. Dies kann umgangen werden, wenn der Root-Zugang über einen Kernel-Bug oder eine Sicherheitslücke erreicht wird.

Ab Android Version 3.0 kann das komplette Dateisystem mit Hilfe von dmccrypt verschlüsselt werden. Dabei wird ein Schlüssel vom Nutzerpasswort abgeleitet. Dadurch soll verhindert werden, dass eine nicht autorisierte Person Zugriff auf die abgelegten Daten des Gerätebesitzers erlangen kann, ohne dessen Passwort zu kennen. Doch Datenverschlüsselung, bei der der Schlüssel auf dem Gerät gespeichert ist, wird durch den Root-Zugang unnütz, da der Root-User Zugriff auf die gespeicherten Schlüssel besitzt. Die Schlüssel außerhalb des Gerätes zu speichern, löst dieses Problem nicht. Zu irgendeinem Zeitpunkt muss der Schlüssel von einem Server abgerufen oder vom Nutzer eingegeben werden. Eine schädliche Anwendung mit Root-Rechten kann den Schlüssel abgreifen [9].

1.3.2 Application Stores

Wie im Kapitel 1.2.2 dargestellt besitzt Android eine Verbindung zu Googles GTalk Server. Dadurch ist Google in der Lage, Nachrichten an das mobile Gerät zu senden. Eine dieser Nachrichten führt dazu, dass eine Applikation installiert wird. Dadurch ist es möglich auf den Play Store über den Computer zuzugreifen und dort, sofern vorher ein Login erfolgt ist, Anwendungen zu installieren. Dabei kann das Gerät, dass mit dem Nutzer-Account verbunden ist, ausgewählt werden. Anschließend können Berechtigungen, die diese Applikation erfordert, eingesehen werden und eine Installation durchgeführt werden. Dazu wird an das Gerät eine Nachricht geschickt, dass dieses die Applikation installieren soll. Bedenklich hieran ist, dass der Nutzer keine weitere Erlaubnis erteilen muss. Problematisch hieran ist, dass ein Angreifer, der Zugriff auf das Google-Benutzerkonto hat, jede beliebige Applikation aus dem Play Store installieren kann [29].

1.3.3 Permission Model

Wie im Kapitel 1.2.5 erläutert, unterliegt jede Applikation in Android dem Permission Model. Will ein Nutzer eine Applikation installieren, so muss er die Rechte, die diese Applikation anfordert, akzeptieren. Dies kann dazu genutzt werden, die Sicherheit zu erhöhen, doch es muss diese nicht zwangsweise erhöhen.

Die erste Möglichkeit für eine Anwendung, die das Permission Model umgehen möchte, ist einfach alle Rechte zu definieren, die sie benötigt, um schädliche Aktionen auszuführen.

Wenn die Applikation einen Zweck erfüllen soll und diese Rechte anfordert, die sie offensichtlich nicht für die Erfüllung dieses Zwecks benötigt, so kann dies unter Umständen leicht von dem Nutzer erkannt werden. Dies könnte zum Beispiel ein Spiel - zum Beispiel Sudoku - sein, das SMS senden möchte. Dem Nutzer sollte an dieser Stelle klar sein, dass dieses Spiel nicht das Recht zum Senden von SMS benötigt. Also wird es dieses vermutlich nutzen, um den Anwender Schaden zuzufügen. Nicht ganz so klar ist, ob eine Anwendung Schaden anrichten möchte, wenn sie zum Beispiel Zugriff auf das Internet anfordert. Dies könnte dazu dienen, Werbung in der Anwendung anzuzeigen, aber auch dazu, Informationen vom Nutzer zu versenden. Ein Problem hierbei dürfte auch sein, dass die wenigstens Nutzer durchlesen, was für Rechte durch eine Applikation angefordert werden und selbst wenn sie dies machen würden, sie die Ausmaße der Kombination der angeforderten Rechte nicht überblicken können. So kann dies zum Beispiel dazu führen, dass ein Nutzer ausspioniert wird. Viele Nutzer wissen auch eine gewisse Ignoranz bezüglich dessen auf, was mit ihren Daten geschieht und es ist ihnen egal, was eine Anwendung mit ihren Daten macht. Es geht ihnen hauptsächlich darum, dass die Applikation den Nutzen erfüllt, für den sie diese geholt haben.

Eine weitere Möglichkeit das Permission Model zu umgehen, ist die Nutzung einer "sharedUserId". Hierbei bekommen zwei Applikationen, sofern sie dies anfordern und vom gleichen Entwickler stammen (also gleich signiert wurden), die gleiche User ID zugewiesen (siehe Kapitel 1.2.3). Android schreibt die Rechte, die angefordert werden der User ID zu und somit allen Applikationen, die diese "sharedUserId" besitzen. Dadurch können Applikationen aus einer Quelle Rechte kombinieren, womit sie Schaden anrichten können. Darf zum Beispiel eine Applikation das Telefonbuch des Nutzers einsehen und hat eine andere die Möglichkeit auf das Internet zuzugreifen, so können beide Applikationen, wenn sie eine "sharedUserId" besitzen, sprich ihre Rechte teilen, zum Beispiel die Kontakte des Nutzers versenden. Der Nutzer hat standardmäßig keine Möglichkeit solche Beziehungen einzusehen. Hier wäre eine Anwendung wichtig, die solche Beziehungen darstellt und die kombinierten Rechte darstellt, damit der Nutzer mehr Kontrolle bekommt.

Ein großes Problem ist, dass die Rechte, die eine Anwendung anfordert, nur akzeptiert oder abgelehnt werden können. Die Sicherheit kann erhöht werden, indem es ermöglicht wird, der Anwendung nur die Rechte zu gewähren, die der Anwender ihr gewähren möchte. Dies impliziert, dass nicht alle Rechte gewährt werden müssen, sondern eine Auswahl, welche Rechte welcher Applikation eingeräumt werden (und welche eben nicht eingeräumt werden) möglich ist. Dadurch könnten Rechte beziehungsweise die Kombination von Rechten, die potentiell gefährlich sind, verboten beziehungsweise entzogen werden und somit die Sicherheit erhöht werden [20]. Es existiert ein Projekt, das dies ermöglicht, ohne dass dafür das mobile Gerät gerootet werden muss. Es setzt darauf, dass Applikationen auf Android in Java Bytecode vorliegen und auf einer virtuellen Maschine ausgeführt werden (siehe Kapitel 1.2.6). Die Applikation wird mit speziellem Überwachungscode versehen. Dadurch können die Applikationen nicht nur überwacht werden, sondern auch schädliche Aufrufe geblockt beziehungsweise so geändert werden, sodass diese keinen Schaden anrichten. Dadurch kann sogar die Nutzung bekannter Sicherheitslücken in den Applikationen sowie in dem Android Betriebssystem verhindert werden¹⁵ [30].

¹⁵The Android Monitor - Real-time policy enforcement for third-party applications <http://www.infsec.cs.uni-saarland.de/projects/android-monitor/android-monitor.pdf>, Februar 2012, (Aufruf Juli 2012)

1.3.4 Rooting

Unter Android werden standardmäßig nur der Kernel und einige Anwendungen mit Root-Rechten ausgeführt. Ein Anwender mit Root-Rechten hat die Möglichkeit auf das komplette System zuzugreifen, also auf alle Applikationen und Daten [9]. Die Schutzmaßnahmen auf Android stützen sich stark auf das Nutzermanagement ab. Dabei bekommen die Applikationen eine User ID zugewiesen, die keine Root-Rechte beinhaltet, wodurch sie gekapselt werden (siehe Kapitel 1.2.4). Hat eine Anwendung auf Android nun Root-Rechte, so läuft diese außerhalb der Sandbox und hat vollen Zugriff auf das gesamte System. Damit können alle Schutzmechanismen, die in Android vorhanden sind, ausgehebelt werden.

Wie ein Angriff aussehen kann, soll anhand der Schadsoftware DroidDream erläutert werden. Der Trend bei mobiler Schadsoftware geht zum Verstecken von Schadsoftware in normalen Anwendungen. So auch bei der Schadsoftware DroidDream, die damals (März 2011) in mehr als 50 verschiedenen Applikationen versteckt wurde. DroidDream kann sich nicht selbst starten, sondern muss durch Nutzerinteraktion gestartet werden. Wird die Applikation geöffnet, startet DroidDream seine eigenen Services und anschließend die Anwendung, in der es sich versteckt. Ein Service namens "com.android.root.Setting" nimmt einmalig Kontakt zu einem Server auf. Dies dient dazu, das Gerät zu identifizieren und dem Server mitzuteilen, dass dieses Gerät infiziert wurde. Daten die dazu übertragen werden, sind IMEI, IMSI, Geräte Modell und SDK Version. Es werden auch noch eine Partner und ProductId übertragen, die die DroidDream Variante identifizieren. DroidDream ist auch so konfiguriert, dass eine weitere Infektion durch eine andere Variante verhindert wird. Die Kommunikation zwischen Server und DroidDream ist verschlüsselt mit einem Schlüssel, der in "com.android.root.adbRoot.crypto" versteckt ist. Verschlüsselt wird unter Verwendung von einer bitweisen XOR-Funktion. Ist das Gerät noch nicht infiziert, wird versucht das Gerät zu rooten. Dabei werden zwei Techniken ausprobiert, die Root-Rechte verschaffen sollen und von Sebastian Kraemer entwickelt wurden. Die erste heißt "exploid"¹⁶, welche Lücken in udev's Event Handling ausnutzt. Wenn diese nicht funktioniert, versucht DroidDream "Rage Against The Cage" zu nutzen. Ist das Gerät gerootet, so installiert DroidDream die Anwendung "com.android.providers.downloadsmanager", sofern diese noch nicht installiert ist. Diese ist in der "sqlite.db" Datenbank-Datei versteckt und wird einfach in das Verzeichnis "/system/app" verschoben, wodurch die Applikation ohne Nutzerinteraktion, also ohne, dass der Nutzer davon etwas bemerkt, installiert wird. Die installierte Anwendung ermöglicht es dem Autor, neue Anwendungen herunterzuladen und ohne Wissen des Geräteinhabers zu installieren beziehungsweise zu aktualisieren, sowie diese dann auch mit Root-Rechten auszuführen [31] [32].

Rage Against The Cage nutzt eine Lücke in adb (Android Debug Bridge¹⁷), wobei ausgenutzt wird, dass nicht überprüft wird, ob die Funktion `setuid()` die User ID erfolgreich gesetzt hat. Es existiert ein Wert (`RLIMIT_NPROC`), welcher festlegt, wie viele Prozesse pro User ID maximal laufen dürfen. Nun werden die Prozesse mit Hilfe der Funktion `fork()` so oft aufgespalten, bis `fork()` signalisiert, dass die maximale Anzahl an Prozessen für die User ID erreicht ist. Anschließend wird "adb" beendet, wodurch dieser neu gestartet wird. Dabei startet dieser mit Root-Rechten, führt einige Initialisierungen durch und

¹⁶<http://cve.mitre.org/cgi-bin/cvname.cgi?name=CVE-2009-1185>, 2009 (Aufruf Juni 2012)

¹⁷<http://developer.android.com/guide/developing/tools/adb.html> (Aufruf Juni 2012)

versucht dann, seine Rechte abzuschwächen, um als Shell-Nutzer zu laufen. Normalerweise würde die Funktion `setuid()` die Anzahl der Root-Prozesse dekrementieren und die Anzahl der Shell-Prozesse inkrementieren. Die maximale Anzahl der Shell-Prozesse ist aber schon erreicht, was dazu führt, dass die Funktion `setuid()` fehlschlägt. Da nicht überprüft wird, ob die User ID gesetzt wurde, läuft der Prozess weiterhin mit Root-Rechten. Wird anschließend über die "adb -d shell" eine Verbindung hergestellt, so erhält der Anwender eine Shell mit Root-Rechten [33].

1.4 Angriffserkennung

In diesem Kapitel wird behandelt, wie Angriffe erkannt werden können. Dazu wird ein Blick auf Intrusion Detection Systeme geworfen (siehe Kapitel 1.4.1). Dabei spielt auch die Intrusion Response eine Rolle (siehe Kapitel 1.4.2). Anschließend wird auf die verschiedenen Arten der Angriffserkennung eingegangen. Darunter fallen die wissensbasierte Erkennung (siehe Kapitel 1.4.3) und die verhaltensbasierte Erkennung (siehe Kapitel 1.4.4). Darauf folgt ein kurzer Abschnitt über Honey pots im Kapitel 1.4.5. Als letztes wird im Kapitel 1.4.6 noch ein Blick auf Cloudbasierte Angriffserkennung geworfen, da sich diese besonders für mobile Geräte eignet.

1.4.1 Intrusion Detection Systeme

Ziel der Intrusion Detection ist es Angriffe und Missbrauch von Computern beziehungsweise Netzen schnellstmöglich zu erkennen und zu melden. Dazu werden die Systeme aktiv überwacht und es wird versucht, Ereignisse zu filtern, welche auf Missbrauch oder Angriffe hindeuten. Dies wiederum bedeutet, dass geeignete Werkzeuge benötigt werden, um die Ereignisse zu protokollieren und im Anschluss bewerten zu können. Intrusion Detection Systeme sind prinzipiell nichts anderes, als eine Zusammenstellung von Werkzeugen, die solche und weitere Funktionen zur Verfügung stellen.

Intrusion Detection Systeme setzen sich aus verschiedenen Komponenten zusammen, die nun näher erläutert werden sollen.

Die erste Komponente ist der *Netzsensor*, welcher prinzipiell zur Überwachung des Netzverkehrs an einer bestimmten Stelle des Netzes dient und somit Ereignisse in einem ganzen Teilnetz untersucht, um verdächtige Ereignisse zu erkennen. Dazu wird in der Regel ein eigener "Rechner" eingesetzt, der als Netzsensor fungiert, wodurch die Endsysteme nicht zusätzlich belastet werden. Manche Hersteller verkaufen Netzsensoren nur noch als Kombination aus Hard- und Software. Diese Art von Sensor eignet sich besonders zur Erkennung von Angriffen, die sich gegen mehrere Systeme im Netz richten (zum Beispiel DoS Angriffe). Netzsensoren können so konfiguriert werden, dass nicht direkt auf sie zugegriffen werden kann, wodurch diese schwerer angreifbar sind als Hostsensoren. Netzsensoren können zusätzlich untereinander verbunden werden, wodurch ein Intrusion Detection Netz aufgebaut werden kann. Ein Problem bei Netzsensoren ist, dass sie bei einer hohen Netzlast das Netz eventuell nicht mehr vollständig überwachen können. Das

liegt hauptsächlich daran, dass die Auswertung, ob gewisse Ereignisse verdächtig sind oder nicht, nur mit relativ hohem Aufwand erfolgen kann. Ein weiterer Kritikpunkt ist, dass Netzsensoren eben nur Angriffe anhand des Netzverkehrs erkennen können. Sobald Kommunikationsdaten verschlüsselt wurden, können sie dies auch nicht mehr beziehungsweise nur sehr eingeschränkt. Auch negativ zu bewerten, ist, dass auftretende Fehler in der Datenübertragung durch Netzsensoren als verdächtig bewertet werden können und dies somit eine gewisse Fehlerrate zur Folge hat.

Als zweite Komponente ist der *Hostsensor* zu nennen. Dieser wird auf dem Host, also dem System, platziert. In der Regel werden diese Sensoren benutzt, um Angriffe auf Anwendungen oder das Betriebssystem zu erkennen, wobei bei manchen auch die Überwachung des hostspezifischen Netzverkehrs mit einbezogen werden kann. Generell bedeutet der Betrieb eines Hostsensors zusätzlichen Aufwand für das betriebene System. Der Hostsensor hat die Möglichkeit mehr Informationen in die Bewertung mit einfließen zu lassen, da er direkt auf dem System ausgeführt wird. Dies ist allerdings auch ein Nachteil, da Hostsensoren im Allgemeinen betriebssystem- wie auch applikationsspezifisch sind und somit die Kosten aufgrund der Vielfalt stark steigen können. Da eine Kopplung zwischen Sensor und System besteht, ist der Angreifer in der Lage, gleichzeitig Sensor und System außer Gefecht zu setzen. Es gibt verschiedene Arten, wie die Überwachung durchgeführt wird. Die Systemüberwachung wird auf Prozessebene direkt durch den Sensor vollzogen. Dabei können Zugriffe auf Dateien und Applikationen, Zugriffsverletzungen und anomale Verhaltensmuster erkannt werden und einem Nutzer zugeordnet werden. Bei der Applikationsüberwachung erfolgt dies in der Regel über Auswertung von Logdaten der Applikation, wobei dies meist von der Anwendung unterstützt werden muss. Die Integritätsüberwachung dient dazu, zu prüfen, ob Dateien geändert wurden, was normalerweise über Checksummen realisiert wird. Dadurch können Veränderungen erkannt werden, was die Untersuchung und Wiederherstellung des Systems erleichtern kann. Das impliziert, dass diese Überwachungsmethode nicht zur Früherkennung genutzt werden kann, da geänderte Dateien meist Resultat eines erfolgreichen Angriffs sind. Als letztes kann, wie bereits erwähnt, auch der hostspezifische Netzverkehr überwacht werden. Das bedeutet, dass der Kommunikationsfluss auf allen Protokollebenen überwacht werden kann, was auch dazu führt, dass verschlüsselte Protokolle die Überwachung nicht behindern. Dadurch, dass ein einzelnes System normaler Weise nicht so viele Daten empfängt und versendet, ist eine vollständige Prüfung des Datenverkehrs möglich. Ein wesentlicher Nachteil ist, dass Angriffe, die sich gegen mehrere Systeme richten, meist nicht erkannt werden können.

Die Dritte Komponente eines Intrusion Detection Systems ist die *Datenbankkomponente*, welche dazu genutzt wird, Ereignisdaten, die bei der Angriffserkennung erzeugt werden, zu speichern, damit diese später weiterverarbeitet werden können. Die Datenmenge, die dabei entsteht, kann durchaus sehr groß sein. Die meisten Hersteller bedienen sich einer SQL-Datenbank.

Die vierte Komponente dient der Verwaltung, also der Konfiguration und Kalibrierung. Unter anderem werden hier die verschiedenen Komponenten erfasst.

Fünfte und letzte Komponente eines Intrusion Detection Systems ist eine Auswertungsstation, die auch mit der Managementkomponente kombiniert sein kann. Diese Komponente dient in der Regel der Auswertung von Ereignissen und der Erstellung von Berichten [34].

Mobile Geräte können in Intrusion Detection Systeme als Hostsensoren integriert werden. Dabei muss wiederum darauf geachtet werden, dass der Anteil, der auf dem mobilen Gerät läuft, nicht so hoch ist, da mobile Geräte einer gewissen Ressourcenknappheit unterliegen.

1.4.2 Intrusion Response

Intrusion Detection Systeme bieten die Möglichkeit auf einen erkannten Angriff automatisiert zu reagieren. Dabei können unterschiedliche Maßnahmen abhängig von dem Ereignis ergriffen werden. Dazu müssen die Ereignisse entsprechend dokumentiert werden. Diese Protokollierung erfolgt durch das Intrusion Detection System selbst. Eine Alarmierung kann abhängig vom Ereignis über unterschiedlichste Kommunikationsmittel - je nach Stand der Technik erfolgen (zum Beispiel per E-Mail, Pager, SMS, usw.). Dabei definieren die meisten Intrusion Detection Systeme unterschiedliche Alarmklassen. Um eine verzugslose Reaktion auf Angriffe zu ermöglichen, können bei der Intrusion Response automatisiert Gegenmaßnahmen ergriffen werden. Dies kann eine automatische Beeinflussung von Netzkomponenten und Rechensystemen beinhalten. Beispiele hierfür sind Änderung der Firewall-Konfiguration (um Zugangsmöglichkeiten für den Angreifer zu versperren und Zeit für Gegenmaßnahmen zu gewinnen), Beenden der Kommunikationsverbindung oder Änderung von Zugriffsrechten auf einem Rechner. Hierbei darf daran erinnert werden, dass Intrusion Detection Systeme häufig Fehlalarme verursachen. Dies sollte bei der Konfiguration der Intrusion Response Funktionen beachtet werden, da sonst die Verfügbarkeit von Systemen negativ beeinflusst werden kann.

Da mobile Geräte in ein Intrusion Detection System eingegliedert werden können, kann auch die Intrusion Response Funktion genutzt werden, welche auf gewisse Ereignisse reagiert. Die Alarmierung kann weiterhin über die genannten Kommunikationsmittel erfolgen - unter anderem nun auch auf einem mobilen Gerät selbst. Zum Beispiel per SMS.

1.4.3 Wissensbasierte Erkennung

Voraussetzung für die wissensbasierte Erkennung ist die Kenntnis über Angriffe, Angriffsvektoren beziehungsweise Verhaltensweisen, die Schäden verursachen können. Dabei wird versucht diese zu charakterisieren. Dies kann von einfachen Mustern in Zeichenketten, die über Pattern Matching erkannt werden können, bis hin zu komplexeren Verhaltensmustern reichen. Wird ein solches Muster erkannt, so wird der Nutzer informiert und gegebenenfalls gefragt, wie weiter vorgegangen werden soll.

Wissensbasierte Erkennung wird in Intrusion Detection Systemen zum Beispiel über Muster von Ereignissen gelöst. Damit können Fehlverhalten vom System, wie auch von einem einzelnen Nutzer erkannt werden. Gern genanntes und einfaches Beispiel dürfte das wiederholte Fehlschlagen von Anmeldeversuchen eines Nutzers sein. Besonderheit bei Intrusion Detection Systemen dürfte sein, dass Signaturen über Skriptsprachen definiert und geändert werden können.

Nachteil solcher wissensbasierten Erkennungsverfahren ist, dass im Regelfall nur bereits bekannte Angriffe registriert werden können. Dies wiederum führt dazu, dass die zugrundeliegende Muster-Datenbank (Signature Database) immer auf einen aktuellen Stand gehalten werden muss. Somit ist diese Erkennungsmethode reaktiv. Sind die Signaturen weniger detailliert ausgeprägt, so steigt die Wahrscheinlichkeit auch bisher unbekannte Angriffe zu erkennen. Allerdings führt dies auch dazu, dass viele Verhaltensweisen für Angriffe gehalten werden, die keine sind ("false positives") [34].

Auch Viren-Schutzprogramme nutzen wissensbasierte Erkennung, welche in diesem Bereich die wichtigste Methode ist. Dabei sind die gesuchten Muster innerhalb einer Datei, die für ein Schadprogramm spezifischen Code-Sequenzen, welche die Signatur bilden. Auch hier ist wieder die Aktualität der Signaturen wichtigstes Kriterium für die Erkennung von Schadprogrammen, welche durch Updates gewährleistet werden muss. Bei der heuristischen Suche wird versucht dieses Problem zu lösen, indem nach verdächtigen Code-Sequenzen gesucht wird, wobei diese allgemeiner gehalten sind. Dies führt auch wieder zu mehr "false positives" [35]. Schlimmer hingegen sind "true negatives" bei denen ein Schadprogramm fälschlicher Weise nicht als dieses eingestuft wird.

Wissensbasierte Erkennung eignet sich durchaus für mobile Geräte. So gibt es mittlerweile schon Virens Scanner, die jede zum Beispiel neu installierte Applikation scannen^{18 19}.

1.4.4 Verhaltensbasierte Erkennung

Bei dieser Art von Erkennung wird prinzipiell versucht zu charakterisieren, wie sich ein System im Normalzustand verhält und Abweichungen (Anomalien) von diesem Verhalten zu erkennen.

Eine Art der Umsetzung ist die Protokollanalyse, bei der der Netzverkehr untersucht wird. In IP Netzwerken können Angriffe relativ gut erkannt werden, bei denen von den Protokollen abgewichen wird. Deshalb wird hier der Normalzustand unter anderem als das Einhalten der Protokollspezifikation definiert, was dann überprüft wird. Diese Überprüfung lässt sich recht performant gestalten. Nicht erkannt werden können dadurch Schwachstellen in der Protokollspezifikation. Deshalb wird diese Art der Erkennung meist mit wissensbasierten Erkennungsverfahren kombiniert.

Eine weitere Möglichkeit ist die Einbeziehung statistischer Daten. Hierbei wird versucht durch statistische Werte zu charakterisieren, wie sich ein System gewöhnlich verhält beziehungsweise wodurch das normale Nutzungsverhalten gekennzeichnet ist. Diese Werte werden in einer Lernphase ermittelt. Es wird davon ausgegangen, dass das System im Falle eines Angriffs von diesen statistischen Werten signifikant abweicht. Problematisch hierbei ist, dass Fehlverhalten während der Lernphase als normal eingestuft wird. Statistische Daten werden für unterschiedlichste Objekte, wie zum Beispiel Nutzer, Dateien und Anwendungen erfasst. Dabei wird ein Augenmerk auf das zugehörige Nutzungsverhalten

¹⁸AVG Antivirus Free: https://play.google.com/store/apps/details?id=com.antivirus&feature=search_result (Aufruf Juni 2012)

¹⁹Lookout Security & Antivirus: https://play.google.com/store/apps/details?id=com.lookout&feature=search_result (Aufruf Juni 2012)

gelegt. Dies können zum Beispiel Anzahl der Zugriffe und Zugriffsdauer auf Dateien sein, welche durch statistische Werte festgehalten werden. Anschließend können diese im laufenden Betrieb aufgenommen Daten gegen die statistischen Werte geprüft werden, welche das Normalverhalten charakterisieren. Problematisch hierbei dürfte sein, die statistischen Werte zu definieren, welche das normale Verhalten eines Systems charakterisieren. Auch muss die Annahme, dass Angriffe von den statistischen Werten - welche normal definieren - abweichen, nicht stimmen. So werden Angriffe, die in dieses als normal charakterisiertes Verhalten fallen, nicht erkannt. Ähnlich wie bei der Erkennung mit Hilfe von statistischen Daten, kann auch künstliche Intelligenz eingesetzt werden, um Angriffe zu erkennen. Doch auch bei dieser Methode ist die Fehlerrate recht hoch.

Vorteil der verhaltensbasierten Erkennung ist, dass auch neuartige beziehungsweise unbekannte Angriffe erkannt werden können.

Die verhaltensbasierte Erkennung eignet sich für den Einsatz auf mobilen Geräten. Je nachdem, wie aufwendig die Analyse ist, ist hier allerdings über eine cloudbasierte Lösung nachzudenken (siehe Kapitel 1.4.6).

1.4.5 Honeypots

Es gibt auch die Möglichkeit Honeypots einzusetzen, welche genutzt werden, um den Angreifer anzulocken. Honeypots sind Server, Netze, Programme und Prozesse, die besonders produktive und sicherheitskritische Systeme vortäuschen und somit ein verlockendes Ziel für einen Angreifer bilden. Das Normalverhalten eines Honeypots zu charakterisieren ist sehr leicht, da in der Regel keine oder nur eine geringe Anzahl an Zugriffen auf diese zu erfolgen hat. Somit sind eigentlich alle Zugriffe, die auf einen Honeypot erfolgen, als Abweichung vom normalen Verhalten zu bewerten. Somit lassen sich Angriffe beziehungsweise Fehlverhalten sehr leicht erkennen, aufzeichnen und auswerten. Problematisch an dieser Stelle ist, dass nur Angriffe auf den Honeypot erkannt werden können und nicht auf produktive Systeme, da ein Honeypot nicht auf einem produktiven System eingesetzt wird. Auch problematisch ist, dass sich das durch einen Angriff verursachte Verhalten auf einem Honeypot von dem Verhalten auf einem produktiven System unterscheiden kann [34].

Der Einsatz von Honeypots ist ähnlich wie bei den Intrusion Detection Systemen in Verbindung mit mobilen Geräten möglich. Dabei kann das mobile Gerät sogar direkt als Honeypot eingesetzt werden²⁰. Herkömmliche Lösung ist aber wohl den Honeypot in einem Netz, auf das ein mobiles Gerät zugreift zu positionieren, sodass schädliche Aktionen, die vom mobilen Gerät ausgehen, erkannt werden können. Es existiert ein Open Source Projekte namens Kippo²¹, in dessen Rahmen ein Honeypot entwickelt wird. Dieses Projekt wurde durch die Sicherheitsabteilung der Deutschen Telekom angepasst, sodass mobile Geräte mit Apple iOS und Android simuliert werden können. Dabei werden bei diesen simulierten Geräten Schwachstellen eingebaut, damit diese ein lohnendes Ziel für

²⁰HoneyDroid - Creating a Smartphone Honeypot: http://www.ieee-security.org/TC/SP2011/posters/HoneyDroid_Creating_a_Smart_Phone_Honeypot.pdf, Mai 2011 (Aufruf Juni 2012)

²¹Kippo - SSH Honeypot <http://code.google.com/p/kippo/> (Aufruf Juli 2012)

einen Angreifer darstellen. Dadurch können Angriffe, die sich gezielt gegen mobile Geräte richten, erkannt werden [36].

1.4.6 Cloudbasierte Erkennung

Ein großes Problem bei mobilen Geräten ist die Beschränkung der Ressourcen - sei es die Prozessorleistung oder die Akkukapazität. Deswegen ist es momentan nicht sinnvoll beziehungsweise möglich, aufwendige Berechnungen auf dem mobilen Gerät durchzuführen. Dieses Problem kann zum Beispiel gelöst werden, indem das mobile Gerät nur Daten sammelt und diese an einen Server zur Auswertung sendet. Zum Beispiel existiert ein Projekt namens "Crowdroid", das der Erkennung von Schadsoftware dient und eine verhaltensbasierte Analyse durchführt. Dabei überwacht Crowdroid die Systemaufrufe. Es wird davon ausgegangen, dass die Systemaufrufe das Verhalten einer Applikation gut wiedergeben. Dabei werden noch Daten wie geöffnete und benutzte Dateien, Ausführungszeiten und die Anzahl der verschiedenen ausgeführten Systemaufrufe aufgenommen. Diese Daten werden benutzt, um zu bestimmen, ob eine Anwendung schädlich ist oder nicht. Crowdroid erreicht dabei recht gute Ergebnisse [37]. Der größte Vorteil ist, dass die Daten auf dem Endgerät nur aufgenommen und ein wenig aufbereitet werden und die Auswertung anschließend auf einem Server stattfindet, was der Ressourcenknappheit auf dem Gerät gerecht wird [37].

Diese Art der Erkennung eignet sich auch im Besonderen für mobile Geräte, da der rechenintensive Teil der Erkennung ausgelagert wird und das Gerät nur als Messstation verwendet wird.

1.5 Fazit

Wie in Kapitel 1.2 gezeigt wurde, besitzt Android einige Schutzmaßnahmen, die teilweise nicht einmal auf Computern gefunden werden können. Die Richtlinie, sich auf erprobte Schutzmechanismen im Kernel abzustützen, ist positiv zu bewerten. Problematisch ist, dass Updates - die auch sicherheitsrelevant sein können - nicht so schnell verteilt werden. Hier könnte die Sicherheit stark erhöht werden, wenn Updates schneller verbreitet werden. Der Play Store von Google dürfte aufgrund der Verwendung des Dienstes Bouncer vergleichsweise sicher sein und an vielen Stellen vor Schadsoftware schützen, was in anderen Application Stores in der Regel nicht gegeben ist. Die Signierung von Applikationen soll die Sicherheit erhöhen, indem Entwickler voneinander abgegrenzt werden und standardmäßig keinen Zugriff auf Daten anderer Applikationen haben. Hier könnte Android jedoch noch sicherer werden, indem eine Zertifizierungsstelle verwendet wird, wodurch Entwickler von Schadsoftware nicht so leicht anonym agieren könnten. Die Sandbox-Technik ist positiv zu bewerten, sofern Geräte nicht gerootet werden können. Hier steht und fällt der Schutz mit den Schutzmaßnahmen des Kernels. Deswegen sind schnelle Updates - gerade, wenn es sich um sicherheitsrelevante Updates handelt - wichtig, um die Sicherheit zu erhöhen. Dies ist momentan noch nicht wirklich gegeben. Das Permission Model bietet auch das Potenzial, die Sicherheit von Android zu erhöhen. Doch wirkt sich hier unter

anderem das Nutzerverhalten stark auf die Sicherheit aus. Die Dalvik Virtual Machine könnte den Schutz prinzipiell erhöhen, doch dient sie nicht wirklich diesem Zweck. In dem Kapitel 1.3 wurde auf verschiedene Angriffsmöglichkeiten eingegangen. Hierbei sollte klar geworden sein, dass sich Faktoren, wie Updates und Nutzerverhalten, direkt auf die Sicherheit des mobilen Gerätes auswirken. Im Kapitel 1.4 wurde dann auf Möglichkeiten der Angriffserkennung eingegangen und auch ein paar Möglichkeiten aufgezeigt, die bei mobilen Geräten Verwendung finden könnten, um die Sicherheit dieser zu erhöhen. Alles in allem sind die Techniken und Ansätze, die in Android Verwendung finden, durchaus positiv zu bewerten. Sie sind jedoch ausbaufähig. Sie gehen in der Regel sogar über das hinaus, was auf einem gewöhnlichen Computer gefunden werden kann. Dennoch gibt es in vielen Bereichen noch Nachholbedarf, damit die Sicherheit auf einem mobilen Gerät erhöht werden kann.

Literaturverzeichnis

- [1] INTEL. *Moore's Law: Raising the Bar*, http://download.intel.com/museum/Moores_Law/Printed_Materials/Moores_Law_Backgrounder.pdf, 2005, Aufruf Juni 2012.
- [2] GARTNER, INC. AND/OR ITS AFFILIATES. *Gartner Says Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth*, <http://www.gartner.com/it/page.jsp?id=1924314>, 15.02.2012, Aufruf Juni 2012.
- [3] IDC CORPORATE USA. *Android- and iOS-Powered Smartphones Expand Their Share of the Market in the First Quarter, According to IDC*, <http://www.idc.com/getdoc.jsp?containerId=prUS23503312>, 24.05.2012, Aufruf Juni 2012.
- [4] MCAFEE LABS. *McAfee Threat Report: Erstes Quartal 2012*, <http://www.mcafee.com/de/resources/reports/rp-quarterly-threat-q1-2012.pdf>, 1. Quartal 2012, Aufruf Juni 2012.
- [5] OPEN HANDSET ALLIANCE. *Google and the Open Handset Alliance Announce Android Open Source Availability*, http://www.openhandsetalliance.com/press_102108.html, 21.10.2008, Aufruf Juni 2012.
- [6] OPEN HANDSET ALLIANCE. *Open Handset Alliance Releases Android SDK*, http://www.openhandsetalliance.com/press_111207.html, 12.11.2007, Aufruf Juni 2012.
- [7] ANDROID DEVELOPERS. *What is Android?*, <http://developer.android.com/guide/basics/what-is-android.html>, Mitte Juni 2012, Aufruf Juni 2012.
- [8] ARM. *ARM1176JZ-S Technical Reference Manual*, <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0333h/Cachfici.html>, Aufruf Juni 2012.
- [9] ANDROID OPEN SOURCE PROJECT. *Android Security Overview*, <http://source.android.com/tech/security/index.html>, Mitte Juni 2012, Aufruf Juni 2012.
- [10] ANDROID OPEN SOURCE PROJECT. *Android Security Overview - Linux Security*, <http://source.android.com/tech/security/index.html#linux-security>, Mitte Juni 2012, Aufruf Juni 2012.
- [11] ANDROID DEVELOPERS. *Platform Versions*, <http://developer.android.com/resources/dashboard/platform-versions.html>, Mitte Juni 2012, Aufruf Juni 2012.

- [12] ANDROID DEVELOPERS. *Android 4.0 Platform - Revisions*, <https://developer.android.com/sdk/android-4.0.html#relnotes>, Mitte Juni 2012, Aufruf Juni 2012.
- [13] ED BOTT - ZDNET. *Why Android updates are a mess: it's the business model*, <http://www.zdnet.com/blog/bott/why-android-updates-are-a-mess-its-the-business-model/4300>, 26.12.2012, Aufruf Juni 2012.
- [14] HEISE ZEITSCHRIFTEN VERLAG. *Google: 10 Milliarden Downloads von Android-Apps*, <http://www.heise.de/newsticker/meldung/Google-10-Milliarden-Downloads-von-Android-Apps-1391244.html>, 07.12.2011, Aufruf Juni 2012.
- [15] ANDROID DEVELOPERS. *Publishing on Google Play*, <http://developer.android.com/guide/publishing/publishing.html>, Mitte Juni 2012, Aufruf Juni 2012.
- [16] HIROSHI LOCKHEIMER, VP OF ENGINEERING, ANDROID. *Android and Security*, <http://googlemobile.blogspot.de/2012/02/android-and-security.html>, 02.02.2012, Aufruf Juni 2012.
- [17] JON OBERHEIDE. *Remote Kill and Install on Google Android*, <http://jon.oberheide.org/blog/2010/06/25/remote-kill-and-install-on-google-android/>, 25.06.2010, Aufruf Juni 2012.
- [18] ANDROID DEVELOPERS. *Security and Permissions*, <http://developer.android.com/guide/topics/security/security.html>, Mitte Juni 2012, Aufruf Juni 2012.
- [19] ANDROID DEVELOPERS. *The AndroidManifest.xml File: <manifest>*, <https://developer.android.com/guide/topics/manifest/manifest-element.html#uid>, Mitte Juni 2012, Aufruf Juni 2012.
- [20] ASAF SHABTAI, YUVAL FLEDEL, URI KANONOV, YUVAL ELOVICI AND SHLOMI DOLEV. *Google Android: A State-of-the-Art Review of Security Mechanisms*, <http://arxiv.org/pdf/0912.5101v1.pdf>, 27.12.2009, Aufruf Juni 2012.
- [21] STEFAN KLEMENT. *Sicherheitsaspekte der Google Android Plattform*, <http://www.informatik.uni-bremen.de/~sohr/papers/DiplomarbeitKlement.pdf>, 18.04.2011, Aufruf Juni 2012.
- [22] CLAUDIA ECKERT. *IT-Sicherheit, Konzepte - Verfahren - Protokolle, 4. Auflage*, Oldenbourg Verlag, München Wien 2006.
- [23] UBUNTU WIKI. *AppArmor*, <https://wiki.ubuntu.com/AppArmor>, 26.04.2012, Aufruf Juni 2012.
- [24] TOMOYO LINUX. *About TOMOYO Linux*, <http://tomoyo.sourceforge.jp/about.html.en>, 07.11.2011, Aufruf Juni 2012.
- [25] ANDROID DEVELOPERS. *The AndroidManifest.xml File: <permission>*, <http://developer.android.com/guide/topics/manifest/permission-element.html#plevel>, Mitte Juni 2012, Aufruf Juni 2012.

- [26] DALVIKVM.COM. *Dalvik Virtual Machine*, <http://www.dalvikvm.com/>, 2008, Aufruf Juni 2012.
- [27] ANDROID DEVELOPERS. *Designing for Security*, <http://developer.android.com/guide/practices/security.html>, Mitte Juni 2012, Aufruf Juni 2012.
- [28] ANDROID DEVELOPERS. *Android 3.0 Platform*, <https://developer.android.com/sdk/android-3.0.html>, Mitte Juni 2012, Aufruf Juni 2012.
- [29] VIRUSLIST.COM - DENIS MASLENNIKOV. *Die dunkle Seite des neuen Android Market*, <http://www.viruslist.com/de/weblog?weblogid=207319380>, 07.02.2011, Aufruf Juni 2012.
- [30] INFORMATIONSDIENST WISSENSCHAFT E. V.. *Neuartiger Ansatz enttarnt Datenmissbrauch auf mobilen Endgeräten*, <http://idw-online.de/pages/de/news486992>, 05.07.2012, Aufruf Juli 2012.
- [31] LOOKOUT MOBILE SECURITY. *Update: Android Malware Droid-Dream: How it Works*, <http://blog.mylookout.com/blog/2011/03/02/android-malware-droiddream-how-it-works/>, 02.03.2011, Aufruf Juni 2012.
- [32] LOOKOUT MOBILE SECURITY. *Technical Analysis - DroidDream Malware*, <http://blog.mylookout.com/droiddream/>, 04.02.2011, Aufruf Juni 2012.
- [33] ANTHONY MCKAY LINEBERRY. *Reversing Latest Exploit Release*, <http://dtors.org/2010/08/25/reversing-latest-exploit-release/>, 25.08.2010, Aufruf Juni 2012.
- [34] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI) / CONSECUR GMBH. *Einführung von Intrusion-Detection-Systemen*, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/IDS/Grundlagenv10_pdf.pdf?__blob=publicationFile, Aufruf Juni 2012.
- [35] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI). *IT-Grundschatz-Kataloge - M 2.157 Auswahl eines geeigneten Viren-Schutzprogramms*, <https://www.bsi.bund.de/ContentBSI/grundschatz/kataloge/m/m02/m02157.html>, Stand 2009, Aufruf Juni 2012.
- [36] HEISE ZEITSCHRIFTEN VERLAG. *Smartphone-Honeypots im Mobilfunknetz der Telekom*, <http://www.heise.de/open/meldung/Smartphone-Honeypots-im-Mobilfunknetz-der-Telekom-1630359.html>, 02.07.2012, Aufruf Juli 2012.
- [37] IKER BURGUERA AND URKO ZURUTUZA, SIMIN NADJM-TEHRANI. *Crowdroid: Behavior-Based Malware Detection System for Android*, <http://www.ida.liu.se/~rtslab/publications/2011/spsm11-burguera.pdf>, 17.10.2011, Aufruf Juni 2012.

Kapitel 2

IPv6-Sicherheit

Stefan Schiller

IPv6 ist heutzutage in nahezu allen Betriebssystemen implementiert. Im Gegensatz zu IPv4, das sich über den jahrelangen Betrieb hinweg weiterentwickelt und stabilisiert hat, kann sich IPv6 nicht auf eine solche fundierte Praxiserfahrung stützen. Dies trägt dazu bei, dass diverse Schwachstellen noch nicht behoben sind und für Angriffe ausgenutzt werden können. Um mit diesem Problem in geeigneter Form umgehen zu können, werden in dieser Arbeit zunächst die Grundlagen von IPv6 erläutert und mit den Techniken von IPv4 verglichen. Bei der Betrachtung der Sicherheit von IPv6 dürfen zudem die diversen Hilfsprotokolle, die den IPv6-Betrieb erst möglich machen, nicht außer Acht gelassen werden. Diese bilden zusammen mit IPv6 selbst ein komplexes System, das viele Möglichkeiten für einen Angriff bietet. Darunter fallen vor allem Denial of Service und Man-In-The-Middle Angriffe. Die einzelnen Angriffstechniken werden im Detail betrachtet und deren Funktionsweise beschrieben. Im Anschluss daran werden die Sicherheitsmechanismen, die IPv6 unterstützt, dargestellt und aufgezeigt, inwiefern diese genutzt werden können, um sich vor derartigen Angriffen zu schützen.

Inhaltsverzeichnis

2.1	Einleitung	35
2.2	IPv6 und Umfeld	36
2.2.1	Grundlagen IPv6	36
2.2.2	ICMPv6	39
2.2.3	Neighbor Discovery Protocol	40
2.2.4	DHCPv6	41
2.3	Angriffstechniken	42
2.3.1	Anonymität	42
2.3.2	Neighbor Discovery Spoofing	42
2.3.3	Duplicate Address Detection DoS	44
2.3.4	Router Faking	45
2.3.5	IPv6 in IPv4-Netzen	47
2.3.6	Schwachstellen bei der Umsetzung des TCP/IP-Stacks	47
2.4	Gegenmaßnahmen	48
2.4.1	Verschlüsselung und Authentifikation	48
2.4.2	IPSec	50
2.4.3	Wirksamkeit	51
2.5	Schluß	55
2.5.1	Fazit	55
2.5.2	Ausblick	56

2.1 Einleitung

In den späten 50iger Jahren erkannte das US-Verteidigungsministerium angesichts der Bedrohungen durch den Kalten Krieg die Notwendigkeit zur Erschaffung eines vom Telefonnetz unabhängigen Systems. Das aus diesen Bestrebungen entstandene ARPANET verknüpfte zu Beginn - im Jahre 1969 - vier verschiedene Forschungseinrichtungen miteinander [2]. Durch die Anbindung weiterer Netzteilnehmer nahm die Größe des ARPANET rasant zu. Das Netz, das anfangs nur einigen wenigen Auserwählten vorbehalten war, entwickelte sich zum weltweit genutzten Internet und ermöglicht heute Milliarden von Menschen die Kommunikation untereinander. Diese schnelle Ausbreitung und der Wandel vom einst privaten, überschaubaren Netz zum größten öffentlichen Netz der Welt birgt neben vielen Möglichkeiten auch Probleme in sich. Beim Austausch von Daten zwischen vier einzelnen Forschungseinrichtungen spielte Netzsicherheit nur eine untergeordnete Rolle, so dass dies in den damals entwickelten Technologien nicht vorgesehen wurde. Heutzutage gibt es eine Vielzahl von Netzteilnehmern, deren Identität nicht immer vollständig bekannt ist. Zudem macht die Nutzung des Internets für finanzielle Geschäfte und andere sicherheitsrelevante Dienste für mögliche Angreifer lukrativ, Informationen zu manipulieren oder zu stehlen. Um diesen Risiken entgegenzutreten, wurden erst im Nachhinein Sicherheitsmechanismen in die bereits etablierten Protokolle integriert. So sind im IPv4 Standard selbst keine Sicherheitsaspekte vorgesehen. Erst optionale Erweiterungen (IPsec) oder Protokolle höherer Schichten (SSL, TLS, SSH) sollen diesen Aspekt abdecken. Das darüber hinaus wachsende Problem der Adressknappheit führte zum Entwurf des IPv6 Standards. Hierbei ist IPsec ein integraler Bestandteil der Architektur, so dass Netzsicherheit bereits auf der Vermittlungsschicht eingebunden ist. Mit IPv6 sollen die zur Erstellung des IPv4 Standards noch nicht bekannten Anforderungen und die während des Betriebs mit IPv4 gewonnenen Erkenntnisse genutzt werden, um sie direkt in das Protokoll zu übernehmen. Die IPv4 Umgebung hat sich über viele Jahre hinweg weiterentwickelt und stabilisiert. Hingegen bildet IPv6 zusammen mit anderen dafür entworfenen Protokollen ein völlig neues System, dessen Komplexität nicht zu unterschätzen ist. Neben neuen Möglichkeiten bringt dies auch neue Risiken mit sich, denn ein neu entworfenes System ist auf Grund mangelnder Erprobung anfällig für Fehler. Ziel dieser Arbeit ist es, die Risiken, die IPv6 mit sich bringt, zu beleuchten und - wo möglich - aufzuzeigen, wie sie durch geeignete Sicherheitsmechanismen überwunden werden können.

In Abschnitt 2.2 werden zunächst die Grundlagen von IPv6 betrachtet und aufgezeigt, welche Neuerungen im Vergleich zu IPv4 hinzu gekommen sind. Dazu werden neben IPv6 selbst die unterschiedlichen Hilfsprotokolle, die zum Umfeld von IPv6 gehören, vorgestellt. Im Anschluß daran, werden in Abschnitt 2.3 verschiedene Angriffstechniken im Detail erläutert. Dazu wird jeweils die zu Grunde liegende Schwachstelle erläutert und darauf aufbauend gezeigt, wie der Angriff umgesetzt wird. Im darauf folgenden Abschnitt 2.4 wird zu Beginn dargestellt, welche Sicherheitsmechanismen IPv6 unterstützt. Daran anschließend wird beschrieben, inwiefern diese genutzt werden können, um die beschriebenen Angriffe zu vermeiden. Den Schluß bildet der Abschnitt 2.5. Darin werden die aus der Arbeit resultierenden Erkenntnisse in einem Fazit zusammengefasst und ein Ausblick auf die weitere Entwicklung des Themas IPv6 und Sicherheit gegeben.

2.2 IPv6 und Umfeld

Bei der Betrachtung von IPv6 spielt nicht nur der IPv6 Standard selbst eine Rolle. Es sind darüber hinaus diverse Hilfsprotokolle und Protokolle höherer Schichten beteiligt, die erst im Zusammenspiel einen IPv6-Betrieb möglich machen. Daher werden im Folgenden IPv6 und die damit in Verbindung stehenden Protokolle eingehender betrachtet.

2.2.1 Grundlagen IPv6

IPv6 befindet sich - ebenso wie IPv4 [8] - auf der Schicht 3 des OSI-Schichten-Modells (Vermittlung). Somit ist es für die Pfadschaltung zwischen zwei Endsystemen zuständig. Dazu werden netzglobale Adressen verwendet, die die Endsysteme eindeutig identifizieren. Darüber hinaus ist diese Schicht für die Aushandlung der Dienstgüte (*Quality of Service*) zuständig [22]. Zunächst werden Adress- und Headeraufbau dargestellt, die sich im Vergleich zu IPv4 wesentlich geändert haben.

Adressaufbau

Eine IPv6-Adresse ist 128 Bit groß und somit wesentlich größer als eine 32 Bit IPv4-Adresse. Bis auf einige Sonderfälle¹ bilden die ersten 64 Bit dabei das *Netzsegment* und die letzten 64 Bit den *Interface Identifier* [11]. Im Gegensatz zu einer IPv4-Adresse, bei der jedes Byte als Dezimalzahl dargestellt wird, wird eine IPv6-Adresse als 2-Byte-Blöcke hexadezimal dargestellt. Abbildung 2.1 zeigt beispielhaft eine IPv6-Adresse.

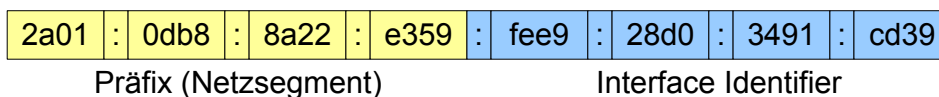


Abbildung 2.1: IPv6-Adresse

Der *Interface Identifier* ist nach dem EUI-64 Format aufgebaut. EUI-64 ist eine eindeutige Identifikation, die vom *Institute of Electrical and Electronics Engineers (IEEE)* definiert wurde. Die Erzeugung eines *Interface Identifiers* aus der EUI-64-Adresse ist in RFC2373 beschrieben [10]. Bei der Verwendung eines Ethernet-Interface werden die 64 Bit des *Interface Identifiers* in der Regel direkt aus den 48 Bit der MAC-Adresse gebildet. Auf die daraus resultierenden Probleme werden wir in Abschnitt 2.3.1 eingehen.

Adressbereiche

Ziel bei dem Entwurf von IPv6 war es, die Struktur der Adressen für das Routing zu optimieren. Die bei IPv4 verwendeten Broadcast-Adressen wurden dabei völlig abgeschafft.

¹Ähnlich wie bei IPv4 gibt es Sonderadressen, denen eine bestimmte Bedeutung zukommt (z.B. stellt 0:0:0:0:0:0:1, abgekürzt ::1, die Loopback-Adresse dar).

Stattdessen werden genauer differenzierbare Multicast-Adressen genutzt. Insgesamt unterscheidet IPv6 drei verschiedene Adresstypen: Unicast, Multicast und Anycast [7].

Unicast-Adressen dienen zur Kommunikation mit genau einem anderen Netzknoten. Mit einer Multicast-Adresse werden alle Netzknoten adressiert, die der jeweiligen Multicast-Gruppe angehören. Multicast-Adressen sind nur lokal. Auch die Anycast-Adresse bezieht sich auf eine Gruppe von Netzknoten. Im Gegensatz zum Multicast wird das Paket aber nur an ein Mitglied der Gruppe gesendet (z.B. den nächsten erreichbaren Knoten der Gruppe). Abbildung 2.2 verdeutlicht die drei Kommunikationsarten.

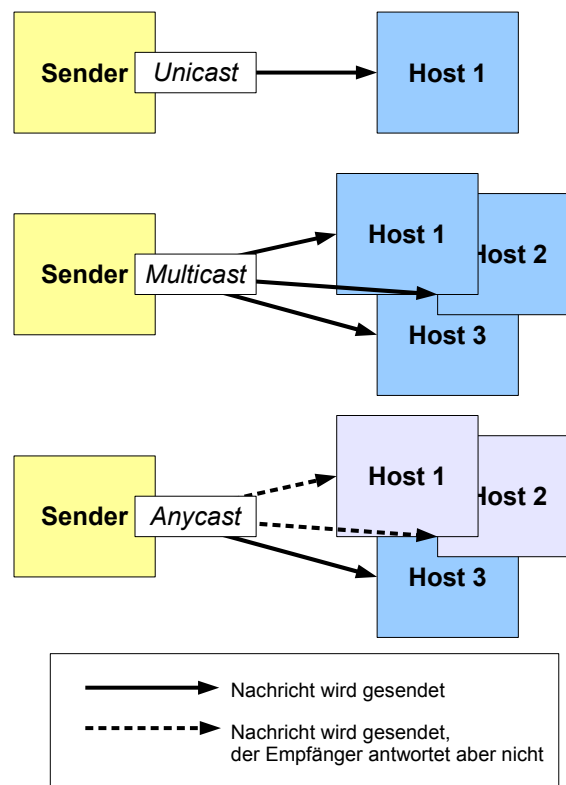


Abbildung 2.2: Uni-, Multi- und Anycast

Einem Endknoten sind folgende Adressen zugewiesen:

- Link-local Adresse des Interfaces
- Aggregierbare Unicast-Adresse
- Loopback-Adresse (::1/128)
- Alle-Knoten-Multicast-Adresse am gleichen Knoten
- Alle-Knoten-Multicast-Adresse am gleichen Link
- Multicast-Adressen aller anderen zugehörigen Gruppen

Tabelle 2.1: Adressbeispiel

Adresse	Beispiel
Link-local	FE80::FEE9:28D0:3491:CD39
Aggregierbarer Unicast	4A00:0000:0122:0001:FEE9:28D0:3491:CD39
Loopback	::1
Alle-Knoten-Multicast gleicher Knoten	FF01::1
Alle-Knoten-Multicast gleicher Link	FF02::1

Die Link-local Adresse wird im lokalen Netz verwendet, um automatisch Adressen zu konfigurieren und andere Netzteilnehmer ausfindig zu machen. Auch für den Fall, dass es im Netz keinen Router gibt, wird sie verwendet. Pakete mit diesem Empfänger werden von Routern nicht weitergeleitet. Die aggregierbare Adresse ersetzt die von IPv4 bekannte öffentliche Adresse. Pakete mit diesem Empfänger verlassen somit auch das eigene Teilnetz. Die Loopback-Adresse dient (wie bei IPv4 [8]) dazu, das eigene Interface zu adressieren. Darüber hinaus verfügt ein Endknoten über verschiedene Multicast-Adressen. Den kleinsten Bereich stellt die Alle-Knoten-Multicast-Adresse am gleichen Knoten dar. Diese bezieht sich nur auf einen Knoten selbst und dient z.B. zum Test bei der Erstellung von Anwendungen, die Multicast-Adressen verwenden. Die Alle-Knoten-Multicast-Adressen am gleichen Link entspricht einem Broadcast innerhalb des lokalen Netzes. Sollte der Knoten weiteren Multicast-Gruppen angehören, so erhält er auch Pakete, die diese Multicast-Adresse als Empfänger enthalten. Falls der Knoten ein Router ist, so besitzt er noch weitere Anycast-Adressen. Tabelle 2.1 zeigt beispielhaft die Adressen eines Endknotens.

Header

Der Headeraufbau hat sich bei IPv6 im Vergleich zu IPv4 fundamental geändert. Wie in Abbildung 2.3 zu sehen ist, enthält der IPv6-Header deutlich weniger Felder, als der IPv4-Header [11]. Dies liegt daran, dass zusätzliche Informationen nur dann eingebunden werden, wenn sie auch benötigt werden.

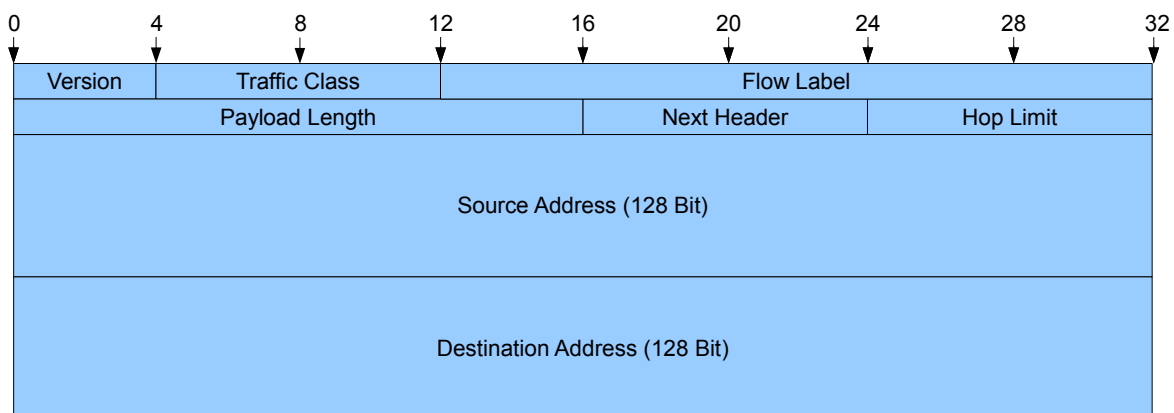


Abbildung 2.3: IPv6-Header

Den einzelnen Feldern kommt folgende Bedeutung zu:

- *Version*: Versionsnummer (hier steht eine 6, bei IPv4: 4).
- *Traffic Class*: Prioritätswert (wird für Quality of Service verwendet).
- *Flow Label*: Wert identifiziert einen Flow (Datenfluss), für den spezielle Behandlung gewünscht wird (z.B. den Video- oder Audio-Datenstrom einer Videokonferenz).
- *Payload Length*: Größe des Paketinhalts in Byte (ohne Header).
- *Next Header*: Typ des nächsten Headers (siehe unten).
- *Hop Limit*: Verbleibende Lebensdauer (entspricht TTL bei IPv4).
- *Source Address*: Adresse des Senders.
- *Destination Address*: Adresse des Empfängers.

Das Feld *Next Header* gibt Auskunft darüber, welcher Header dem IPv6-Header folgt. Dies kann entweder ein *Extension Header* sein (für zusätzliche Informationen wie z.B. Routing oder Fragmentierung) oder ein Header der nächst höheren Schicht (z.B. TCP oder UDP). Die *Extension Header* selbst besitzen auch ein *Next Header* Feld, so dass die Hintereinanderkettung mehrerer Header möglich ist. Abbildung 2.4 zeigt ein IPv6-Paket, das über zusätzliche Routing- und Fragmentierungsinformationen verfügt. Dahinter folgt ein UDP-Header und die eigentlichen Nutzdaten [5].

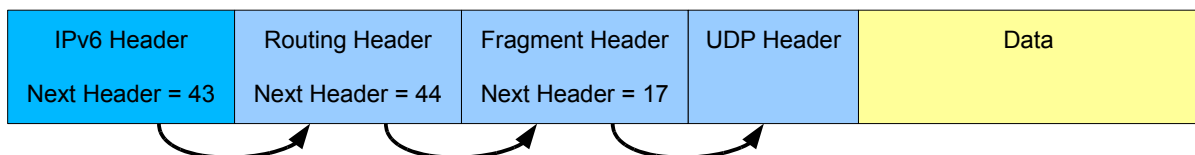


Abbildung 2.4: IPv6-Paket nach van Hauser

2.2.2 ICMPv6

Das Internet Control Message Protocol Version 6 (ICMPv6) ist eins der Hilfsprotokolle, das zum Umfeld von IPv6 gehört. Es dient der Übertragung von Statusinformationen und Fehlermeldungen. Es stellt die IPv6-Variante des schon bei IPv4 verwendeten ICMP dar. Das Protokoll arbeitet wie IPv6 selbst auf Schicht 3 (Vermittlung). Im Gegensatz zu IPv4 ist ICMPv6 für den IPv6-Betrieb zwingend notwendig [6] [18].

ICMP definiert verschiedene Nachrichtentypen, die aufgeteilt sind in Fehlermeldungen (Werte von 0-127) und Statusinformationen (128-255). Tabelle 2.2 zeigt die für die weitere Betrachtung relevanten Nachrichtentypen.

Tabelle 2.2: ICMP-Nachrichtentypen

Typ	Name	Bedeutung
1	<i>Destination Unreachable</i>	Die angegebene Zieladresse ist nicht erreichbar.
2	<i>Packet Too Big</i>	Die MTU (Maximum Transmit Unit) wurde überschritten.
...		
133	<i>Router Solicitation</i>	Router-Anfrage.
134	<i>Router Advertisement</i>	Bekanntgabe eines Routers.
135	<i>Neighbor Solicitation</i>	Anfrage nach MAC-Adresse eines Nachbarn am gleichen Link.
136	<i>Neighbor Advertisement</i>	Bekanntgabe der MAC-Adresse eines Nachbarn am gleichen Link.
...		

2.2.3 Neighbor Discovery Protocol

Das Neighbor Discovery Protocol wird im lokalen Netz (LAN) von Netzgeräten und Routern verwendet, um folgende Zwecke zu erfüllen [14]:

1. Die Anwesenheit anderer Netzteilnehmer feststellen (*ND - Neighbor Discovery*).
2. Die zugehörige physikalische Adresse (Link-Layer-Adresse) ermitteln (*Address Resolution*).
3. Verwalten von Erreichbarkeitsinformationen zu aktiven Nachbargeräten (*NUD - Neighbor Unreachability Detection*).
4. Router im lokalen Netz ermitteln (*RD - Router Discovery*).
5. Automatische Adressbeziehung (*Address Autoconfiguration*).
6. Erkennung von Adressüberschneidungen (*DAD - Duplicate Address Detection*).
7. Umleitung (*Redirection*).

Die dafür notwendigen Informationen werden mittels ICMPv6 ausgetauscht. Dazu werden die in Abschnitt 2.2.2 erwähnten ICMPv6-Typen genutzt. NDP ersetzt das bei IPv4 verwendete ARP, das eine Zuordnungstabelle von MAC- zu IP-Adressen verwaltet. Auch bei IPv6 wird diese Zuordnung durchgeführt. Soll ein Paket an eine noch unbekannte IP-Adresse gesendet werden, sendet der Host ein ICMPv6-Paket vom Typ 135 an die Link-Layer-Multicast-Adresse. Dies entspricht einer *ARP Request* an die Broadcast-Adresse FF-FF-FF-FF-FF-FF bei ARP. Der Typ 135 beschreibt eine Nachbaranfrage (*Neighbor Solicitation*). Abbildung 2.5 verdeutlicht die Anfrage in einem Beispiel: Host 1 möchte an Host 3 ein Paket senden. In der Zuordnungstabelle von Host 1 ist noch kein Eintrag für die IP von Host 3 vorhanden, so dass er eine Nachbaranfrage stellt.

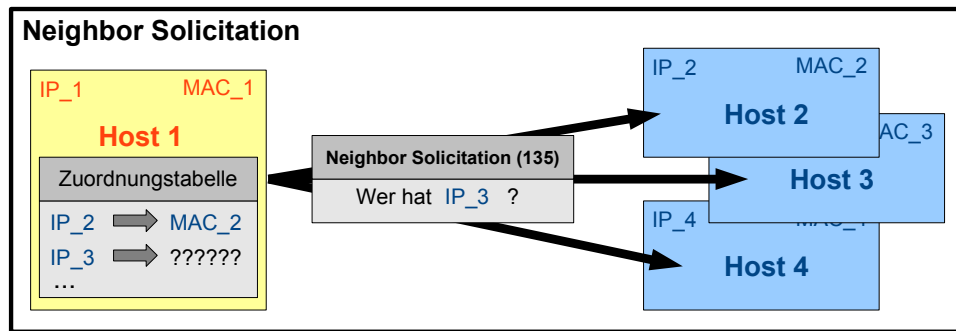


Abbildung 2.5: Neighbor Solicitation

Auf die Anfrage antwortet der zur IP-Adresse gehörige Host mittels eines *Neighbor Advertisement* Pakets (Typ 136). Bei ARP antwortet der Host mit einem *ARP Reply*. Die jeweilige IP-Adresse wird vom anfragenden Host in die Zuordnungstabelle eingetragen. Abbildung 2.6 zeigt die Antwort von Host 3 für obiges Beispiel.

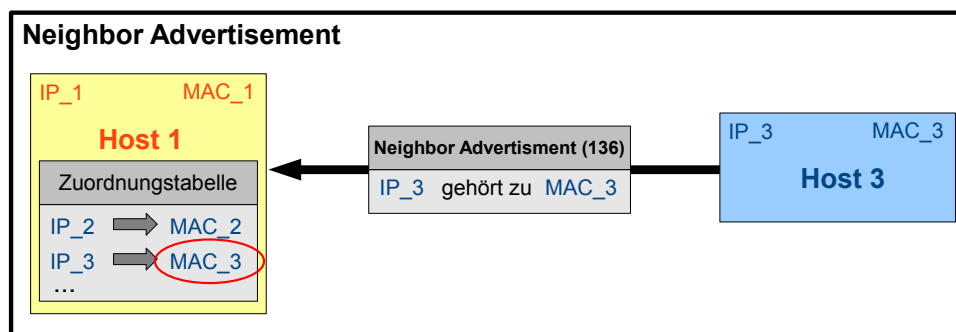


Abbildung 2.6: Neighbor Advertisement

Auf diese Weise kann der Host das Paket an die zur MAC-Adresse gehörenden IP-Adresse senden. Wie bereits an oben stehender Aufzählung zu sehen ist, umfasst NDP weitere Funktionalitäten, auf deren Schwachstellen in Abschnitt 2.3 näher eingegangen wird.

2.2.4 DHCPv6

Wie in Abschnitt 2.2.3 beschrieben, ist die automatische Adressbeziehung (*Autoconfiguration*) Teil des Neighbor Discovery Protocol. Diese Aufgabe wurde bei IPv4 von DHCP übernommen. Die Autokonfiguration sorgt zwar für die Zuweisung einer IP-Adresse, allerdings ermöglicht es nicht, weitere Informationen an den Client zu übertragen. Dies ist allerdings notwendig, um den Client über den zu verwendenden DNS-Server zu informieren. Bei IPv4 ist dies mittels DHCP möglich, so dass für IPv6 auch ein eigenes dafür entworfenes Protokoll - DHCPv6 [12] - umgesetzt wurde. Eine Alternative, die auf DHCPv6 verzichtet, ist in RFC 6106 [21] beschrieben.

2.3 Angriffstechniken

Nach der Betrachtung der verschiedenen Eigenschaften und Verfahren bei IPv6 widmet sich das folgende Kapitel nun den unterschiedlichen Angriffstechniken. Ein Teil der grundsätzlichen Prinzipien ist bereits aus dem IPv4-Umfeld bekannt. Der andere Teil ist erst durch die Einführung von IPv6 entstanden.

2.3.1 Anonymität

Ein Problem, das bei IPv4 so noch nicht bestand, betrifft die Anonymität. Jedes Netzinterface besitzt eine global eindeutige MAC-Adresse, mit der das Interface identifiziert werden kann. Bei IPv4 bestand keine feste Verbindung zwischen IP-Adresse und MAC-Adresse. Beim Routing von Paketen durch das Internet wird die MAC-Adresse stets durch die MAC-Adresse des jeweiligen Routers ersetzt. Auf Grund dessen bleibt die MAC-Adresse des Senders einem möglichen Angreifer, der ein Paket über das Internet empfängt, verborgen.

Wie in Abschnitt 2.2.1 beschrieben, ist ein Teil der IPv6-Adresse (der *Interface Identifier*) direkt aus der MAC-Adresse abgeleitet. Somit kann ein Host über mehrere Netze hinweg eindeutig identifiziert werden. Da sich der *Interface Identifier* nicht ändert, bleibt die Adresse eindeutig. Beispielsweise könnten Webseiten auf diese Weise ein detailliertes Profil des jeweiligen Gerätes erstellen. Abschnitt 2.4.3 geht auf Gegenmaßnahmen zu diesem Sicherheitsproblem ein.

2.3.2 Neighbor Discovery Spoofing

Wie in Abschnitt 2.2.3 beschrieben, ersetzt NDP bei IPv6 die Funktionalität von ARP. Eine seit langem bekannte Schwachstelle von ARP ist das ARP-Spoofing [1]. Dabei gibt ein Angreifer, der sich im lokalen Netz befindet, dem Opfer vor, dass die IP-Adresse des Internet-Gateways (Router) über die MAC-Adresse des Angreifers zu erreichen ist. Darüber hinaus gibt er dem Router vor, dass das Opfer ebenfalls über die MAC-Adresse des Angreifers zu erreichen ist. Pakete, die vom Opfer an den Router gesendet werden sollen, erhält stattdessen der Angreifer. Dieser kann die Pakete mitschneiden oder sogar manipulieren und anschließend an den Router weiterleiten. Da Pakete vom Router an das Opfer ebenfalls an den Angreifer gesendet werden, erhält der Angreifer auch die entsprechenden Antwort-Pakete und kann diese an das Opfer weiterleiten. Auch hierbei können die Pakete vorher manipuliert werden. Auf diese Weise merkt das Opfer nichts von dem Mitschneiden bzw. der Manipulation des Angreifers. Es handelt sich um einen klassischen Man-In-The-Middle-Angriff.

Auch bei IPv6 und NDP ist ein derartiger Angriff möglich. Dazu wird ein Beispiel dargestellt, bei dem der Datenfluss zwischen einem Opfer und einem Router umgeleitet wird. Abbildung 2.7 zeigt den unbeeinflussten Datenfluss zwischen Opfer und Router sowie deren Zuordnungstabellen.

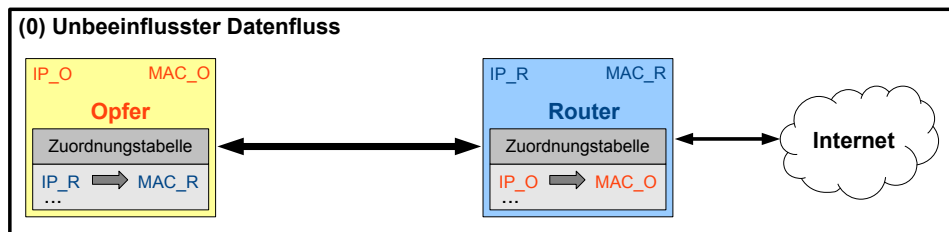


Abbildung 2.7: Opfer und Router tauschen Daten aus

Der Angreifer sendet nun dem Opfer ein gespooftes *Neighbor Advertisement* Paket (ICMPv6-Type 136), in das er als Quelladresse die zu übernehmende IPv6-Adresse (im Beispiel die des Routers) einträgt. Das Opfer übernimmt diesen Eintrag in die Zuordnungstabelle (Abbildung 2.8).

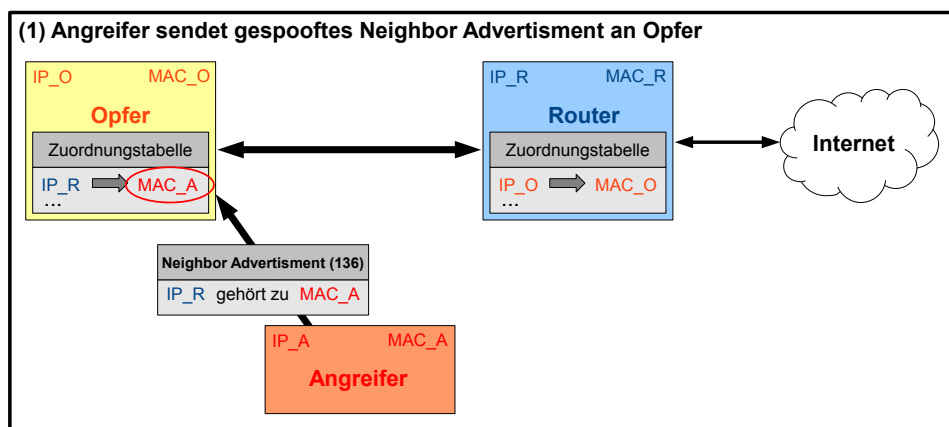


Abbildung 2.8: Angreifer nimmt Einfluss auf Zuordnungstabelle des Opfers

Anschließend sendet der Angreifer auch dem Router ein gespooftes *Neighbor Advertisement* Paket, so dass der Router in seiner Zuordnungstabelle für die IP-Adresse des Opfers die MAC-Adresse des Angreifers einträgt (Abbildung 2.9).

Auf diese Weise sorgt der Angreifer dafür, dass Pakete, die vom Opfer an den Router gesendet werden sollen, an ihn gelangen. Auch Pakete vom Router an das Opfer werden an ihn geleitet. Dies kann entweder als *Denial of Service (DoS)* Angriff eingesetzt werden oder als *Man-In-The-Middle* Angriff. Dazu leitet der Angreifer die Pakete lediglich an den eigentlichen Empfänger weiter (Abbildung 2.10).

Wie im Beispiel gezeigt übernimmt ein Host auch *Neighbor Advertisement* Einträge, die er nicht explizit angefragt hat. Dies dient dazu seine Zuordnungstabelle möglichst aktuell zu halten. Der Angreifer muss also gar nicht erst auf eine Anfrage (*Neighbor Solicitation*) warten.

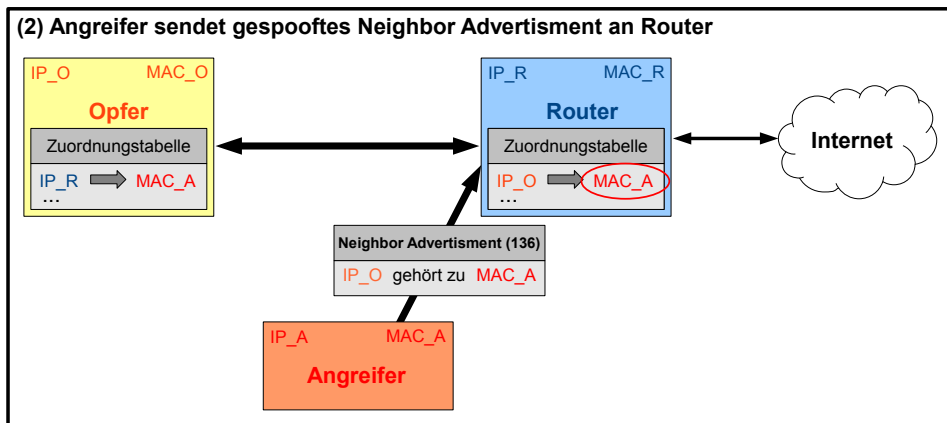


Abbildung 2.9: Manipulation des Routers

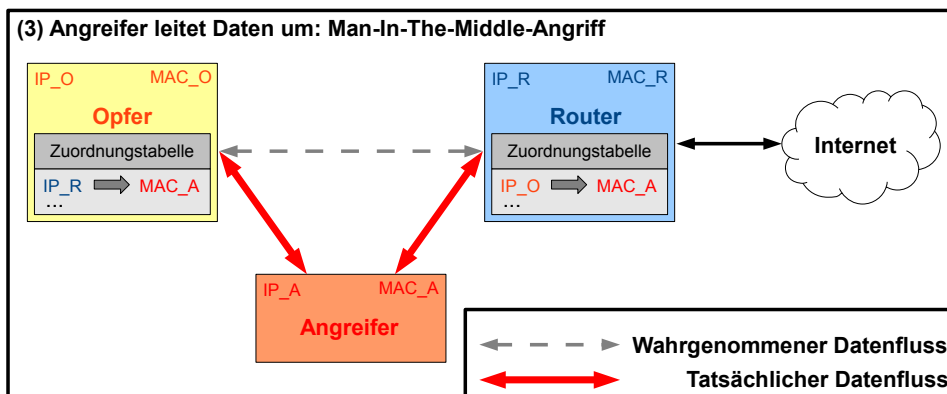


Abbildung 2.10: Neighbor Discovery Spoofing

2.3.3 Duplicate Address Detection DoS

Eine IP-Adresse muss einem einzigen Interface eindeutig zugewiesen sein. Es darf also jede IP-Adresse nur einmal im jeweiligen Netz vergeben sein. Um Adresskonflikte zu erkennen, wurde bei IPv4 ARP genutzt. Dazu sendet ein Host, der dem Netz beitreten möchte, eine ARP-Request [9]. So ermittelt er, ob die IP-Adresse, die er verwenden möchte, bereits vergeben ist. Wenn er keine Antwort auf die Anfrage erhält, scheint die Adresse noch nicht vergeben zu sein und er kann sie nutzen. Wie in Sektion 2.2.3 aufgelistet, sorgt bei IPv6 NDP für die Erkennung von Adressüberschneidungen (*Duplicate Address Detection*). Das Prinzip ähnelt dem oben beschriebenen Verfahren bei IPv4. Sobald ein neuer Host dem Netz beitrifft, sendet er eine *Neighbor Solicitation* an die Link-Layer-Multicast-Adresse und fragt auf diese Weise an, ob die IP-Adresse, die er verwenden möchte, bereits vergeben ist. Abbildung 2.11 zeigt die Anfrage des neuen Hosts.

Sollte ein Gerät antworten, so weiss der neue Host, dass die IP-Adresse bereits vergeben ist und muss eine andere wählen. Beim Duplicate Address Detection DoS antwortet ein Angreifer auf eine solche Anfrage mit einem *Neighbor Advertisement* Paket, wie in Abbildung 2.12 dargestellt.

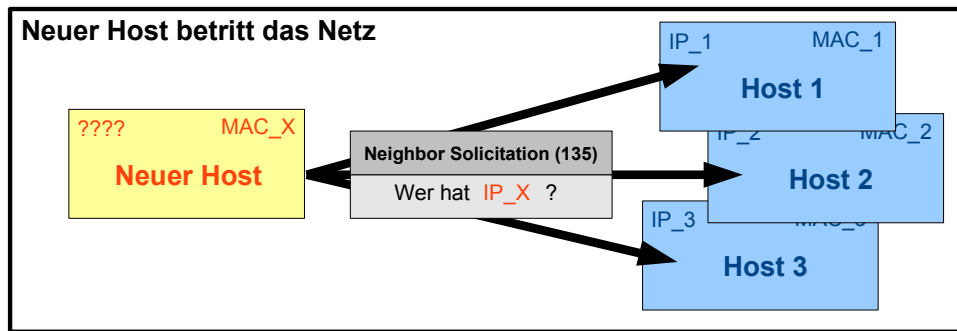


Abbildung 2.11: IP-Adresse frei zur Nutzung?

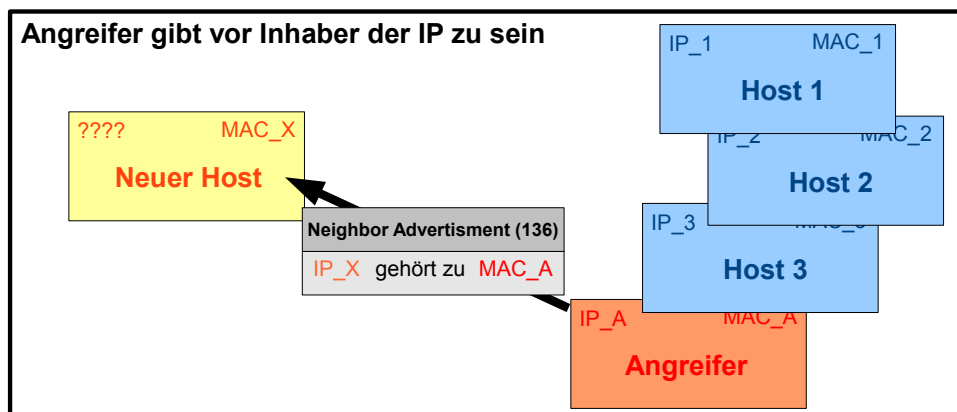


Abbildung 2.12: Angreifer blockiert neuen Host (DoS)

Er gibt also vor, die Adresse, die der Host verwenden möchte, selbst bereits zu verwenden. Der neue Host wird nun eine andere IP-Adresse wählen und erneut eine *Neighbor Solicitation* senden, um festzustellen, ob die neue IP-Adresse verwendet werden kann. Auch auf diese Anfrage und weitere darauf folgende Anfragen antwortet der Angreifer mit einem *Neighbor Advertisement*. Auf diese Weise kann sich der Host keine gültige Adresse zuweisen und somit dem Netz nicht beitreten [13].

2.3.4 Router Faking

Ein Host, der einem lokalen Netz beitreten und das Internet nutzen möchte, muss den Default-Router ausfindig machen. Auch dies geschieht mittels NDP (siehe 2.2.3, *Router Discovery*).

Abbildung 2.13 zeigt, in welchen Schritten dies geschieht. Zunächst sendet der Host eine *Router Solicitation* Nachricht an die Router-Multicast-Adresse (1). Der Router antwortet dem Host mit einem *Router Advertisement* und gibt dabei eine Prioritätsstufe an (2). Die Prioritätsstufe kann niedrig, mittel oder hoch sein und ist in RFC 4191 [16] spezifiziert. Standardmäßig ist sie mittel. Nach dem Erhalten der Antwort übernimmt der Host den Router als seinen Default-Router (3).

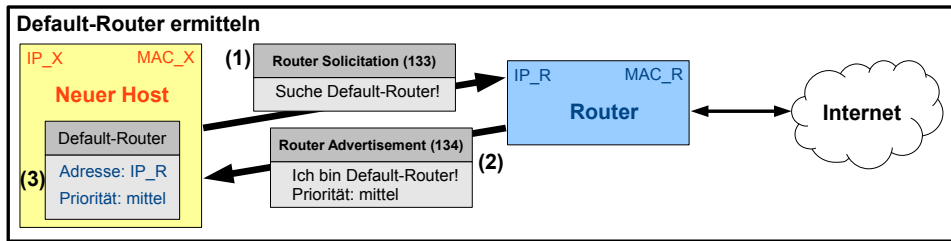


Abbildung 2.13: Router Discovery

Ein Angreifer kann vertäuschen selbst ein Router zu sein, indem er ein *Router Advertisement* Paket sendet. Dabei setzt er die Priorität auf hoch, um sicherzustellen, dass der Host ihn als Default-Router übernimmt. Abbildung 2.14 stellt diesen als Router Faking bekannten Vorgang dar.

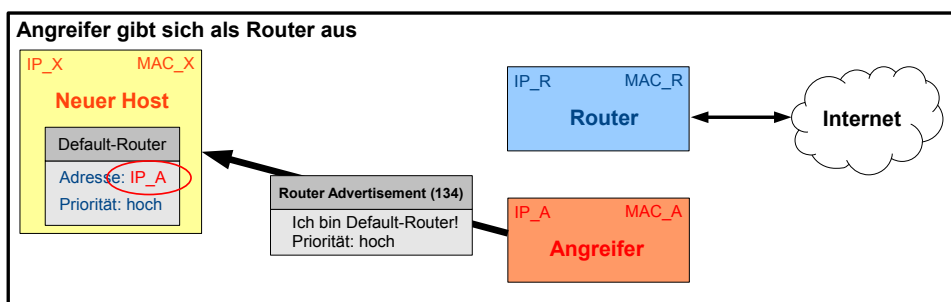


Abbildung 2.14: Router Faking

Dieser Angriff funktioniert nur, wenn der Default-Router des Hosts nicht schon selbst die Prioritätsstufe hoch besitzt. In einem solchen Fall ändert der Host seinen Default-Router nicht. Doch auch das kann der Angreifer umgehen. Das ICMPv6 Packet vom Type 134 (*Router Advertisement*) enthält ein Feld *Router Lifetime*, in dem der Router angibt, wie lange er in der Liste der Default Router verbleiben soll [19]. Der Angreifer spooft ein *Router Advertisement* Paket, in das er als Absender den eigentlichen Default-Router schreibt. Das Feld *Router Lifetime* setzt er auf 0. Der Host, der das Paket empfängt, geht davon aus, dass sein Default-Router nicht länger Default-Router ist (*Lifetime*: 0) und entfernt ihn (Abbildung 2.15).

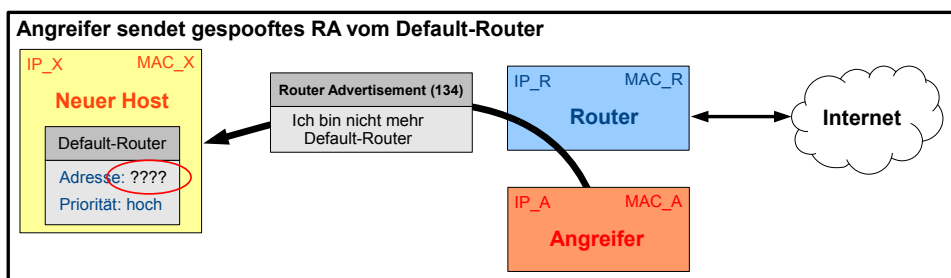


Abbildung 2.15: Default-Router entfernen

Nun braucht der Angreifer lediglich erneut ein *Router Advertisement* Paket senden und damit vorgeben, selbst Default-Router zu sein. Diesmal übernimmt der Host den Eintrag.

2.3.5 IPv6 in IPv4-Netzen

Ein Großteil der zur Zeit bestehenden Netze nutzt noch IPv4. Dennoch ist IPv6 bereits in den meisten Betriebssystemen implementiert. Die Systeme können also mit beiden Protokollen umgehen (Dual-Stack). Dies führt dazu, dass auch Netze, die nicht speziell für die Verwendung von IPv6 konfiguriert sind, angreifbar sind. Zum einen betrifft das Firewalls. Eine Firewall, die nur mit IPv4 arbeitet, kann leicht umgangen werden. Eine Art dies zu tun ist *Tunneling*. Viele Provider bieten Ihren Kunden noch keinen eigenen IPv6-Zugang an. Um dennoch mittels IPv6 im Internet zu agieren, kann beispielsweise das Kommunikationsprotokoll Teredo genutzt werden. Dabei werden die Daten des IPv4-Hosts mittels IPv4 und UDP an einen Teredo-Server gesendet, der an das IPv4- und IPv6-Netz angeschlossen ist. Dieser speist sie dann als IPv6-Pakete in das IPv6-Netz ein. Über einen Teredo-Relay werden die Daten dann wieder an den IPv4-Host gesendet [17]. Durch das Kapseln der Daten wird die IPv4-Firewall umgangen. Auch die generelle Einschränkung, dass Hosts hinter einem NAT-Router keine Verbindungen von außen akzeptieren können, wird damit umgangen [24].

Ein weiteres Problem sind Man-In-The-Middle-Angriff. Dabei täuscht ein Angreifer durch das Senden von *Router Advertisement* Paketen vor, ein IPv6-Router zu sein und leitet sämtlichen Datenverkehr ins Internet weiter. Auf diese Weise kann der Angreifer alle Daten mitlesen, ohne dass die Betroffenen etwas davon mitbekommen. Eine Umsetzung für Windows 7 nennt sich *SLAAC Attack* [28].

2.3.6 Schwachstellen bei der Umsetzung des TCP/IP-Stacks

Die Grundlage der bisher dargestellten Angriffstechniken bilden Schwachstellen im IPv6-Protokolldesign selbst. Im Folgenden Abschnitt werden beispielhaft Schwachstellen vorgestellt, die sich durch die Umsetzung des Protokolls ergeben.

RA Flooding

Eine Angriffstechnik nutzt *Router Advertisement* Pakete, um Teilnehmer im Netz lahm zu legen. Ein IPv6-Netz kann mehrere Router beinhalten. Sobald ein Router in einem *Router Advertisement* Paket ein Netz bekannt gibt, konfiguriert ein Host eine IPv6-Adresse für dieses Netz. Der Angreifer sendet nun eine Vielzahl von *Router Advertisement* Paketen, in denen er unterschiedliche Netze bekannt gibt. Dies führt dazu, dass der Host überlastet wird und unter Umständen völlig lahmgelegt ist. Betroffen von diesem Problem waren Cisco ASA/PIX, Cisco IOS, Windows 2008, 7, Vista und einige ältere Versionen von Linux. Mittlerweile ist diese Schwachstelle größten Teils behoben [5].

Linux Kernel: IPv6-Stack

Die IPv6 Implementierung des Linux Kernels vor Version 3.1 generiert beim Versand von fragmentierten Paketen nicht für jeden Empfänger eigenständige Identifikationsnummern. Ein Angreifer kann dies ausnutzen, um die Nummer vorherzusagen und präparierte Pakete an das Opfer zu senden, die seine Netzressourcen erheblich belasten [29].

Cisco ASA/ASASM

Verschiedene Cisco Geräte verarbeiten IPv6-Pakete auf unsichere Weise. Wenn eine bestimmter Modus aktiviert ist, kann ein Angreifer mittels präparierter IPv6-Pakete das Gerät zum Neustart bringen [30].

Windows: TCP/IP Stack

Ein Fehler in der Verarbeitung von IPv6-Paketen kann den Kernel von Windows zum Absturz bringen. Dabei werden die Pakete mit einem speziell aufbereiteten Extension Header erweitert und an das Opfer gesendet. Bereits die Zusendung einer kleinen Zahl von präparierten Paketen reicht, um einen Denial of Service auszulösen [31].

2.4 Gegenmaßnahmen

Nach der Betrachtung der verschiedenen Angriffstechniken bei IPv6, werden nun möglichen Gegenmaßnahmen vorgestellt. Dazu werden zunächst zwei generelle Verfahren zum Schutz vor Angriffen vorgestellt: Verschlüsselung und Authentifikation. Im Anschluss daran wird dargestellt, wie diese Verfahren in IPv6 mittels dem integralen Bestandteil IPSec verwendet werden können.

2.4.1 Verschlüsselung und Authentifikation

Verschlüsselung

Ein Problem bei der Übertragung von Daten über ein unsicheres Medium ist das Mithören der Daten durch einen Angreifer. Um dies zu verhindern, sollten die Daten verschlüsselt werden. Die Verschlüsselung erfolgt mittels eines Schlüssels, den der Angreifer nicht kennt oder nur mit extrem hohem Aufwand ermitteln könnte. Es wird zwischen symmetrischer und asymmetrischer Verschlüsselung unterschieden. Bei der symmetrischen Verschlüsselung verwenden beide Kommunikationspartner den selben Schlüssel. Dieser muss zuvor

ausgetauscht werden². Bei der asymmetrischen Verschlüsselung besitzt jeder Kommunikationspartner einen öffentlichen Schlüssel (*Public Key*) und einen privaten Schlüssel (*Private Key*). Der öffentliche Schlüssel wird von einer vertrauenswürdigen Zertifizierungsstelle vergeben und ist jedem zugänglich. Der private Schlüssel ist nur dem Besitzer bekannt. Möchte Alice eine Nachricht an Bob versenden, so nutzt sie den öffentlichen Schlüssel von Bob um die Nachricht zu verschlüsseln. Bob, der die verschlüsselte Nachricht empfängt, nutzt seinen privaten Schlüssel um die Nachricht wieder zu entschlüsseln. Im Allgemeinen sind asymmetrische Verschlüsselungsverfahren deutlich rechenaufwändiger als symmetrische Verfahren.

Authentifikation

In Abschnitt 2.3.2 und Abschnitt 2.3.4 wurden mit *Neighbor Discovery Spoofing* und *MITM with Redirects* zwei Man-In-The-Middle-Angriffe bei IPv6 dargestellt. Dabei kommuniziert das Opfer nicht mehr mit dem eigentlichen Kommunikationspartner, sondern mit dem Angreifer selbst. Könnte das Opfer die Identität seines Kommunikationspartner verifizieren, so wäre ein solcher Angriff nicht mehr möglich. Ein Verfahren, um dieses Problem zu lösen, ist die Authentifikation. Möchte Alice eine Nachricht an Bob versenden, muss sie glaubwürdig machen, dass sie auch tatsächlich Absender dieser Nachricht ist. Dies kann mittels asymmetrischer Verschlüsselung³ durch folgende Schritte sichergestellt werden [4]:

1. Alice berechnet mittels einer *One-Way-Hashfunction* den Hashwert der Nachricht: $H(m)$.
2. Alice verschlüsselt den Hashwert mit ihrem privaten Schlüssel d : $E_d(H(m))$.
3. Alice sendet Bob die Nachricht und die Authentifikationsdaten: $(m, E_d(H(m)))$.
4. Bob berechnet ebenfalls den Hashwert der Nachricht: $H(m)$.
5. Bob prüft, ob die mit dem öffentlichen Schlüssel von Alice (e) entschlüsselten Authentifikationsdaten dem Hashwert entsprechen: $E_e(E_d(H(m))) = H(m)$.

Da nur Alice ihren privaten Schlüssel kennt und den Hashwert damit so verschlüsselt, dass nach dem Entschlüsseln mit ihrem öffentlichen Schlüssel wieder der Hashwert entsteht, ist auf diese Weise sichergestellt, dass die Nachricht tatsächlich von ihr stammt.

²Hierfür kann z.B. der Diffie-Hellman-Schlüsselaustausch verwendet werden. Das Verfahren basiert auf dem diskreten Logarithmus-Problem und dient zum Austausch eines Schlüssels über ein unsicheres Medium. Durch alleiniges Mithören der Kommunikation kann ein Angreifer den Schlüssel nur mit großem Aufwand ermitteln.

³Hierzu kann z.B. das RSA-Verfahren verwendet werden.

2.4.2 IPSec

IPSec ist eine Protokoll-Suite, die verschiedene Sicherheitsfeatures zur Datenübertragung mit IP bereitstellt. Auch IPv4 kann bereits mit IPSec betrieben werden. Bei IPv6 wurde IPSec allerdings fest in den Standard mit übernommen und ist im Gegensatz zu IPv4 nicht bloß optionale Erweiterung. Dennoch muss IPSec auch bei IPv6 entsprechend der jeweiligen Anforderungen konfiguriert werden. Ohne Konfiguration besteht auch kein Schutz und das System kann durch die in Abschnitt 2.3 erläuterten Techniken angegriffen werden. In IPSec sind beide in Abschnitt 2.4.1 vorgestellten Verfahren - Verschlüsselung und Authentifikation - unter der Bezeichnung *Encapsulating Security Payload (ESP)* und *IP Authentication Header (AH)* umgesetzt. Es gibt zwei verschiedene Modi, in denen IPSec betrieben werden kann: der *Tunnel Mode* und der *Transport Mode*. Auch eine Kombination von beiden ist möglich.

Tunnel Mode

Der *Tunnel Mode* erzeugt ein virtuelles privates Netz (VPN) zwischen zwei Gateways. Dies dient dazu, um zwischen zwei Netzen, die über ein unsicheres Netz miteinander verbunden sind (z.B. das Internet), eine sichere Verbindung zu schaffen. Durch Verschlüsselung und Authentifikation können so z.B. die Netze von zwei verschiedenen Firmenstandorten sicher miteinander verbunden werden. Wie in Abbildung 2.16 zu sehen ist, wird die Sicherheitsbeziehung zwischen Gateway1 und Gateway2 hergestellt. Somit sind die Daten bei der Übertragung über das Internet zwar geschützt, jedoch bietet diese Variante keinen Schutz vor Angriffen innerhalb des eigenen Teilnetzes.

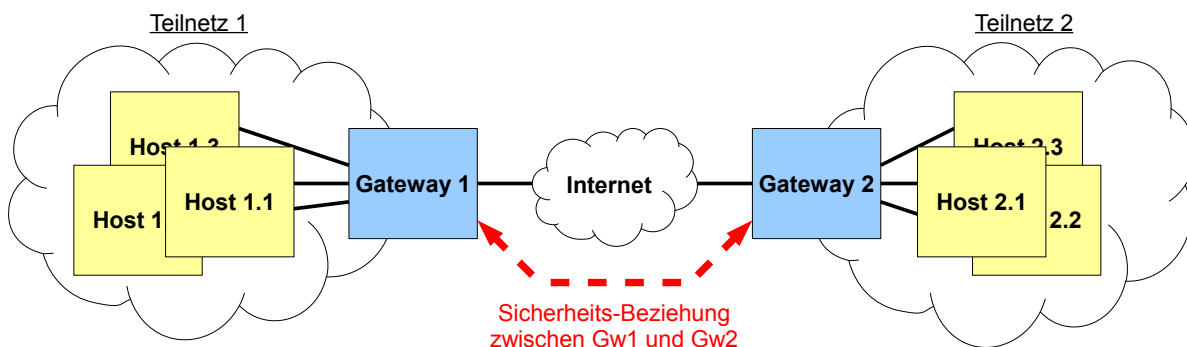


Abbildung 2.16: Tunnel Mode

Abbildung 2.17 zeigt den logischen Aufbau eines Pakets mit Tunnel Mode, das bei der Übertragung über das Internet mitgeschnitten wurde. Wie zu sehen ist, sind die IP-Adressen der Hosts und die Nutzdaten (bzw. weitere Extensionsheader) verschlüsselt. Auf diese Weise bleibt der interne Aufbau des eigenen Teilnetzes verborgen.

Transport Mode

Im *Transport Mode* wird eine Sicherheitsbeziehung direkt zwischen den Hosts aufgebaut. Dieser Modus ist in Abbildung 2.18 dargestellt.



Abbildung 2.17: Paketaufbau bei Tunnel Mode

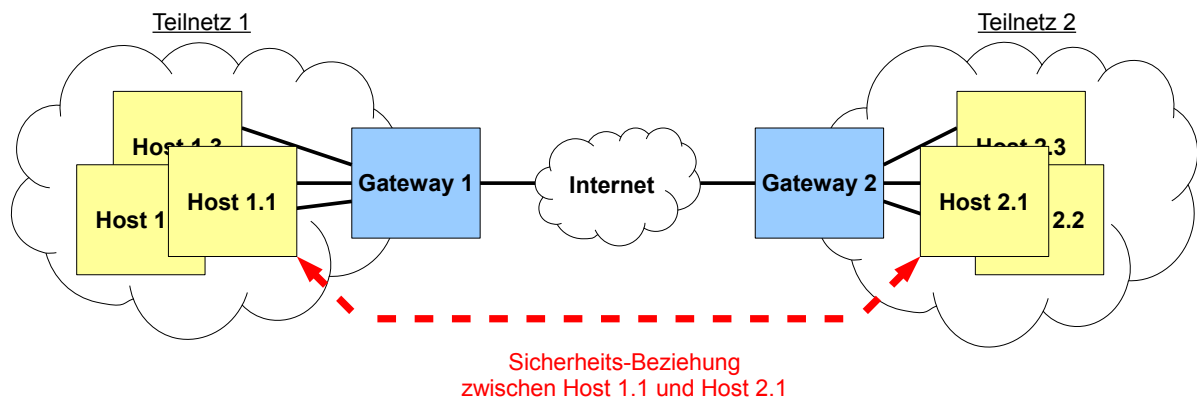


Abbildung 2.18: Transport Mode

Dieser Modus bietet auch Schutz vor Angriffen aus dem eigenen Teilnetz. Der Nachteil dieser Variante ist allerdings, dass die IP-Adressen der Hosts unverschlüsselt über das Internet übertragen werden. Der Paketaufbau in Abbildung 2.19 verdeutlicht dies.



Abbildung 2.19: Packetaufbau bei Transport Mode

Transport und Tunnel Mode

Eine sicherere Variante ist die Kombination von beiden Modi. Dazu führt Host 1.1 die Verschlüsselung zu Gateway2 und die Authentifikation von Host 1.1 zu Host 2.1 durch. So wird sichergestellt, dass die IP-Adressen der Hosts bei der Übertragung über das Internet verschlüsselt sind und zusätzlich ein Schutz vor Angriffen aus dem eigenen Netz besteht. Der Aufbau eines solchen Pakets ist in Abbildung 2.20 dargestellt.

2.4.3 Wirksamkeit

Nach der generellen Betrachtung der durch IPSec angebotenen Sicherheitsverfahren wird im folgenden Abschnitt konkret dargestellt, welche Angriffstechnik aus Abschnitt 2.3 mit entsprechenden Gegenmaßnahmen verhindert werden können.

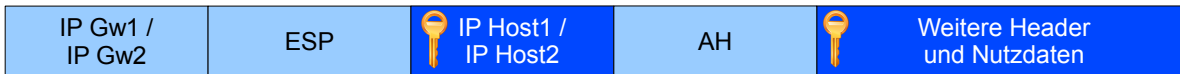


Abbildung 2.20: Packetaufbau bei Transport und Tunnel Mode

Anonymität

Angriff: Ein Host kann eindeutig identifiziert werden, da der *Interface Identifier* aus der MAC-Adresse des Netz-Interfaces gebildet wird.

Gegenmaßnahme: Die in RFC4941 [20] beschriebenen Privacy Extensions sorgen dafür, dass der *Interface Identifier* nicht aus der MAC-Adressen, sondern einer Zufallszahl erzeugt wird. Die Adresse wird durch die *Autoconfiguration* automatisch vergeben und ändert sich mit der Zeit. Auf diese Weise wird verhindert, dass ein Gerät eindeutig identifiziert werden kann.

Neighbor Discovery Spoofing

Angriff: Mittels gespoofter *Neighbor Advertisement* Pakete schaltet sich der Angreifer zwischen Opfer und Gateway.

Gegenmaßnahme: Ein Weg dieses Problem zu lösen, ist die Verwendung von IPSec. Dies ist auch als Lösungsmaßnahme in der ursprünglichen Spezifikation von NDP angegeben [19]. Es wird allerdings nicht genau beschrieben, wie dies umgesetzt werden kann. In der Praxis kann die Anzahl von notwendigen Sicherheitsbeziehungen, die für IPSec eingerichtet werden müssen, sehr groß sein. Diese Sicherheitsbeziehungen müssen manuell eingerichtet werden, so dass dies mit sehr viel Aufwand verbunden ist und für die meisten Fälle keine angemessene Lösung darstellt.

Eine weitere Möglichkeit ist das Einrichten von statischen Einträgen. Dies ist auch mit manuellem Konfigurationsaufwand verbunden, der sich gerade in großen Netzen und vor allem mit veränderlichen IP-Adressen⁴ nicht eignet. In einigen Fällen ist es aber eine Möglichkeit, schnell eine Lösung zu etablieren. So kann beispielsweise das Standard-Gateway statisch konfiguriert werden, um zu verhindern, dass ein Angreifer den Datenverkehr ins Internet mitschneidet. In Abbildung 2.21 ist die Zuordnungstabelle (IPv6 -> MAC-Adresse) unter Windows 7 dargestellt. Diese kann mit dem Kommando `netsh interface ipv6 show neighbors` angezeigt werden. Um einen statischen Eintrag hinzuzufügen, kann das Kommando `netsh interface ipv6 add neighbors <Schnittstelle> <IPv6-Adresse> <MAC-Adresse>` verwendet werden.

Eine Alternative zu den beiden genannten Methoden bildet *SEcure Neighbor Discovery (SEND)*. Es erweitert NDP um zusätzliche Optionen, um die verschiedenen Funktionen zu schützen. Bevor ein Host eine Adresse anfordern kann, muss er ein Paar aus privaten und öffentlichem Schlüssel generieren. Diese werden verwendet, um sicherzustellen, dass

⁴Bei Verwendung der Privacy Extension sind IP-Adressen nicht statisch und können sich mit der Zeit ändern.

```

C:\Windows\system32\cmd.exe

C:\Users\st>netsh interface ipv6 show neighbors

Schnittstelle 1: LAN-Verbindung

Internetadresse                               Physische Adresse   Typ
-----
ff02::2                                       33-33-00-00-00-02   Permanent
ff02::16                                      33-33-00-00-00-16   Permanent
ff02::1:3                                     33-33-00-01-00-03   Permanent
ff02::1:ff85:81ad                             33-33-ff-85-81-ad   Permanent

C:\Users\st>netsh interface ipv6 add neighbors "LAN-Verbindung" "2a01:0db8:8a22:
e359:fee9:28d0:3491:cd39" "00-AA-BB-CC-DD-EE"

```

Abbildung 2.21: Hinzufügen eines statischen Eintrags

der Absender einer *Neighbor Discovery* Nachricht auch wirklich der Inhaber dieser Adresse ist. Auf diese Weise wird verhindert, dass ein Angreifer eine bereits vorhandene Adresse übernimmt. Die Erweiterung ist detailliert in RFC3971 [14] beschrieben.

Desweiteren kann diese Art von Angriff durch intelligente Switches mittels des unter dem Begriff *Port Security* benannten Sicherheitsfeatures verhindert werden. Dabei kann jede Schnittstelle des Switches so konfiguriert werden, dass nur bestimmte Pakete überhaupt weitergeleitet werden. Auf diese Weise kann die Schnittstelle eines gewöhnlichen Hosts so konfiguriert werden, dass er keine *Router Advertisement* Pakete senden kann. Ebenso kann verhindert werden, dass ein Host gespoofte *Neighbor Advertisement* Pakete ins Netz sendet [26].

Duplicate Address Detection DoS

Angriff: Der Angreifer gibt vor, Inhaber aller IP-Adressen zu sein, die ein neuer Host verwenden möchte, so dass dieser keine gültige Adresse beziehen kann.

Gegenmaßnahme: Dieser Angriff kann ebenfalls durch die Verwendung von *SEcure Neighbor Discovery (SEND)* verhindert werden. Dazu muss beim Senden eines *Neighbor Advertisement* Pakets die verwendete IP-Adresse mittel RSA authentifiziert werden. Der Sender muss also nachweisen, dass er wirklich Inhaber der IP-Adresse ist. In einem Netz muss allerdings nicht jeder Host zwangsweise SEND verwenden. Wenn ein Host mit SEND eine IP-Adresse beziehen möchte und er von einem Host ohne SEND eine unauthentifizierte *Router Advertisement* Nachricht erhält, so kann er einmalig eine andere IP-Adresse wählen und erneut eine Anfrage stellen. Darauf folgende Meldungen von unauthentifizierten Hosts werden jedoch mit dem Risiko eines Adresskonfliktes ignoriert. Die Wahrscheinlichkeit und Auswirkungen eines Adresskonfliktes sind in RFC 3972 [15] beschrieben. Auch die Verwendung von intelligenten Switches kann diesen Angriff verhindern.

Router Faking

Angriff: Der Angreifer gibt einem neuen Host gegenüber vor ein Router zu sein. Indem er vortäuscht, dass andere Router nicht länger als Router fungieren, sorgt er dafür, dass der Host nur den Angreifer als Default-Router wählen kann.

Gegenmaßnahme: Auch bei diesem Angriff ist die Verwendung von SEND wirksam. Dabei müssen *Router Advertisement* Pakete eine RSA-Signatur enthalten, die mit dem öffentlichen Schlüssel eines Knotens erstellt wurde, der nachweisen kann, dass er zum Routen berechtigt ist. *Router Advertisement* Pakete ohne diesen Sicherheitsmechanismus werden verworfen. Ebenso kann mittels Port Security das Senden der *Router Advertisement* Pakete ins Netz direkt verhindert werden.

IPv6 in IPv4-Netzen

Angriff: (1) Durch Tunneling (Teredo) wird die gewöhnliche IPv4-Firewall umgangen und die durch NAT eingeschränkte Erreichbarkeit von außen aufgehoben.

(2) Durch Einführung eines vorgetäuschten IPv6-Routers kann ein Angreifer in einem IPv4-Netz sämtlichen Datenverkehr mithören.

Gegenmaßnahme: (1) Um zu verhindern, dass ein Tunnel mittels Teredo aufgebaut wird, kann der Teredo Port 3544 komplett gesperrt werden [17]. Damit die Firewall mit dem Protokoll umgehen kann, muss dies erst eigens implementiert werden. Für weitere Informationen liefert Cisco eine detaillierte Beschreibung, wie IPv6 Tunnel in einem Netz erkannt werden können [25].

(2) Ein Netzadministrator sollte in einem Netz, indem kein IPv6 genutzt wird, das Senden von *Router Advertisement* Paketen durch die Verwendung von Port Security komplett verhindern. Als Anwender ist es empfehlenswert, IPv6 zu deaktivieren, wenn es nicht verwendet wird. Bei Windows kann dies über die Eigenschaften des Netzwerkadapters deaktiviert werden (Abbildung 2.22). Für weitere Informationen siehe [27].

Schwachstellen bei der Umsetzung des TCP/IP-Stacks

Angriff: Diverse Schwachstellen beziehen sich auf die Umsetzung des TCP/IP IPv6 Stacks. Dazu vorgestellt wurden das RA Flooding (betrifft mehrere Betriebssysteme) und jeweils eine Sicherheitslücke in Linux, Cisco ASA/ASASM und Windows.

Gegenmaßnahme: Als Gegenmaßnahme zu Schwachstellen in der Umsetzung existiert prinzipiell nur eine Möglichkeit: die Software aktualisieren. Dafür ist es natürlich notwendig, dass ein Softwareupdate existiert, das die betroffene Sicherheitslücke behebt. Dem Netzadministrator und Anwender bleiben hier nur die Möglichkeit, sich regelmäßig zu informieren und bei Neuerungen schnell zu reagieren. In der endgültigen Verantwortung der Umsetzung stehen die Betriebssystem-Entwickler.

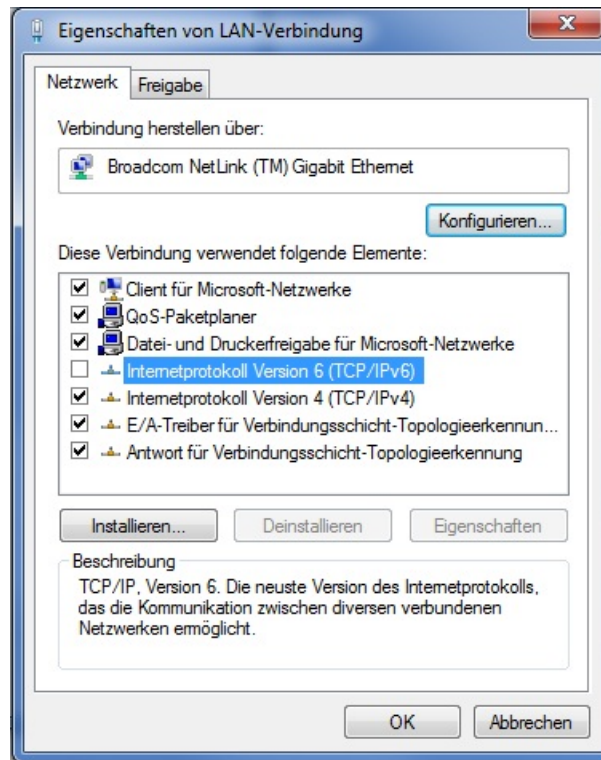


Abbildung 2.22: Deaktivieren von IPv6 unter Windows

2.5 Schluß

Nach Betrachtung der Grundlagen, verschiedener Angriffstechniken und den entsprechenden Gegenmaßnahmen wird in den folgenden Abschnitten nun dargestellt, welche Erkenntnisse daraus gewonnen werden können und wie sich IPv6 voraussichtlich weiterentwickeln wird.

2.5.1 Fazit

Die dargestellten Angriffstechniken finden primär in lokalen Netzen Anwendung. Ein wichtiger Aspekt ist daher die Frage nach der Vertrauenswürdigkeit der Netzteilnehmer in diesem Netz. Es sollte sichergestellt werden, dass nur berechtigte Personen physischen Zugang zum Netz besitzen. Auf diese Weise können zumindest private Netze geschützt werden. Jedoch sind viele Netze so groß, dass nicht jeder Netzteilnehmer als vertrauenswürdig eingestuft werden kann. Dazu gehören z.B. Firmennetze und Netze an Universitäten oder anderen Einrichtungen. Auch die in den letzten Jahren stark angewachsene Anzahl von Hotspots und die verbreite Verwendung von Smartphones in öffentlichen, drahtlosen Netzen bieten einem möglichen Angreifer eine Plattform für die verschiedenen Angriffstechniken.

In Abschnitt 2.4 wurden mit IPsec und SEND zwei wichtige Sicherheitsmechanismen zum Schutze vor Angriffen dargestellt. IPsec erfordert eine recht aufwändige Konfiguration und liegt zum größten Teil in der Hand des Netzadministrators. Auch SEND bietet Schutz vor

einer Vielzahl von Angriffen. Das Problem ist, dass in den meisten Betriebssysteme SEND noch nicht implementiert ist. Daher ist es nötig, zusätzliche Programme zu installieren. Für Linux gibt es das Programm *NDprotector*, das auf *iprout2* und *ip6tables* basiert. Auch für Windows gibt es mit dem Programm *WinSEND* eine Möglichkeit von SEND zu profitieren [23].

2.5.2 Ausblick

Es bleibt abzuwarten, wie lange es noch dauern wird, bis IPv6 endgültig IPv4 ersetzt. Bis dahin wird voraussichtlich noch einige Zeit vergehen, in der vor allem viel Arbeit in die Stabilisierung der Umsetzungen von IPv6 gesetzt werden muss. An den Schwachstellen im Protokolldesign selbst wird sich voraussichtlich kaum noch etwas ändern. Die damit in Verbindung stehenden Probleme sind jedoch durch die genannten Gegenmaßnahmen in den Griff zu bekommen. Schwieriger verhält es sich mit Schwachstellen in den Umsetzungen. Bei IPv4 hat es jahrelang gedauert, bis das Protokoll stabil in den Betriebssystemen implementiert war. Und selbst hier gilt natürlich der Grundsatz, dass durchs Testen zwar Fehler gefunden werden können, deren Abwesenheit jedoch nie bewiesen werden kann. Bis der Stabilitätsgrad der IPv6-Umsetzungen ein vergleichbares Niveau erreicht hat, wird es noch dauern. Aufgrund dessen sind beim IPv6-Betrieb zwei Aspekte besonders wichtig: Aufmerksamkeit und Aktualität. Dies gilt nicht nur für Netzadministratoren, sondern auch für gewöhnliche Anwender. Durch ständiges Informieren über sicherheitsrelevanten Themen und Softwareupdates kann angemessen auf neue Erkenntnisse im Bereich IPv6-Sicherheit reagiert werden. So können die neuen Möglichkeiten von IPv6 sicherheitsbewusst genutzt werden.

Literaturverzeichnis

- [1] JON ERICKSON. *Hacking: The Art of Exploitation*, No Starch Press, San Francisco, 2008.
- [2] JOHN NAUGHTON. *A Brief History of the Future: Origins of the Internet*, Orion, London, 2000.
- [3] GABI DREO RODOSEK. *Vorlesungsfolien: Einführung in Rechnernetze*, München, 2012.
- [4] MARIO ROMSY. *Vorlesungsskript: Elementare Zahlentheorie und Kryptographie*, München, 2012.
- [5] MARC HEUSE. *CCC Congress 2010: Recent advances in IPv6 insecurities*, Berlin, 2010.
- [6] MATHIAS HEIN, MICHAEL REISNER. *IPv6 - Das Migrationsbuch*, Franzis' Verlag, Poing, 2003.
- [7] HERBERT WIESE. *Das neue Internetprotokoll IPv6*, Carl Hanser Verlag, München Wien, 2002.
- [8] J. POSTEL. *Internet Protocol*, RFC 791, 1981.
- [9] D. PLUMMER. *Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*, RFC 826, 1982.
- [10] R. HINDEN, S. DEERING. *IP Version 6 Addressing Architecture*, RFC 2737, 1998.
- [11] S. DEERING, R. HINDEN. *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460, 1998.
- [12] R. DROMS, ED., J. BOUND, B. VOLZ, T. LEMON, C. PERKINS, M. CARNEY. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, RFC 3315, 2003.
- [13] P. NIKANDER, ED., J. KEMPF, E. NORDMARK. *IPv6 Neighbor Discovery (ND) Trust Models and Threats*, RFC 3756, 2004.
- [14] J. ARKKO, ED., J. KEMPF, B. ZILL, P. NIKANDER. *SEcure Neighbor Discovery (SEND)*, RFC 3971, 2005.

- [15] T. AURA. *Cryptographically Generated Addresses (CGA)*, RFC 3972, 2005.
- [16] R. DRAVES, D. THALER. *Default Router Preferences and More-Specific Routes*, RFC 4191, 2005.
- [17] C. HUITEMA. *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)*, RFC 4380, 2006.
- [18] A. CONTA, S. DEERING, M. GUPTA, ED.. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, RFC 4443, 2006.
- [19] T. NARTEN, E. NORDMARK, W. SIMPSON, H. SOLIMAN. *Neighbor Discovery for IP version 6 (IPv6)*, RFC 4861, 2007.
- [20] T. NARTEN, R. DRAVES, S. KRISHNAN. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, RFC 4941, 2007.
- [21] J. JEONG, S. PARK, L. BELOEIL, S. MADANAPALLI. *IPv6 Router Advertisement Options for DNS Configuration*, RFC 6106, 2010.
- [22] ZIMMERMANN, H.. *OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection*, IEEE, 1980.
- [23] REIKO KAPS. *SEND: Sicherheit für IPv6-Autokonfiguration*, <http://www.heise.de/netze/artikel/SEND-Sicherheit-fuer-IPv6-Autokonfiguration-1389090.html> Stand 28.06.2012, heise.de, 2011.
- [24] JOHANNES ENDRES, REIKO KAPS. *Teredo bohrt IPv6-Tunnel durch Firewalls*, <http://www.heise.de/netze/artikel/Teredo-bohrt-IPv6-Tunnel-durch-Firewalls-221537.html> Stand 08.07.2012, heise.de, 2009.
- [25] CISCO. *Detecting IPv6 Tunnels in an Enterprise Network*, Cisco, 2010.
- [26] CISCO. *Layer 2 Security Features on Cisco Catalyst Layer 3 Fixed Configuration Switches Configuration Example*, http://www.cisco.com/en/US/products/hw/switches/ps5023/products_configuration_example09186a00807c4101.shtml, Cisco, 2007.
- [27] MICROSOFT. *So deaktivieren Sie bestimmte IPv6-Komponenten (Internet Protocol Version 6) in Windows Vista, Windows 7 und Windows Server 2008*, <http://support.microsoft.com/kb/929852>, Microsoft, 2012.
- [28] ALEC WATERS. *SLAAC Attack - 0day Windows Network Interception Configuration Vulnerability*, <http://resources.infosecinstitute.com/slaac-attack/>, infosec institute, 2011.
- [29] DFN-CERT. *DFN-CERT-2012-0239: Mehrere Schwachstellen im Linux Kernel*, <https://portal.cert.dfn.de/adv/DFN-CERT-2012-0239/>, DFN-CERT, 2012.
- [30] DFN-CERT. *DFN-CERT-2012-1216: Schwachstelle in Cisco ASA und Cisco ASASM*, <https://portal.cert.dfn.de/adv/DFN-CERT-2012-1216/>, DFN-CERT, 2012.

- [31] DFN-CERT. *DFN-CERT-2010-1025: Schwachstellen im Microsoft Windows TCP/IP Stack ermöglichen Privilegienskalation*, <https://portal.cert.dfn.de/adv/DFN-CERT-2010-1025/>, DFN-CERT, 2010.

Kapitel 3

IT-Sicherheitsmanagement

Nico Hamann

IT-Sicherheit nimmt für moderne Unternehmen einen immer höheren Stellenwert ein, und so entsteht der Bedarf nach einer gemanagten IT-Sicherheitsumgebung. Für diesen Anwendungsfall bietet es sich an, ein IT-Sicherheitsmanagementsystem zu etablieren. Diese Arbeit liefert einen Überblick über solche IT-Sicherheitsmanagementsysteme, und zeigt darin enthaltene Elemente und Strukturen auf. Außerdem werden allgemein anerkannte Verfahren vorgestellt. Der Standard ISO 27001:2005 und der darauf aufbauende IT-Grundschatz, sowie die Systeme OCTAVE und COBIT werden in ihren Grundzügen erklärt. So soll den Verantwortlichen ein Anhalt für die Wahl des für das Unternehmen passenden Verfahrens zu geben.

Inhaltsverzeichnis

3.1	Einführung	63
3.1.1	Grundlagen	63
3.2	Elemente und Strukturen	65
3.2.1	Richtlinien	66
3.2.2	Planung	68
3.2.3	Programme	69
3.2.4	Schutzmaßnahmen	69
3.2.5	Menschen	71
3.2.6	Projektmanagement	72
3.3	Verfahren	73
3.3.1	ISO/IEC 27001:2005	73
3.3.2	IT-Grundschutz	75
3.3.3	OCTAVE	76
3.3.4	COBIT	78
3.4	Zusammenfassung	79

3.1 Einführung

Moderne Kommunikations- und Informationsverarbeitende Systeme sind aus der heutigen Gesellschaft nicht mehr wegzudenken. Sei es das Smartphone, welches quasi zu jeder Zeit und an jedem Ort eine Verbindung in das Internet bereitstellt, oder der Personal Computer, der mehrfach in einem modernen Haushalt vorhanden ist. Nutzer und Systemadministratoren haben es mit immer komplexer werdenden Systemen zu tun, deren Zusammenspiel nicht immer offensichtlich ist.

Genauso wie im Privaten, steigt die Nutzung von Informations-Technik (IT) auch in Unternehmen rapide an. Wo zuletzt noch Schreibmaschinen und handschriftliche Notizen auf Bürotischen verteilt waren, stehen heute Computer-Monitore und VoIP-Telefone. Auch die Unternehmensprozesse an sich werden immer mehr von IT durchzogen, sodass in den meisten Unternehmen heute eine Abhängigkeit besteht.

Fällt nun die IT in so einem Unternehmen vollständig oder in Teilen aus, kann dadurch das Tagesgeschäft massiv gestört und im Extremfall sogar unmöglich werden. Dadurch entstehen den Unternehmen teils sehr hohe Kosten. So kostet ein einstündiger Ausfall der IT einer Studie zufolge 54% der teilnehmenden Unternehmen mehr als 51.000 US-Dollar.[1, S. 9]

Neben den hohen Kosten sinkt natürlich auch das Ansehen eines Unternehmens. Ein Datenleck bei einer großen Internetsuchmaschine zum Beispiel kann sich nicht positiv auf die Publicity auswirken, genauso wie ein E-Mail-Provider, der die Erreichbarkeit seiner Server und somit der E-Mail-Postfächer seiner Kunden nicht sicherstellen kann. Oder das Versicherungsunternehmen, welches seine Schadensfälle nicht regulieren kann, weil der zentrale Kundendaten-Server ausgefallen ist. Diese Liste könnte man beliebig erweitern.

Zu guter Letzt folgen natürlich noch gesetzliche Vorgaben, wie beispielsweise zum Datenschutz, welche eingehalten werden müssen, da deren Nichteinhaltung hohe Kosten in Form von Bußgeldern nach sich ziehen kann.

3.1.1 Grundlagen

Infolge dieser Situation nimmt die IT-Sicherheit einen immer höheren Stellenwert im Unternehmen ein – entsprechend steigen auch die Ausgaben zur Herstellung und Aufrechterhaltung eines angemessenen Sicherheitsniveaus.

Doch Sicherheit wird nicht nur durch den Einsatz von Virenschaltern und Firewalls erreicht. Bereits durch die Einführung einer geregelten IT-Sicherheitsstruktur und der geregelten Verteilung von Verantwortungen kann die Sicherheit in einem Unternehmen erhöht werden. Genauso kann die Sensibilisierung von Mitarbeitern und die Einhaltung einiger Grundregeln Gefahren abwehren.

Alle diese möglichen Maßnahmen werden in der modernen Unternehmensstruktur unter dem IT-Sicherheitsmanagement zusammengefasst. Dieses befaßt sich vor allem mit den

organisatorischen und administrativen Belangen, aber auch die technische Durchführung ist ein Teil dieses Aufgabenfeldes.

Anders als in der allgemein gebräuchlichen Form ist, wenn man hier von IT spricht, nicht nur die digitale Information gemeint, sondern Informationen im Allgemeinen. In der Fachliteratur hat sich für das Informations-Sicherheitsmanagement jedoch der etwas kürzere Begriff „IT-Sicherheitsmanagement“ etabliert, weshalb auch in dieser Arbeit darauf zurück gegriffen wird.

Die Komplexität des Gebietes des IT-Sicherheitsmanagements macht es notwendig und empfehlenswert, auf systematisches Vorgehen zurückzugreifen. Ein in einem Unternehmen etabliertes System zur Erreichung und Aufrechterhaltung eines Sicherheitsniveaus wird IT-Sicherheitsmanagementsystem (ISMS) genannt. Gleichzeitig macht die Nutzung eines ISMS die Standardisierung und somit auch die Zertifizierung der IT-Sicherheitsstruktur möglich.

Das im Bundesministerium des Inneren (BMI) für die IT-Sicherheit verantwortliche Bundesamt für Sicherheit in der Informationstechnik (BSI) beschreibt ISMS folgendermaßen:

„Ein Managementsystem für die Informationssicherheit (ISMS) ist das geplante und organisierte Vorgehen, um ein angemessenes Sicherheitsniveau für die Informationssicherheit zu erzielen und aufrechtzuerhalten.“[2, S. 55]

Nach [2, S. 24] bestehen ISMS aus den vier Feldern Strategie, Management-Prinzipien, Ressourcen und Mitarbeiter. Ziel ist es, die klassischen Schutzziele sicherzustellen, welche im Folgenden kurz erklärt werden.[3, S. 6 ff.][4, S. 188]

Authentizität Die Echtheit und Glaubwürdigkeit von Informationen ist eindeutig überprüfbar, und die Identität eines Nutzers kann festgestellt werden.

Integrität Informationen können nicht unauthorisiert und unbemerkt manipuliert werden.

Vertraulichkeit Unauthorisierter Zugriff auf Informationen ist nicht möglich.

Verfügbarkeit Authorisierte Handlungen werden nicht durch nicht-authorisierte Handlungen beeinträchtigt.

Verbindlichkeit Veränderungen können nachgewiesen und im Nachhinein vom Nutzer nicht abgestritten werden.

Datenschutz Personenbezogene Daten sind gesondert entsprechend der geltenden Gesetze geschützt.

Noch elementarer ist die Einteilung in die drei Schutzziele *Vertraulichkeit*, *Integrität* und *Verfügbarkeit*. Nach ihren englischen Bezeichnungen ist dieses Dreieck auch als C. I. A (Confidentiality, Integrity, Availability) bekannt.[5, S. 6 ff.]

3.2 Elemente und Strukturen

Wie kann nun also IT-Sicherheit erreicht werden? In diesem Kapitel sollen nun die einzelnen Elemente und Strukturen eines ISMS vorgestellt werden.

Um die eher allgemeine Einteilung in die vier Felder Strategie, Management-Prinzipien, Ressourcen und Mitarbeiter zu erweitern, wird hier auf die *6 P's* nach [5, S. 18 ff.] zurückgegriffen. Der Titel kommt von den englischen Begriffen, die alle mit P beginnen, und im Folgenden vorgestellt werden.

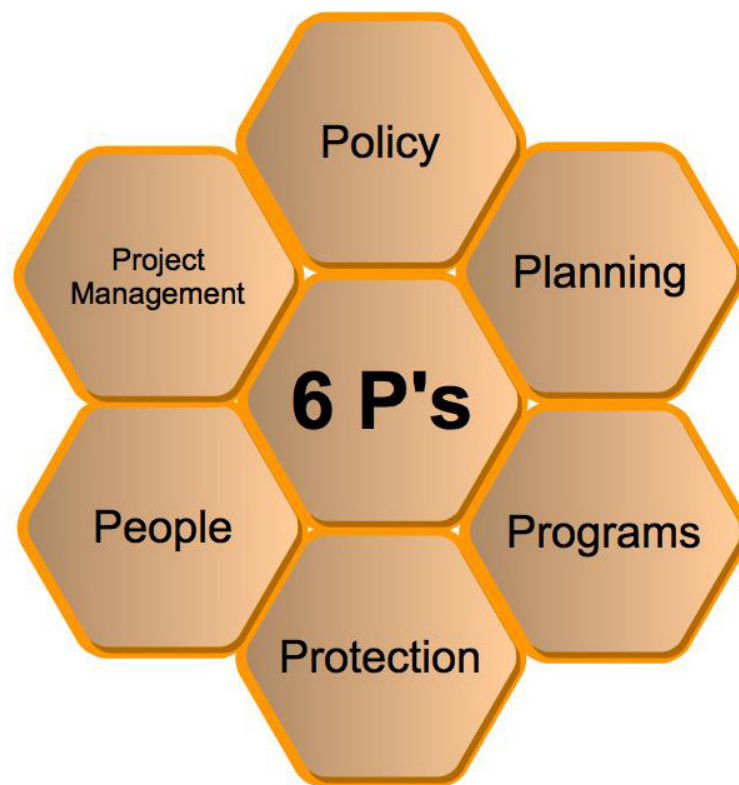


Abbildung 3.1: Die 6 P's

Policy (Richtlinien) Hierbei geht es vor allem darum, dass sich die Leitungsebene des betreffenden Unternehmens zu seinem Bestreben nach IT-Sicherheit bekennt. Aber auch Regeln zur Verwendung der IT sind unter diesem Punkt eingeordnet.

Planning (Planung) Planen für den Notfall ist genauso wichtig wie das koordinierte Vorgehen beim Herstellen des Sicherheitsniveaus.

Programs (Programme) Unter Programme sind gesonderte, abgeschlossene Maßnahmen zur Erhöhung der Sicherheit zu sehen, wie zum Beispiel Mitarbeiterschulungen zur Security Awareness, oder aber Übungen des Ernstfalls.

Protection (Schutzmaßnahmen) Sowohl auf organisatorischer als auch auf technischer Seite müssen Schutzmaßnahmen getroffen werden, wie zum Beispiel effektives Risikomanagement oder die Einrichtung von Firewalls und Viren-Scannern.

People (Menschen) Der Mitarbeiter sollte einen sicheren Umgang mit der IT wahren. Bei diesem Punkt geht es aber auch um die Verteilung von Rollen, wie zum Beispiel der eines Sicherheitsbeauftragten oder Sicherheitsteams.

Project Management (Projektmanagement) Koordiniertes Vorgehen ist ein wichtiger Bestandteil zum Erreichen eines Sicherheitsniveaus. Darum ist das Projektmanagement hier unerlässlich. Aber genauso muss die Einführung einer neuen Technologie in einem gemanagten Prozess erfolgen, um unerwünschte Seiteneffekte zu verhindern.

Im Folgenden sollen nun diese Elemente genauer betrachtet und einzelne Aspekte des IT-Sicherheitsmanagements in diese Gliederung eingeordnet werden.

3.2.1 Richtlinien

Ein Sicherheitsniveau kann nicht erreicht werden, wenn nur ein halbherziges Interesse dazu besteht. Deswegen ist vor allem die Leitungsebene eines Unternehmens gefragt, damit diese ihr Bestreben nach IT-Sicherheit verbindlich festhält. Dies geschieht in der *Sicherheitsleitlinie*. Desweiteren müssen zur Sicherstellung der IT-Sicherheit bestimmte *Richtlinien* für die Nutzer aufgestellt werden. So kann das Schadensrisiko durch Fehlbenutzung minimiert werden.

Sicherheitsziele Am Anfang eines jeden Sicherheitsprozesses muss das ausführende Unternehmen festlegen, welche Sicherheitsziele erreicht werden sollen. Andererseits besteht die Gefahr, dass die später erarbeitete Strategie und die Programme zwar einen Sicherheitsprozess umsetzen, dieses jedoch nicht den Anforderungen und den Wünschen, oder sogar dem Budget des Unternehmens entspricht.

An dieser Stelle ist es wichtig, das benötigte Sicherheitsniveau mit dem vorhandenen Budget zu vereinbaren oder ggf. sinnvolle Einschränkungen zu treffen. Wichtig in diesem Zusammenhang ist, wie sehr der Geschäftsprozess des Unternehmens von der IT abhängt und welcher Schaden durch den Ausfall der IT verursacht werden kann.[2, S. 66 f.]

Sicherheitsleitlinie Die Sicherheitsleitlinie enthält die groben IT-Sicherheitsziele eines Unternehmens. In ihr wird festgelegt, welchen Bezug der Sicherheitsprozess zu den Aufgaben und Zielen des Unternehmens hat. Außerdem wird hier die Sicherheitsstrategie des Unternehmens, sowie das angestrebte Sicherheitsniveau festgelegt.

Die Sicherheitsleitlinie solle zudem noch Informationen enthalten, wie das angestrebte Sicherheitsniveau erreicht werden soll und welche Organisationsstruktur dafür vorgesehen ist – zum Beispiel die Ernennung eines IT-Sicherheitsbeauftragten. Die Leitungsebene des Unternehmens sollte sich darin zum IT-Sicherheitsprozess bekennen und die Durchsetzung des Prozesses zusichern.[2, S. 70]

Nach Fertigstellung der Sicherheitsleitlinie soll diese allen Bereichen des Unternehmens eröffnet werden. Die Mitarbeiter sollten die Inhalte kennen und nachvollziehen können.

Weiterhin ist es wichtig, dass die IT-Sicherheitsleitlinie kontinuierlich an neue Gegebenheiten und Anforderungen angepasst wird. Aufgrund der schnellen Weiterentwicklung der IT empfiehlt es sich, die Sicherheitsleitlinie ca. alle zwei Jahre zu aktualisieren.[2, S. 72]

Das Fehlen einer Sicherheitsleitlinie kann dazu führen, dass dessen Inhalte durch die für die Durchführung des Sicherheitsprozesses verantwortlichen Mitarbeiter selbst definiert werden. Dadurch werden sie schwer nachvollziehbar und es können Inkonsistenzen auftreten. Im Worst-Case können Teile dieser Informationen auch verloren gehen.[1, S. 83]

Sicherheitsrichtlinien Ist die Sicherheitsleitlinie erstellt gilt es Richtlinien für die Nutzung der IT-Systeme des Unternehmens festzulegen. Zum Beispiel kann durch die Einschränkung der Nutzung privater Notebooks im Unternehmens-LAN oder USB-Speichermedien an Unternehmens-PC's die unbemerkte Einschleusung von Viren und anderer Schadsoftware verhindert werden. Weiterhin seien einfache Prinzipien, wie das Gebot zum „aufgeräumten Arbeitsplatz“ oder dem gesperrten Bildschirm,¹ genannt. Neben der Einrichtung dieser übergreifenden Gebote können und sollten aber auch anwendungsspezifische Richtlinien erstellt werden.

Die Sicherheitsrichtlinien sollten eine wichtige Eigenschaft erfüllen: Sie müssen von jedem Mitarbeiter, der mit dem entsprechenden System in Berührung kommt, verstanden werden. Weiterhin sollte aber auch beachtet werden, dass die Richtlinien die Arbeitsfähigkeit nicht zu sehr einschränken. Der Geschäftsprozess muss auch unter Einhaltung der Richtlinien effizient weitergeführt werden können.

Eine weitere wichtige, und einfach umzusetzende Richtlinie soll noch erwähnt werden. Passwortsicherheit kann einen großen Teil zur sicheren Nutzung der IT beitragen. Einfache (bzw. nicht ausreichend komplexe) Passwörter machen es einem Angreifer einfach, in das System einzudringen und seine „Arbeit zu erledigen“. Ist ein Passwort zum Beispiel ein einfaches deutsches Wort, kann es mithilfe spezieller Programme², die frei im Internet erhältlich sind, binnen Sekunden geknackt und somit der Zugang zu dem IT-System erlangt werden. Es drängt sich hier das Verlangen nach einer Passwort-Richtlinie auf. In einer solchen Richtlinie sollte also die Art des Passwortes (keine deutschen oder englischen Wörter) sowie die Form (mindestens 8 Zeichen lang, Groß- und Kleinschreibung, mindestens ein Sonderzeichen) festgelegt werden.[3, S. 462 f.]

¹Die Prinzipien des *aufgeräumten Arbeitsplatzes* und des *gesperrten Bildschirms* sagen aus, dass sämtliche Informationen und Medien am Arbeitsplatz für unbefugte unzugänglich aufbewahrt werden müssen. Dazu gehört auch das Sperren des Bildschirms, wenn man den Arbeitsplatz (auch nur für kurze Zeit) verlässt.

²Als Beispiel sei hier das für Unix/Linux- und Windows-basierte Systeme erhältliche Programm *John the Ripper* genannt, erhältlich unter <http://www.openwall.com/john/>

3.2.2 Planung

Der Abschnitt Planung gliedert sich in zwei elementare Teilbereiche. Auf der einen Seite steht die Planung zum Erreichen eines IT-Sicherheitsniveaus, welches für ein strukturiertes Vorgehen unerlässlich ist. Auf der anderen Seite steht die Notfallplanung, welche bei Eintreten eines Sicherheitsvorfalls die schnelle Wiederaufnahme des Geschäftsprozesses sicherstellen soll. Im Folgenden wird näher auf die beiden Felder eingegangen.

Planen der Sicherheitsstruktur Die Herstellung eines Sicherheitsniveaus ist ein komplexes Vorgehen, welches ohne ausführliche Planung nur schwerlich durchzuführen ist. Dementsprechend gründlich sollte die Planung vorgenommen werden. Halbherziges Vorgehen kann an dieser Stelle das Erreichen des gewünschten Sicherheitsniveaus gefährden. Standardisierte Verfahren für das IT-Sicherheitsmanagement liefern einen Ansatz, welche Planungsvorgänge für welchen Bereich empfehlenswert sind (siehe 3.3).

Zuallererst muss die grundsätzliche Strategie für die IT-Sicherheit entwickelt werden. Dazu empfiehlt es sich, bei der Unternehmensstrategie zu beginnen und diese dann Schritt für Schritt auf den entsprechenden Teilbereich herunterzubrechen. Anschließend sollte für jeden Punkt der Strategie festgestellt werden, welche Anforderungen sich dadurch für die IT bzw. IT-Sicherheit ergeben und auf welche Weise diese erfüllt werden können. Siehe dazu auch 3.2.1.

Ist die Strategie erstellt, sollten die Elemente der IT-Infrastruktur erfasst werden. Anschließend ist der Schutzbedarf der einzelnen Komponenten festzulegen. Anhalt dafür ist die Wichtigkeit der Komponente für den Geschäftsprozess. Fragen wie „Ist das Unternehmen noch geschäftsfähig, wenn die Komponente ausfällt?“ oder „Würden gesetzliche Vorgaben gebrochen?“ können helfen, den Schutzbedarf festzustellen. Wichtig ist hierbei auch zu beachten, welches Sicherheitsniveau erreicht werden soll und den festgestellten Schutzbedarf evtl. noch einmal zu überdenken.

Notfallplanung Die Notfallplanung ist eines der wichtigsten Elemente des IT-Sicherheitsmanagement. Schließlich lassen sich nicht alle möglichen Sicherheitsvorfälle vorhersehen. Genauso kann das Unerwartete nur schwer verhindert werden. Aber wenn ein Vorfall auftritt kann lageangepasst reagiert und gehandelt werden, um so den Geschäftsprozess schnellstmöglich wieder aufzunehmen und Kosten gering zu halten, denn jede Minute, in der ein stark von der IT abhängiges Unternehmen nicht auf diese zugreifen kann, kostet Geld.

Solche Pläne sind zum Beispiel der *Incident Response Plan (IRP)* oder der *Disaster Recovery Plan (DRP)*. Der IRP hat die Hauptaufgabe, bei eintretenden Sicherheitsvorfällen diese zu identifizieren, und schnellstmöglich zu stoppen oder zu minimieren. Danach folgt die Untersuchung des Vorfalls und die Wiederherstellung der betroffenen Systeme. Zuletzt sind nach der Analyse des Vorfalls angemessene Maßnahmen zu entwickeln, um einen gleichartigen Vorfall in Zukunft verhindern zu können. Für die Erfüllung dieser Aufgaben bietet sich zum Beispiel die Einrichtung eines *Computer Emergency Response Team (CERT)* an.[6]

Es kommt vor, dass der Sicherheitsvorfall nicht mehr durch den IRP gelöst werden kann. Wenn das Unternehmen nicht in der Lage ist, die Folgen des Vorfalls einzudämmen und zu kontrollieren, oder wenn der verursachte Schaden so groß ist, dass sich das Unternehmen nicht kurzfristig wieder davon erholen kann, tritt der DRP in Kraft. Er sollte Anweisungen enthalten, wie weiterer Schaden verhindert und die operative Fähigkeit des Unternehmens schnellstmöglich wiederhergestellt werden kann. Weiterhin ist eine genaue Rollenverteilung festzulegen, um Kompetenzen klar zu regeln und einen zielführenden und effizienten Wiederherstellungsprozess sicher zu stellen.

3.2.3 Programme

Bei Programmen handelt es sich um eigenständige, unabhängige und abgeschlossene Maßnahmen im Rahmen des ISMS, welche die Sicherheit verbessern sollen. Dazu gehören zum Beispiel Mitarbeiterschulungen zum Thema *Security Awareness*³ oder Übungen. Letzteres ist eine praktische Möglichkeit, die Umsetzung und die Effektivität der IT-Sicherheitsstruktur zu überprüfen. Außerdem werden Vorgänge durch mehrmaliges wiederholen automatisiert und können so instinktiv ausgeführt werden.

Neben Schulungen und Übungen gehören aber auch Maßnahmen zum Erhöhen der physikalischen Sicherheit in das Teilgebiet Programme, wie z. B. Zutrittskontrollen, den Einsatz von Sicherheitsdiensten und Brandschutzmaßnahmen.

Die Programme sind grundsätzlich als in sich abgeschlossene Projekte gemanaged, verfolgen also auch ein eigenes Projektmanagement (siehe 3.2.6).

3.2.4 Schutzmaßnahmen

Eine große Schutzmaßnahme ist die Risikoanalyse und –vorsorge. Neben der Schutzbedarfsfeststellung (siehe 3.2.2) sind auch äußere Gefährdungen zu betrachten. Dazu zählen zum Beispiel Naturkatastrophen, Stromausfälle oder Wasserschäden. Ist die Risikoanalyse abgeschlossen, müssen entsprechend Konsequenzen gezogen werden, um die Gefährdung zu minimieren. Einige Beispiele sind der Einsatz einer Unterbrechungsfreie Stromversorgung (USV)⁴ oder Datensicherungsmechanismen.

Obwohl oben bereits beschrieben wurde, dass IT-Sicherheit nicht *nur* durch technische Maßnahmen herzustellen ist, behandelt dieser Abschnitt eben jene technische Möglichkeiten, die dem IT-Sicherheitsteam bzw. den Administratoren eines Unternehmens zur Verfügung stehen.

³Die Mitarbeiter sollen also aktiv auf die Notwendigkeit der IT-Sicherheit hingewiesen und entsprechend geschult werden. Sie sollen mögliche Gefahren erkennen und so weniger anfällig für Angriffe werden und die ihnen anvertrauten Systeme sicher verwenden können.

⁴Eine USV versorgt Server mithilfe von Batterien solange mit Strom, bis ein sicheres Herunterfahren abgeschlossen ist. Somit werden Datenverluste durch nicht abgeschlossene Schreibvorgänge verhindert.

Diese Technologien sind in hardware- und softwareseitige Lösungen zu unterteilen, welche sich im Idealfall sinnvoll ergänzen und die Usability⁵ der Systeme nicht unangemessen beeinträchtigen. Aber es ist zu beachten, dass nicht immer die Technologie mit dem höchsten Anschaffungspreis auch automatisch die Beste ist – vor allem im Software-Bereich kann hier oft auf freie, oder sogar kostenlose Produkte zurückgegriffen werden.

Hardwareseitig sei vor allem die altbewährte Firewall erwähnt. Mit ihr können unbefugte Zugriffe sowohl von außerhalb der Unternehmens-IT-Infrastruktur nach innen, als auch von innen in das Internet verhindert und ggf. protokolliert werden. Letzteres kann dann sinnvoll sein, wenn ein Client-System bereits durch Schadsoftware infiziert wurde und diese Software (z. B. ein trojanisches Pferd⁶) nun versucht, den Kontakt zu seinem Ersteller herzustellen. Weiterhin können Intrusion Detection System (IDS) unbefugte Zugriffe aufdecken und Administratoren ggf. benachrichtigen.

Als Softwarelösung sei unter anderem der Virens Scanner genannt. Weiterhin ist die Nutzung von Monitoring-Software wie Nagios⁷ weit verbreitet. Mit derartiger Software kann der Zustand der eigenen IT, wie z. B. die Erreichbarkeit der Server, überwacht werden. Gleichzeitig ist bei Ausfall die Abarbeitung zuvor definierter Eskalationsstufen möglich, sowie die einfache Benachrichtigung eines verantwortlichen Administrators.

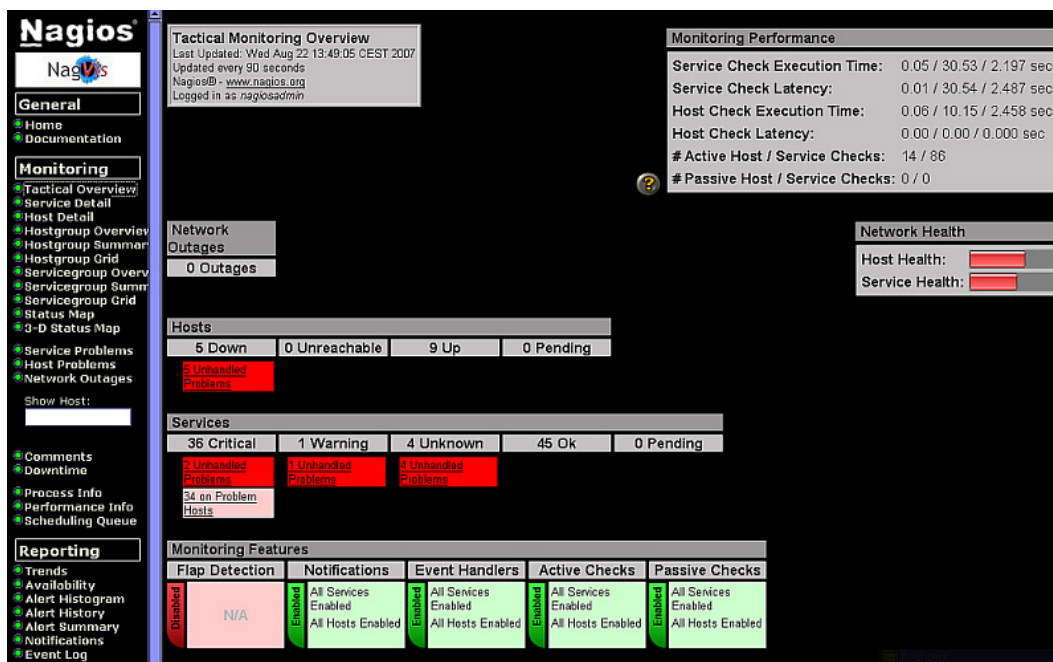


Abbildung 3.2: Nagios Taktische Übersicht[7]

⁵Also die möglichst unkomplizierte Nutzbarkeit und Bedienung des Systems durch den Nutzer.

⁶Ein Trojanisches Pferd, oder auch *Trojaner*, ermöglicht einem Angreifer den Zugriff auf das infizierte System. Der Name ist an die an die griechische Mythologie angelehnt. Ein mit griechischen Soldaten gefülltes Holzpferd wird als Geschenk in die Stadt Troja gebracht. Von dort öffneten die Soldaten die Stadtmauern Trojas und ließen das griechische Heer ein.

⁷<http://www.nagios.org/>, veröffentlicht als freie Software unter der GNU GPL-Lizenz

3.2.5 Menschen

Ein großer, wenn nicht der entscheidende, Faktor das IT-Sicherheitsmanagement betreffend ist der Mensch. Sowohl im Negativen, als auch im Positiven.

Beginnen wir mit dem Negativen. Obwohl der Mensch der eine Faktor ist, welcher in allen Unternehmen und in allen Sicherheitsprozessen gleich ist, ist es auch der am schwierigsten zu bewältigende Faktor. Oft unterschätzt, ist hier sowohl mit Tätern aus dem Kreis von Terroristen bis Kleinkriminellen zu kämpfen, als auch mit dem wütenden Ex-Mitarbeiter, der dem Unternehmen noch einmal etwas Böses will bevor er es verlässt. Und zu guter Letzt gibt es noch den unbedarften, unwissenden Mitarbeiter. Sein Nutzungsverhalten ist, wenn überhaupt, nur schwer vorhersehbar und das ein oder andere Mal nichteinmal rational zu erklären. Ein weiterer Punkt ist die Anfälligkeit des Menschen für Aspekte des Social Engineering.[8, S. 48 f.]

Die bereits oben beschriebenen Sicherheitsrichtlinien sollen helfen, den Nutzer in geordnete Bahnen zu lenken und zum *richtigen Handeln* zu bewegen. Weiterhin ist der Mitarbeiter durch Schulungen zu sensibilisieren und ihm die Sicherheitsrisiken klar zu machen. Auf diese Weise soll auch die Einhaltung der Richtlinien bewirkt werden. Außerdem wird auf diese Weise ein arbeitsrechtliches Mittel geschaffen. Denn erst ein durch Schulungen sensibilisierter Mitarbeiter kann bei Nichtbeachtung der Richtlinien auch entsprechend belangt werden. Die Belangbarkeit von Mitarbeitern ist auch insofern wichtig, dass die Mitarbeiter diese Schulungen weniger als Bestrafung sehen, sondern als Hilfe für ihre tägliche Arbeit. Wird die Belangbarkeit nicht angestrebt und durchgesetzt, ist es wahrscheinlicher, dass es zu schweren Sicherheitsvorfällen kommt.[5, S. 187]

Nicht erwähnt wurden bis jetzt jene Innentäter, die als Mitarbeiter im Unternehmen tätig sind und von dort mit kriminellen Hintergedanken Sabotage betreiben oder Daten stehlen. Hierbei kann es sich sowohl um langjährige Mitarbeiter handeln, als auch um solche, die erst vor Kurzem und genau zu diesem Zweck in das Unternehmen gekommen sind. Zur Prävention können zum Beispiel Sicherheitsüberprüfungen sinnvoll sein, sowie die Anwendung des einfachen Prinzips „*Kenntnis nur wenn nötig*“⁸.

Nun zu dem Positiven, also zu den Menschen, die die Umsetzung des Sicherheitsprozesses unterstützen und vorantreiben. Hier sollte in jedem Unternehmen gleich welcher Größe ein *IT-Sicherheitsbeauftragter (IT-SIBE)*, oder *Chief Information Security Officer (CISO)*, eingesetzt werden. Er ist der Hauptverantwortliche für die Umsetzung und Einhaltung des IT-Sicherheitsprozesses. Je nach Größe des Unternehmens kann der IT-SIBE durch ein IT-Sicherheitsteam unterstützt werden. Bei kleinen Unternehmen wird die Rolle des IT-SIBE ggf. durch die Geschäftsführung mit übernommen. Es bietet sich jedoch an, vor allem während der ersten Einführung eines ISMS einen hauptamtlichen IT-SIBE einzusetzen, damit sich dieser voll und ganz mit der Aufgabe beschäftigen kann.

Desweiteren wurde bereits das CERT genannt, welches als schnelle Eingreiftruppe bei Sicherheitsvorfällen aktiviert und zur Schadensbegrenzung und zur Analyse dieser Vorfälle eingesetzt wird. Aber auch hier muss je nach Unternehmensgröße entschieden werden, ob

⁸ „*Kenntnis nur wenn nötig*“ beschreibt den Grundsatz, dass Mitarbeiter nur Zugriff auf diejenigen Daten bekommen, welche sie auch unbedingt zum Arbeiten benötigen.

und in welchem Umfang ein solches Team aufgestellt wird und ob es sich um hauptamtliche oder nebenamtliche „Retter“ handelt.

3.2.6 Projektmanagement

Sowohl bei der Einführung des ISMS, als auch von neuen Technologien ist ein strukturiertes und geplantes Vorgehen absolut empfehlenswert. Dabei können die Grundsätze des Projektmanagements eine praktische Orientierungshilfe sein.

Von den vielen Modellen, welche für das Projektmanagement in der Fachliteratur erwähnung finden, wird im IT-Sicherheitsmanagement auf den Deming-Kreis, also dem Zyklus *plan-do-check-act* (PDCA), zurückgegriffen.[1, S. 393 ff.]

P plan beschreibt die Planung einer Verbesserung oder eines neuen Vorgehens. Näheres zur Planung wurde bereits in Kapitel 3.2.2 beschrieben.

D do beschreibt die Durchführung des Geplanten. Dazu gehört auch die Erfolgskontrolle und das Monitoring, also die Überwachung der Einhaltung und die Dokumentation der Ergebnisse.

C check stellt die regelmäßige, bedarfsorientierte Überprüfung des Sicherheitsstruktur dar.

A act beschreibt die unmittelbare beseitigung von Mängeln. Anschließend wird für umfangreichere Verbesserungen wieder bei *plan* gestartet.

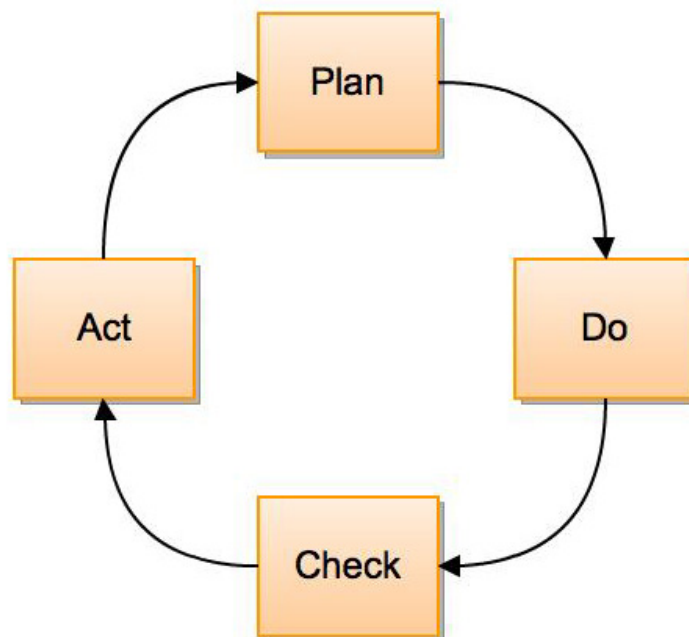


Abbildung 3.3: PDCA-Kreis

Dieser Ablauf ist so auch in dem ISO-Standard 27001:2005 zu finden (siehe 3.3.1). Er lässt sich nahezu auf alle Aspekte und Elemente des IT-Sicherheitsmanagement anwenden.[2, S. 27]

Da man mit dem Thema Projektmanagement sicher noch einige Kapitel füllen könnte, sei hier für weitere Informationen auf die entsprechende Fachliteratur (wie [9]) verwiesen.

3.3 Verfahren

Im letzten Kapitel wurde gezeigt, welche Elemente in einem ISMS enthalten sind. Man konnte bereits erkennen, dass es sich bei der Einrichtung eines neuen ISMS um ein komplexes Unterfangen handelt. Um diesen Vorgang zu vereinfachen, kann auf etablierte Verfahren und Systeme zurückgegriffen werden. Der Anfang wurden von kleinen, nur national gültigen, Systemen gemacht. Später wurden einige dieser Systeme abgeglichen und zusammengefasst. Dadurch entstanden die heute etablierten, international gültigen Verfahren, wovon im Folgenden einige kurz vorgestellt werden.

Durch die Verwendung von Standards wird gleichzeitig eine Zertifizierung ermöglicht. Während einige Zertifikate einen eher ideologischen Wert haben – weil sie z. B. derartig allgemein gefasst sind, dass durch die Zertifizierung keine Aussage über die Qualität des ISMS getroffen werden kann – können andere z. B. von Banken für die Vergabe von Krediten oder von öffentlichen Auftraggebern für die Vergabe von Aufträgen verlangt werden.

3.3.1 ISO/IEC 27001:2005

Die International Organization for Standardization (ISO) bzw. International Electrotechnical Commission (IEC) hat 2005 den Standard ISO 27001:2005 veröffentlicht, und zuletzt 2008 aktualisiert. Dieser Standard, sowie die komplette 27000er-Reihe beschäftigt sich allein mit dem IT-Sicherheitsmanagement.

Inhalt der 27001 ist der Aufbau und Betrieb eines ISMS. Er orientiert sich an dem bereits vorgestellten PDCA-Zyklus und spezifiziert dessen Phasen. So werden für jede Phase diverse Aktivitäten angegeben, die mit dieser verbunden sind und entsprechend durchgeführt werden sollen. Einige dieser Aktivitäten wurden in dieser Arbeit bereits vorgestellt.

Plan Der Bereich *plan* besteht aus insgesamt elf Unterpunkten, PLAN-a bis PLAN-k. Dazu gehört neben der Festlegung des Anwendungsbereichs des ISMS auf die Erstellung einer Sicherheitsleitlinie und die Identifizierung und Bewertung von Risiken. Sind die Risiken analysiert, sollen Optionen zur Risikobehandlung und die daraus resultierenden Maßnahmen bestimmt werden. Das Restrisiko wird akzeptiert, und der Auftrag zur Einführung des ISMS wird gegeben. Am Ende dieser Phase muss eine ausführliche Dokumentation zu allen in der Phase getroffenen Entscheidungen erstellt und in der „*Erklärung der Anwendbarkeit*“ festgehalten werden.[10, S. 39 ff.]



Abbildung 3.4: Zertifikat nach ISO 27001, auf Basis der „*Erklärung der Anwendbarkeit*“ [11]

Do Eingeteilt in acht Unterpunkte, geht es in dieser Phase darum, das vorher geplante umzusetzen. Begonnen werden soll mit der Aufstellung eines Risikobehandlungsplans. Obwohl diese Aktivität vom Namen eher zur *Plan*-Phase zuzuordnen wäre, ist sie aufgrund ihrer praktischen Anteile erst in der *Do*-Phase zu finden. Dazu gehört unter anderem die Verteilung von Verantwortlichkeiten und die Durchführung von Audits⁹ und Bewertungen. Als nächste Schritte folgen die Umsetzung des Risikobehandlungsplans und der daraus resultierenden Maßnahmen. Die Wirksamkeit der Maßnahmen soll abgeschätzt werden, und Mitarbeiter in Sicherheitsbelangen sensibilisiert und geschult werden. Außerdem wird die Verwaltung und der Betrieb des ISMS dieser Phase zugeordnet, sowie das Ressourcenmanagement. Aber auch die Erkennung und Bearbeitung von Sicherheitsvorfällen ist Teil dieser Phase. Entsprechende Konzepte, wie der IRP wurden bereits erläutert. [10, S. 54 ff.]

⁹Also die Bewertung des ISMS auf Basis der ISO-Norm. Audits werden auch durchgeführt, wenn eine Zertifizierungsstelle die Einhaltung der Norm überprüft, um ggf. ein Zertifikat auszustellen.

Check Nach der Durchführung folgt die ebenfalls acht Aktivitäten umfassende Phase der Überprüfung. Dazu gehört die Überwachung, also z. B. die schnelle Erkennung von Sicherheitsvorfällen, genauso wie die regelmäßige Überprüfung der Wirksamkeit des ISMS. Dazu sollen ebenfalls die getroffenen Maßnahmen anhand ihrer Wirksamkeit bewertet werden. Weiterhin wird eine regelmäßige Wiederholung der Risikobewertung vorgeschrieben, sowie die Durchführung interner Audits. Aber nicht nur die einzelnen Komponenten sollen regelmäßig kontrolliert werden, sondern auch das Management des ISMS an sich. Wenn die Sicherheitspläne aktualisiert und Handlungen und Ereignisse, die sich auf das ISMS auswirken (können), aufgezeichnet werden, ist auch diese Phase nach ISO-Norm abgeschlossen.[10, S. 62 ff.]

Act Zu guter Letzt folgt die *Act*-Phase, die aus vier Aktivitäten besteht. Identifizierte Verbesserungsmöglichkeiten sollen umgesetzt werden. Außerdem ist aus Sicherheitsvorfällen zu lernen und die entsprechenden Schlüsse für die Weiterentwicklung des ISMS zu ziehen. Die (geplanten) Verbesserungen sind genauso wie die Einrichtung eines ISMS an sich an die Betroffenen zu kommunizieren – ggf. sind die Änderungen auch mit Vertragspartnern zu klären, vor allem, wenn durch die Veränderungen Vertragsverhältnisse berührt werden. Bei und nach Einführung einer Verbesserung ist wieder die Erfolgskontrolle unerlässlich. Es ist je nach Unternehmen und Verbesserung abzuwägen, ob die Erfolgskontrolle zeitnah oder im Rahmen des nächsten regelmäßigen Audits vorzunehmen ist.[10, S. 68 ff.]

Neben der Einhaltung verlangt die ISO 27001 außerdem die ausführliche Dokumentation aller Vorgänge und Ergebnisse. Hier sei noch einmal besonders auf die „*Erklärung der Anwendbarkeit*“ verwiesen, welche unter anderem auch von Auditoren zur Prüfung des ISMS herangezogen werden kann.

3.3.2 IT-Grundschutz

Der im deutschsprachigen Raum wohl bekannteste Standard bzgl. der IT-Sicherheit ist der vom BSI entwickelte IT-Grundschutz. Dieser ist aufgeteilt in die BSI-Standards zur IT-Sicherheit, bestehend aus den Standards 100-1 bis 100-3, sowie die IT-Grundschutzkataloge. Letzteres, ein als Loseblatt-Sammlung vertriebenes Werk, umfasst mittlerweile über 4.000 Seiten und kann vom BSI als Ordner-Sammlung oder kostenfrei als pdf-Download bezogen werden. Das Gesamtwerk zum IT-Grundschutz ist auf die ISO 27001 abgestimmt. Somit kann ein nach Grundschutz umgesetztes ISMS gleichzeitig nach ISO 27001 zertifiziert werden. Auf der anderen Seite wird eine Zertifizierung nach Grundschutz zusätzlich von der ISO anerkannt. Auch der PDCA-Zyklus der ISO-Norm findet beim Grundschutz wieder Anwendung.

BSI-Standards zur IT-Sicherheit Der Standard 100-1 beschreibt die allgemeinen Anforderungen an ein ISMS. Es wird eine leicht verständliche Vorgehensweise vorgestellt, womit ein ISMS nach ISO-Norm umgesetzt werden kann.

Um die konkrete IT-Grundschutz-Vorgehensweise geht es in dem BSI-Standard 100-2. Hier wird auf Basis der in 100-1 festgestellten Rahmenbedingungen und Anforderungen

der Aufbau eines IT-Sicherheitsmanagements beschrieben. Er geht intensiv auf die Erstellung eines Sicherheitskonzeptes, sowie der Entwicklung angemessener IT-Sicherheitsmaßnahmen ein.

Im Standard 100-3 wird eine vom BSI entwickelte Methodik zur Risikoanalyse erklärt, welche auf dem IT-Grundschutz basiert. Sie bietet sich für Unternehmen an, welche bereits erfolgreich ein ISMS nach IT-Grundschutz etabliert haben und eine ergänzende Sicherheitsanalyse durchführen wollen.[2, S. 12 f.]

Das Notfallmanagement wird im BSI-Standard 100-4 beschrieben. Er behandelt sowohl die Konzipierung und Planung, als auch Übungen zum Überprüfen des Notfallmanagements.

Neben den Standards stellt das BSI auch Beispiel-Konzepte für Unternehmen unterschiedlicher Größe zur Verfügung, welche zur Umsetzung und Etablierung des eigenen IT-Sicherheitsmanagements verwendet werden können. Außerdem ist noch ein Beispiel für das produzierende Gewerbe vorhanden.

IT-Grundschutzkataloge Die Grundschutzkataloge bestehen aus derzeit 85 abgeschlossenen Bausteinen. Jeder dieser Bausteine behandelt jeweils ein abgeschlossenes Thema. Zu diesem werden technische Zusammenhänge erklärt, dazu passende Gefahren aufgelistet und in einem umfangreichen Gefahrenkatalog beschrieben. Dazu passende Lösungsmöglichkeiten werden ebenfalls aufgelistet und in einem Maßnahmenkatalog erläutert. Die Kataloge stellen, im Gegensatz zu vielen anderen Verfahren, einen technischen Leitfaden dar, der geprägt ist durch Übersichten, Checklisten und Beispielen. Ziel ist eine möglichst einfache Umsetzung des IT-Grundschutz. Durch die Modularität ist es auch möglich, das Sicherheitsniveau schrittweise zu erreichen, denn die einzelnen Bausteine können unabhängig voneinander bearbeitet werden.[4, S. 196]

Als weitere Hilfestellung für die Einrichtung eines ISMS bietet die BSI die Software *GSTOOL* an. Mit diesem (für Behörden kostenlosen) Tool kann der Anwender die IT-Infrastruktur seines Unternehmens erfassen, den Schutzbedarf feststellen und Elemente der IT-Infrastruktur den Bausteinen aus den IT-Grundschutzkatalogen zuordnen. Das Tool unterstützt bei der Abarbeitung aller notwendigen Schritte zur Einrichtung eines ISMS nach IT-Grundschutz.

3.3.3 OCTAVE

Die von der Carnegie Mellon University entwickelte Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE)-Methode beschäftigt sich mit dem Teilaspekt der Risikoanalyse. Mit Hilfe von OCTAVE kann eine bestehende IT-Infrastruktur analysiert, und festgestellt werden, wo Handlungsbedarf besteht. OCTAVE besteht aus drei Phasen. Die Ergebnisse jeder dieser Phasen können in mehreren Workshops erarbeitet werden und resultieren in einem Handlungsplan, auf dessen Basis ein IT-Sicherheitsmanagement initiiert werden kann. Die Entwicklung wurde von dem US Department of Defense gesponsort, und das Verfahren ist auf Unternehmen und Organisationen ab einer Größe von 300 Mitarbeitern ausgelegt.

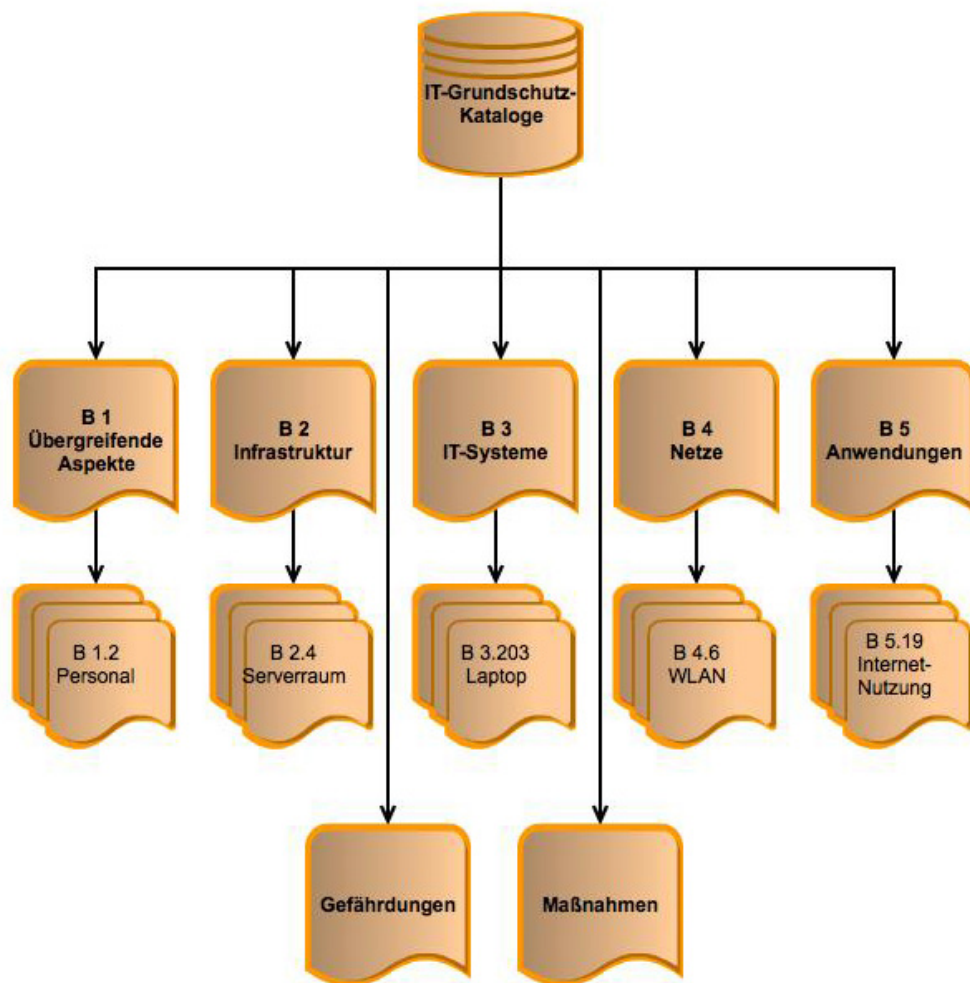


Abbildung 3.5: Aufbau der IT-Grundschutzkataloge

Phase 1: Schutzbedarfsfeststellung In dieser Phase gilt es zu definieren, welche Unternehmenswerte schützenswert sind, und in welchem Umfang dies geschehen soll. Dazu wird in vier Prozessen festgestellt, welche Elemente für das obere und untere Management schützenswert sind. Anschließend wird auch die Meinung ausgewählter Mitarbeiter, z. B. aus der IT-Abteilung, eingeholt. Ist dieser Vorgang abgeschlossen, werden den festgestellten Elementen Schutzbedarfsstufen zugeordnet und die Gefährdungen analysiert.

Phase 2: Infrastrukturelle Schwachstellen identifizieren In der zweiten Phase soll die Infrastruktur auf Schwachstellen untersucht werden. Dazu wird für jedes der Elemente aus Phase 1 untersucht, welche Komponenten der IT-Infrastruktur dafür von großer Bedeutung sind. Die so identifizierten Komponenten werden einzeln auf Schwachstellen untersucht. Diese Schwachstellen werden dann mit in das Ergebnis der Schutzbedarfsfeststellung integriert, um ihre Bedeutung für die einzelnen Elemente festzustellen.

Phase 3: Sicherheitsstrategie und –pläne entwickeln Die dritte Phase verarbeitet nun all die vorher gesammelten Informationen zu einer Sicherheitsstrategie. Dazu müssen Risiken für die einzelnen schützenswerten Elemente festgestellt und analysiert werden. Anschließend müssen Kriterien entwickelt werden, anhand derer diese Risiken bewertet und evaluiert werden können. Außerdem muss festgelegt werden, wie auf diese Risiken reagiert werden soll. Ergebnis dieses Prozesses sind Pläne zur Risikominderung, welche in der späteren Entwicklung eines ISMS berücksichtigt werden sollten.[12, S. 3 f.]

Die Carnegie Mellon University stellt dem Anwender von OCTAVE ein umfassendes Werk an (englisch-sprachigen) Zusatzmaterialien zur Verfügung. So werden bereits vorgefertigte Präsentationen für die Workshops, sowie Fragebögen für die in Phase 1 stattfindenden Erhebungen geliefert. Durch die Anwendung von OCTAVE allein wird jedoch noch kein ISMS etabliert. Dieser Prozess beginnt erst *nach* OCTAVE.

3.3.4 COBIT

Control Objectives for Information and related Technology (COBIT) ist ein vom IT-Governance Institute entwickeltes, und von der Information Systems Audit and Control Association (ISACA) herausgegebenes Standardwerk, welches sich hauptsächlich mit dem Audit von IT-Umgebungen und Prozessen beschäftigt, also mit der Messung und Bewertung von IT-Prozessen. Anders als bei den bereits vorgestellten Standards ist COBIT nicht alleine auf die IT-Sicherheit fokussiert – aufgrund des umfassenden Ansatzes wird die IT-Sicherheit jedoch nicht vernachlässigt.

COBIT besteht aus Kontrollzielen und einer Struktur zur Klassifizierung des IT-Systems. Um die Ziele zu erreichen, werden Aktivitäten angegeben, welche zu Prozessen verknüpft werden. Diese Prozesse werden wiederum thematisch zu Domänen zusammengefasst. Bei COBIT sind die Prozesse zu vier Domänen zusammengefasst, welche im Folgenden kurz vorgestellt werden.[13, S. 27]

Planung und Organisation In der ersten Domäne *Planung und Organisation* steht die Entwicklung einer Strategie für die IT im Vordergrund. Ziel dieser Strategie soll es sein, die Geschäftsziele zu erreichen. So wird sie auch laufend von der Geschäftsstrategie beeinflusst. Weiterhin soll die IT-Architektur, die technische Ausrichtung und die IT-Organisation definiert werden. Da diese Bereiche aktiv voneinander beeinflusst werden, sollte die Entwicklung Hand-in-Hand ablaufen. Die Ergebnisse dieser Prozesse fließen in das Investitionsmanagement, welches sich mit der finanziellen Seite beschäftigt, und in das Personalführungsmanagement. Die Management-Ziele sollen an die betroffenen Personen herangetragen werden. Zu guter Letzt stehen noch Qualitäts-, Risiko- und Projektmanagement. Diese Aktivitäten betreffen jedoch nicht nur diese Domäne, sondern zusätzlich auch alle Nachfolgenden.

Akquisition und Implementierung Die Ergebnisse der letzten Domäne fließen nun in die Domäne *Akquisition und Implementierung* ein. Aufgrund jener sollen automatisierte

IT-Lösungen erworben bzw. entwickelt, und anschließend in die IT-Infrastruktur implementiert werden. Dabei geht es sowohl um Hardware-, als auch um Software-Lösungen. Sind die Systeme erworben, hat die Schulung der Anwender stattzufinden, bevor mithilfe des Change Management¹⁰ das System in die IT eingeführt wird.

Delivery und Support Kernaufgabe der Domäne *Delivery und Support* ist die Inbetriebhaltung, also die Administration der IT-Infrastruktur. Dazu gehören auch Teilbereiche wie IT-Sicherheit und die Schulung von Mitarbeitern. Dazu sollen die Anforderungen im Bereich IT-Sicherheit und Betrieb durch das Service Level Management erfasst und in eine Betriebsorganisation überführt werden. So können interne Service Level Agreements (SLAs)¹¹ entwickelt werden. Hier soll auch die Zusammenbarkeit mit externen Lieferanten, wie Service-Dienstleistern, festgelegt werden. Aus dem Service Level Management ergeben sich Data-, Facility und Operationsmanagement. Außerdem ist über die Einrichtung eines Service Desk für die IT-Probleme der Mitarbeiter nachzudenken. Zu guter Letzt ist die nicht zu vernachlässigende Dokumentation in diese Domäne eingeordnet.

Monitoring und Evaluation Die regelmäßige Überwachung und Kontrolle festgelegter Ziele ist Teil der Domäne *Monitoring und Evaluation*. Die IT-Infrastruktur soll anhand der festgelegten Standards und Qualitätskriterien kontrolliert, sowie die Einhaltung gesetzlicher Vorgaben sichergestellt werden. Dies kann auch durch interne bzw. externe unabhängige Audits geschehen. Zu beachten sind auch Veränderungen in externen Anforderungen sowie den sonstigen Geschäftsanforderungen, da sich z. B. gesetzliche Vorgaben ändern und somit direkt auf die bestehende IT-Infrastruktur auswirken können.

3.4 Zusammenfassung

Es wurde gezeigt, dass IT-Sicherheit ein komplexes Thema ist, und dass die Verbesserung der IT-Sicherheit nicht allein durch die Einführung neuer Technologien sichergestellt wird. Vielmehr muss eine geregelte IT-Sicherheitsstruktur entwickelt werden. Für den Entwurf und Betrieb solcher ISMS ist heutzutage ein großer Fundus an Verfahren verfügbar. Diese Verfahren dienen als Beispiel und Vorlage für die Entwicklung eines eigenen, auf das Unternehmen zugeschnittenen ISMS.

Die Wahl des für das Unternehmen passenden Verfahrens ist von dem Sicherheitsniveau abhängig, welches man erreichen will. Die ISO 27001:2005 stellt einen international anerkannten, zertifizierbares Verfahren vor, welches ein grundlegendes Sicherheitsniveau herstellen kann. Sie ist vor allem in Verbindung mit dem Vorgehen nach IT-Grundschutz

¹⁰Im Change Management geht es um die Integration neuer Systeme in eine bestehende IT-Infrastruktur. Dabei muss darauf geachtet werden, dass der Betriebsablauf möglichst wenig und so kurz wie möglich gestört wird.

¹¹SLAs halten die verlangte Verfügbarkeit von Systemen fest. Wird ein SLA mit einem externen Dienstleister geschlossen, kann so z. B. die Reaktionszeit bei Ausfällen vertraglich festgelegt, oder die maximal geduldete Ausfallzeit eines Systems definiert werden.

interessant, da hier das ISO-Zertifikat mit praktischen Hinweisen zur Umsetzung kombiniert wurde. Für große Unternehmen, welche besonderen Wert auf die Risikoanalyse legen, bietet sich OCTAVE an. Hierbei ist jedoch zu beachten, dass OCTAVE allein noch kein ISMS entwickelt. Hier ist das Verfahren ggf. mit einem anderen zu verbinden. Einen ganzheitlichen Ansatz für die IT-Umgebung liefert COBIT. Es beschäftigt sich nicht nur mit der IT-Sicherheit, sondern mit allen Aspekten der IT-Infrastruktur.

Aufgrund der immer stärkeren Verflechtung von IT in die Unternehmensprozesse, ist eine umfangreiche Betrachtung des Themenkomplexes *IT-Sicherheit* für das Management von Unternehmen unumgänglich. Diese Arbeit konnte jedoch nur einen Anhalt über die Komplexität dieses Feldes geben. Aus diesem Grund sei für das weitergehende Studium auf die referenzierte Literatur verwiesen.

Literaturverzeichnis

- [1] MÜLLER, Klaus-Rainer: *IT-Sicherheit mit System Sicherheitspyramide – Sicherheits-, Kontinuitäts- und Risikomanagement – Normen und Practices – SOA und Softwareentwicklung*. Wiesbaden : Vieweg, 2008
- [2] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (Hrsg.): *IT-Sicherheitsmanagement und IT-Grundschatz : BSI-Standards zur IT-Sicherheit*. Köln : Bundesanzeiger-Verlag, 2005
- [3] ECKERT, Claudia: *IT-Sicherheit : Konzepte – Verfahren – Protokolle*. München : Oldenbourg Verlag, 2006
- [4] DINGER, Jochen ; HARTENSTEIN, Hannes: *Netzwerk- und IT-Sicherheitsmanagement*. Karlsruhe : Universitätsverlag Karlsruhe, 2008
- [5] WHITMAN, Michael E. ; MATTORD, Herbert J.: *Management of Information Security*. Boston : Thomson Course Technology, 2008
- [6] *Red Hat Enterprise Linux 4: Sicherheitshandbuch: Erstellen eines Incident-Response-Plans*. <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-de-4/s1-response-plan.html>, Abruf: 01.07.2012
- [7] *Dreiling-Datentechnik GmbH – Nagios*. <http://www.dreiling-datentechnik.com/ex/nagios.html>, Abruf: 11.07.2012
- [8] KLIPPER, Sebastian: *Information Security Risk Management : Risikomanagement für ISO/IEC 27001, 27005 und 31010*. Wiesbaden : Vieweg + Teubner, 2011
- [9] KRAUS, Georg ; WESTERMANN, Reinhold: *Projektmanagement mit System : Organisation, Methoden, Steuerung*. Wiesbaden : Gabler, 2010
- [10] KERSTEN, Heinrich ; REUTER, Jürgen ; SCHRÖDER, Klaus-Werner: *IT-Sicherheitsmanagement nach ISO 27001 und Grundschatz : der Weg zur Zertifizierung*. Wiesbaden : Vieweg, 2008
- [11] *Host Europe GmbH – Unternehmen – ISO 27001*. <http://www.hosteurope.de/content/Unternehmen/ISO27001>, Abruf: 12.07.2012
- [12] ALBERTS, Christopher J. ; DOROFEE, Audrey J. ; ALLEN, Julia H.: *OCTAVE Catalog of Practices, Version 2.0 / Carnegie Mellon Software Engineering Institute*. 2001.

- [13] GOLTSCHÉ, Wolfgang: *COBIT kompakt und verständlich : der Standard zur IT-Governance ; so gewinnen Sie Kontrolle über Ihre IT ; so steuern Sie Ihre IT und erreichen Ihr Ziele*. Wiesbaden : Vieweg, 2006

Kapitel 4

Einblick in den Aufbau und Analysemöglichkeiten von Speichermedien und Standarddateisystemen vor einem forensischen Hintergrund

Sandy-Dorothea Hein

Diese Seminararbeit behandelt das Thema "IT-Forensik" mit besonderem Fokus auf den Aufbau und Analysemöglichkeiten von Speichermedien und Standarddateisystemen. Im Verlauf der Arbeit wird ein grundlegender Überblick über die Thematik IT-Forensik und ihre gerichtliche Verwertbarkeit gegeben. Weiterhin werden die für eine forensische Untersuchung notwendigen Grundkenntnisse über Speichermedien und Standarddateisysteme behandelt und eine Auswahl an nützlichen Tools aufgelistet.

Inhaltsverzeichnis

4.1	Einleitung	85
4.2	Begriffsdefinitionen	86
4.3	Digitale Spuren	86
4.4	Verwertbarkeit vor Gericht	87
4.5	Incident Response	88
4.5.1	IT-Forensik vs. Incident Response	88
4.6	Ausgewählte Speichermedien	89
4.6.1	Festplattenlaufwerk	89
4.6.2	Flashspeicher	91
4.7	Aufbau und Analyse von Standarddateisystemen	92
4.7.1	File Allocation Table	94
4.7.2	New Technology File System	96
4.7.3	Extended File System	98
4.8	Auswahl nützlicher Tools	99
4.9	Zusammenfassung	101

4.1 Einleitung

Innerhalb der letzten Jahrzehnte haben Computersysteme sowohl in unserem privaten als auch beruflichen Umfeld zunehmend an Bedeutung gewonnen. Neben den offensichtlichen Computersystemen, wie PCs, Notebooks, Supercomputer, Handys, Smartphones und Tablets, handelt es sich u.a. auch bei Überwachungskameras und Geldautomaten um solche. Letztere können bspw. bei der Aufklärung von Straftaten, wie Einbruch und Diebstahl, und der dazugehörigen Täterüberführung von hohem Nutzen sein.

Nun bringt die Tatsache, dass heutzutage Computer unentbehrlich und zu dem Hauptspeicher- und Hauptkommunikationsmedium unserer Gesellschaft geworden sind, auch gewisse Gefahren mit sich. Der Fakt, dass wir nahezu alles auf unseren Computersystemen speichern – ob bewusst oder unbewusst – macht sie v.a. im beruflichen Umfeld und beim Umgang mit sensiblen Daten zu einem wichtigen Zielobjekt krimineller Handlungen.

Um solche kriminellen Handlungen professionell zu untersuchen und nachzuweisen, kam es zur Entstehung einer neuen Wissenschaft, der IT-Forensik.

Die IT-Forensik umfasst diverse Teilgebiete und setzt sehr viele Grundkenntnisse voraus. Diese Arbeit setzt den Fokus auf einige Grundlagen der IT-Forensik mit besonderem Augenmerk auf das notwendige Grundwissen über Speichermedien und Standarddateisysteme in Bezug auf forensische Ermittlungen.

Dazu wird zunächst auf die grundlegenden Begriffsdefinitionen eingegangen, gefolgt von einem Einblick in die Bedeutung digitaler Spuren. Anschließend wird die Notwendigkeit der Verwertbarkeit vor Gericht und die dafür unumgänglichen Prinzipien genauer betrachtet. Nach einer Erklärung des Begriffs "Incident Response", wird diese der IT-Forensik gegenüber gestellt.

Nach der Vermittlung dieser Grundlagen richtet sich der Fokus der Arbeit auf den Aufbau und die Analysemöglichkeiten von Speichermedien und Standarddateisystemen. Im Zuge der Erläuterungen zu den Speichermedien wird genauer auf das Festplattenlaufwerk und den Flashspeicher eingegangen, während hierbei näher auf das "File Allocation Table", "New Technology File System" und "Extended File System" eingegangen wird. Abschließend werden einige nützliche Analyse-Tools aufgelistet und die Resultate der Arbeit zusammengefasst dargestellt.

4.2 Begriffsdefinitionen

Bei der IT-Forensik, auch digitale Forensik genannt (engl. computer forensics), handelt es sich um ein Teilgebiet der Forensik.

Der Begriff "Forensik" stammt von dem lateinischen Wort "forum" (dt. Marktplatz) ab und beschreibt im Allgemeinen den Vorgang der Verbrechensaufklärung. Der Zusammenhang zum Begriff "Marktplatz" lässt sich dadurch begründen, dass es früher üblich war, Gerichtsverfahren sowie den Strafvollzug öffentlich auf dem Marktplatz durchzuführen. [3, 9]

Die Methoden der IT-Forensik befassen sich mit der Aufklärung und dem Nachweis von strafbaren Handlungen in Bezug auf IT-Systemen und führen im Zuge dieser eine digitale Spurensicherung durch. Die forensische Informatik erweitert die IT-Forensik um die Anwendung von wissenschaftlichen Methoden zur Zielerfüllung.

Die IT-Forensik lässt sich in die zwei Spezialgebiete Computer-Forensik und Netzwerk-Forensik einteilen. Bei der Computer-Forensik steht die Analyse von digitalen Speichermedien im Vordergrund, während sich die Netzwerk-Forensik mit der Überwachung von Netzwerken und der Untersuchung von flüchtigen und dynamischen Daten eines Netzwerks, wie bspw. dem E-Mail-Verkehr oder Chat-Sitzungen befasst. [1, 2, 8, 9] Der Fokus dieser Arbeit liegt auf dem Bereich der Computer-Forensik und somit der Auswertung von Spuren auf digitalen Speichermedien.

4.3 Digitale Spuren

Der französische Mediziner Edmund Locard (1877-1966) prägte die Aussage: "Jeder und alles am Tatort nimmt etwas mit und lässt etwas zurück". [8] Dieser Satz wurde zu dem Grundprinzip der forensischen Wissenschaft und trägt mittlerweile den Titel "Locard's Exchange Principle". Das lässt sich auch auf die Computer-Forensik übertragen und wird dort unter dem Titel "Das digitale Austauschprinzip" geführt: "In jedem hinreichend komplexen digitalen System hinterlässt Datenverarbeitung notwendigerweise Spuren." [8]

Eine Spur (engl. evidence) ist ein Nachweis zur Unterstützung oder Widerlegung einer Theorie über einen Tathergang. Im forensischen Zusammenhang haben solche Spuren die Eigenschaft, vor Gericht als Beweis verwertbar zu sein und werden daher auch oft als Beweismittel bezeichnet. Eine Spur wird aber erst vor Gericht zu einem tatsächlichen Beweis.

Spuren, die auf Daten basieren, welche auf einem Computersystem gespeichert oder auf ein solches übertragen worden sind, werden als digitale Spuren (engl. digital evidence) bezeichnet.

Prinzipiell werden zwei grundlegend verschiedene Arten von digitalen Spuren unterschieden. Auf der einen Seite gibt es die technisch vermeidbaren Spuren. Diese wurden um ihrer selbst Willen erzeugt und sind z.B. Log-Dateien oder Zeitstempel in Dateisystemen. Die technisch unvermeidbaren Spuren andererseits sind Spuren, die unweigerlich anfallen und nicht "wegkonfigurierbar" sind, wie bspw. gelöschte Dateien im Dateisystem. Letztere Art stellt die für einen IT-Forensiker besonders interessanten Spuren dar.

Anfallen können digitale Spuren nicht nur auf dem heimischen Rechner (Personal Firewall, Virensch scanner, Cache- und Verlaufs-Speicher des Browsers, Log-Dateien, etc.), sondern auch auf dem Handy oder Telefon (Wahlwiederholungsfunktion, Abrechnungsdaten des Mobilfunk anbieters, etc.) oder an Geldautomaten und digitalen Überwachungskameras – eben immer dort, wo es sich um ein komplexes Computersystem handelt.

Für einen Täter ist es fast unmöglich seine Spuren restlos zu zerstören, da selbst gelöschte Dateien in der Regel noch über einen längeren Zeitraum auf dem Datenträger rekonstruiert werden können. Somit verschaffen digitale Spuren dem Ermittler einen alles entscheidenden Vorteil bei der Beweissicherung und Täterüberführung. [8, 9]

4.4 Verwertbarkeit vor Gericht

Das Hauptziel einer jeder forensischen Untersuchung ist, dass die Auswertung der Spuren auch vor Gericht verwertbar ist. Um dies zu gewährleisten, bedarf es einiger grundlegenden Prinzipien.

Eine wichtige Rolle bei der forensischen Analyse spielt die Wahrung der Persönlichkeitsrechte des Einzelnen inkl. aller dazugehörigen datenschutzrechtlichen Aspekte. Bis das zu untersuchende System in die Hände eines Forensikers gelangt, muss es zuerst rechtmäßig sichergestellt bzw. beschlagnahmt werden, um die Zulässigkeit vor Gericht zu gewährleisten. Da es sich dabei immer um einen Eingriff in die Grundrechte einer Person handelt, wenn dessen Haus, Computer, Systeme, etc. durchsucht werden, muss der Ermittler auf der Basis einer rechtlichen Ermächtigungsgrundlage handeln, die in der Strafprozessordnung festgehalten ist. Dazu nimmt der Ermittler in der Praxis Kontakt zu Juristen bzw. Staatsanwälten auf, um einen ordnungsgemäßen Ablauf zu gewährleisten. Zur Durchsuchung und Beschlagnahmung ist in der Regel ein Gerichtsbeschluss notwendig. Darin muss genau festgehalten werden, wo und welche Spuren sichergestellt werden sollen, mit der Begründung, warum wahrscheinlich genau dort Spuren einer bestimmten Straftat zu finden sein sollen. Bei einem zusätzlichen Spurenfund, der auf eine andere Tat hinweist, ist ein weiterer Gerichtsbeschluss zu beantragen, um diese sicherzustellen. [10, 16]

Handelt es sich bei dem sicherzustellenden Objekt um dienstliche Hardware, verkompliziert dies das Vorgehen. Prinzipiell ist der Arbeitgeber zwar in diesem Fall der Besitzer der Hardware und darf somit sogar selbst bspw. den dienstlichen Computer seiner Mitarbeiter durchsuchen. Jedoch gibt es hierbei gewisse Einschränkungen, v.a. bei dem Umgang mit personenbezogenen und privaten Daten, die bei Nichtbeachtung zu juristischen Konsequenzen führen können. Daher sollten in einem solchen Fall stets Juristen zu Rate gezogen werden.

Ein lückenloser Nachweis über jede Person, die mit dem untersuchten System zu tun hatte, ist unter Angabe von Ort, Zeit und Grund unerlässlich zur Beweisführung vor Gericht. Unabhängig vom Ermittler muss auch der Forensiker einige Prinzipien wahren, um eine Verwertbarkeit vor Gericht zu garantieren. Ein entscheidender Grundsatz ist: never touch original. Dies bedeutet, dass das Originalsystem sicher verwahrt wird und nicht an diesem die Analysen durchgeführt werden. Stattdessen werden forensische Kopien erstellt (engl. forensic sound imaging). Die nötigen forensischen Untersuchungen sollten nur auf diesen Kopien und zudem nur im read-only-Modus durchgeführt werden, um einer Verfälschung der Untersuchungsergebnisse durch versehentliche Änderungen innerhalb der

Datenbestände vorzubeugen. Durch den "never touch original"-Grundsatz ist die Möglichkeit einer späteren, unabhängigen Überprüfung stets gewährleistet. [1, 8, 16]

Daneben ist das wohl wichtigste Prinzip der forensischen Untersuchung die ordnungsgemäße Dokumentation. Jeder einzelne Schritt sollte von Hand (zur Erleichterung der Authentifikation) auf Papier geschrieben werden. Dabei muss jedes Blatt mit dem Namenskürzel signiert werden und den aktuellen Ort und die Zeit (genaue Uhrzeit der Atomuhr mit Hinweis auf die Zeit auf dem Beweisstück) beinhalten. Zudem ist es zur Vorbeugung von Manipulationen empfehlenswert, die Seiten durchzunummerieren und bestenfalls zu binden. Die GUI zu dokumentieren, stellt dabei schon eine kleine Herausforderung dar. Videoaufnahmen und Screenshots sind zusätzlich zum Vier-Augen-Prinzip die beste Dokumentationsart, um die Vorgehensweise nachzuweisen. Wichtige Erkenntnisse und Ergebnisse sollten trotzdem schriftlich festgehalten werden. Allgemein gilt der Grundsatz, dass der Forensiker nie zu detailliert dokumentieren kann. [1, 8, 16]

Bei der Analyse des Systems sollte der forensische Ermittler sich an Fakten halten und keine voreiligen Schlüsse aufgrund von Vermutungen und Theorien ziehen.

Vor Gericht ist es wichtig, dass auch wenn wissenschaftliche Untersuchungen kaum auslegbar sind, der Forensiker, der die digitalen Spuren als Beweis erbringt, diese auch seriös, einfach, nachvollziehbar und verständlich präsentiert, damit auch Juristen diese nachvollziehen können. Jedoch sollte er auch in der Lage sein, alle Details des Vorgehens und alle Ergebnisse einem Sachverständigen fachlich sicher vorzutragen. [1, 8, 16]

4.5 Incident Response

In den Bereich der Incident Response gehören alle Aufgaben und Funktionen, die im technischen oder organisatorischen Bereich angefallen sind, um auf einen Sicherheitsvorfall einer Organisation zu reagieren. Das umfasst alle Aktivitäten - von Beginn der Analyse über notwendige Entscheidungsprozesse während der Bewältigung bis hin zur Beseitigung der durch den Sicherheitsvorfall hervorgerufenen Veränderungen - die ohne das Eintreten des jeweiligen Incidents nicht notwendig geworden wären. [4, 14]

4.5.1 IT-Forensik vs. Incident Response

Das IT-forensische Vorgehen hat oberste Priorität bei der gerichtsverwertbaren und detaillierten Sicherung der Spuren. Bei der Incident Response ist es jedoch meist wichtiger, das betroffene System schnellstmöglich wieder in den Tagesbetrieb überlaufen zu lassen. Der Sicherheitsvorfall wird so gehandhabt, dass die Untersuchungen an die Bedürfnisse der Organisation anpassen werden.

Das Zusammenführen der IT-Forensik und der Incident Response ist sehr sinnvoll, da dadurch die Flexibilität erhöht wird. Je nach Sachlage muss die Entscheidung getroffen werden, ob eine vollständige forensische Untersuchung Priorität hat oder nicht. Zudem lässt sich das forensische Vorgehen je nach Bedrohungslage (Bedrohung = Angreifer oder Bedrohung = großer Schaden) individuell anpassen. Ein Beispiel für eine Vereinigung der

IT-Forensik und der Incident Response stellt das Common Model von Schwittay und Freiling dar. Es erweitert das IT-forensische Vorgehen um eine Management-Komponente und ergänzt die Incident Response um eine detaillierte Analysekomponente. [8, 9, 14]

4.6 Ausgewählte Speichermedien

In jedem Computersystem kommen verschiedene Speichermedien zum Einsatz. Diese sind im Bereich der Computer-Forensik der Hauptträger digitaler Spuren, weswegen der forensische Ermittler zur professionellen Analyse Grundkenntnisse in diesem Bereich aufweisen muss. Zwei repräsentative Speichermedien sind das Festplattenlaufwerk und der immer bedeutsamer werdende Flashspeicher. In den folgenden Abschnitten wird besonders auf den Aufbau und die Funktionsweise dieser näher eingegangen.

4.6.1 Festplattenlaufwerk

Ein Festplattenlaufwerk - kurz Festplatte (engl. hard disk drive = HDD) - ist ein magnetisches Speichermedium. Auf einem Computer ist es der Hauptspeicher für nicht flüchtige Daten. Daher bietet es eine erste Grundlage zur forensischen Analyse und zur Suche nach Daten.

Die Festplatte besteht aus mehreren (bis zu zwölf) kleinen, aufeinandergesteckten und rotierenden Scheiben, die mit einem magnetischen Film beidseitig beschichtet sind. Auf beiden Seiten einer Scheibe werden Daten gespeichert. Jedoch enthält im Normalfall eine der beiden Seiten lediglich Positionierungsangaben und kann daher nicht für die eigentliche Datenspeicherung verwendet werden. Die Dichte einzelner Scheiben liegt zur Zeit bei 130GB. [7]

Sowohl die Scheiben, als auch die Plattenarme inklusive der Schreib- und Leseköpfe befinden sich zum Schutz vor Feuchtigkeit und Schmutz in einem hermetisch abgeschlossenen Gehäuse. Die heute gebräuchlichste Festplattenschnittstelle zum Datentransfer ist ATA (Advanced Technology Attachment). Mithilfe von Adaptern lassen sich ATA-Festplatten an nahezu jede andere Schnittstelle anschließen, wie z.B. an eine Netzwerk- oder USB (Universal Serial Bus) - Schnittstelle. [7]

Aufbau und Funktionsweise

Die einzelnen Scheiben rotieren mithilfe eines kleinen Antriebsmotors mit einer konstanten Geschwindigkeit. Sowohl gelesen als auch geschrieben werden die Daten auf der Festplatte von kleinen, sehr dicht über der Scheibenoberfläche schwebenden Köpfen, die an Plattenarmen befestigt sind und sich aufgrund des vorhandenen kleinen Schrittmotors (Aktuator) ähnlich wie die Nadel eines Schallplattenspielers vor und zurück bewegen. Die Köpfe verändern beim Schreibprozess die Ausrichtung der magnetischen Partikel auf der Oberfläche der rotierenden Scheiben. Das Auslesen von Informationen erfolgt durch die Induktion eines Stromes im entsprechenden Kopf aufgrund der Rotation. Berührt einer dieser Köpfe eine Scheibe, wird dies als "Head-Crash" bezeichnet, der zu starken

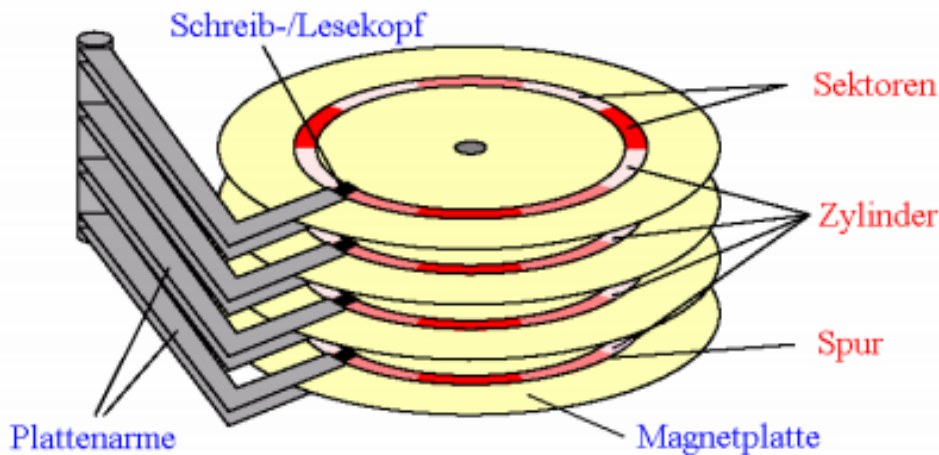


Abbildung 4.1: Aufbau einer Festplatte (hier für eine Festplatte mit vier Magnetscheiben) [12]

Zerstörungen dieser führen kann.

Zur Suche bestimmter Daten muss der Kopf an die richtige Stelle bewegt werden. Dabei heißt jede Position, die der Kopf einnehmen kann, während sich die Scheibe unter ihm bewegt, Spur. Spuren wiederum sind in weitere Sektoren unterteilt. Als Zylinder wird ein Satz von Spuren auf verschiedenen Seiten bezeichnet, die denselben Abstand zur Achse haben. Bewegen sich alle Köpfe zusammen, können die in einem einzelnen Zylinder gespeicherten Daten ohne eine weitere Bewegung gelesen werden. Jeder Sektor lässt sich somit durch Kopfnummer, Zylindernummer und Sektornummer eindeutig indentifizieren. [6, 7]

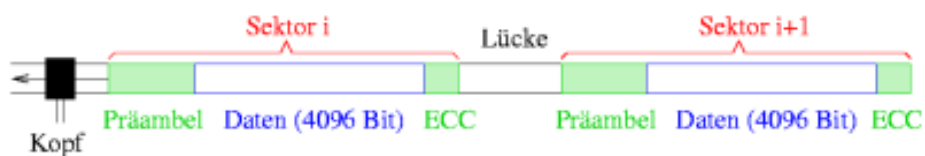


Abbildung 4.2: Organisation der Daten auf einer Festplatte) [12]

Sektoren haben in der Regel eine Länge von 512 Byte (4096 Bit) Nutzdaten. Zusätzlich enthält jeder Sektor wiederum 40 bis 100 Byte für Verwaltungsinformationen (Präambel) und zur Fehlererkennung/-korrektur (ECC–Error Correction Control). Beim Speichern von Dateien werden diese in untereinander verkettete Sektoren abgelegt. In jedem Sektor befindet sich die Information, wo der darauffolgende Sektor der Datei liegt. In einer File Allocation Table (s. 2.6.1) wird für das Betriebssystem der erste Sektor einer jeden Datei vermerkt. Die Menge aufeinanderfolgender Sektoren in einem Laufwerk wird als Partition definiert. In dem ersten Sektor eines Laufwerks befindet sich der Master Boot Record, in dem ab dem 446 Byte die Partitionstabelle zu finden ist, die u.a. Informationen über den Anfangs- und Endsektor, die Anzahl der Sektoren pro Partition und den Dateisystemtyp bereitstellt. Der forensische Ermittler überprüft die Partitionstabelle auf Konsistenz, also ob es bspw. Sektoren gibt, die keiner Partition angehören. Partitionen enthalten ein

Dateisystem, welches im ersten Sektor jeder Partition identifizierbar ist. Wurde die Partitionstabelle gelöscht, um Spuren zu beseitigen, kann der Forensiker nach entsprechenden Spuren eines Dateisystems suchen, um die einzelnen Partitionen zu identifizieren. [6, 7] Es werden immer ganze Sektoren ausgelesen oder beschrieben. Dies kann zu einem sogenannten Slack führen. Wird ein Sektor mit einer Datei beschrieben, die geringer als 4096 Bit groß ist, bleiben die restlichen Bit unbenutzt und werden als "Slack" bezeichnet. Früher wurde dieser mit Daten aus dem RAM (Random Access Memory) überschrieben, weswegen er als "RAM-Slack" bezeichnet wird, doch mittlerweile wird dieser Bereich vom Betriebssystem mit Nullen gefüllt, um diese Sicherheitslücke zu schließen. Die Lücke zwischen den einzelnen Sektoren wird "Drive-Slack" genannt und kann als Versteck für Daten dienen oder auch Daten von vorhergehenden Dateisystemen enthalten. Daher ist es für den Forensiker sehr von Interesse, den Slack auszulesen. [6, 7]

4.6.2 Flashspeicher

Der Flashspeicher, auch Flash-EEPROM (Electrically Erasable Programmable Read Only Memory) genannt, ist ein elektronisches semi-permanentes Speichermedium aus digitalen Speicherchips. Er gewährleistet eine nichtflüchtige Speicherung bei gleichzeitig niedrigem Energieverbrauch. Flashspeicher sind portabel und miniaturisiert, jedoch lassen sich bei neuen Flash-EEPROM – im Gegensatz zu gewöhnlichen EEPROM-Speichern – Bytes, die kleinsten adressierbaren Speichereinheiten, nicht einzeln löschen. Flashspeicher haben u.a. eine Verwendung in Speicherkarten für Digitalkameras und Mobiltelefonen, in USB-Sticks, MP3-Playern und sogenannten SSDs (Solid State Drives) – einer "Festplatte aus Flashzellen". Die SSD ist der Nachfolger der magnetischen Festplatte, der u.a. aufgrund der mechanischen Robustheit, sehr kurzer Zugriffszeiten und des Fehlens jeglicher Geräusentwicklung, die Festplatte ablösen wird. Das Hauptargument, das vor der Anschaffung einer SSD abschreckt, ist der aktuell noch hohe Preis. [7]

Wie bereits erwähnt, besteht auch ein USB-Stick aus Flashzellen. Dieser wird häufig zur Datenübertragung zwischen verschiedenen Computersystem genutzt und bietet sich somit als Überträger von jeglicher Schadsoftware (Würmer, Trojaner, Viren, usw.) an. U.a. zur Analyse solcher krimineller Fälle muss ein Forensiker auch Kenntnisse über den Flashspeicher besitzen.

Aufbau und Funktionsweise

Der Aufbau und die Funktionsweise eines Flashspeichers basiert auf dem des MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor).

Ein MOSFET lässt sich in die drei Anschlüsse Gate, Drain und Source, einer Oxydisolierschicht, der Sperrzone und dem Bulk aufteilen. Zu Beginn ist der Transistor nicht leitend, da durch die npn-Übergänge kein Stromfluss zugelassen wird. Durch die bestehende Oxydschicht, wird zunächst zwischen Gate und Bulk ein Kondensator gebildet, der beim Anlegen einer Spannung aufgeladen wird. Dadurch kommt es zu einer Rekombination

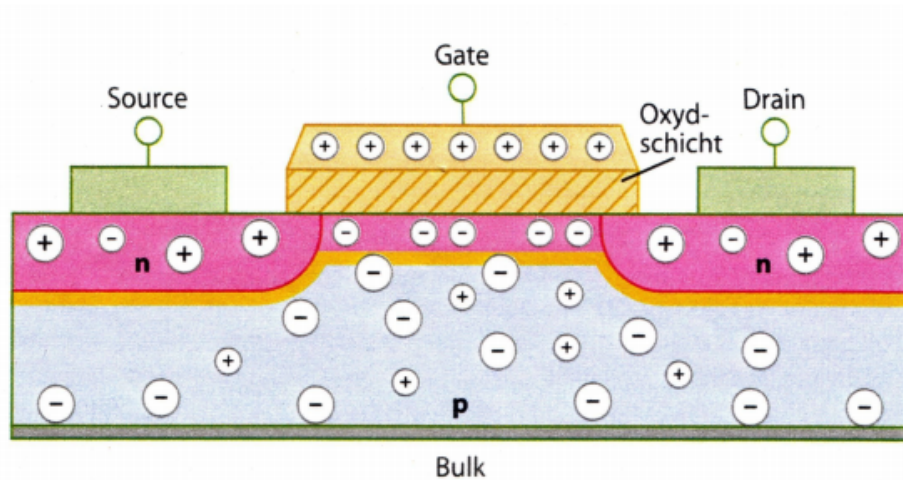


Abbildung 4.3: Aufbau eines MOSFET [11]

der Ladungsträger an den Übergängen. Das dadurch entstehende elektrische Feld bewegt die Ladungsträger zur Grenzschicht. Dabei bildet die Grenzspannung (Threshold) den n-leitenden Kanal zwischen Source und Drain. [13]

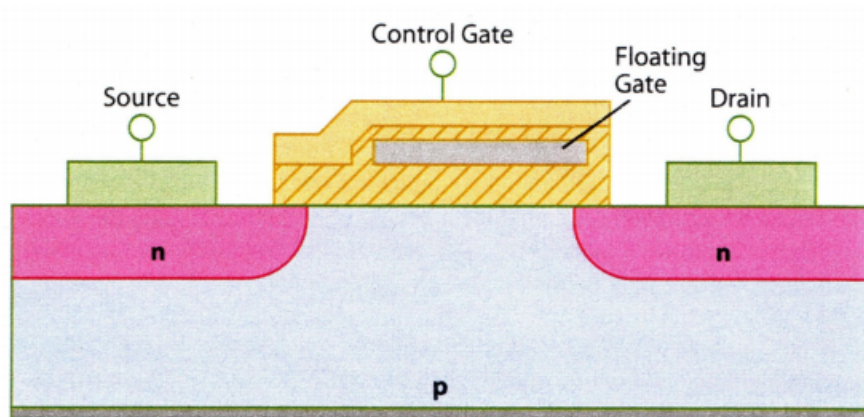


Abbildung 4.4: Aufbau einer Flash-Zelle [11]

Der Unterschied vom MOSFET zu einer Flashzelle ist das zusätzliche Floating Gate, das eigentliche Speicherelement. Es ist wie auch das Control Gate durch eine Oxydschicht isoliert.

4.7 Aufbau und Analyse von Standarddateisystemen

Das Ziel eines Dateisystems ist die Organisation abgespeicherter Daten in einer Hierarchie von Verzeichnissen und Daten. Im Normalfall sind alle Dateisysteme unabhängig von einem speziellen Betriebssystem. Bei der Dateisystemanalyse werden die Daten in einem

Laufwerk genauer betrachtet und als ein Dateisystem interpretiert. Der Forensiker weiß durch seine Kenntnisse über die Strukturen einzelner Dateisysteme, wo und wie er nach bestimmten Daten suchen muss. Daten lassen sich einer der folgenden fünf verschiedenen Kategorien zuordnen, für die es jeweils individuelle Analysetechniken gibt [7].

- Dateisystemdaten

Die Kategorie "Dateisystemdaten" enthält allgemeine Dateisysteminformationen wie bspw. die Größe des Dateisystems oder den Aufenthaltsort bestimmter Datenstrukturen. Daher wird sie auch als "Karte des Dateisystems" bezeichnet.

Die Dateisystemdaten befinden sich meist in den ersten Blöcken des Laufwerks und müssen bei der Suche nach der Art des Dateisystems oder dem Ort zusätzlicher wichtiger Datenstrukturen immer analysiert werden. Gehen Daten in dieser Kategorie verloren, ist dies schwierig zu kompensieren. [7]

- Inhaltsdaten

Zu dieser Kategorie gehören alle Dateien, bei denen der Inhalt im Vordergrund steht. Da dies der Hauptgrund für die Existenz von Dateisystemen ist, gehört dieser Kategorie der Großteil der Daten in einem Dateisystem an. Organisiert werden diese Inhaltsdaten normalerweise in Blöcken ("Cluster", "Datenblock") mit einer bestimmten Standardgröße. Diese Cluster sind entweder als "benutzt" oder "unbenutzt" markiert, abhängig davon, ob sie Dateien beinhalten oder nicht. Je nach Betriebssystem wird die Belegungsreihenfolge von Blöcken und der Löschvorgang unterschiedlich durchgeführt. Üblicherweise ist eine Datei nicht komplett in einem Block gespeichert, sondern auf mehrere Blocksequenzen aufgeteilt. Dieser Vorgang wird als Fragmentierung bezeichnet. Beim Löschen von Dateien werden bei einem Standardlöschverfahren lediglich die Verweise auf den Aufenthaltsort der einzelnen Fragmente gelöscht und der Status der betroffenen Cluster auf "unbenutzt" gesetzt. Dadurch lassen sich diese Dateien sehr leicht wiederherstellen, sofern die Datenblöcke nicht durch neue Dateien überschrieben werden. Wurde jedoch mit einem Programm zum "sicheren Löschen" gelöscht, ist dies schwer nachvollziehbar, außer der Forensiker findet das entsprechende Programm auf der Festplatte und kann daraus bspw. die Zeit der letzten Verwendung lesen.

Zur Analyse von Inhaltsdaten gibt es prinzipiell vier verschiedene Möglichkeiten. Während sich spezielle Datenblöcke im Hex-Editor anzeigen lassen, können auch alle Datenblöcke durchsucht werden. Weiterhin lässt sich die Suche auf unbenutzte Datenblöcke oder benutzte Datenblöcken ohne oder mit mehreren Metadateneinträgen eingrenzen. [7]

- Metadaten

Zu den Metadaten gehören Dateien mit Informationen über bestimmte Dateien, wie z.B. deren Größe, Erst- und Änderungsdatum, etc. – nicht aber deren Inhalte. Ebenso wie bei den Inhaltsdaten werden auch die Metadaten in "benutzt" und "unbenutzt" unterteilt, wobei letztere durchaus wertvolle Spuren enthalten können, da die Verweise oft noch vorhanden sind, falls die Metadaten nicht überschrieben wurden. Andernfalls sucht der forensische Ermittler nach Datenblöcken, die so aussehen wie der Dateityp, nach dem er sucht (z.B. Word-Datei), und fügt diese dann zusammen.

Um die entsprechenden Metadaten auszuwerten, muss zuerst ein Dateiname auf diese verweisen, mit anschließendem Zugriff auf die einzelnen Cluster der Datei. Es besteht die Möglichkeit einer logischen Suche anhand von Schlüsselwörtern. Metadaten können sehr hilfreich bei der Spurensuche sein. Bspw. lassen sich mithilfe von Tools alle Dateien auflisten, die in einem bestimmten Zeitraum verändert wurden. [7]

- Dateinamen

In der vierten Kategorie werden die Namen von Dateien gespeichert, meist in Verbindung mit einem Metadateneintrag. Metadaten und Dateinamen stehen in einem ähnlichen Zusammenhang wie der Namen eines Rechners und seine zugehörige IP-Adresse. Während der Dateiname, vergleichbar mit dem Rechnernamen, eine symbolische Beschreibung liefert, stellen Metadaten, vergleichbar mit der IP-Adresse eines Rechners, eine eindeutige Beschreibung dar.

Manchmal kann es dazu kommen, dass Dateien gelöscht werden, weil ihr Dateiname gelöscht wurde. Der trotzdem noch existierende Verweises auf die Metadaten führt zu der entsprechenden Datei, sofern die Metadaten noch nicht an andere Dateien übertragen wurden. Die gewöhnliche Methode zur Auflistung aller Dateinamen ist das Aufrufen aller Verzeichniseinträge. Weiterhin lässt sich nach Schlüsselwörtern suchen. Dabei ist die Verlässlichkeit der Dateieindungen anzuzweifeln, da diese leicht gefälscht werden können. [7]

- Anwendungsdaten

Anwendungsdaten sind zusätzliche Daten, die Funktionen bereitstellen, jedoch für den korrekten Betrieb eines Dateisystems nicht notwendig sind, wie z.B. Quota-Statistiken oder Dateisystem-Journale. Ein Journal ist in diesem Zusammenhang eine Liste aller zuletzt durchgeführten Änderungen und bietet zugleich die wichtigste Analysequelle. Jedoch sind diese Journale für forensische Zwecke kaum noch auswertbar. [7]

Anhand der Einteilung in diese Kategorien besteht die Möglichkeit einzelne Dateisysteme einfach miteinander zu vergleichen, wobei diese nicht immer genau in dieses Schema hineinpassen müssen.

Zur allgemeinen Analyse kann der Forensiker immer in typischen Sektoren ("unbenutzt", "defekt", etc.) nach Verstecken und Spuren gelöschter Dateien suchen. Zudem sind einzelne Dateitypen anhand ihrer anwendungsabhängigen Strukturen zu identifizieren, da jeder Dateityp einen standardmäßigen Header und Footer hat. [7]

4.7.1 File Allocation Table

Das File Allocation Table FAT (zu deutsch etwa "Dateizuordnungstabelle") ist ein von Microsoft entwickeltes und einfaches Dateisystem aus der DOS-Zeit. Heutzutage sind mobile Datenträger, wie USB-Sticks in FAT formatiert, da jedes Betriebssystem damit arbeiten kann. Die Datenstruktur bei diesem Dateisystem entspricht nicht komplett der Aufteilung in die o.g. Kategorien, was bspw. dadurch deutlich wird, dass Verzeichniseinträge nicht zwischen Dateinamen und Metadaten unterscheiden.

Prinzipiell existiert für jedes Verzeichnis und jede Datei ein Verzeichniseintrag mit Metadaten, wie Dateigröße, Dateiname und Verweis auf den ersten Cluster. Die Verweise von dem ersten Dateisystemblock auf die Folgenden sind in der FAT hinterlegt. [7, 15]

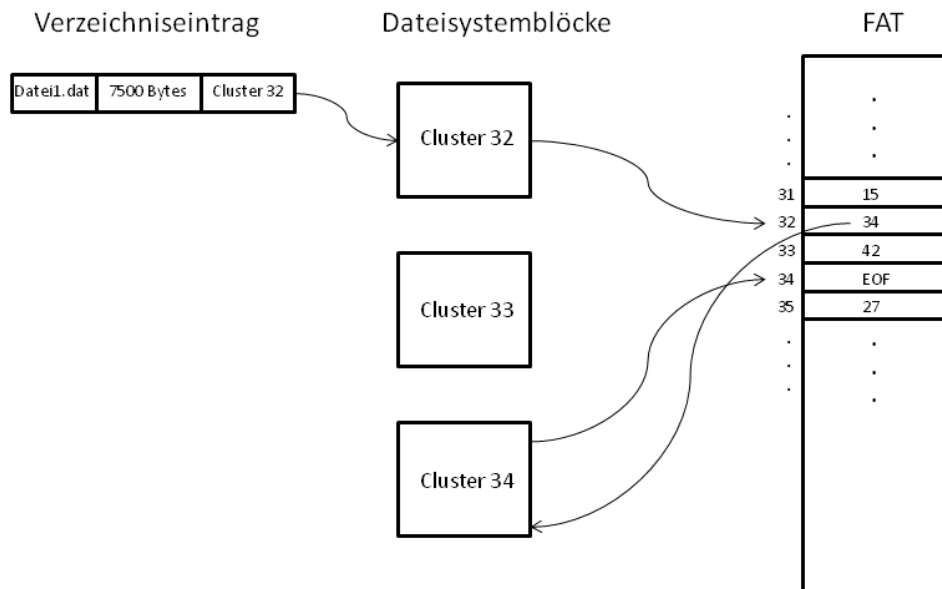


Abbildung 4.5: Übersicht Fragmentierung

Aufbau und Analyse

Ein FAT-Dateisystem gliedert sich in folgende physische Bereiche: Die reservierten Sektoren (inkl. Bootsektor), FAT und der Datenbereich. Das FAT-Dateisystem muss mit einem bestimmten Bytemuster beginnen, um ein bootbares Dateisystem als Laufwerk anzeigen zu lassen. Abhängig von der Größe eines FAT-Eintrags werden u.a. drei verschiedene Versionen unterschieden: FAT12, FAT16 und FAT32. Diese Versionen wurden nacheinander entwickelt und erweitern jeweils den Adressraum, dessen Größe die jeweilige Zahl hinter FAT angibt. [7, 15]

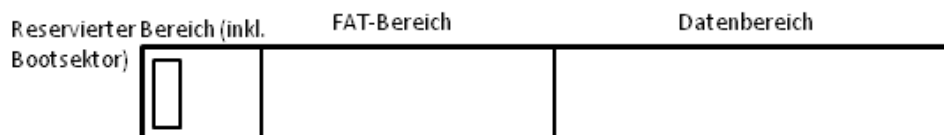


Abbildung 4.6: Layout des FAT

- **Reservierte Sektoren**

Sektoren, die vom Dateisystem nicht genutzt werden, können zwischen dem Bootsektor und der ersten FAT reserviert werden. Dieser Bereich steht dann für die Nutzung von betriebssystemspezifischen Erweiterungen oder eines Bootmanagers zur Verfügung. Da nicht alle reservierten Sektoren genutzt werden müssen, dienen sie als geeignetes Versteck. [7, 15]

- **Bootsektor**
 Der Bootsektor ist der erste Sektor im reservierten Bereich und beinhaltet Dateisystemdaten mit Informationen über das FAT-Dateisystem, u.a. die Anzahl der reservierten Sektoren (in FAT12/16 nur ein Sektor, in FAT32 viel mehr Sektoren), die Anzahl der FATs und ihre jeweilige Größe, die Anzahl der Einträge im Wurzelverzeichnis und bei FAT32 zusätzlich den Ort des Wurzelverzeichnisses. In FAT32 gibt es eine Backup-Kopie des Bootsektors.
 Bei der Analyse des Bootsektors spielt die Bestimmung des genauen Dateisystemtyps (FAT12, FAT16 oder FAT32) und des Ortes, an denen FAT und Cluster beginnen eine wesentliche Rolle. Im Bootsektor lassen sich kaum Daten verstecken. Durch Analyse des Backup-Bootsektors in FAT32 lassen sich Manipulationen feststellen. [7, 15]
- **FAT**
 Der FAT-Bereich kann aus einem oder mehreren FATs bestehen. Die Größe ergibt sich aus dem Produkt von der Anzahl der FATs und der Größe eines FAT. In diesem Bereich wird der Status der einzelnen Cluster festgehalten: 0 steht für unbelegt, 0xFF7 (FAT12), 0xFFF7 (FAT16), 0xFFFF FFF7 (FAT32) für beschädigt und die restlichen Werte bedeuten, dass das Cluster belegt ist.
 Zur Analyse wird der FAT-Bereich nach unbelegten und beschädigten Clustern durchsucht. Es besteht die Möglichkeit, dass hinter dem letzten Cluster noch der ein oder andere unbenutzte Sektor aufzufinden ist, ebenso wie das Ende der FAT noch ein potentiell Versteck bietet. [7, 15]
- **Datenbereich**
 Der Datenbereich ist in mehrere Dateisystemblöcke, sogenannte Sektoren aufgeteilt. In FAT12/16 ist das Wurzelverzeichnis am Anfang des Datenbereichs. In FAT32 ist das Wurzelverzeichnis an dem im Bootsektor angegebenen Ort.
 Es gibt keine Cluster mit den Adressen 0 und 1, was ein entscheidendes Kriterium zur Erkennung des Dateisystems darstellt. Der erste Cluster (mit der Adresse 2) liegt nicht notwendigerweise am Anfang des Dateisystems. In FAT12/16 befindet sich der erste Cluster in dem auf das Wurzelverzeichnis folgenden Sektors. In FAT32 wiederum ist der Cluster 2 im ersten Sektor des Datenbereichs zu finden. [7, 15]

Als zusätzliches eindeutiges Erkennungsmerkmal des Dateisystems, befindet sich in den Bytes 510 und 511 des ersten Sektors eine Signatur des FAT (510: 0x55, 511: 0xAA). Der Forensiker kann mithilfe der Anzahl der Datenblöcke ebenfalls herausfinden, um welche Version es sich handelt. FAT12 besteht aus weniger als 4085 Datenblöcken, FAT16 aus 4085 bis 65525 Datenblöcken und FAT32 aus mehr als 65525 Datenblöcken. [7, 15]

4.7.2 New Technology File System

Das New Technology File System (NTFS) bezeichnet ein proprietäres Dateisystem von Microsoft, das als Nachfolgedateisystem von FAT ab Windows NT nutzbar ist. Dieses Standarddateisystem für heutige Windows-Rechner ist im Vergleich mit FAT nicht nur

sehr zuverlässig und sicher, sondern auch skalierbar und für größere Dateisysteme verwendbar. Bei diesem Dateisystem sind alle Daten in Dateien abgelegt. Dateisystemdaten werden dabei unter Nutzung des Hidden-Attributs vor dem User versteckt. [7, 15]

Aufbau und Analyse

Das NTFS besteht lediglich aus einem Bootsektor mit Bootcode und dem Datenbereich, der wie auch beim FAT-Dateisystem in Cluster aufgeteilt ist und ähnelt im Aufbau dem der FAT. Zudem ist auch wie bei seinem Vorgänger der Beginn der Dateisystemdaten im Bootsektor festgelegt. [7, 15] Zur zentralen Datenstruktur gibt es die im Datenbereich

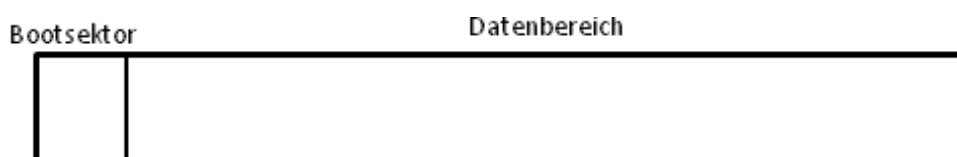


Abbildung 4.7: Layout des NTFS

untergebrachte Master File Table (MFT) mit Informationen über alle Dateien und Verzeichnisse. Dabei hat auch jede Datei und jedes Verzeichnis mindestens einen Eintrag. Ein einzelner MFT-Eintrag hat eine Größe von 1 KB und besteht aus einem Kopf, gefolgt von einer Reihe von Attributen und dem unbenutzten Speicher. Die ersten 42 Byte sind auf 12 Felder aufgeteilt, während die restlichen 982 Byte unstrukturiert sind. Der erste Cluster beginnt mit der Adresse 0, was ein eindeutiges Unterscheidungsmerkmal zum FAT darstellt. [7, 15] Der allererste Eintrag in der MFT ist ein Eintrag mit Informationen über die

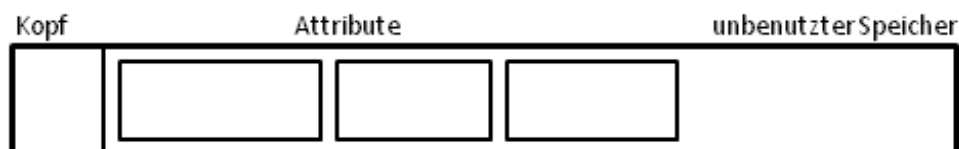


Abbildung 4.8: MFT Eintrag

MFT selbst mit dem Namen "\$MFT". Die folgenden 15 MFT-Einträge sind für weitere Metadaten bestimmt, deren Titel immer mit "\$" beginnt, gefolgt von mindestens einem Großbuchstaben (z.B. Eintrag 3: \$Volume, \$Boot \$BadClus).

MFT-Attribute bestehen auch aus einem Kopf und weiterem Inhalt. Jedem Attribut ist eine charakteristische Nummer und ein Name (ebenfalls beginnend mit "\$") zugeordnet. Es werden zwei Arten von Attributen unterschieden: Die residenten Attribute, die den Inhalt im MFT-Eintrag abspeichern und die nicht-residenten Attribute, deren Inhalt in einem Cluster ist, auf den verwiesen wird. Im Attributkopf befindet sich u.a. die Attributnummer, die Größe, der Namen und ein Biteintrag bzgl. des Status des Inhalts (komprimiert, verschlüsselt, resident). Der Attributinhalt kann jede Größe und Form haben.

Zur Analyse liest der Forensiker die gewünschten Metadaten aus. Auf der Position 0 befindet sich der Eintrag "\$MFT", welcher zum Finden aller Dateien dient und somit der Wichtigste ist. Weitere nützliche Metadaten befinden sich bspw. im Eintrag 3 (\$Volume)

mit Informationen über den Laufwerksnamen und die NTFS-Versionsnummer, im Eintrag 7 (\$Boot) mit Daten des Bootsektors und im Eintrag 8 (\$BadClus). In letzterem sind die beschädigten Cluster abgespeichert, die beim Lesen direkt übersprungen werden und somit auch ein potentiell Versteck darstellen. Weitere Versteckmöglichkeiten bietet der Bootsektor, unbenutzter Speicher in der MFT, unbenutzte Sektoren oder auch die Zuordnung ungewöhnlicher Attribute zu einem MFT-Eintrag.

Wird ein Eintrag gelöscht, verändert sich lediglich der Kopf - der Inhalt bleibt. Zudem lässt sich der Cluster einer gelöschten Datei besser als bei FAT finden. [7, 15]

4.7.3 Extended File System

Das Extended File System (EXT) ist ein offenes Journaling-Dateisystem, welches speziell für Linux entwickelt wurde und dort auch häufig als Standarddateisystem verwendet wird. EXT2 und die Nachfolger EXT3 und EXT4 sind die aktuell genutzten Dateisysteme. Die Priorität liegt im Gegensatz zu Microsoft-Dateisystemen in der Zuverlässigkeit und in der Geschwindigkeit. Die einzelnen Fragmente einer Datei werden bewusst auf nah beieinander liegenden Clustern gespeichert, um die Wege des Lesekopfes kurz zu halten.

Aufbau und Analyse

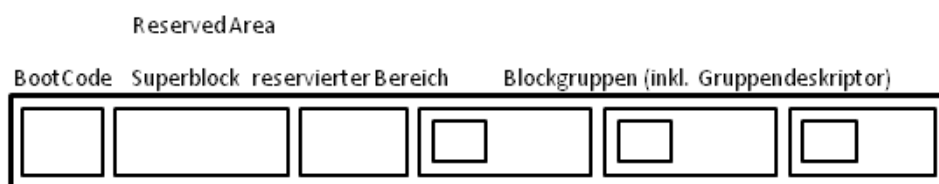


Abbildung 4.9: Layout des EXT

Das EXT besteht grundlegend aus einem Superblock und einer Folge von Blockgruppen. Der Begriff "Block" bezeichnet eine Folge zusammenhängender Sektoren auf der Platte. Der Superblock enthält ausschließlich Konfigurationsdaten, wie die Anzahl der Sektoren eines Clusters, die Gesamtanzahl an Blöcken im Dateisystem und pro Blockgruppe, sowie die Anzahl der reservierten Blöcke vor der allerersten Blockgruppe. Er beginnt ab dem Byte 1024 und ist 1024 Byte groß. Davon sind viele unbenutzt. Eine Blockgruppe hat mehrere Cluster, wobei jede Blockgruppe bis auf die letzte aus gleich vielen Datenblöcken besteht. Ein Block ist je nach Angabe im Superblock 1024, 2048 oder 4096 Byte groß. Die erste Blockadresse beginnt wie bei NTFS bei 0. Bis auf die Blöcke in der reserved area, gehören alle weiteren Blöcke zu einer Blockgruppe. In jedem Gruppensdeskriptor befindet sich ein Backup des Superblocks. Weiterhin enthält er u.a. die Menge von Indexknoten und Datenblöcken, sowie die Gruppensdeskriptortabelle mit Verweisen auf alle anderen Blockgruppen im Dateisystem. Noch vor dem Superblock befindet sich bei einem bootbaren Dateisystem der Bootcode. [7, 15]

Metadaten werden im Indexknoten gespeichert. Verzeichniseinträge enthalten Dateinamen mit einem Verweis auf einen Indexknoten, der wiederum auf nah beieinander liegende Datenblöcke verweist. [7, 15]

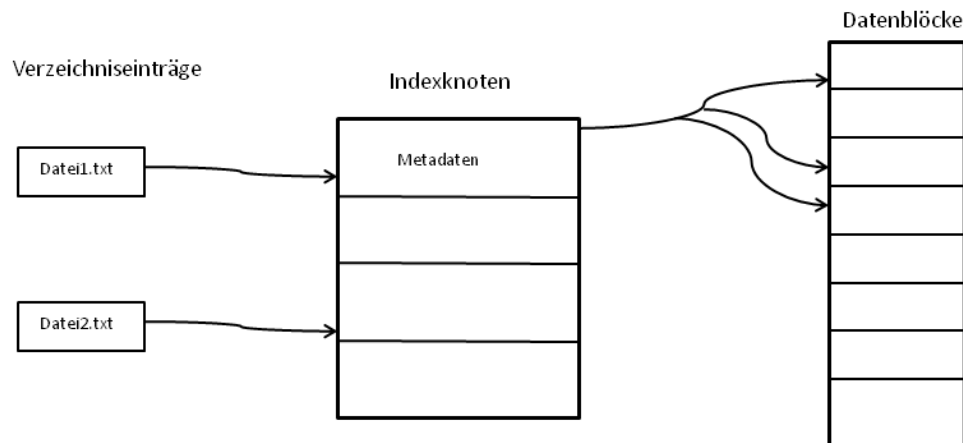


Abbildung 4.10: Übersicht Fragmentierung

Das EXT bietet eine Vielzahl an Versteckmöglichkeiten: Zum Einen steht genügend freier Speicherplatz im Superblock und seinen Backups zur Verfügung und zum Anderen ist der reservierte Bereich für den Bootcode meist unbenutzt. Die Tatsache, dass es sich um ein offenes Dateisystem handelt und somit zusätzliche Erweiterungen eingearbeitet werden können, erschwert dem Forensiker zumeist die Analyse.

4.8 Auswahl nützlicher Tools

Im Folgenden werden beispielhaft einige Tools aufgelistet, die zu Analysezwecken genutzt werden können.

- Entschlüsselung geschützter Dateitypen
 - Advanced Password Recovery Software Tool Kit (New Technologies Inc.), kommerziell
 - Decryption Collection (Paraben), kommerziell
 - Distributed Network Attack (AccessData), kommerziell
 - Distributed Password Recovery, (Elcomsoft), kommerziell
 - Password Recovery Tools, (Elcomsoft), kommerziell
 - Password Recovery Toolkit, (AccessData), kommerziell
- Datenträgeranalyse
 - Computer Incident Response Suite (New Technologies Inc.), kommerziell
 - DiskInvestigator (Kevin Solway), kostenlos: unter Windows inkl. Recovery-Funktionen
 - Encase Forensic Edition (Guidance Software), kommerziell
 - Forensic Toolkit (= FTK) (AccessData), kommerziell

- P2 Software Pack (Paraben), kommerziell
- X-Ways Forensics (X-Ways Software Technology AG), kommerziell
- Datenträgersicherung
 - Forensic Acquisition Utilities, Open Source
 - Forensic Replicator (Paraben), kommerziell: bitgenaues Sichern
 - SafeBack (New Technologies Inc.), kommerziell: bitgenaues Sichern
 - X-Ways Capture (X-Ways Software Technology AG), kommerziell: auch Online-Sicherung
- Dateisystemanalyse
 - Fatback, Open Source: Datei-Recovery für FAT-Dateisysteme
 - Filemon (Sysinternals), kostenlos: Realtime-Analyse der Zugriffe auf das Dateisystem
 - Forensic Toolkit (Foundstone), kostenlos
 - RemoteRecover (Sysinternals), kostenlos: Recovery-Tool für Windows-Dateisysteme (mit Remote-Funktionalität)
 - Sleuthkit, Open Source
 - The Coroners Toolkit, Open Source: Analyse von Unix-Dateisystemen (Images)
- Browser-Analyse
 - Galleta (Foundstone), kostenlos: Analyse von Internet Explorer Cookie-Dateien
 - Pasco (Foundstone), kostenlos: Analyse der Internet-Explorer History
 - X-Ways Trace (X-Ways Software Technology AG), kommerziell: Auswertung von Internet-Browser-Benutzung unter Windows)
- Datei-Recovery
 - Foremost, Open Source
 - Rifiuti (Foundstone), kostenlos: Analyse des Windows-Papierkorbes
 - X-Ways Davory (X-Ways Software Technology AG), kommerziell
- Sonstiges
 - Incident Response Collection Report, Open Source: Skript zur Sammlung von flüchtigen Daten unter Windows
 - PDA Seizure (Paraben), kommerziell: Sichern und Auswerten von Handhelds
 - PDD, Open Source: Sicherung von Palm-Geräten

4.9 Zusammenfassung

Die Wissenschaft der IT-Forensik ist äußerst komplex. Daher kann diese Arbeit auch nur einen kleinen Einblick in diese geben. Zur erfolgreichen Durchführung forensischer Untersuchungen, bedarf es vielen Basiswissens, um überhaupt die konkreten Abläufe einer solchen Analyse in der Praxis nachvollziehen und anwenden zu können. Auch die Schnelllebigkeit bei der Weiterentwicklung im Hard- und Softwarebereich, setzt voraus, dass ein Forensiker sein Wissen regelmäßig auffrischt und auch fähig ist, sich schnell in neue Systeme einzuarbeiten. Auch dies verdeutlicht die notwendige Voraussetzung eines breit gefächerten und intensiven Grundwissens, von dem ein Anteil in dieser Arbeit vermittelt wird.

Durch den immer bedeutsamer werdenden hohen Stellenwert der Computersysteme in der heutigen Gesellschaft wird auch die IT-Forensik weiterhin eine immer wichtigere Rolle spielen. Computersysteme werden in immer mehr Bereichen eingesetzt und genießen in diesen auch ein zunehmendes Vertrauen v.a. in Bezug auf Zuverlässigkeit. Dies macht sie weiterhin äußerst attraktiv als Zielobjekt krimineller Handlungen jeglicher Art. Häufig werden diese von Tätern verübt, die Fachwissen aufweisen und mit hochwertigen Tools arbeiten, sodass diese lediglich mit mehr Fachwissen und leistungsstärkeren Tools zu überführen sind. Dies unterstreicht die Notwendigkeit und den Wert hervorragender fachlicher Kompetenzen im Bereich der IT-Forensik.

Literaturverzeichnis

- [1] CLAUDIA ECKERT. *IT-Sicherheit: Konzepte - Verfahren - Protokolle*, Oldenbourg Verlag, 02.11.2011, 7. Auflage
- [2] *Modulhandbuch-MINF2011*, 29.12.2011,
http://www.unibw.de/inf/studium/inf/ma/modhbminf2011/at_download/down1
- [3] VANESSA WIEDT. *Maßregelvollzug*, GRIN Verlag, 07.11.2006, 1. Auflage, S.2 (1. Was ist Maßregelvollzug?)
- [4] KLAUS-PETER KOSSAKOWSKI. *Information Technology Incident Response Capabilities*, BoB-Books on Demand, 2001, S.13 (1.1 Wichtige Begriffsdefinitionen)
- [5] LORENZ KUHLEE, VIKTOR VÖLZOW. *Computer-Forensik Hacks*, O'Reilly Germany, 2012,
- [6] EVI NEMETH, GARTH SNYDER, TRENT R. HEIN. *Linux-Administrations-Handbuch*, Pearson Deutschland GmbH, 2009, S.202f., S. 208f.
- [7] BRIAN CARRIER. *File System Forensic Analysis*, Addison-Wesley, 17.03.2005
- [8] EOGHAN CASEY. *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet*, Academic Press, 20.04.2011, 4. Auflage
- [9] ANDREAS DEWALD, FELIX C. FREILING. *Forensische Informatik*, BoD – Books on Demand, 10.2011, 1. Auflage
- [10] *Strafprozessordnung StPO*, Fassung vom 07.04.1987, letzte Änderung zum 01.04.2012, §94ff
- [11] MATTHIAS MÜLLER. *SSDs und Flash Memory*, Universität Ulm, 16.06.2010, S.6-10
https://www-vs.informatik.uni-ulm.de/teach/ss10/rb/docs/flash_folien.pdf
- [12] A. STREY. *Festplatten und Dateisysteme*, Universität Ulm, WS 2003/2004, D4-D6, D10 <http://www.informatik.uni-ulm.de/ni/Lehre/WS03/TechInf2/2003w-TI2-D1-4.pdf>
- [13] NARAIN ARORA. *Mosfet Modeling for VLSI Simulation: Theory And Practice*, World Scientific, 14.02.2007

- [14] CHRIS PROSISE, KEVIN MANDIA. *Incident Response and Computer Forensics*, McGraw-Hill Prof Med/Tech, 2003, 2. Auflage
- [15] DAN FARMER, WIETSE VENEMA. *Forensic Discovery*, Addison-Wesley, 2005
- [16] ALEXANDER GESCHONNECK. *Computer-Forensik: Computerstraftaten erkennen, ermitteln, aufklären*, Dpunkt.Verlag GmbH, 2011, 5. Auflage

Kapitel 5

Sicherheit Sozialer Netzwerke

Julian Petery

In dieser Seminararbeit geht es um aktuelle Themen in Bezug auf Social Engineering. Zunächst werden Informationen vorgestellt, die zur Durchführung eines Angriffs benötigt werden. Darauf folgend werden die verschiedenen Angriffsformen vorgestellt beginnend mit Computer basierten Verfahren, wie z.B. Phishing. Anschließend wird auf von Menschen ausgehende Angriffe eingegangen und als drittes wird die Königsdisziplin vorgestellt, die Menschen dazu bringt die gewünschten Informationen selbstständig zu liefern. Zu Ende werden Ansätze zur Verhinderung des Erfolgs solcher Angriffe vorgestellt.

Inhaltsverzeichnis

5.1	Einführung	107
5.2	Benutzte Informationen	107
5.2.1	Informationen über das berufliche Umfeld	107
5.2.2	Informationen über das private Umfeld	107
5.3	Computer Based Social Engineering	108
5.3.1	Phishing	108
5.3.2	Präparierte Internetseiten	109
5.4	Human Based Engineering	110
5.4.1	Der "Hilfsbedürftige"	110
5.4.2	Der "Moralische"	110
5.4.3	Erzeugen von Schuldgefühlen	110
5.4.4	Der "Insider"	111
5.4.5	Der Fachchinese	111
5.4.6	Der Vorgesetzte	111
5.5	Reverse Social Engineering	111
5.5.1	Angriff auf Vertrauensbasis	112
5.6	Schutzmaßnahmen	112
5.6.1	Technische Maßnahmen	112
5.6.2	Awareness schaffen	112
5.6.3	Schriftliche Vereinbarung	113
5.6.4	Betriebsklima	113
5.7	Zusammenfassung	113

5.1 Einführung

"Hallo, Sie haben 1.000.000 EUR gewonnen. Um den Gewinn einzulösen überweisen Sie bitte 1 000 EUR an folgende Adresse." oder "Prüfen Sie hier ob Ihre Kreditkarte gestohlen wurde. Geben Sie nur die Nummer und das Ablaufdatum ein". Wer kennt sie nicht, die manchmal lächerlich dilettantischen Versuche unbescholtenen Kunden ihre Daten aus der Tasche zu ziehen? Kaum einer würde jemals auf solch einen Trick hereinfallen. Was aber wäre, wenn man aus der Personalabteilung die Bitte erhält diese Daten wegen einer neuen Art der Nebenkostenabrechnung zu versenden? Hier dürfte die Quote schon deutlich höher liegen.

Die Idee Menschen zu manipulieren und Aufträge durchführen zu lassen, die man selbst nicht machen kann, ist so alt wie die Menschheit selbst. In früheren Zeiten hießen solche Leute Hochstapler oder Trickbetrüger oder aber, wenn sie im staatlichen Auftrag handelten, Spione. Viele der unten vorgestellten Methoden werden auch heute noch immer von staatlichen Stellen im nachrichtendienstlichen Alltag eingesetzt. [1]

Aufgrund der immer weiter zunehmenden Verbreitung Sozialer Netzwerke und Vernetzung der Menschheit werden Angriffe aus dem Internet und insbesondere weiter zunehmen. Diese Seminararbeit will den Leser sensibilisieren und aufklären.

5.2 Benutzte Informationen

Grundsätzlich lässt sich sagen, dass sich jede Information über eine Person oder über ihr Umfeld gegen sie verwenden lässt. Je mehr ein Angreifer über sein Opfer weiß, desto leichter wird es ihm fallen den Angriff erfolgreich durchzuführen. Diese Informationen kann man in zwei Gruppen unterteilen, zum einen in Informationen über das berufliche Umfeld und zum anderen in Informationen über das private Umfeld. In beiden Fällen ist dies öffentlich zugängliches Material, aber auch vertrauliche Daten finden Verwendung.

5.2.1 Informationen über das berufliche Umfeld

Hierunter fallen sowohl Informationen, die für einen Angriff genutzt werden als auch meistens die Daten die das Ziel des Angriffs sind. Dass insbesondere im zweiten Fall dies vertrauliche Daten sind, dürfte ersichtlich sein, aber auch belanglose Informationen, wie z.B. der Speiseplan der Kantine, können einen Angriffspunkt darstellen. [2] S.27

5.2.2 Informationen über das private Umfeld

Hier schlägt die Stunde der Sozialen Netzwerke. Auch wenn nach einer Studie des Pew Research Institutes die Zahl der Nutzer, die restriktive Privatsphäreneinstellungen verwenden, am steigen ist [3], sind noch immer Informationen, die für einen Social Engineer

interessant sind, wie z.B. bei Facebook der Benutzername, aus dem die E-Mailadresse gebaut wird, öffentlich einsehbar. Und selbst wenn das Opfer sich optimal schützen sollte, das Netz vergisst nichts [6].

Insbesondere wird dies problematisch, sobald Angreifer diese Informationen zusammentragen und sich dadurch ein deutlich genaueres Bild des Opfers machen können [6]. Diese Aggregation muss heutzutage nicht mehr von Hand durchgeführt werden, sondern kann automatisiert durchgeführt werden. In Deutschland wird ein solcher Dienst beispielsweise von yasni.de angeboten.

5.3 Computer Based Social Engineering

Hierunter fallen Angriffe, die nicht auf ein spezielles Opfer ausgerichtet sind, sondern auf die breite Masse abzielen. Darunter fallen die klassischen Phishing-Mails, die Opfer dazu bringen sollen ihre Daten preiszugeben, oder auch besondere Arten von Würmern, die gerade in Sozialen Netzwerken sich verbreiten und ebenfalls Daten ausspionieren [6].

5.3.1 Phishing

Es handelt sich um ein Kunstwort, welches sich aus *Password* und *Fishing* zusammensetzt und das *Abfischen* von Zugangsdaten beschreibt. [4]

Diese Angriffsform ist auch für ungeübte Angreifer leicht umzusetzen, da sich die Absenderinformationen leicht fälschen lassen und somit dem Empfänger leicht ein anderer Absender vorspielen lassen. Der untenstehende PHP-Code veranschaulicht, wie dies funktioniert:

```

1 <?php
2 $nachricht = "<b>Herzlich Willkommen...<b>";
3 $an = "empfaenger@test.de";
4 $betreff = "Betrefftext";
5 $extra = "From: mail@sender.de (Mr. Sender)\r\n";
6 $extra .= "Content-Type: text/html\r\nContent-Transfer-Encoding: 8bit\r\n";
7 $extra .= "X-Mailer: PHP ". phpversion();
8
9 mail($an, $betreff, $nachricht, $extra);
10 ?>

```

Hierbei ist die Zeile 5 die entscheidende, da man hier jede beliebige Adresse eintragen kann. Hierbei ist gerade bei Facebook aktuell ein neues Risiko entstanden, da Facebook für jeden Benutzer eine E-Mailadresse eingerichtet hat, die sich aus dem öffentlich einsehbaren Benutzernamen und '@facebook.com' zusammensetzt [8]. Dadurch kann dem Empfänger vorgespielt werden, dass die Nachricht von einem Bekannten stammt. Gerade durch die im RFC2076 spezifizierte Funktionalität des Reply-To im E-Mailheader kann dadurch leicht eine ganze Konversation mit einem vermeintlichen Bekannten durchgeführt

werden. Ein Mittel gegen diese Angriffsart kann die Signatur von E-Mails darstellen. Dabei erzeugt der Absender die Mail mit seinem privatem Schlüssel einen Hashwert und fügt diesen in die Nachricht ein, der Empfänger wiederum kann mit Hilfe des öffentlichen Schlüssel des Absenders und dem Hashwert die Integrität der Nachricht überprüfen. Ein gängiges Verfahren dabei ist PGP. Allerdings sorgt diese Signatur in dieser Form nur für den Nachweis, dass die Nachricht nicht während der Übertragung verändert wurde. Für die Verifizierung des Absenders ist es nötig, dass sich dieser einer Signierstelle gegenüber ausgewiesen haben. Dies hat bis jetzt einen Großteil aller Nutzer abgeschreckt, allerdings ist es möglich im neuen Elektronischen Personalausweis eine solche Signatur zu hinterlegen. Es bleibt zu hoffen, dass mit der zunehmenden Verbreitung auch die Zahl der signierten Mails zunimmt.

5.3.2 Präparierte Internetseiten

Das Hauptziel dieser Phishing-Mails ist es Opfer auf eine Webseite zu leiten um dort persönliche Daten abzugreifen oder Schadsoftware zu installieren. Hier werden häufig existierende Internetseiten bis ins kleinste Detail nachgebaut, so dass es für den unbedarften Nutzer schwierig ist, die Fälschung zu erkennen, insbesondere da durch URL-Spoofing auch eine falsche Adresse vorgetäuscht werden kann. Bei der Installation von Schadsoftware gibt es zwei Vorgehensweisen: Den Drive-By-Download und das überreden des Nutzers diesen anzustoßen. Bei erster Variante werden oftmals Sicherheitslücken insbesondere in PDF-Readern und Flash-Playern ausgenutzt [9]. Bei der anderen werden Techniken des Social Engineerings eingesetzt um das Opfer zu einem Download einer Software zu überreden. Dies wurde und wird gerade auf Sozialen Plattformen häufig eingesetzt, als ein Beispiel soll die Facebook-Anwendung "We catch Stalkers" dienen [7]. Hierbei wurde dem Nutzer vorgegaukelt, dass er herausfinden könnte, welche Nutzer wie oft sein Profil besucht haben. Allerdings generierte sie nur eine zufällige Liste seiner Freunde, postete aber an die Pinnwände seiner Freunde eine Werbung für diese Anwendung. Es laut diesem Artikel noch nicht bekannt, welchen Zweck die Anwendung hat, allerdings lässt die rasche Verbreitung die berechtigte Sorge zu, dass in naher Zukunft Kriminelle diesen Weg vermehrt nutzen werden. Gerade da es vielen Personen nicht bekannt ist, dass Inhalte von fremden Servern auf Facebook geladen werden können und dies auch an Stellen geschehen kann, an denen man es nicht erwartet, ist auch der Drive-By-Download auf Sozialen Netzwerken eine nicht zu unterschätzende Gefahr. Es wird nur bei Anwendungen vor externen Inhalten gewarnt, allerdings können auch normale Seiten Inhalte von externen Servern nachladen. [10]

Das Problem der Seiten, die sich als eine andere ausgeben, könnte durch die zunehmende Verbreitung von https-Verbindungen verringert werden, da diese von einer weiteren Stelle verifiziert werden, ähnlich der Signatur bei E-Mails. Auch die Browser-Hersteller unterstützen dies mit farblichen Hinweisen auf die Sicherheitsstufe einer Webseite. Allerdings bedarf es hier einer weiteren Sensibilisierung der Nutzer, da viele nicht auf diese Möglichkeiten zurückgreifen.

5.4 Human Based Engineering

Dieser Abschnitt beschäftigt sich mit zielgerichteten Angriffsformen. Da diese sehr aufwendig sind werden diese nur bei Personen durchgeführt, die für den Angreifer eine besondere Bedeutung aufweisen und eine hohe Erfolgsquote versprechen. Da viele Benutzer von Sozialen Netzwerken viele Informationen leichtfertig preisgeben, wird den Angreifern eine Möglichkeit gegeben, sich auf das Opfer vorzubereiten. Dieser Abschnitt basiert auf einer Fibel, die SAP im Jahre 2007 veröffentlichte [4]

5.4.1 Der "Hilfsbedürftige"

Hierbei wird sich der Angreifer als ein Kamerad auf einem neuen Dienstposten vorstellen, der noch nicht alles weiß und dem man als Kamerad gerne hilft. Am einfachsten gibt sich der Angreifer zusätzlich noch als Angehöriger einer anderen Teilstreitkraft aus, denn so ist es unauffällig, wenn der Social Engineer eine truppengattungsspezifische Information nicht hat oder einen bestimmten Ausdruck nicht kennt. Zusätzlicher Druck wird ausgeübt, wenn man dem Opfer klarmacht, dass von der verlangten Hilfe oder Information die Erfüllung eines Befehls abhängt Zum Beispiel: "Oh nein, ich kann doch nicht im ersten Monat hier schon zu meinem S6 gehen und ihm berichten, dass ich meine Zugangsdaten verlegt habe. Können wir das mit Rücksicht auf meine Probezeit nicht irgendwie unbürokratisch lösen? Ich wäre Ihnen ewig dankbar!"

5.4.2 Der "Moralische"

Bei dieser Methode wird einem Mitarbeiter des Zielunternehmens vermittelt, dass doch alle ein Team sind und an einem Strang ziehen sollten. In aller Regel läuft dies so ab, dass der Angreifer von jemandem eine Information oder bestimmte Handlung verlangt, dieser sich aber aufgrund von Arbeitsanweisungen weigert, die Information zu liefern oder die Tätigkeit auszuführen. Der Angreifer wird dann subtil versuchen, den Ansprechpartner als nicht teamfähig und egoistisch darzustellen. Oft lenkt derjenige dann ein, um den Erfolg eines Projektes nicht zu vereiteln. Zum Beispiel: "Alles klar, ich verstehe, dass Sie mir diese Information nicht geben können. Dann muss ich Herrn Dr. Wichtig eben ausrichten, dass wir den Termin nicht halten können. Schade um das Teamziel für dieses Quartal."

5.4.3 Erzeugen von Schuldgefühlen

Bei diesem Vorgehen wird dem Opfer deutlich gemacht, was passieren kann, wenn eine bestimmte Information nicht bereitgestellt oder eine Handlung nicht ausgeführt wird. Häufig wechselt der Angreifer stark in den persönlichen Bereich und beschreibt negative Folgen aus dem privaten Umfeld.

Zum Beispiel: "Ich kann es wirklich nicht nachvollziehen, dass Sie es verantworten können, den Job eines Familienvaters wegen einer Arbeitsanweisung der 'Orga' zu gefährden."

5.4.4 Der "Insider"

Der Insider versucht durch seine Wortwahl, internes Wissen und Informationen, die nur Angehörigen des Unternehmens bekannt sein können, jeden Zweifel an der Rechtmäßigkeit seines Anliegens zu beseitigen. Häufig handelt es sich um Informationen zu aktuellen Projekten, stattgefundenen Meetings oder einfach nur die Kenntnis des aktuellen Speiseplans der Kantine woher soll ein Externer auch wissen, dass der Fisch mit Kartoffelsalat am Freitag nicht geschmeckt hat.

5.4.5 Der Fachchinese

"Wir haben im Moment große Probleme im Netz. Irgendein Bot-Ausbruch sorgt für so viel Traffic, dass wir unseren Quality of Service nicht mehr in den Griff bekommen, was zu extrem schlechten Response-Zeiten führt. In manchen Abteilungen kann kaum noch gearbeitet werden. Wir vermuten, dass das von Ihrem System kommt."

Dieser Satz ist ein typisches Beispiel für die Technik des Fachchinesen. Der unbedarfte Anwender wird mit einer technischen Aussage so verwirrt, dass er nicht versteht, worum es geht und keine Chance hat, den Wahrheitsgehalt der Aussage zu verifizieren. Das Ergebnis ist oft, dass der Anwender aufgibt und bedingungslos das tut, was der vermeintliche Techniker am Telefon verlangt. Trauen Sie sich nachzufragen oder teilen Sie einem Anrufer mit, dass Sie seine Aussagen nicht verstehen. Sagen Sie im Zweifelsfall, dass Sie, bevor Sie Eingriffe an Ihrem System vornehmen, mit dem Vorgesetzten sprechen oder erst genau wissen möchten, worum es geht. Ein Social Engineer wird in aller Regel den Angriff abbrechen und nach einem leichtgläubigeren Opfer suchen. Informieren Sie die IT über solche Anrufe. Eine seriöse IT-Abteilung ist immer in der Lage, ihr Anliegen zu begründen und verständlich zu beschreiben!

5.4.6 Der Vorgesetzte

Diese Methode wird meist in Unternehmen mit stark hierarchischer Struktur verwendet. Das Opfer wird mit angeblichen Aussagen von Mitarbeitern höherer Hierarchieebenen überzeugt. Angestellte haben dann nicht immer den Mut, den jeweiligen Vorgesetzten anzusprechen, um die Aussage bestätigen zu lassen. Dies setzt voraus, dass der Angreifer über ein Organigramm verfügt und weiß, wessen Anweisungen im Unternehmen nicht hinterfragt werden.

5.5 Reverse Social Engineering

Bei dieser Vorgehensweise versucht der Angreifer zu seinem Opfer eine Vertrauensbasis aufzubauen, um dieses dazu zu bewegen die benötigten Informationen freiwillig und aus eigener Initiative herauszugeben.

5.5.1 Angriff auf Vertrauensbasis

Gerade in sozialen Netzwerken ist besonders problematisch, dass hier die viele Menschen leichtfertig von Problemen berichten und auch Hilfsangebote von vermeintlichen "Freunden" leichter annehmen als dies bei anderen Kommunikationsformen der Fall sein mag. Muss ein Angreifer im Normalfall selbst einen Fehler beim Angegriffenen verursachen, sondern bekommt diese Notsituationen quasi "frei Haus" geliefert. Sobald das Opfer einen ähnlichen Fehler erleidet, wird es sich an eine unbürokratische, schnelle und kompetente Beratung in der Angelegenheit erinnern. Allerdings ist diese Angriffsform sehr langwierig und aufwendig und ist somit nicht für die Gewinnung einer bestimmte Information einsetzbar.

5.6 Schutzmaßnahmen

Vorneweg muss gesagt werden, dass es keinerlei 100%-igen Schutz gegen Social Engineering gibt. Man kann den Unsicherheitsfaktor Mensch nicht durch das Einspielen von Software und Zwischenschalten von Hardware ausschalten, da es ja gerade Prinzip des Social Engineerings ist einen Innentäter zu erzeugen.

5.6.1 Technische Maßnahmen

Diese Maßnahmen können die unten aufgeführten Schritte nicht ersetzen, sondern nur ergänzen. Hier ist vorallem die Verwendung von Signatur und Verschlüsselung der elektronischen Kommunikation, wie oben erläutert, zu erwähnen. In Verbindung mit einer entsprechenden Konfiguration der Mailserver kann somit schon ein Einfallstor geschlossen werden.

5.6.2 Awareness schaffen

Eine erste Reaktion wäre sicherlich, nun sämtliche Kommunikation mit der Außenwelt zu verbieten. Allerdings wird sich kaum jemand daran halten, weshalb eine Schulung der Mitarbeiter wesentlich effektiver ist. Stefan Schuhmacher beschreibt in seinem Artikel in [5] einen Versuch mit zwei Gruppen von Kindern, bei dem einer Gruppe ein Spielzeug verboten, die anderen aber mit einer Begründung davon ferngehalten wurden. Das Ergebnis war, dass die Gruppe mit der begründeten Enthaltung auch nach dem Ablauf von ein paar Wochen dieses Spielzeug für uninteressant hielten, während die anderen ein gewisses Verlangen entwickelt hatten. Daraus kann man unser Thema folgern, dass es notwendig ist, die Mitarbeiter über die Gefahren im Internet aufzuklären, als nur bestimmte Teile zu verbieten.

5.6.3 Schriftliche Vereinbarung

Ein weiterer Aspekt, der im oben genannten Artikel beschrieben wird, ist, dass eine Vereinbarung deutlich effektiver ist, sobald sie schriftlich niedergelegt wird. Die Wahrscheinlichkeit, dass diese eingehalten wird, kann noch weiter gesteigert werden, sobald sie öffentlich gemacht wird [5]. Dies liegt insbesondere daran, dass sich dann bei einem Nichteinhalten das Bild des Delinquenten in der Öffentlichkeit ändern würde. Deshalb sollten Richtlinien zum Datenschutz öffentlich ausgehangen werden, so dass die Mitarbeiter immer wieder daran erinnert werden, zu was sie sich verpflichtet haben.

5.6.4 Betriebsklima

Auch wenn es eine Selbstverständlichkeit in der Führung sein sollte, kann man nicht oft genug betonen, dass ein Austausch von Informationen über alle Ebenen hinweg nötig ist. Bei Social Engineering ist dies besonders wichtig, denn sobald ein Mitarbeiter weiß, dass er seinen Kollegen oder Vorgesetzten leicht erreichen kann, hat er die Möglichkeit, sich bei dieses rückzuversichern und schon durch die Ankündigung den Angreifer abzuschrecken [11].

5.7 Zusammenfassung

Wie gezeigt wurde, kann jedes Mittel und jede Information als ein Mittel zu einem Angriff gegen eine Person oder Unternehmen genutzt werden. Insbesondere Soziale Netzwerke stellen hierbei ein Sicherheitsrisiko dar, da hier der Nutzer eine vertrauliche Umgebung erwartet und deshalb die Hemmschwelle Informationen preiszugeben gesenkt ist.

Auf technischer Ebene gibt es vielfältige Möglichkeiten an Daten des Opfers zu gelangen, zum einem durch Vorspiegelung falscher Tatsachen und die darauf folgende freiwillige Herausgabe der Daten und zum anderen die Abfischung durch kompromitierte Webseiten oder Software.

Von Menschen ausgehende Angriffe basieren darauf das Opfer unter Druck zu setzen und dadurch zur Herausgabe von Daten zu zwingen. Eine weitere Möglichkeit ist es eine Vertrauen sbasis aufzubauen und darauf zu warten, dass die Informationen freiwillig und aus eigenem Antrieb geliefert werden.

Es gibt keine Sicherheit gegen Social Engineering, allenfalls Möglichkeiten das Risiko zu senken:

- Signatur und Verschlüsselung der elektronischen Kommunikation
- Mitarbeiter aufklären

- öffentliche und schriftliche Vereinbarung zum Umgang mit vertraulichen Daten
- Möglichkeiten zur Rückversicherung bei Kollegen/Vorgesetzten schaffen

Literaturverzeichnis

- [1] DIE VERFASSUNGSSCHUTZBEHÖRDEN DES BUNDES UND DER LÄNDER. *Wirtschaftsspionage - Risiko für ihr Unternehmen*, Bundesamt für Verfassungsschutz für die Verfassungsschutzbehörden in Bund und Ländern, Düsseldorf 2008.
- [2] LIPSKI, MARCUS. *Social Engineering- Der Mensch Als Sicherheitsrisiko in Der It*, Diplomica Verlag GmbH, Hamburg 2009.
- [3] MADDEN, MARY. *Privacy management on social media sites*, Pew Research Center's Internet & American Life Project, Washington, D.C. 2012.
- [4] UWE BAUMANN, KLAUS SCHIMMER, ANDREAS FENDEL. *SAP Pocketseminar. "Faktor Mensch - Die Kunst des Hackens oder warum Firewalls nichts nützen"*, SAP, s.l. 2007.
- [5] STEFAN SCHUMACHER. *Admins Albtraum-Die psychologischen Grundlagen des Social Engineering, Teil I* in IT-Grundschutz 2009/7, SecuMedia Verlags-GmbH, Ingelheim 2009
- [6] WORTMANN, TILL. Focus Online. [Online] 8. August 2007. [Zitat vom: 13. Juli 2012.] http://www.focus.de/finanzen/karriere/bewerbung/internet/tid-7051/bewerbung_aid_69071.html.
- [7] SEBAYANG, ANDREAS. Golem.de. [Online] [Zitat vom: 1. Juli 2012.] <http://www.golem.de/1102/81218.html>.
- [8] FACEBOOK INC.. [Online] [Zitat vom: 1. Juli 2012.] <https://www.facebook.com/help/?faq=224049364288051>.
- [9] BACHFELD, DANIEL. Heise Security. [Online] [Zitat vom: 1. 07 2012.] <http://www.heise.de/security/artikel/Zweifelhafte-Antiviren-Produkte-270094.html>.
- [10] FACEBOOK INC. [Online] [Zitat vom: 12. Juli 2012.] <https://developers.facebook.com/docs/appsonfacebook/pagetabs/>.
- [11] DR. JOHANNES WIELE. *Social Engineering erkennen* [Online] [Zitat vom: 12. Juli 2012.] <http://www.lanline.de/fachartikel/social-engineering-erkennen.html?page=3/>.

Kapitel 6

Sicherheit von Smart Grids

Johann Dreßler

Die folgende Abhandlung mit dem Thema „Sicherheit von Smart Grids“ geht der Fragestellung nach, ob und für wen Smart Grids ein neues Angriffsziel darstellen. Dazu werden die Begrifflichkeiten „Smart Grid“ und „Sicherheit“ näher beleuchtet. Danach folgt ein Überblick über wesentliche technische Smart Grid-Komponenten. Schließlich werden Angriffsmöglichkeiten und von welchen Tätern eine Gefahr ausgeht zusammenfassend erläutert.

Inhaltsverzeichnis

6.1	Einführung	119
6.2	Das Smart Grid	120
6.2.1	Was ist das Smart Grid?	120
6.2.2	Der Sinn und Zweck des Smart Grids	121
6.3	Der Begriff Sicherheit und dessen Vielfalt	123
6.3.1	Vertraulichkeit	123
6.3.2	Integrität	124
6.3.3	Authentizität	124
6.3.4	Verfügbarkeit	124
6.4	Die Smart Grid-Technik	124
6.4.1	Das Smart Meter	124
6.4.2	Kommunikationssysteme	125
6.4.3	IKT-gestützte Energiemanagementsysteme	126
6.5	Angriffsmöglichkeiten	127
6.5.1	Der Cyber-Terrorist	128
6.5.2	Die Organisierte Kriminalität	128
6.5.3	Der Dienstleister als Täter	128
6.5.4	Der Endkunde in der Täterrolle	129
6.6	Schlußfolgerung	129

6.1 Einführung

Es gibt wohl wenige Themen die so bedeutend für die Zukunft sind wie die Energieversorgung. Deutschland hat sich gegen die Atomkraft und für die fossilen und regenerativen Energieträger entschieden. Da eines Tages nur noch die regenerativen Energieträger zur Verfügung stehen werden und diese in ihrer zeitlichen Verfügbarkeit variieren, müssen Energiemanagementsysteme zum Einsatz kommen. Diese werden nicht nur der Energiebereitstellung dienen, sondern auch der Energieeinsparung.

Der Einzug der intelligenten Stromzähler in die deutschen Haushalte ist im Gange. Mit dem Energiewirtschaftsgesetz von 2010 ist dies für Neubauten und Totalsanierungen verpflichtend. Diese zukunftsweisenden Geräte steuern und verwalten unseren Energieverbrauch. Allerdings geben wir dem Energielieferanten damit auch die Möglichkeit zum Beispiel ein Verbrauchsprofil unseres Haushalts zu erstellen oder eine Fernabschaltung durchzuführen. Diese Stromzähler sind aber nur ein kleines, wenn auch sehr wichtiges Zahnrad im großen Uhrwerk, das den Namen „Smart Grid“ trägt. Das intelligente Stromnetz umfasst neben den Smart Meter auch Energieerzeuger und -speicher, sowie Übertragungssysteme. Mehr über das intelligente Stromnetz und dessen Sinn und Zweck wird im Abschnitt 2.2 erläutert. Was ein Smart Meter genau ist, warum man es braucht und was es außerdem für wichtige, technische Bausteine in einem Smart Grid gibt, wird im Abschnitt 2.4 betrachtet.

Der gesamte Energiefluss, von der Quelle bis zum Verbraucher, wird dann durch Informations- und Kommunikationstechnologie gesteuert. Dies ist notwendig um die Waage zwischen Energieangebot und Energienachfrage möglichst im Gleichgewicht zu halten und so eine optimale Energieversorgung zu jedem Zeitpunkt sicherzustellen. Auf der anderen Seite bietet diese weitverzweigte Systemstruktur auch Angriffspunkte um sie zu stören und zu manipulieren. Es stellt sich also dem Verbraucher die Frage, wie sicher das ganze Smart Grid ist. Dieser Sicherheitsaspekt wird in Abschnitt 2.3 beleuchtet. Ein kleine Auswahl an möglichen Angriffsszenarien wird dann im Abschnitt 2.5 dargestellt.

6.2 Das Smart Grid

6.2.1 Was ist das Smart Grid?

Der Begriff „Smart Grid“ stammt aus dem Englischen und bedeutet „Intelligentes Stromnetz“. Es umfasst im Wesentlichen die Ergänzung des bereits vorhandenen Stromnetzes um ein Kommunikationsnetz. Dies stellt die Übermittlung von Informationen zwischen den Verbrauchern und den Anbietern sicher. Weiterhin werden Informationen, die der Steuerung der einzelnen Bausteine des Smart Grids (Abbildung 6.2) dienen, übertragen. Allein schon an der Definition (Abbildung 6.1) des US-amerikanischen NIST erkennt man die Komplexität eines Smart Grids. Die Abbildung 6.3 verdeutlicht außerdem die Abhängigkeiten zwischen den einzelnen Komponenten

„A Smart Grid is the Modernization of the electricity delivery system so it monitors, protects and automatically optimizes the operation of its interconnected elements, from the central and distributed generator through the high-voltage transmission network and the distribution system, to industrial users and building automation systems, to energy storage installations and to end-use consumers and their thermostats, electric vehicles, appliances and other household devices.“

Abbildung 6.1: Definition Smart Grid [1, S. 3]

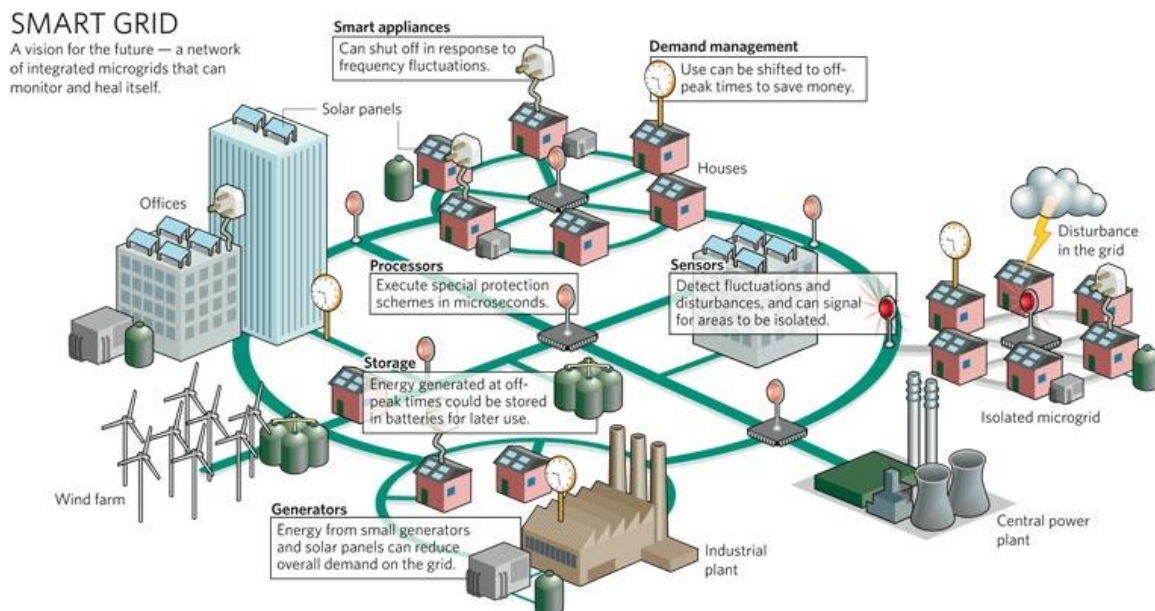


Abbildung 6.2: Komponenten eines Smart Grid
[1, S. 4]

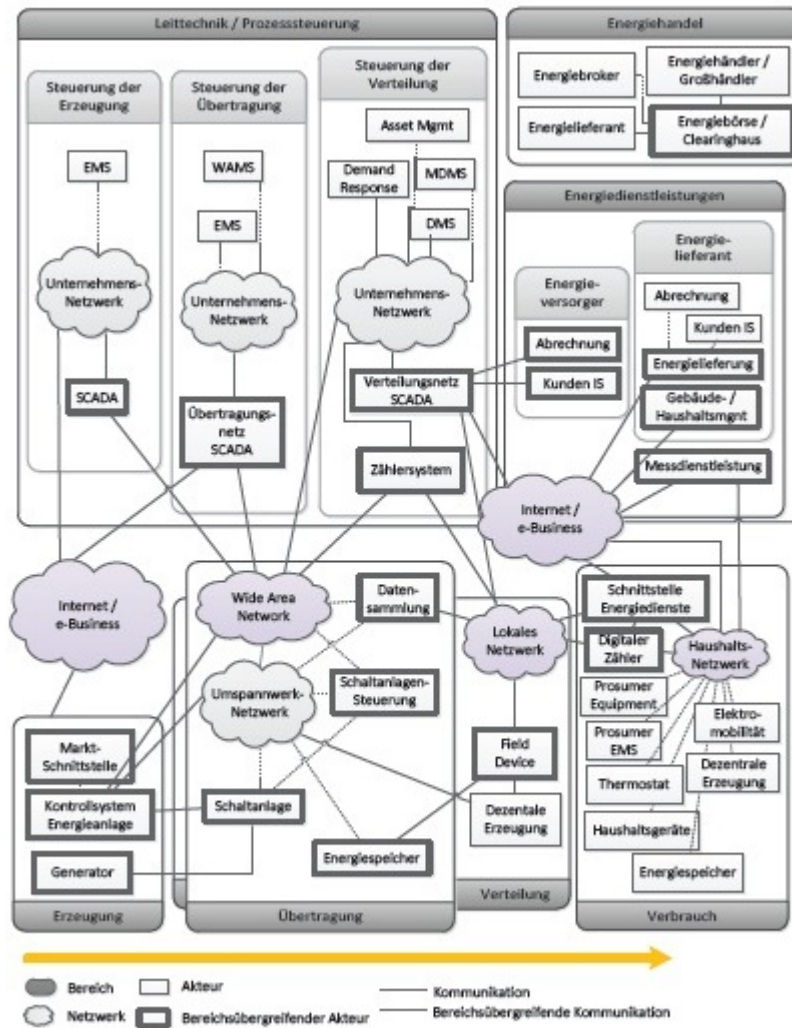


Abbildung 6.3: Konzeptuelles Modell des Smart Grid [4]

6.2.2 Der Sinn und Zweck des Smart Grids

Die zeitliche Verfügbarkeit regenerativer Energiequellen ist nicht vergleichbar mit der von fossilen Energiequellen. Wenn der Himmel bewölkt ist oder kein Wind weht, kann mit Solarzellen oder Windkraftträdern keine nutzbare Energie gewonnen werden. Um dies ausgleichen zu können, werden Energiespeicher, wie zum Beispiel Pumpspeicherkraftwerke, benötigt. Solche Energiespeicher stehen aber nur in sehr geringer Anzahl zur Verfügung. Somit muss der Strom in dem Augenblick genutzt werden, in dem er erzeugt wird. Wenn das nicht möglich ist, kann es zu Auswirkungen auf die Netzstabilität kommen.

Um diesen Folgen schon bei der Energieerzeugung entgegenzuwirken, gibt es zwei Herangehensweisen. Bei der ersten orientiert sich die Energiegewinnung am aktuellen, realen Verbrauch des Endkunden, der durch die Smart Meter erfasst wird. Das Standardlastprofil (Abbildung 6.4) verliert damit an Bedeutung. Die zweite Möglichkeit bezieht sich

auf die Verbrauchsregelung durch präzises Ein- und Ausschalten elektrischer Geräte. Dabei werden thermische Verbraucher wie Klimaanlage und Kühlaggregate aufgrund ihrer eigenen Pufferwirkung vorzugsweise eingesetzt.

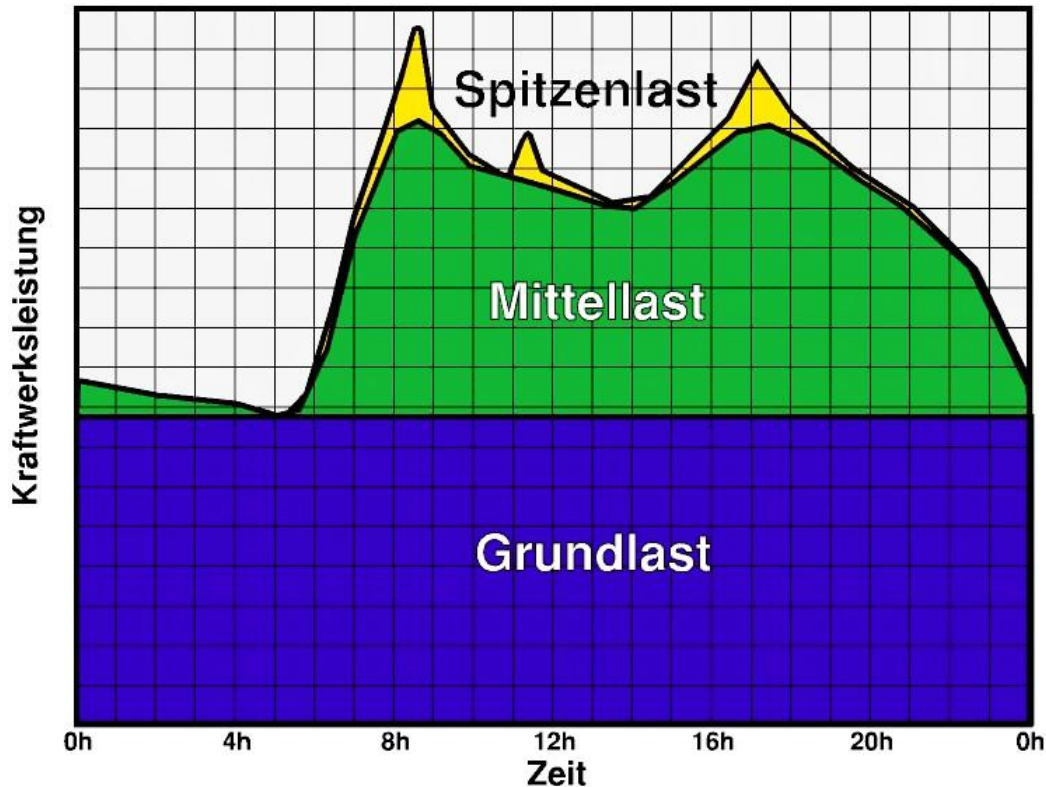


Abbildung 6.4: Das Standardlastprofil
[2, S. 2]

Die bisherige Denkweise, dass sich die Energiegewinnung an dem Bedarf ausrichtet wird nun durch die Energiewende umgekehrt. Die Verfügbarmachung regenerativer Energien durch die Nutzung von Smart Grids ist unerlässlich für die erfolgreiche Einbeziehung eben dieser neuen Energieträger in das bestehende Energiesystem. Des weiteren verändert sich die bisher hierarchisch organisierte Energiegewinnung durch einzelne Kraftwerke zu einer dezentralen Verteilung, die durch eine Vielzahl von Windkraftanlagen, Erdwärme- und Wasserkraftwerken gespeist wird. Beispielhaft kann man sagen, dass jede Photovoltaikanlage die beim Bürger auf dem Grundstück steht, zum Energieerzeuger für alle wird. Der Strom fließt dann also in beide Richtungen. (Abbildung 6.5) Um diese Vorgänge steuern zu können benötigt man Kommunikationsstrukturen die einen Datenaustausch zwischen den einzelnen Komponenten ermöglichen. Folglich müssen die Haushaltsgeräte auch über externe Steuerungsmöglichkeiten verfügen. Ist dies erreicht, „ist der Schritt zum „Smart Home“ auch nicht mehr groß: Der Wäschetrockner, der eine E-Mail verschickt, sobald das Flusensieb verstopft ist, ist da sicher nur der Anfang.“ [2, S. 3]

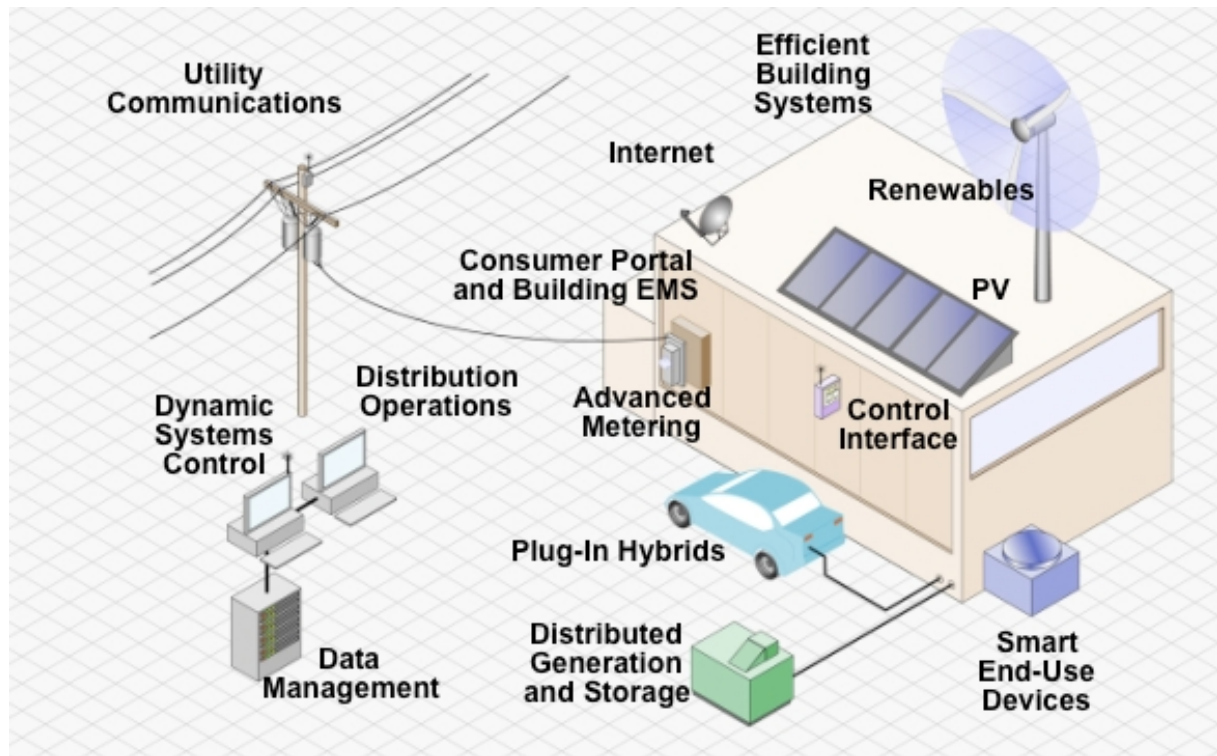


Abbildung 6.5: Vernetzungsbeispiel im Smart Grid
[1, S. 5]

6.3 Der Begriff Sicherheit und dessen Vielfalt

Der Begriff „Sicherheit“ wird in der Informationstechnik und auch in der Elektrotechnik verwendet. Jedoch gibt es Unterschiede in der Bedeutung in den jeweiligen Bereichen. In der Informationstechnik werden nach dem BSI besonders die Teilaspekte der Vertraulichkeit, der Integrität, der Authentizität und der Verfügbarkeit hervorgehoben. Dagegen konzentriert sich der Verband der Elektrotechnik, Elektronik und Informationstechnik e.V. (VDE) auf die Verhinderung von Stromunfällen. Die für das Smart Grid relevantesten Aspekte sind die der Informationstechnik und deshalb werden diese nachfolgend kurz erläutert.

6.3.1 Vertraulichkeit

Hinter dem Begriff „Vertraulichkeit“ steht die Verwehrung des unerlaubten Informationszugriffs. Smart Meter bieten die Möglichkeit die Verbrauchsdaten häufig zu erfassen. Aus diesen Daten können Verbrauchsprofile erstellt werden, die wiederum genaue Folgerungen bezüglich der Lebensverhältnisse des Endkunden zulassen. Hinzu kommt, dass nicht nur die Smart Meter Daten erzeugen. Jedes Smart Grid-Gerät, ob Waschmaschine, Kühlschrank oder die Heizung, produziert Daten über die Nutzung des Geräts. Somit können auch daraus Rückschlüsse auf den Alltag des Endkunden gezogen werden.

6.3.2 Integrität

Ein Synonym für Integrität ist die Zuverlässigkeit oder auch die Vertrauenswürdigkeit. Smart Meter verarbeiten unterschiedliche Daten, wie zum Beispiel den aktuellen Strompreis und den bisherigen Verbrauch. Die Integrität dieser Daten und des Smart Meters an sich sind sehr wichtig für den Energieanbieter, weil damit die Verbrauchskosten berechnet werden.

6.3.3 Authentizität

Bei dem Aspekt der Glaubwürdigkeit wird darauf geachtet, dass der richtige Endverbraucher auch mit seinem entsprechendem Energieversorger kommuniziert. Das gilt natürlich ebenso für die Gegenrichtung. Der Endverbraucher möchte seine sensiblen Daten nicht an ein beliebiges Unternehmen senden. Genauso müssen Energieanbieter sicher gehen, dass sie nicht mit absichtlich gesendeten, falschen Messdaten ihre Energiebereitstellung planen.

6.3.4 Verfügbarkeit

Dieser Faktor entspricht im weiteren Sinne der Versorgungssicherstellung, wobei dessen wohl schwerwiegendste Auswirkung der Ausfall der Stromversorgung ist. In diesem Fall kann der Endverbraucher seine elektrischen Geräte nicht mehr in Betrieb nehmen. Unter anderem ist davon auch die Kommunikation betroffen. Analogtelefone mögen dann noch funktionsfähig sein, aber die Internetkommunikation hingegen nicht. Aus diesem Grund ist eine Steuerung der Smart Grid-Komponenten über die Kommunikationsinfrastruktur, selbst bei funktionierender Stromversorgung, nicht mehr gegeben.

6.4 Die Smart Grid-Technik

6.4.1 Das Smart Meter

Der Zugang zum Smart Grid wird in jedem angeschlossenen Haushalt durch ein Smart Meter (Beispiel: Abbildung 6.6) sichergestellt. Diese „intelligenten Stromzähler“ wurden mit dem Energiewirtschaftsgesetz (EnWG) Anfang 2010 deutschlandweit eingeführt und sind bei Neubauten und Totalsanierungen Pflicht. Mithilfe der Smart Meter lässt sich der Stromverbrauch in kurzen Zeitabständen messen und die erfassten, digitalen Werte können dann direkt verarbeitet und übermittelt werden. Somit ermöglichen diese Geräte eine äußerst detaillierte und verzögerungsfreie Lastermittlung. Die bei der Verbrauchserfassung bis zur Rechnungserstellung verarbeiteten Daten unterteilen sich wie folgt(vgl. [3]):

- Verbrauchsdaten, die von der Messstelle erfasst und zum Messstellenbetreiber zum Zwecke der Abrechnung übertragen werden

- Gerätedaten, die zur Erstellung und Übermittlung der Verbrauchsdaten notwendig sind und den Nutzer für die Abrechnung eindeutig identifizieren,
- Nutzerdaten, die den Verbraucher kennzeichnen und für die Abwicklung der Abrechnung erforderlich sind oder ihm einen Zugang zu einem möglichen Webinterface beim Betreiber der Messstelle gestatten und
- Daten, die Informationen über das zur Abrechnung genutzte System liefern, wie z.B. Verbrauchsumsätze, Systemzustand und Topologie, Zulieferer des Wirksystems oder einzelner Geräte, Ausbreitung und geographische Verteilung des Systems.



Abbildung 6.6: Smart Meter der EVB Energie AG
[6]

Smart Meter können außerdem mit einem sogenannten „Unterbrecher“ ausgestattet sein. Das macht es dem Energieanbieter möglich, die Stromversorgung abzuschalten, ohne dass er vor Ort sein muss. Dafür gibt es zwei wesentliche Gründe von Seiten der Smart Meter-Hersteller. Zum einen kann dadurch die größtmögliche Leistungsaufnahme beschränkt werden. In der Praxis kann das nur mit einer Trennung der Stromversorgung sinnvoll realisiert werden. Zum anderen dient es dazu, Kunden, die ihre Stromrechnung unpünktlich oder gar nicht bezahlen, von der Stromversorgung auszuschließen. So werden Kosten für den Vor-Ort-Einsatz eines Mitarbeiters gespart und dieser wird gleichzeitig auch vor eventuell aufgebracht Kunden geschützt.

6.4.2 Kommunikationssysteme

Für die benötigten Kommunikationssysteme kann auf die bereits bestehenden Technologien (DSL, Mobilfunk, (W)LAN) zurückgegriffen werden. Somit ist dafür keine völlige Neukonzeption nötig. Andererseits müssen für die Vernetzung von Endgeräten wie zum Beispiel Waschmaschine, Klimaanlage und Kühlschrank Anschlussmöglichkeiten geschaffen werden. Auch die Übermittlung von Steuerinformationen bedarf neuer Innovationen um deren Sicherheit zu garantieren. Dabei steht im Speziellen die Sicherheit von SCADA-Netzen im Vordergrund, weil sie wichtige Komponenten in einem Smart Grid darstellen.

Die ursprüngliche Aufgabe von SCADA-Netzen ist das Überwachen und Steuern technischer Prozesse mithilfe eines PC-Systems. Hierbei sind Aktoren und Sensoren so miteinander verbunden, dass Steuerung und Kontrolle mittels Computern und PLC über das SCADA-Netz erfolgen. In so einem Netz müssen in Echtzeit Daten verarbeitet werden. Dabei sind die Ressourcen wie Speicher und CPU-Leistung beschränkt. Obwohl sie in hochsicherheitskritischen Umgebungen verwendet werden, sind sie eher selten mit Sicherheitsmaßnahmen ausgestattet, da der Betrieb von SCADA-Netzen meist isoliert ist. Außerdem würden etwaige Ver- und Entschlüsselungsmaßnahmen die Echtzeitdatenverarbeitung behindern. Weiterhin werden Authentifizierungsmaßnahmen der Benutzer meist zugunsten des schnellen Notfallzugriffs vernachlässigt. Ein längeres Passwort, welches vergessen wurde, oder eine scheiternde biometrische Zugangskontrolle stellen für solche Netze ein zu hohes Risiko dar. Die Entwicklung geeigneter Authentifizierungsmöglichkeiten muss dafür vorangetrieben werden.

Die Verlangsamung der Kommunikation in SCADA-Netzen kann auch durch den Einsatz von Firewalls ausgelöst werden. Diese müssen nämlich an die besonderen SCADA-Protokolle angepasst sein. Einen möglichen Angriffspunkt stellt hier die nötige Fernadministration dar.

Die Rahmenbedingungen der Ressourcenbeschränkung und der Echtzeitdatenverarbeitung stehen auch im Konflikt zu der Kommunikationssicherheit in SCADA-Netzen. Die Identifizierung einzelner Komponenten kann nicht garantiert werden, weshalb das Abhören und Manipulieren von übertragenen Daten möglich ist.

6.4.3 IKT-gestützte Energiemanagementsysteme

Gegenüber den vorhandenen Kommunikationsstrukturen sind die Energiemanagementsysteme erst im Aufbau. Hier erstellt und erprobt das E-Energy-Programm des BMWi erste Konzepte. So ein Managementsystem hat verschiedene Schichten. Dies beginnt mit den Systemen auf Haushaltsebene (Abbildung 6.7). Auf dem nächsten Level befinden sich die örtlichen Management-Zentralen, wo die Haushalte verwaltet werden. Diese lokalen Zentralen stehen untereinander, sowie mit überregionalen Zentralen in Verbindung und steuern die Energieflüsse. Besonders herausfordernd ist die internationale Koordinierung des Energiemanagements, da die Staaten verschiedene Ansprüche an die Verarbeitung von Daten stellen. Man erkennt, dass hohe Ansprüche an die Energiemanagementsysteme gestellt werden, die über den technischen Aspekt hinausgehen. So werden auch Fähigkeiten aus den Bereichen Informatik, Steuerungs- und Regelungstechnik sowie der Rechts- und Wirtschaftswissenschaft benötigt.

Aufgrund der komplexen und schwergewichtigen Anforderungen eines Smart Grids, die besonders bei der Sicherheit und im Datenschutz liegen, können bereits im Einsatz befindliche Systeme (Produktionssteuersysteme, Logistiksysteme) nicht verwendet werden. Ausschlaggebend dafür sind die Merkmale des Smart Grids, „wie Heterogenität und Vielzahl der beteiligten Parteien, die Dezentralität der Verwaltung, die Sensibilität der erfassten Daten etc.“ [1, S. 18]

Eine weitere Komponente der Energiemanagementsysteme sind die Energiemarktplätze, bei denen man sogenannte Mehrwertdienste in Anspruch nehmen kann. Anpassungsfähige Tarifmodelle gehören dabei zu den einfacheren Diensten. Wesentlich umfassendere Dienstleistungen stellen zum Beispiel die Fernsteuerung der Haushaltsgeräte mittels Mobilfunk oder auch medizinische Unterstützungsleistungen dar.

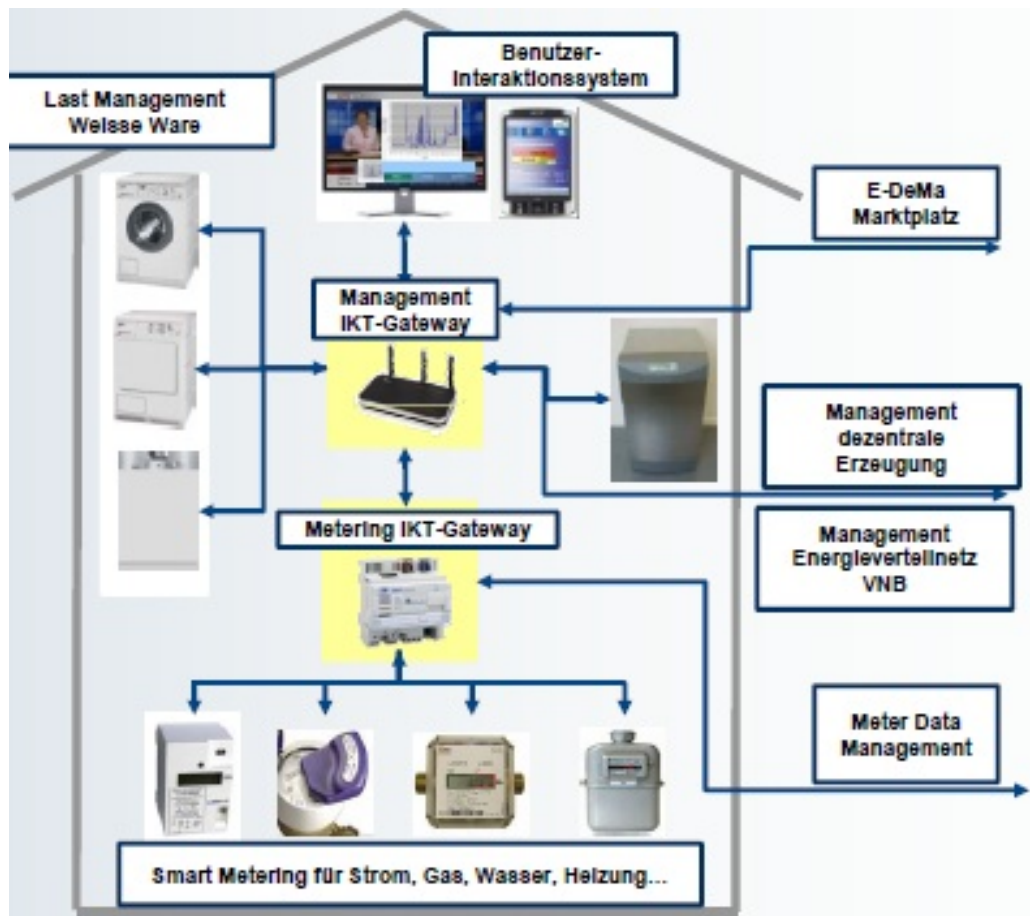


Abbildung 6.7: Energiemanagementsystem in Privathaushalten
[1, S. 17]

6.5 Angriffsmöglichkeiten

Eine US-amerikanische Studie [5] kam 2008 zu folgendem Resultat: „the energy industry has turn to the most vulnerable to cyberattacks“ Zukünftig werden also besonders Smart Grids im Fokus von Angriffen und Bedrohungen stehen. Ein Grund dafür sind unter anderem die vielen großflächig vorhandenen und schlecht gesicherten Smart Meter. Eine Empfehlung des Europäischen Parlaments sieht vor, dass 80 Prozent der Endverbraucher bis zum Jahr 2020 über intelligente Stromzähler verfügen. Der Netzbetrieb der Smart Grid-Komponenten bringt eine hohe Anzahl von ungenügend abgesicherten Zugangspunkten mit sich die wiederum sicherheitskritische Zugänge zu den Smart-Grid Systemen ermöglichen.

Wie schon erwähnt ist es dem Nutzer in einem Smart Grid möglich Mehrwertdienste über Energiemarktplätze zu beziehen. Um die Sicherheit dieser Dienste garantieren zu können, fehlen allerdings noch die technischen Überprüfungsöglichkeiten. Somit sind die übertragenden Daten vom Missbrauch und der unerlaubten Weiterleitung gefährdet. Daraus können sich Probleme für die Vertraulichkeit ergeben.

Sogenannte Cyber-Physical Systems entstehen durch die Verbindung von IKT-gestützten Anlagen mit physikalischen Elementen. Also sind auch Smart Grids Cyber-Physical Systems. Die Schnittstellen zwischen den Systemen stellen mögliche Angriffspunkte für Denial-of-Service- und terroristische Attacken dar. In einem Smart Grid sind auch Energieversorgungsnetze und Business-IT-Netze, welche zum Beispiel für die Mehrwertdienste genutzt werden, verbunden. Für den sicheren Betrieb von Energiemanagementsystemen birgt diese Vernetzung ein Risiko, da die Business-IT-Netze oft nur über ungenügende Kontrolle und Isolierung verfügen.

Aus diesen Gründen bieten Smart Grids verlockende Angriffspunkte, weshalb immer mit zunehmenden, gesetzwidrigen Angriffen zu rechnen ist. Im Folgenden werden die Gefahren, die von denkbaren Angreifer-Klassen ausgehen, kurz erläutert.

6.5.1 Der Cyber-Terrorist

Auf dem Cyber Security Gipfel 2010 wurden mögliche Angriffsszenarien durch Cyber-Terroristen diskutiert. Eine Möglichkeit ist das Einbringen von gefälschten Daten in die Kommunikationsinfrastrukturen mit dem Ziel, die Energieversorgung zu unterbrechen. Eine Zweite das Beeinflussen von Energiemanagementsystemen um die Weiterleitung von Daten aufzuhalten. Weiterhin könnten durch inkorrekte Daten Notfallabschaltungen ausgelöst werden. Durch Denial-of-Service Attacken könnte auch die Nutzbarkeit von Diensten beeinträchtigt werden. Der Einsatz von Bot-Netzen kann auch zu absichtlichen Stromausfällen und damit verbundenen wirtschaftlichen Schäden führen.

6.5.2 Die Organisierte Kriminalität

Die Verbrechen der Organisierten Kriminalität sind meist auf den finanziellen Gewinn ausgerichtet. In einem Smart Grid gibt es viele wertvolle Objekte und Informationen, weshalb es ein attraktives Angriffsziel ist. Da wäre zum einen der Diebstahl von Leistungen wie zum Beispiel der unentgeltliche Energieverbrauch. So können auch günstig Bot-Netze zur Kontrolle des Smart Grids etabliert werden, um daraus wieder finanziellen Gewinn zu schlagen. Außerdem könnte auch mit dem Verkauf von sensiblen Nutzer- und Verbrauchsdaten, die illegal beschafft wurden, Geld verdient werden.

6.5.3 Der Dienstleister als Täter

Bei den Mitarbeitern von Energiedienstleistern besteht am ehesten die Gefahr, dass diese Opfer von Social Engineering-Angriffen werden. Die Schadsoftware wird dabei mittels gängigen Wechselmedien oder fingierten E-Mails auf die Computer von Mitarbeitern

übertragen. Dort können dann eventuell gespeicherte Passwörter oder andere wichtige Informationen ausgelesen werden. Weiterhin könnten Angreifer durch Phishing-Methoden an die Zugangsdaten von Mitarbeitern kommen, um auf diese Weise unerlaubten Zugriff zu sicherheitsrelevanten Programmen zu erhalten. Mittels ungenügend gesicherten Remote-Zugängen können Angreifer über die Fernwartung Zugang zu den Smart Metern von Privathaushalten erlangen und dann beliebig Daten verändern.

6.5.4 Der Endkunde in der Täterrolle

Smart Grid-Komponenten im Privathaushalt sind direkten physischen Angriffen ausgeliefert, da sie meist einfach zu erreichen sind. Denkbar sind Angriffe zur eigenen Vorteilsnahme, wie zum Beispiel die Verbrauchsdatenmanipulierung zum eigenen geldwerten Vorteil. Aber auch Verbrauchsdaten und Lastprofile von Mietern und Nachbarn können damit ausspioniert werden. Des Weiteren könnten Verbrauchswerte umgeleitet werden, um die Konten anderer zu belasten. Zwischen den Energiemanagementsystemen und den Smart Metern fließen die Kommunikationsdaten in beide Richtungen. Deshalb muss man damit rechnen das die Zählstände Dritter von außerhalb des Privathaushalts verändert werden oder sogar der Strom abgeschaltet werden könnte.

6.6 Schlußfolgerung

Zumindest in Deutschland sind die Weichen für eine umweltfreundliche Energiewirtschaft durch die forcierte Nutzung regenerativer Energien gestellt. Der Einzug von Smart Metern in Privathaushalte ist im Gange und die Netzstrukturen sind teils vorhanden und teils im Aufbau sowie der Entwicklung. Die Bereitstellung von sicheren und vertrauenswürdigen Smart Grid- Komponenten und Netzen stellt die Energieanbieter und die Entwicklung vor bedeutende Aufgaben. Nur wenn diese gewissenhaft und gründlich erledigt werden, ist die Akzeptanz und Leistungsfähigkeit solcher Energiesysteme gegeben und die Ziele des Energiesparens und des Umweltschutzes erfüllt.

Schließlich bleibt zu sagen, dass Smart Grids attraktive Angriffsziele darstellen. Dementsprechend muss damit gerechnet werden, dass es immer wieder Versuche gibt die Smart Meter und die dahinter stehenden Netzinfrastrukturen anzugreifen und zu manipulieren. Dabei ist mit allen erdenklichen Auswirkungen zu rechnen. Die Energieanbieter und Netzbetreiber, aber auch die Endverbraucher sollten und müssen darauf vorbereitet sein um etwaige Folgen minimieren zu können.

Literaturverzeichnis

- [1] CLAUDIA ECKERT. *Sicherheit im Smart Grid*, Alcatel-Lucent-Stiftung, Stuttgart 2011.
- [2] KLAUS J. MÜLLER. *Sicherheit im Smart Grid*, Secorvo Security Consulting GmbH, Karlsruhe 2011.
- [3] PETER SCHOO. *Smart-Energie: Wieviel Datenschutz und Datensicherheit wollen wir uns leisten?*, Fachtagung Smart Energy 2010, Dortmund 2010.
- [4] PETRA BEENKEN, HANS J. APPELRATH, CLAUDIA ECKERT. *Datenschutz und Datensicherheit in intelligenten Energienetzen*, DACH Security 2010, Wien 2010.
- [5] <http://www.secpoint.com/Survey-revealed-that-Energy-industry-at-risk-of-cyberattack.html>
- [6] http://de.wikipedia.org/w/index.php?title=Datei:Intelligenter_zae_hler_-_Smart_meter.jpg&filetimestamp=20081129180313

Kapitel 7

Analyse von Botnetzen

Dominik Holzapfel

Botnetze sind heutzutage der Ursprung für allerlei unerwünschte Aktivitäten im Internet, wie beispielsweise Spam-Versand oder Distributed Denial of Service (DDoS)-Attacken. Sie sind verantwortlich für einen Großteil des unerwünschten Traffics und stellen somit auf vielerlei Ebene eine Bedrohung dar. In dieser Ausarbeitung werden Botnetze deswegen in ihrer Gesamtheit analysiert, von der Verbreitung und den typischen Anwendungen über die Erkennung bis zur endgültigen Abschaltung.

Inhaltsverzeichnis

7.1	Einleitung	133
7.2	Botnetze	134
7.2.1	Definition	134
7.2.2	Aufbau/Struktur	135
7.2.3	Verbreitung	136
7.2.4	Anwendung	138
7.3	Erkennung	140
7.3.1	Honeypots	140
7.3.2	Traffic-Analyse	141
7.3.3	DNS-Monitoring	142
7.4	Abschaltung	143
7.4.1	Simultane C&C Abschaltung	143
7.4.2	Honeypots bei P2P	143
7.4.3	Active Worm Defense	144
7.5	Ausblick	144

7.1 Einleitung

Schadsoftware für Computer ist fast genauso alt wie Computer selber, nur wenige Jahre nach der Verbreitung der ersten Heimcomputer Ende der 1970er Jahre traten die ersten Schadprogramme auf. Heutzutage gilt Brain aus dem Jahr 1986 als der erste Computervirus¹. Eine der ersten wissenschaftlichen Arbeiten, die sich mit Computerviren beschäftigte, stammt aus dem Jahr 1987 [1]. Zur damaligen Zeit waren Computerviren vor allem eine Möglichkeit, die eigenen Programmierkenntnisse auszureizen und hatten hauptsächlich humoristischen Charakter [2]. Doch schon bald tauchten die ersten Viren mit Schadcharakter auf, die Dateien löschten oder System unbrauchbar machten. Mit der Zeit wandelte sich der Charakter der ursprünglichen Viren und es tauchten neue Formen wie beispielsweise Würmer oder Trojaner auf. Eines der wohl populärsten Beispiele für diese neue Kategorie von Schadsoftware mit Zerstörungspotential ist der "I LOVE YOU"-Virus, der sich zu Beginn des neuen Jahrtausends schlagartig verbreitete und auf den befallenen Systeme zahlreiche Dateien löschte^{2,3}. Die verursachten Schäden sind schwer zu beziffern, Schätzungen gehen jedoch von einer Milliardenhöhe aus.

Schlussendlich kam es auch zur Entwicklung von Botnetzen. Die Anfänge liegen vermutlich im Internet Relay Chat (IRC) [3]. Dort ist es möglich, mithilfe von sogenannten Bots einfache Funktionen zu automatisieren, wie etwa die Vergabe von Rechten, die Ausgabe von Statistiken und Wetterdaten oder auch einfache Spiele-Bots für automatische Quize oder Poker⁴. Aufbauend auf dieser Idee von einfachen, fernsteuerbaren und verteilten Bots entstanden so die ersten Botnetze, die für schädliche Zwecke benutzt wurden [4].

Wann genau es zum ersten Botnetz kam, spielt im Endeffekt auch gar keine Rolle. Tatsache ist, dass Botnetze heute eine der größten Bedrohungen im Internet darstellen, Schätzungen gehen davon aus, dass bis zu einem Viertel aller Computer Teil eines Botnetzes sind⁵. Die Idee hinter einem Botnetz ist es, die befallenen Computer - im Gegensatz zu herkömmlicher Schadsoftware - nicht direkt zu schädigen, sondern sich die schiere Rechenpower zu Nutzen zu machen, die sich aus der Unzahl an befallenen Systemen ergibt. Damit lassen sich dann weitere Angriffe realisieren, wie beispielsweise eine DDoS-Attacke gegen Server oder der Massenversand von Spam. Weitere Beispiele folgen in einem späteren Abschnitt.

Im Folgenden werden Botnetze zuerst klassifiziert und typische Strukturen beschrieben. Danach folgt eine Erklärung der Verbreitungsarten und der häufigsten Einsatzmöglichkeiten. In den weiteren Abschnitten geht es um die Erkennung eines Botnetzes, sowie seiner anschließenden Abschaltung. Als Abschluss geht es im Ausblick um die Zukunft der Botnetze.

¹<http://campaigns.f-secure.com/brain/>

²<http://heise.de/-19551>

³<http://nakedsecurity.sophos.com/2009/05/04/memories-love-bug-worm/>

⁴Eine Auswahl an Bots findet sich unter <http://www.dmoz.org/Computers/Software/Internet/Clients/Chat/IRC/Bots/>

⁵<http://news.bbc.co.uk/2/hi/business/6298641.stm>

7.2 Botnetze

Im folgenden Abschnitt werden zuerst die Begriffe "Bot" und "Botnetz" definiert. Danach folgt eine Übersicht über die beiden vorherrschenden Strukturformen von Botnetzen, wie sie heutzutage anzutreffen sind. Den Abschluss bilden Möglichkeiten zur Verbreitung, sowie typische Anwendungen von Botnetzen.

7.2.1 Definition

Eine Definition von Bots und Botnetzen findet sich in [5]:

The term botnets is used to define networks of infected end-hosts, called bots, that are under the control of a human operator commonly known as a botmaster.

Eine weitere, ähnliche Definition liefert [6]:

A "botnet" consists of a network of compromised computers ("bots") connected to the Internet that is controlled by a remote attacker ("botmaster").

Zusammengefasst kann man also sagen, dass ein Botnetz nichts anderes ist, als eine Ansammlung an infizierten Systemen, die unter einer gemeinsamen Kontrolle von außerhalb stehen.

Einen wichtigen Zusatz zur Definition findet man in [7]:

Bots [...] [operate the infected computer] without any knowledge of the legitimate owner.

Dieser Zusatz ist insofern wichtig, als dass er ein Botnetz in dem Sinne, in dem es hier verwendet werden soll, abgrenzt von freiwilligen Aktionen der Nutzer, die die sonstigen Merkmale eines Botnetzes erfüllen. Ein solches Beispiel aus jüngster Zeit war die sogenannte "Operation Payback" der Netzaktivistengruppe "Anonymous", bei der sich tausende Sympathisanten freiwillig ein Programm installierten, um damit DDoS-Angriffe gegen Webserver zu ermöglichen [8]. Diese "freiwilligen Botnetze" sind im folgenden Verlauf nicht Teil der Betrachtungen.

7.2.2 Aufbau/Struktur

Wie soeben erwähnt, ist es für die Existenz eines Botnetzes unabdingbar, dass die Bot-Rechner zentral gesteuert werden können. Typische Kommandos, die der Botmaster an seine Bots versendet, sind Organisationsbefehle, wie *Update* um den Schadcode zu aktualisieren, oder Ausführungsbefehle, wie *Flood* um einen DDoS-Angriff zu starten⁶.

Eine direkte Kommunikation des Botmasters mit den Bots ist aus mehrerer Hinsicht nicht praktikabel, weshalb eine Zwischenstelle für die Kommunikation benötigt wird. Um dies zu ermöglichen, gibt es im Wesentlichen zwei Arten; die zentrale Steuerung über einen oder mehrere Command and Control Server (C&C-Server) oder eine dezentrale Peer-to-Peer (P2P) Struktur. Weitere Möglichkeiten sind denkbar, zum Beispiel eine Steuerung mithilfe von Instant Messaging (IM) Diensten, in der Praxis jedoch aufgrund diverser Probleme nur sehr selten anzutreffen.

zentral

Wie bereits in der Einleitung erwähnt, entstanden Botnetze vermutlich erstmals im Umfeld von IRC. Vor diesem Hintergrund ist es auch nicht verwunderlich, dass die Mehrheit der Botnetze IRC als Kommunikationsmittel einsetzen [9]. Die Vorteile dieses Ansatzes liegen klar auf der Hand: IRC ist ein Chat-Protokoll, das es ermöglicht, auf einem Server oder einem Serververbund Chaträume (Channels) einzurichten, zu denen sich dann ein Client verbinden kann. Die Nutzung mit einer sehr großen Anzahl an Clients gleichzeitig ist problemlos möglich. Außerdem ist das Protokoll textbasiert und sehr einfach, weshalb für nahezu jede Programmiersprache passende IRC-Bibliotheken existieren. Die Integration in eine Schadsoftware ist also einfach zu realisieren.

Der Botmaster setzt also einen Verbund aus IRC-Servern auf und codiert die Server-Adressen in seinen Schadcode. Die Bots verbinden sich nach der Infektion des Zielsystems zum IRC-Server und erhalten von dort weitere Anweisungen. Ist der Botmaster aktuell ebenfalls online im Channel, kann er seinen Bots direkte Befehle erteilen. Ansonsten gibt es die Möglichkeit, die Message of the Day (MOTD) des Channels zu benutzen. Die MOTD ist eine Nachricht, die jedem Client nach der Verbindung zugestellt wird und kann somit zum Beispiel für die Bekanntgabe des aktuellen DDoS-Ziels genutzt werden.

Aber auch über die reine Kommunikation hinaus bietet IRC noch ein paar weitere, für den Botmaster interessante Funktionen. So gibt die Anzahl der aktuell zum Channel verbundenen Bots Aufschluss darüber, wie groß das gesamte Botnetz inzwischen ist. Weiterhin unterstützt IRC mit der Erweiterung Xabi Direct Client-to-Client oder eXtended DCC (XDCC) Dateiübertragungen [10], womit es dem Botmaster ermöglicht wird, Updates der Botsoftware auf dem selben Kanal zu verteilen.

Aus dieser zentralen Struktur folgt natürlich auch direkt eine Verwundbarkeit, da die IRC-Server einen Single Point of Failure darstellen. Deshalb kommen meistens mehrere Server parallel zum Einsatz, um die Überlebensfähigkeit zu erhöhen.

⁶http://www.securelist.com/en/analysis/204792003/The_botnet_business

dezentral

Um die Schwachstellen einer zentralen Infrastruktur zu umgehen, begannen Botnetz-Autoren, eine dezentralisierte P2P-Architektur zu implementieren. Die ersten Ansätze dazu finden sich bereits 2003 im Wurm Slapper [11]. Das erste große dezentrale Botnetz war das Storm Botnetz, das 2007 aktiv war und vom dem vermutet wird, das es hinter den Cyber-Angriffen auf Estland im selben Jahr steht [12].

Eine P2P-Architektur ist ungleich schwieriger zu implementieren und erfordert sehr viel mehr Aufwand als die Nutzung von IRC. So muss beispielsweise dafür gesorgt werden, dass neu infizierte Systeme dem Netz beitreten können. Da eine zentrale Schwachstelle ausgeschlossen werden soll, ist es also nicht möglich, einen Server bereitzustellen, der bekannte Client-Adressen verwaltet, wie es zum Beispiel beim BitTorrent-Protokoll mithilfe sogenannter Tracker realisiert wird⁷. Stattdessen müssen die Clients ihre IP-Adressen untereinander austauschen. Dabei muss aber stets darauf geachtet zu werden, dass ein einzelner Client nur eine streng begrenzte Anzahl an anderen Clients kennen darf. Ansonsten bestünde die Gefahr, dass im Falle einer Kompromittierung des Botnetzes durch einen einzigen Client eine Übernahme der Kontrolle oder eine Abschaltung droht.

Mögliche Implementierungen einer P2P-Struktur finden sich in [6] oder [13].

7.2.3 Verbreitung

Die Verbreitung von Botnetzen unterscheidet sich nicht von der traditioneller Malware. Einziger Zusatzaspekt ist, dass das Herstellen einer Verbindung zum C&C-Server oder zum P2P-Netz notwendig ist, um die Infizierung eines Systems abzuschließen. Trotzdem sollen im Folgenden die gängigsten Verbreitungsmechanismen kurz vorgestellt werden.

verseuchte Dateien

Die mit Abstand häufigste Verbreitungsart von Malware ist die direkte Weitergabe oder die Einbettung des Schadcodes in andere Dateien, über drei Viertel aller Malware wird auf diese Art übertragen [14]. Am einfachsten ist die Einbettung in eine ausführbare Datei, da der eingeschleuste Schadcode hier sehr einfach zur Ausführung gebracht werden kann. Doch auch die Einbettung in andere Dateien, wie zum Beispiel Dokumente oder Multimedia-Inhalte ist möglich. Das Programm, mit dem die betroffenen Dateien geöffnet werden, muss in diesem Fall einen Fehler besitzen, über den es möglich ist, eigenen Code einzuschleusen. Beliebteste Angriffsziele sind die Programme, die auf nahezu jedem Rechner installiert sind, also Microsoft Office, Internet Explorer, Adobe Reader, Adobe Flash Player und Java, um nur einige Beispiele zu nennen [15].

⁷http://bittorrent.org/beps/bep_0003.html

Scanning und Exploiting

Diese zweithäufigste Verbreitungsart zielt auf Schwachstellen in Software ab, die über das Netz angesprochen werden können. Typische Angriffsziele sind Server für Dienste wie Hypertext Transport Protocol (HTTP), File Transfer Protocol (FTP), Structured Query Language (SQL) oder Server Message Block (SMB). Jeder dieser Dienste hat bekannte Standard-Ports, für HTTP beispielsweise Port 80. Ein Bot, der auf einem bereits infizierten System läuft, überprüft nun, ob auf einem potentiellen Zielsystem ein möglicherweise verwundbarer Dienst läuft, indem er eine Anfrage an den zugehörigen Port sendet (Scanning). Antwortet das Zielsystem, so sendet es dabei in den meisten Fällen auch eine Versionskennung der verwendeten Software mit. Existiert für die gefundene Software in der gemeldeten Version eine Schwachstelle, die es ermöglicht, beliebigen Code einzuschleusen und zur Ausführung zu bringen, so kann sich die Schadsoftware replizieren und auf einem weiteren System einnisten (Exploiting).

Scanning kann in zwei Arten auftreten.

Lokal Bei der lokalen Variante wird nur das jeweilige Subnetz gescannt, in dem sich das bereits infizierte System befindet. Der Vorteil ist, dass ein Subnetz oftmals unter einer gemeinsamen Verwaltung steht und auf vielen Systemen die gleiche Software zum Einsatz kommt. Somit ist es wahrscheinlich, dass eine ausgenutzte Schwachstelle auf vielen weiteren System funktioniert. Allerdings ist es ebenso wahrscheinlich, dass alle Bots gleichzeitig funktionsunfähig gemacht werden, beispielsweise durch eine gemeinsame Firewall oder einen Patch, der im gesamten Subnetz eingespielt wird.

Global Im Gegenzug dazu generiert der Bot bei einem globalen Scan zufällige IP-Adressen und überprüft diese daraufhin auf Schwachstellen. Die Aussicht auf Erfolg ist dementsprechend auch deutlich kleiner, da sehr viele IP-Adressen entweder nicht vergeben sind oder nicht aus dem Internet erreichbar sind. Allerdings wird bei einem erfolgreichen Fund der Schadcode sehr weiträumig verteilt.

Drive-By Downloads

Unter dem Ausdruck Drive-By Downloads versteht man, wenn der Nutzer dazu gebracht wird, eine Webseite mit schadhaften Inhalten aufzurufen, die dann automatisch, also ohne weiteres Zutun, auf dem System des Nutzers installiert werden [16]. Meist handelt es sich um JavaScript, mit dem eine Schwachstelle des Browsers ausgenutzt wird oder es wird direkt eines der Browser-Plugins angegriffen, wie zum Beispiel der Flash Player. Diese Angriffe sind deshalb so extrem effektiv, da bis zu 45% aller Internetnutzer mit einer veralteten und damit verwundbaren Version eines Browsers im Internet surfen [17]. Plugins wie der Flash Player haben darüber hinaus eine Verbreitung von etwa 98%, wodurch sich Schadcode-Entwickler nahezu darauf verlassen können, auf einem potentiellen Zielsystem mindestens ein verwundbares Plugin zu finden.

7.2.4 Anwendung

Sobald sich das Bot-Programm auf genügend Rechnern eingenistet hat, kann der Botmaster damit beginnen, es für seine Zwecke einzusetzen. Wie bereits in der Einleitung erwähnt, ist das Ziel nicht, die befallenen System zu schädigen. Stattdessen wird der Botmaster versuchen, das Botnetz zu monetarisieren. Möglichkeiten, um mit einem Botnetz Geld zu "verdienen" gibt es zahlreiche, im Folgenden sollen nur ein Teil davon ausschnittsweise dargestellt werden. Zu beachten ist dabei, dass sich die Möglichkeiten gegenseitig nicht ausschließen, ein Botnetz kann sehr wahrscheinlich ebenso eine Kombination an Anwendungen aufweisen.

Spam

Spam-Mails, also unerwünschte Werbe-Emails machen inzwischen den größten Teil aller gesendeten Email aus. [18] beziffert den Prozentsatz der Spam-Mails bei etwa 90%, andere Schätzungen gehen von ähnlich hohen Zahlen aus. Laut [19] wurden ungefähr 77% davon durch Botnetze versendet. Der Versand von Werbe-Mails mag auf den ersten Blick extrem ineffizient und nicht lukrativ erscheinen. Und tatsächlich ist der Prozentsatz der Spam-Mails, die erfolgreich zugestellt werden, den Spam-Filter passieren, vom Nutzer gelesen werden und schlussendlich zum Verkauf eines Produktes führen verschwindend gering und bewegt sich im Bereich von 0.00001% [20]. Diese irrwitzig kleine Anzahl wird dadurch ausgeglichen, dass große Botnetze durchaus in der Lage sein können, mehrere Milliarden Mails pro Tag zu versenden. Daraus ergibt sich, dass sich die Gewinne eines Botnetzes derartiger Größe durchaus im Bereich von bis zu 10.000\$ pro Tag bewegen können.

DDoS

Mithilfe eines DDoS-Angriffs kann ein Zielsystem vorübergehend unbrauchbar gemacht werden, indem es mit mehr Anfragen überlastet wird, als es verkraftet. Eine DDoS-Attacke ist auf mehrere Arten möglich, die verbreitetste und einfachste ist jedoch der sogenannte SYN-Flood, der auf Transmission Control Protocol (TCP) [21] aufbaut. Um damit einen Server tatsächlich zum Überlasten zu bringen, ist jedoch eine große Anzahl an Angreifern notwendig.

Das grundlegende Problem von TCP, das diese Art von Angriff ermöglicht, ist die Art und Weise, wie Verbindungen aufgebaut werden. Da TCP ein zuverlässiges, verbindungsorientiertes Protokoll ist, kommt zu diesem Zweck ein Drei-Wege-Handshake zum Einsatz. Im ersten Schritt wendet sich der Client an den Server und sendet ein SYN-Paket mit einer zufälligen Sequenznummer. Der Server antwortet im zweiten Schritt dem Client daraufhin seinerseits mit einem SYN-Paket, mit einer eigenen Sequenznummer und der um eins erhöhten Sequenznummer des Clients zur Bestätigung. Im dritten Schritt bestätigt der Client ebenfalls, indem er die Sequenznummer des Servers um eins erhöht und zurücksendet. Danach ist die Verbindung hergestellt und die Nutzdaten können gesendet werden.

Kernstück des Problems ist nun, dass der Server nach seiner ersten Antwort den Status der bis dahin halboffenen Verbindung im Speicher vorhalten muss. Dazu muss er mindestens die Internet Protocol (IP)-Adresse des Clients und die Sequenznummer zwischenspeichern, oftmals wird zusätzlich bereits ein Teil des Arbeitsspeichers reserviert. Verhält sich der Client nun absichtlich schadhaf, so kann er diese Tatsache verwenden, um den Server zu überlasten, man spricht dann vom bereits erwähnten SYN-Flooding [22]. Dazu sendet der Client massenhaft Verbindungsanfragen an den Server, ohne jedoch den dritten Teil des Handshakes abzuschließen. Der Server hält also immer mehr halboffene Verbindungen vor, auf die keine weitere Antwort mehr erfolgt. Um seine Anonymität zu erhöhen und die Erkennung eines Angriffs zu erschweren, kann der Client darüber hinaus seine Absender-Adresse fälschen und stattdessen eine zufällige Adresse in das TCP-Paket eintragen. Eine gültige Anfrage an den Server ist dann von einem Angriff nicht mehr zu unterscheiden.

Die Verteidigung gegen einen DDoS-Angriff ist grundsätzlich möglich, jedoch nicht immer einfach, sondern unter Umständen sehr komplex und teuer.

Mit DDoS-Angriffen werden regelmäßig Webseitenbetreiber zu Schutzgeldzahlungen erpresst. Als Beispiele seien hier nur Angriffe auf Pizza-Bestelldienste⁸ oder auf Wettanbieter⁹ genannt.

Phishing

Phishing zielt darauf ab, Anmeldedaten von Nutzern zu erhalten. Die Bandbreite reicht dabei von relativ unwichtigen Foren-Logins, über die Anmeldedaten von Onlineshops bis hin zu Bank- oder Email-Account-Daten. Was ein Botmaster mit einer großen Anzahl an Kreditkartendaten anfangen kann, ist offensichtlich und braucht hier nicht gesondert erwähnt zu werden. Doch auch Email-Zugangsdaten stellen ein lohnenswertes Ziel dar, da es einem Angreifer hiermit unter anderem meist möglich ist, das Passwort für viele andere Dienste, die der geschädigte Nutzer einsetzt, zurückzusetzen. Auch der Versand von Mails an die Kontakte des Opfers ist möglich, da die Kontakte davon ausgehen, dass es sich um eine vertrauenswürdige Mail handelt und somit unbedachter Links anklicken oder Anhänge öffnen.

Um das Phishing mithilfe eines Schadprogramms durchzuführen, kann dieses beispielsweise aufgerufene Webseiten eines Opfers auf gefälschte Duplikate umleiten. Das direkte Mitschneiden von Tastatureingaben eines Nutzers durch einen Keylogger ist aber ebenso möglich. Je nachdem, wie aufwändig die Phishing-Webseite gestaltet wurde, hat sie das Potential, mehr oder weniger Nutzer in die Irre zu führen und zur Eingabe von vertraulichen Daten zu verleiten. Einer Studie zufolge liegt der Anteil an Nutzern, die eine Phishing-Webseite irrtümlich für glaubwürdig halten, je nach Seite zwischen 20 und 90% [23].

⁸<http://heise.de/-1329787>

⁹<http://www.golem.de/1106/84164.html>

7.3 Erkennung

Der erste Schritt zur Bekämpfung eines Botnetzes ist die Erkennung einer Infektion, diese lässt sich grob in proaktive und reaktive Maßnahmen aufteilen. Unter die proaktiven Vorkehrungen fallen alle Möglichkeiten, die dazu dienen sollen, die Infektion zu verhindern, bevor es dazu kommen kann. Darunter fallen zum Beispiel die Blockierung von schadhafte Webseiten¹⁰ oder die Erkennung von Botnetz-Software durch Antiviren-Scanner aufgrund von Signaturen oder Heuristiken. Diese Maßnahmen unterscheiden sich für Botnetze aber nicht von denen für allgemeine Malware und werden deshalb im Folgenden nicht genauer betrachtet. Es soll stattdessen um Botnetz-typische, reaktive Maßnahmen der Erkennung gehen, die also einsetzen, nachdem die Infektion bereits erfolgt ist. Es werden die Möglichkeiten der Nutzung von Honeypots, der Traffic-Analyse und des DNS-Monitorings vorgestellt.

7.3.1 Honeypots

Unter einem Honeypot versteht man eine Installation von realen oder virtuellen Maschinen, die einen Angreifer von seinem eigentlichen Ziel ablenken soll, beziehungsweise das Vortäuschen eines Zieles, das eigentlich gar keines ist. Diese sind über das Netz zwar erreichbar, werden im normalen Betrieb aber niemals angefragt. Ein Zugriff auf diese Systeme kann also immer als Angriffsversuch gewertet werden, da ein Angreifer von außen die Struktur des Netzes nicht kennt und somit zuerst alle Systeme ermittelt. Die Kommunikation mit den Honeypot-Systemen wird aufgezeichnet und ermöglicht dann Rückschlüsse auf verwendete Angriffsvektoren. Auch Hersteller von Anti-Malware Software setzen Honeypots ein, um eine möglichst große Anzahl an neuem Schadcode zu erhalten.

Honeypots werden in zwei Kategorien unterteilt, die im Folgenden kurz erläutert werden.

Low Interaction

Low Interaction Honeypots stellen die einfachere Form der Implementierung dar. Sie bieten nach außen mehrere Dienste an, wie zum Beispiel einen Web- und einen SQL-Server, emulieren aber nur die nötigsten Grundfunktionen und ermöglichen es so, Kopien von Malware zu erhalten, die automatisiert nach Schwachstellen sucht und diese ausnutzt.

Aufgrund der Einfachheit der zu emulierenden Systeme lassen sich Low Interaction Honeypots sehr gut als virtuelle Maschinen realisieren. Oft eingesetzte Umgebungen sind Nepenthes¹¹, dessen Nachfolger dionaea¹² oder honeyd [24]. honeyd ermöglicht die Emulation kompletter Netzstrukturen inklusive der darin befindlichen Systeme. Sobald ein Zugriff auf ein virtuelles System verzeichnet wird, kann der gesamte Traffic aufgezeichnet und zur späteren Analyse gespeichert werden. Nach einer Infektion wird das System mit

¹⁰Eine Möglichkeit dazu bietet Google Safe Browsing, dass in den Browsern Google Chrome und Mozilla Firefox verwendet wird, siehe <http://code.google.com/intl/de/apis/safebrowsing/>.

¹¹<http://nepenthes.carnivore.it/>

¹²<http://dionaea.carnivore.it/>

einer frischen Installation vergleichen, um den Schadcode zu extrahieren. Je nach Anwendungsfall kann das System dann entweder neu aufgesetzt werden, oder der Schadcode wird weiter beobachtet, um aus seinem Verhalten und seinem Traffic Rückschlüsse auf die Gesamtfunktion zu erhalten.

High Interaction

Ein High Interaction Honeypot bietet den vollen Umfang eines echten Systems und eignet sich deshalb besonders zur Aufzeichnung manuell ausgeführter Angriffe, die über das bloße Einschleusen von Schadcode hinausgehen. Eine Firma kann beispielsweise einen High Interaction Honeypot aufsetzen und mit Daten befüllen, die denen der Produktivsysteme gleichen, aber verändert wurden. Hält ein Angreifer den Honeypot für glaubwürdig und attackiert diesen, so lässt sich nachvollziehen, nach welchen Daten er sucht.

Ein High Interaction Honeypot ist selbstverständlich deutlich aufwändiger, da eine reine Emulation der Protokolle nicht mehr ausreicht. Allerdings ist er, wie bereits erwähnt, speziell geeignet für die Analyse manueller Angriffe und daher für die Betrachtung von Botnetzen weniger relevant.

7.3.2 Traffic-Analyse

Eine Möglichkeit der Erkennung von bereits infizierten Systemen bietet die Traffic-Analyse, die dazu notwendigen Daten kommen meist aus Honeypots. Sind die aktuell verwendeten C&C-Server eines Botnetzes bekannt, so ist eine Erkennung schnell und einfach implementierbar, Listen von entsprechenden Servern sind öffentlich verfügbar¹³. Da die Server von Botmastern aber regelmäßig gewechselt werden, ist diese Möglichkeit nicht immer nutzbar und so kommen auch andere Techniken zum Einsatz.

Signaturen

Botzilla [25] ist eine Software, die Botnet-Traffic anhand charakteristischer Signaturen erkennen kann. Zur Initialisierung wird die Software mit zwei großen Mengen an Datenpaketen trainiert, die einerseits zulässigen Traffic und andererseits Traffic von Botnetzen enthalten. Aus den Trainingsmengen werden dann die Strings extrahiert, die mit sehr hoher Wahrscheinlichkeit in schädlichen Paketen auftreten. Sind die Signaturen erstellt, kann jedem unbekanntem Datenpaket eine Wahrscheinlichkeit zugeordnet werden, dass es sich um Botnetz-Traffic handelt. Liegt diese Wahrscheinlichkeit über einem Schwellwert, so wird das Paket verworfen und nicht weitergeleitet. Die Erkennung für bekannte Botnetze funktioniert relativ zuverlässig. Dieser Ansatz hat allerdings sowohl Probleme mit verschlüsseltem Traffic, als auch mit unbekannter Software, von der noch nicht ausreichend Pakete für eine Signaturgenerierung vorhanden sind.

¹³Ein Beispiel ist <http://www.cyber-ta.org/releases/malware-analysis/public/SOURCES/CandC.Cumulative.Summary>

IRC Traffic

Wie bereits mehrfach erwähnt, kommen für zentralisierte Botnetze überwiegend IRC-Server zum Einsatz. Ein Ansatz zur Erkennung, der sich auf die Analyse von IRC-Traffic spezialisiert, ist in [26] beschrieben. Das Verhalten von Bots in IRC-Channeln unterscheidet sich grundlegend vom Verhalten menschlicher Nutzer und kann deswegen zur Differenzierung dienen. Unterscheidungsfaktoren sind beispielsweise die durchschnittliche Anzahl an Wörtern in einer Nachricht (Menschen schreiben längere Sätze als die Botmaster-Kommandos), Umfang des verwendeten Vokabulars (Bots nutzen nur wenige Schlüsselwörter für die gesamte Kommunikation), aktive Nutzer im Channel (Bots senden meist keine Nachrichten an den Channel, sondern lauschen nur) und weitere. Dieser Ansatz funktioniert ebenfalls für unbekannte Botnetze.

Generalisierter Flow Ansatz

Die beiden vorangegangenen Ansätze analysieren den Inhalt der zu untersuchenden Datenpakete. Eine solche Deep Packet Inspection ist für große Router nicht mehr praktisch durchführbar. In [27] findet sich deshalb eine Möglichkeit, Botnetz-Traffic nicht anhand der Paket-Inhalte, sondern anhand der Flow Charakteristiken selber zu erkennen. Die Flows werden anhand ihrer Parameter, wie Dauer, Anzahl der Pakete, Geschwindigkeiten und anderen zuerst gefiltert und die Flows verworfen, die vermutlich zu keinem Botnetz gehören. Im Anschluss werden die verbleibenden Flows analysiert und in Verbindung mit anderen Flows im Netz gebracht. Finden sich mehrere Flows von verschiedenen Systemen, die eine hohe Korrelation aufweisen, so ist dies ein Indiz für gemeinsame Aktivitäten, wie sie unter anderem in Botnetzen auftreten. Damit lässt sich die Topologie eines Botnetzes erkennen und es lassen sich zum Beispiel zentrale C&C-Server ausfindig machen.

7.3.3 DNS-Monitoring

Um einer Erkennung und einer Abschaltung zu entgehen, wechseln Botmaster die verwendeten C&C-Server sehr häufig, oftmals kommen einzelne Server nur wenige Tage zum Einsatz. Damit auch Bots, die einige Tage offline waren, eine Migration auf die neuen Server vollziehen können, wird im Regelfall auf die Nutzung von Domain Name System (DNS) zurückgegriffen. Der Bot hat nur die Hostnamen einiger Server gespeichert und ruft die (ständig wechselnden) IP-Adressen vor einer Verbindung ab. Aufgrund der Struktur von Botnetzen entstehen dadurch Korrelationen bei den DNS-Anfragen, ähnlich denen bei der oben erwähnten Flow Analyse [28]. Bei einer normalen DNS-Nutzung rufen vollkommen zufällige Clients beliebige Einträge ab, Bots hingegen rufen in einer annähernd konstanten Gruppe nahezu zeitgleich den selben Hostnamen ab. Durch die Erkennung solcher Gruppenzusammenhänge lassen sich infizierte Systeme identifizieren.

7.4 Abschaltung

Nach einer umfassenden Erkennung und Analyse ist das nächste Ziel natürlich die Abschaltung eines Botnetzes. Während die Verbreitung, Anwendung und Erkennung von Botnetzen in der Literatur hinreichend untersucht sind, gibt es zur Abschaltung nur sehr wenige Untersuchungen. Die Gründe dafür sind zahlreich, umfassen aber unter anderem die Tatsache, dass der letzte Schritt, zum Beispiel die Abschaltung einzelner Server, im Aufgabenbereich von Strafverfolgungsbehörden liegt¹⁴. Die konkrete Vorgehensweise bleibt also in den meisten Fällen geheim. Die Behörden arbeiten dabei oftmals nicht alleine, sondern in Kooperation mit großen Firmen, wie zum Beispiel Microsoft¹⁵ oder Herstellern von Anti-Viren-Software.

Trotzdem sollen im Folgenden grundlegende Möglichkeiten aufgezeigt werden, ein Botnetz abzuschalten.

7.4.1 Simultane C&C Abschaltung

Eine Variante zur Abschaltung zentraler Botnetze ist die simultane Abschaltung aller aktuellen C&C-Server. Eine solche Aktion ist jedoch alles andere als trivial und erfordert unter Umständen monatelange Vorbereitung. Das generelle Problem ist, dass absolut alle Server gleichzeitig abgeschaltet werden müssen. Überlebt auch nur ein einziger Server, so kann das Botnetz durch den Botmaster eventuell vollständig wieder hergestellt werden. Durch die Verteilung der Server über die ganze Welt ist die Kooperation einer Vielzahl an Behörden notwendig. Trotzdem ist die simultane Abschaltung eine Vorgehensweise, die in der Praxis regelmäßig eingesetzt wird.

7.4.2 Honeypots bei P2P

Ein dezentralisiertes Botnetz verzichtet auf einen einzelnen Single Point of Failure genau zu dem Zweck, um eine Abschaltung zu erschweren. Eine Möglichkeit, wie mit dem Einsatz von Honeypots und sehr viel Glück ein P2P-Botnetz abgeschaltet werden könnte, wird in [6] beschrieben. Sollte es mit einem Honeypot gelingen, ein frisch entstandenes Bot-Programm in der ersten Phase der Verbreitung einzufangen, so kann das entstehende Botnetz eventuell massiv beeinflusst werden. Dazu werden möglichst viele Systeme des Honeypots absichtlich infiziert, damit neue Knoten außerhalb des Honeypots diese absichtlich infizierten Systeme in ihren Peer-Listen eintragen. Gelingt dieser Schritt, wird danach ein großer Teil der Kommunikation innerhalb des Botnetzes durch die Systeme des Honeypots abgewickelt. Somit kann das Netz analysiert und modifiziert werden. Da es einem Botmaster aber möglich sein kann, einen Honeypot zu erkennen [29] und er diese Techniken gerade zu Beginn der Verbreitung einsetzen wird, ist diese Variante eher theoretischer Natur als praktisch relevant.

¹⁴<http://heise.de/-1376540>

¹⁵Zwei Beispiele aus der jüngeren Vergangenheit finden sich unter <http://www.golem.de/1103/82187.html> und <http://www.golem.de/1109/86718.html>.

7.4.3 Active Worm Defense

Natürlich lässt sich ein Botnetz auch dadurch abschalten, dass alle infizierten Systeme bereinigt und gepatcht werden. In vielen Fällen passiert dies aber nicht, weil sich beispielsweise die Nutzer des Rechners der Gefahr durch Malware allgemein gar nicht bewusst sind und deshalb auf Schutzmechanismen und Software-Updates verzichten. Um infizierte Systeme aber doch zu bereinigen, beziehungsweise noch nicht infizierte Systeme vorsorglich zu schützen, gibt es die Methode der Active Worm Defense. Darunter versteht man die Verbreitung einer Software, die die selbe Schwachstelle benutzt, wie der Schadcode, auf den Systemen aber dann die Sicherheitslücke patcht oder die bereits vorhandene Malware entfernt [30]. Diese Maßnahme wirft jedoch neben den technischen Fragen auch jede Menge ethischer und moralischer Fragen auf. Außerdem muss eine solche Art der Verteidigung nicht zwangsläufig besser funktionieren als andere Techniken [31]. Es gab allerdings in der Vergangenheit bereits Einsätze solcher Konter-Würmer¹⁶.

7.5 Ausblick

Botnetze werden auch in mittelfristiger Zukunft eines der Hauptprobleme im Internet bleiben. Heutige Schadsoftware wird immer professioneller und immer schwerer zu erkennen und zu entfernen. Neuartige Botnetze, wie das aktuelle TDL-4 (Trojan Downloader) nutzen eine Vielzahl von Techniken, darunter verschlüsselte P2P-Kommunikation, Rootkit-Funktionalitäten in Kombination mit versteckten Dateisystemen, um jeder Entdeckung durch das System oder den Nutzer zu umgehen, oder eigenen Antiviren-Komponenten, um konkurrierende Schadsoftware auf infizierten Systemen zu entfernen¹⁷. TDL-4 gilt deshalb nach aktuellem Stand als unzerstörbar und wird vermutlich noch sehr lange zu den aktiven Botnetzen zählen.

Darüber hinaus suchen Botmaster ständig nach neuen Wegen, einer Abschaltung zu umgehen und probieren laufend neue Techniken aus. Botnetze, die keine klassische Struktur aufweisen, sondern stattdessen ihre Befehle über Twitter-Nachrichten erhalten, sind nur ein Beispiel für diese neue Art von Botnetzen¹⁸.

In Zukunft wird auch die mobile Komponente einen immer größeren Einfluss in der Malware-Szene allgemein und der der Botnetze im Speziellen einnehmen. Aktuelle Smartphones haben ein Leistungsniveau erreicht, das als Plattform für mögliche Botnetze vollkommen ausreichend ist. Die Tatsache, dass Smartphones, im Gegensatz zu klassischen Computern, nahezu dauerhaft ans Internet angebunden sind und die bisherige Nicht-Verbreitung von Anti-Viren-Software, machen diese für Botmaster nur noch interessanter.

Mittelfristig wird der einzig wirksame Schutz gegen eine immer stärkere Verbreitung von Botnetzen nur die Steigerung des Sicherheitsbewusstseins aller Nutzer sein.

¹⁶Welchia nutzte die selbe Schwachstelle wie Blaster und entfernte diesen vom System: http://www.symantec.com/security_response/writeup.jsp?docid=2003-081815-2308-99

¹⁷http://www.securelist.com/en/analysis/204792180/TDL4_Top_Bot

¹⁸<http://ddos.arbornetworks.com/2009/08/twitter-based-botnet-command-channel/>

Literaturverzeichnis

- [1] COHEN, F. *Computer viruses - Theory and experiments*, Computers & security Vol. 6, 1987
- [2] LEE, A. AND BUREAU, P.M. *From Fun to Profit - The Evolution of Malware*, Virus Bulletin Conference, 2007
- [3] OIKARINEN, J. AND REED, D. *RFC 1459: Internet Relay Chat Protocol*, 1993
- [4] CANAVAN, J. *The evolution of malicious IRC bots*, Virus Bulletin Conference, 2005
- [5] RAJAB, M. ET AL *A multifaceted approach to understanding the botnet phenomenon*, Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, 2006
- [6] WANG, P. AND SPARKS, S. AND ZOU, C.C. *An advanced hybrid peer-to-peer botnet*, IEEE Transactions on Dependable and Secure Computing, 2010
- [7] ONO, K. AND KAWAISHI, I. AND KAMON, T. *Trend of botnet activities*, 41st Annual IEEE International Carnahan Conference on Security Technology, 2007
- [8] PRAS, A. ET AL *Attacks by "Anonymous" WikiLeaks proponents not anonymous*, 2010
- [9] ZHUGE, J. ET AL *Characterizing the IRC-based botnet phenomenon*, 2007
- [10] PATEL, M. *IRC XDCC - Red Alert for University Administrators*, 2006 IEEE International Conference on Electro/information Technology, 2006
- [11] ARCE, I. AND LEVY, E. *An analysis of the slapper worm*, Security & Privacy, IEEE, 2003
- [12] WILSON, C. *Botnets, cybercrime, and cyberterrorism: Vulnerabilities and policy issues for congress*, DTIC Document, 2008
- [13] GRIZZARD, J.B. ET AL *Peer-to-peer botnets: Overview and case study*, Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, 2007
- [14] FOSSI, M. ET AL *Symantec Internet Security Threat Report - Trends for 2010*, 2011
- [15] SOPHOS *Security Threat Report: 2011*

- [16] COVA, M. AND KRUEGEL, C. AND VIGNA, G. *Detection and analysis of drive-by-download attacks and malicious JavaScript code*, Proceedings of the 19th international conference on World wide web, 2010
- [17] FREI, S. ET AL *Understanding the Web browser threat: Examination of vulnerable online Web browser populations and the "insecurity iceberg"*, 2008
- [18] MESSAGING ANTI-ABUSE WORKING GROUP *Email Metrics Program, Report #15 – First, Second and Third Quarter 2011*, 2011
- [19] WOOD, P. ET AL *Symantec MessageLabs Intelligence: 2010 Annual Security Report*, 2011
- [20] KANICH, C. ET AL *Spamalytics: An empirical analysis of spam marketing conversion*, Proceedings of the 15th ACM conference on Computer and communications security, 2008
- [21] POSTEL, J. *RFC 793: Transmission control protocol*, 1981
- [22] EDDY, W. *RFC 4987: TCP SYN Flooding Attacks and Common Mitigations*, 2007
- [23] DHAMIJA, R. AND TYGAR, J.D. AND HEARST, M. *Why phishing works*, Proceedings of the SIGCHI conference on Human Factors in computing systems, 2006
- [24] PROVOS, N. *A virtual honeypot framework*, Proceedings of the 13th conference on USENIX Security Symposium-Volume 13, 2004
- [25] RIECK, K. ET AL *Botzilla: detecting the phoning home of malicious software*, Proceedings of the 2010 ACM Symposium on Applied Computing, 2010
- [26] MAZZARIELLO, C. *IRC traffic analysis for botnet detection*, Fourth International Conference on Information Assurance and Security, 2008
- [27] STRAYER, W. ET AL *Botnet detection based on network behavior*, Botnet Detection, 2008
- [28] CHOI, H. AND LEE, H. AND LEE, H. AND KIM, H. *Botnet detection by monitoring group activities in DNS traffic*, 7th IEEE International Conference on Computer and Information Technology, 2007
- [29] ZOU, C.C. AND CUNNINGHAM, R. *Honeypot-aware advanced botnet construction and maintenance*, International Conference on Dependable Systems and Networks, 2006
- [30] NICOL, D.M. AND LILJENSTAM, M. *Models of active worm defenses*, Urbana, 2004
- [31] LILJENSTAM, M. AND NICOL, D.M. *Comparing passive and active worm defenses*, Proceedings. First International Conference on the Quantitative Evaluation of Systems, 2004

Kapitel 8

Multicast-Encryption

Christian Marciniak

Multicast ist das Versenden von Nachrichten von einer Quelle an eine bestimmte Gruppe von Empfängern. Dabei wird die Nachricht nur einmal verschickt und erst auf dem Weg repliziert, statt je eine Kopie der Nachricht an die jeweiligen Empfänger zu schicken. Möchte man Daten verschicken, die nicht für jeden zugänglich sein sollen, so müssen diese Nachrichten verschlüsselt werden. Um dies gewährleisten zu können, müssen die Schlüssel zur Ver- und Entschlüsselung möglichst effizient verteilt werden. Beachtet man, dass die Empfängergruppe bei Multicast-Übertragungen u. U. sehr dynamisch ist, so ist die Schlüsselverteilung nicht trivial.

Diese Arbeit gibt zunächst einen Überblick über verschiedene Multicast-Verfahren. Nach einer Vorstellung eines möglichen Anwendungs-Szenarios werden einige Verfahren zur Schlüssel-Verteilung dargestellt und an Hand des Szenarios bewertet.

Inhaltsverzeichnis

8.1	Einleitung	149
8.2	Multicast-Grundlagen	149
8.2.1	Einführung	149
8.2.2	IP-Multicast	151
8.2.3	Overlay-Multicast	152
8.2.4	Hybrid-Multicast	152
8.3	Anforderungen an die Verschlüsselung	152
8.3.1	Szenario	153
8.3.2	Abgeleitete Anforderungen	153
8.4	Lösungsansätze für Multicast-Encryption	154
8.4.1	Überblick über drei grundlegende Verteilungsverfahren	154
8.4.2	Proxy-Encryption	155
8.4.3	L-Layer Encryption Trees	156
8.4.4	Verfahren bei Teil-Kompromittierung	158
8.4.5	Überblick	159
8.5	Zusammenfassung und Ausblick	159

8.1 Einleitung

Multimedia-Anwendungen erlangen eine immer höhere Bedeutung im Internet. Doch trotz stetiger Fortschritte in der Codierungs-Technik erfordern Multimedia-Übertragungen eine hohe Bandbreite. Sollen gleichzeitig viele Empfänger erreicht werden, vervielfacht dies den Bandbreiten-Bedarf der Übertragung. Ein möglicher Weg, den Bandbreiten-Bedarf zu senken, ist die Verwendung von Multicast, wobei nur eine Kopie der Daten ins Netz geschickt wird und sich erst später vervielfältigt, so dass die einzelnen Netzabschnitte nicht mit mehreren Kopien des selben Materials überlastet werden.

Nun gibt es gerade im Bereich der Video-Übermittlung einige Szenarien, bei denen der Empfängerkreis eingeschränkt werden soll.¹ Um sicherzustellen, dass auch tatsächlich nur die gewünschten Empfänger etwas mit dem gesendeten Material anfangen können, wird dieses verschlüsselt. Die Verschlüsselung für sich selbst genommen ist kein schwieriges Problem und bereits gut erforscht. Schwieriger ist es, die Schlüssel für die Entschlüsselung effizient an die Empfänger zu verteilen. Daher beschäftigen sich viele Arbeiten über Multicast-Verschlüsselung genau mit diesem Thema.

Die vorliegende Arbeit gibt zunächst einen Überblick über Multicast-Verfahren in Abschnitt 8.2. Es werden die drei Verfahren IP-Multicast, Overlay-Multicast und Hybrid-Multicast vorgestellt und deren Vor- und Nachteile aufgezeigt. Es folgt in Abschnitt 8.3 eine Beschreibung eines Anwendungsszenarios für verschlüsselte Multicast-Übertragung. Daraus werden Anforderungen allgemeiner Art an Multicast-Verschlüsselungen abgeleitet. Abschnitt 8.4 gibt einen Überblick über einige Verfahren zur Schlüsselverteilung, welches als Hauptproblem der Multicast-Verschlüsselung identifiziert wurde.

8.2 Multicast-Grundlagen

Dieser Abschnitt stellt Grundlagenwissen zu Multicast bereit. Dies umfasst die Gründe für die Entwicklung von Multicast, eine Beschreibung der verschiedenen Ausprägungen von Multicast und eine kurze Erläuterung einiger Multicast-Anwendungen.

8.2.1 Einführung

Die einfachste denkbare Kommunikationsform stellt Unicast dar. Bei dieser Kommunikationsform gibt es genau einen Sender und einen Empfänger (siehe auch Abbildung 8.1). Um eine Nachricht von einem Sender an viele Empfänger zu senden gibt es Broadcast-Verbindungen, bei denen die Nachricht an alle Teilnehmer eines Netzes geschickt wird (siehe Abbildung 8.2). Dies ist vor allem dann nötig, wenn die spezifischen Empfänger-Adressen noch nicht bekannt sind. Ist das der Fall, kann es selbst dann notwendig werden, Nachrichten per Broadcast zu verschicken, wenn nur ein Empfänger erreicht werden soll.

¹Hier ist nicht nur die Einschränkung auf eine bestimmte Gruppe gemeint, sondern auch die Zugriffsverhinderung für Nicht-Gruppenmitglieder.

Nun ist es aber auch denkbar, dass eine Nachricht zwar viele Empfänger haben soll, jedoch die Gruppe der Empfänger eingeschränkt können werden soll. Zu diesem Zweck gibt es Multicast (Abbildung 8.3), bei dem eine Nachricht von einem Sender an eine vorher bestimmte Gruppe von Empfängern gesendet wird.

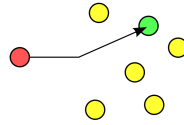


Abbildung 8.1: Unicast - ein Sender, ein Empfänger (aus: [20])

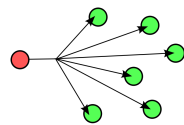


Abbildung 8.2: Broadcast - ein Sender, alle Hosts empfangen (aus: [20])

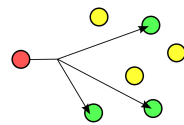


Abbildung 8.3: Multicast - ein Sender, nur eine bestimmte Gruppe empfängt (aus: [20])

Abbildung 8.4 verdeutlicht den Unterschied zwischen Uni- und Multicast. In Teil (a) ist der Übermittlungsweg für mehrere Unicast-Verbindungen dargestellt. Man erkennt gerade bei den quellenahen Routern die hohe Last. In Teil (b) ist Multicast dargestellt, bei dem nur noch eine Kopie auf jeder Teilstrecke verschickt wird.

[11] nennt u. a. Internet Radio bzw. Fernsehen als Anwendung für Multicast. Diese Anwendungen, ebenso wie die in Abschnitt 8.2.3 erwähnte Information-Spreading Anwendung scheinen sehr gut geeignet für Multicast, da sie das klassische Prinzip mit einem Sender und vielen Empfängern perfekt widerspiegeln. [9] nennt als weiteren Anwendungsfall Video-Konferenzen. Bei diesen wechselt zwar der Sender, ist also nicht fix, jedoch gibt es auch hier eine größere Anzahl an Empfängern, die durch eine Gruppe beschrieben werden kann.

Nach [1] gibt es zwei wesentliche Management-Funktionalitäten, die von Multicast-Protokollen abgedeckt werden müssen: Gruppen-Mitgliedschafts-Management und Zustellungspfad-Management. Erstes beschäftigt sich mit der Verwaltung der Mitglieder, also z. B. dem Hinzufügen und Entfernen von Mitgliedern zu bzw. aus der Gruppe. Letzteres beschäftigt sich mit dem Erstellen eines Pfades, der alle vorgesehenen Gruppenmitglieder erreicht. Dies sollte möglichst effizient geschehen.

[1] beschreibt weiterhin Herausforderungen, die das Multicast-Verfahren mit sich bringt:

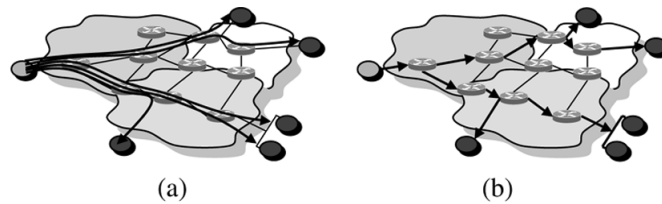


Abbildung 8.4: Unterschied zwischen Unicast und Multicast (aus: [1])

- Konstruktion eines effizienten Verteilungs-Pfades
- Gruppen-Dynamik
- Skalierbarkeit (Anzahl der Sender und Anzahl der Empfänger pro Gruppe)

Die Gruppen-Dynamik ist insofern ein Problem, da im Falle der Nutzung von Verschlüsselung eine hohe Gruppen-Dynamik einen häufigen Schlüsselaustausch bedingt. Dies wird in Abschnitt 8.3 näher erläutert. Die Skalierbarkeit wird durch die Anzahl der Sender bzw. Empfänger pro Gruppe behindert. Auch wenn Multicast grundsätzlich eine 1-zu-n-Kommunikation vorsieht, so ist es teilweise notwendig, mehrere Sender zu haben, so dass verschiedene Verteilungspfade notwendig werden. Eine hohe Anzahl an Empfängern kann, vor allem wenn sie geografisch stark verteilt sind, den Effizienzgewinn durch Multicast zu Nichte machen, da der Verteilungsbaum sehr groß wird.

8.2.2 IP-Multicast

Eine Ausprägung von Multicast ist IP-Multicast. Dabei wird nach [1] und [6] in den Routern zusätzlich zur Unicast-Tabelle eine Multicast-Tabelle angelegt. Pro Link wird nur eine Kopie des Paketes geschickt. Eine Vervielfältigung erfolgt bei entsprechenden Abzweigungen im Pfad. Bei IP-Multicast kann jeder Host Pakete an die Multicast-Gruppe schicken, in dem er sie an die Gruppen-Adresse sendet, welche nach [5] aus dem Bereich 224.0.0.0 - 239.255.255.255 stammt.² Daher kann dieses Verfahren als Empfänger-orientiert bezeichnet werden, da die Empfänger sich bei ihrem lokalen Router anmelden müssen, die Sender aber selbst kein Mitglied der Gruppe sein müssen und auch keine Kenntnis über die Gruppenmitglieder besitzen (müssen).

Diese Variante des Multicast bringt einige Probleme mit sich. Die Multicast-Tabellen sind wesentlich dynamischer als die Unicast-Tabellen, da sich die Mitglieder einer Multicast-Gruppe ändern können. Außerdem sind die Multicast-Tabellen komplexer als ihre Unicast-Pendants, da u. U. für eine Adresse mehr als ein Ausgang gespeichert werden muss. Des Weiteren ist die Implementierung komplex. Probleme ergeben sich u. a. bei der Abrechnung oder der Zugangskontrolle und in der Router-Hardware, die aufwändig angepasst werden muss. Trotz jahrelanger Forschung ist IP-Multicast kaum im Internet verfügbar. Lediglich in einigen Firmen-Netzen finden sich Implementierungen, da hier die Dynamik weniger stark ausgeprägt ist. [1]

²Dabei sind die Adressen im Bereich 224.0.0.0/24 für die Nutzung durch Routing-Protokolle reserviert und sollten laut Anforderung der IANA von Multicast-Routern nicht weitergeleitet werden.

8.2.3 Overlay-Multicast

Während beim IP-Multicast die Router die Verteilungslogik übernehmen, wird beim Overlay-Multicast ein gänzlich anderes Konzept verfolgt. Wie in [1] beschrieben, betreiben die Router nur Unicast-Forwarding. Die Hosts übernehmen die Multicast-Logik. Dabei werden die End-Systeme direkt durch Pfade miteinander verbunden. Ferner verwalten die Einzelsysteme nur Gruppen, in denen sie selbst Mitglied sind, was den Verwaltungsaufwand für den einzelnen Host deutlich reduziert.

In [1] werden drei Arten von Overlay-Multicast-Verfahren beschrieben: Dedizierte Infrastruktur, Applikations-Endpunkt und Wegpunkt. Beim erstgenannten Verfahren schließen sich die Endsysteme an spezifizierte Hosts an, welche am Rand des Netzes platziert sind. Diese Hosts übernehmen dann die Multicast-Logik und arbeiten ähnlich wie IP-Multicast-Router. Da die Infrastruktur hierbei ähnlich zum IP-Multicast ist, sind auch die Probleme ähnlich. Die zweite Variante stellt das Applikations-Endpunkt-Verfahren dar, welches auch in [7] beschrieben wird. Hier übernehmen die normalen End-Hosts sämtliche Funktionalität. Das bedeutet, dass es keine dedizierten Geräte für den Multicast mehr gibt, sondern alles auf den End-Systemen verarbeitet wird. Diese Variante skaliert gut, da die verfügbare Bandbreite mit der Anzahl der Hosts im System zunimmt, jedoch bildet sich ein Flaschenhals bei der Upload-Bandbreite der Hosts, da diese häufig gering ist. Das Wegpunkt-Verfahren ist eine Mischung aus beiden Verfahren, bei dem Wegpunkte einer Gruppe hinzugefügt oder auch von ihr entfernt werden können, die dann auf einem bestimmten Abschnitt das Forwarding übernehmen. Die Wegpunkte (Router) übernehmen eine ähnliche Funktion wie beim IP-Multicast für einen bestimmten Streckenabschnitt. Die Netze, welche über keine Wegpunkte verfügen, bewerkstelligen die Nachrichtenweiterleitung über die Hosts wie beim Applikations-Endpunkt-Verfahren. Nach [2] wird Overlay-Multicast vor allem in Streaming Media oder Information Spreading Anwendungen eingesetzt, um die Last der Server zu verringern.

8.2.4 Hybrid-Multicast

Beide bisher vorgestellten Ansätze haben ihre Vorteile, aber auch Nachteile. Wie häufig kommt man zu dem Schluss, dass eine Verbindung beider Ansätze sinnvoll erscheint; genau das leistet Hybrid-Multicast. Hierbei wird IP-Multicast in abgeschotteten Insel-Netzen verwendet, die wiederum über Overlay-Multicast-Tunnel miteinander verbunden werden. Dies reduziert die Auswirkungen der Gruppendynamik, da Mitgliedschaftsänderungen u. U. nur eine Insel anstatt das gesamte System betreffen. Als Problem ergibt sich jedoch, dass nun Protokolle aus beiden Welten verwendet werden müssen, was die Komplexität dieser Lösung erhöht. [1]

8.3 Anforderungen an die Verschlüsselung

Dieser Abschnitt beschreibt ein Szenario, in dem verschlüsselte Multicast-Ströme notwendig sind. Aus diesem Szenario werden im Anschluss Anforderungen an die Verschlüsselung

abgeleitet, die zur besseren Einordnung der in Abschnitt 8.4 vorgestellten Lösungen dienen sollen.

8.3.1 Szenario

In dieser Arbeit wollen wir einen Pay-TV-Anbieter betrachten, der sein Angebot über das Internet bereitstellt. Technische Details zur Umsetzung spielen hier insofern keine Rolle, da es lediglich um einen möglichen Anwendungsfall geht, der demonstriert, worauf bei der Verschlüsselung von Multicast-Strömen zu achten ist.

Dieser Pay-TV-Anbieter hat einen festen Kundenstamm. Jedoch kommt es vor bzw. nach Großereignissen, wie z. B. einer Fußball-Weltmeisterschaft, zu vielen Neu-Abonnements bzw. danach wieder zu Kündigungen derjenigen Verträge, die solch eine Flexibilität mit ihrer geringen Laufzeit ermöglichen. Regelmäßig erfolgt eine Veränderung des Kundenstamms zum Jahreswechsel, da viele Verträge Geschenk-Verträge sind, die zum 1. Januar starten und ein Jahr Gültigkeit haben.

Der Anbieter möchte seine Daten möglichst effizient verschicken und nutzt daher Multicast. Damit niemand, der keinen Vertrag mit ihm abgeschlossen hat, in der Lage ist, das Fernseh-Programm zu nutzen, verschlüsselt er die Daten.

8.3.2 Abgeleitete Anforderungen

[8] benennt Anforderungen, die auch für unseren Pay-TV-Anbieter wesentlich sind: Forward und Backward Secrecy. Erste meint, dass Mitglieder, welche die Multicast-Gruppe verlassen, nicht in der Lage sein dürfen, in der Zukunft versandte Pakete zu entschlüsseln. Analog meint Backward Secrecy, dass ein Mitglied, welches neu zur Gruppe stößt, den Inhalt der vergangenen Kommunikation nicht entschlüsseln darf. Gerade für den Pay-TV-Anbieter sind diese beiden Eigenschaften wesentlich, da sonst seine Kunden nur für einen kurzen Zeitraum einen Vertrag abschließen müssten und anschließend jede jemals gesendete oder in Zukunft gesendete Übertragung mit ansehen könnten - der sichere Ruin seines Unternehmens.

Daraus ergibt sich die Notwendigkeit eines so genannten Re-Keying-Mechanismus. Dieser sorgt für eine neue Verteilung der Schlüssel zur Entschlüsselung der Kommunikation. Dieser muss laut [8] skalierbar sein, was in diesem Fall Folgendes bedeutet: Performance (Geschwindigkeit des Algorithmus) und Betrachtung der Mitglieder in Gruppen (keine Betrachtung des einzelnen Gruppenmitglieds, sondern Abstraktion). [9] nennt weiterhin die Anzahl der Schlüssel in Servern und Clients, Anzahl der Re-Key-Nachrichten sowie die Anzahl der generierten Schlüssel bei einem Mitgliedschaftswechsel als Kriterien für Skalierbarkeit. Skalierbarkeit ist für den Pay-TV-Anbieter notwendig, da seine Server sonst mit der Neu-Berechnung von Schlüsseln überlastet werden würden und er viel Traffic mit der Neu-Vergabe von Schlüsseln erzeugen würde.

Gesetzt den Fall, dass unser Pay-TV-Anbieter viele Kunden hat, so kann es vorkommen, dass ein Schlüssel an potentielle Angreifer fällt. Der Verlust dieses einen Schlüssels darf

jedoch nicht dazu führen, dass die gesamte Kommunikation innerhalb der Gruppe unsicher wird. Diese Anforderung wird Containment genannt und ist ebenfalls in [8] genannt.

[12] nennt weiterhin Authentizität als Kriterium. Auch dies ist für den Pay-TV-Anbieter wichtig, da seine Kunden sicherstellen müssen, dass die Nachrichten von seinen Servern kommen und nicht von Angreifern verschickt wurden. Idealerweise möchte auch der Anbieter sicherstellen können, dass Vertragsänderungen auch wirklich von ihnen initiiert wurden.

8.4 Lösungsansätze für Multicast-Encryption

Dieser Abschnitt skizziert diverse Lösungsansätze zur Multicast-Encryption. Dabei werden die unterschiedlichen Verfahren vorgestellt und verglichen. Die Verfahren werden nach den Anforderungen, welche in Abschnitt 8.3.2 vorgestellt wurden, bewertet.

Das grundsätzliche Problem bei der Multicast-Verschlüsselung besteht dabei nicht in der Verschlüsselung an sich; damit beschäftigen sich unzählige hier nicht betrachtete Arbeiten. Das grundlegende Problem ist die effiziente und sichere Schlüsselverwaltung zur Entschlüsselung der Daten. Diese Schlüsselverwaltung darf nicht zu aufwändig sein, muss aber trotzdem Secrecy gewährleisten. Um überhaupt Schlüssel verteilen zu können, müssen also die Empfänger der jeweiligen Nachrichten vorher bekannt sein, anders als es z. B. bei IP-Multicast der Fall ist. Daher werden hier verschiedene Verfahren zur Schlüsselverteilung vorgestellt.

8.4.1 Überblick über drei grundlegende Verteilungsverfahren

[18] nennt drei verschiedene Verfahren zur Verteilung der Schlüssel. Zum einen gibt es die Möglichkeit, Schlüssel manuell zu verteilen. Diese Lösung skaliert aus offensichtlichen Gründen nicht. Dennoch ist sie u. U. in militärischen Kontexten einsetzbar, wenn vorher alle Geräte bekannt sind. Der Vorteil liegt darin, dass kein Rechenaufwand bei der Schlüsselverteilung entsteht, sondern die Schlüssel nur ein einziges Mal fest vergeben werden müssen. Forward bzw. Backward Secrecy und Authentizität spielen bei dieser Methode keine Rolle.³ Containment ist nicht gegeben, da alle Mitglieder über den gleichen Schlüssel verfügen müssen.

Das Verfahren des paarweisen Verschlüsseln läuft wie folgt ab: Zunächst einigt sich die Quelle mit jedem Empfänger auf einen anderen Schlüssel. Mit diesen Schlüsseln wird dann der Gruppenschlüssel verschlüsselt, so dass nach dem Austausch des Gruppenschlüssels alle Empfänger die Multicast-Nachrichten entschlüsseln können. Dieses Verfahren benötigt einen geringen Rechenaufwand, vor allem dann, wenn die Anzahl der Empfänger konstant bleibt. Bei häufigen Mitglieder-Wechseln muss jedoch immer mit der gesamten Gruppe ein neuer Schlüssel ausgehandelt werden, was zu erhöhter Netzlast führt. [13] bezeichnet

³Genauer gesagt werden sie durch den Schlüsselverwalter sichergestellt, der die Schlüssel manuell vergibt.

dieses Verfahren als Stern-Graph. Beim Hinzufügen eines neuen Empfängers muss ein neuer Gruppenschlüssel generiert werden, der an die alte Gruppe verschlüsselt mit dem alten Gruppenschlüssel geschickt wird. Wird ein Mitglied aus der Gruppe der Empfänger gelöscht, so wird ebenfalls ein neuer Schlüssel generiert. Dieser wird dann per Unicast, mit den individuellen Schlüsseln jedes Mitglieds verschlüsselt, an die verbliebenen Mitglieder geschickt. Forward und Backward Secrecy sind durch den Re-Keying-Mechanismus bei Veränderung der Mitgliederzahl gegeben. Containment ist ebenfalls nicht gegeben, da alle Mitglieder den gleichen Gruppenschlüssel besitzen. Über Authentizität werden keine Aussagen getroffen.

Als drittes Verfahren werden hierarchische Bäume in [18] vorgestellt. Dabei werden entlang der Knoten-Ebenen verschiedene Schlüssel verteilt, mit denen der Gruppenschlüssel verschlüsselt wird. Dies hat den Vorteil, dass im Falle einer Kompromittierung eines Teils des Baums, nicht alle Mitglieder einen neuen Schlüssel benötigen, sondern nur die betroffenen Teile, deren Schlüsselanteile beschädigt wurden. Dies führt zu einer verbesserten Skalierung des Verfahrens. Allerdings erzeugt dieses Verfahren auch die größte Rechenlast beim Schlüssel verwaltenden System. Auch dieses Verfahren wird in [13] genannt. Beim Hinzufügen eines Mitglieds muss hierbei ein neuer Gruppenschlüssel generiert werden. Außerdem ist ein neuer Sub-Gruppen-Schlüssel nötig für die Teilgruppe, in der das neue Mitglied aufgehängt wird. Beim Löschen müssen entlang des Graphen bis hin zur Wurzel neue Sub-Gruppen-Schlüssel und anschließend ein neuer Gesamt-Schlüssel verteilt werden. Auch hier ist Containment wieder nicht gegeben, da wieder ein einzelner Gruppenschlüssel verwendet wird. Authentizität wird in den Beschreibungen des Verfahrens nicht betrachtet. Forward und Backward Secrecy sind wieder durch den Re-Keying-Mechanismus gegeben.

8.4.2 Proxy-Encryption

Der im Folgenden beschriebene Lösungsansatz stammt aus [19]. Die grundsätzliche Idee ist, die Rechenlast zur Verschlüsselung an die Zwischen-Knoten zu verteilen, um eine möglichst gute Skalierbarkeit zu erzeugen. Dies wäre z. B. möglich, indem jeder Zwischenknoten die Nachricht entschlüsselt und anschließend wieder neu verschlüsselt. Dabei müsste jedoch jedem Zwischen-Knoten vertraut werden, was eine unrealistische Anforderung darstellt.

Daher wählt man in dem vorgestellten Schema einen anderen Weg: Jeder Zwischenknoten wendet den Algorithmus mit einem so genannten Segment-Schlüssel, welcher von einem Trusted Server generiert und verteilt wird, an. Dabei wird also aus einer verschlüsselten Nachricht eine neue verschlüsselte Nachricht erzeugt, ohne dass zwischenzeitlich der Klartext vorgelegen hat. Durch eine Abwandlung des El-Gamal-Verschlüsselungsverfahrens wird sichergestellt, dass am Ende die Nachricht entschlüsselt werden kann.

Abbildung 8.5 gibt beispielhaft den Ablauf des Verfahrens an. Im Folgenden werden kurz die wesentlichen Elemente beschrieben, ohne auf die Feinheiten des Verschlüsselungsalgorithmus einzugehen. S verschlüsselt die Nachricht mit Hilfe eines Generators g , einer Nonce⁴ r und des ersten Segmentschlüssels S_1 zu mg^{S_1r} und verschickt diese an R_1 . R_1

⁴Number used once, eine zufällig gewählte Zahl

verschlüsselt mit $S_1 - S_2$, so dass mg^{S_2r} an R_2 geschickt werden kann. Dieser verschlüsselt nun analog mit $S_1 - S_M$, so dass $mg^{S_M r}$ und sendet die Nachricht an den Empfänger M . M kann die Nachricht mit Hilfe von S_M entschlüsseln und erhält den Klartext m .

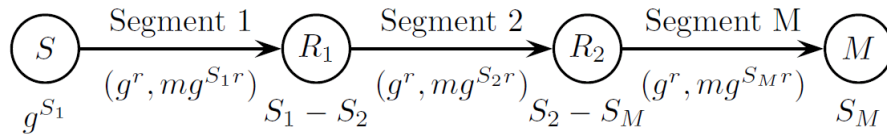


Abbildung 8.5: Schematische Darstellung des Proxy-Verfahrens (aus: [19])

Eine wichtige Annahme, die getroffen werden muss, damit das Verfahren funktioniert, ist die, dass die Router / Zwischenknoten die Daten korrekt weiterverarbeiten und neu verschlüsseln. Verarbeiten die Router die Daten nicht korrekt, ist das Verfahren wirkungslos und es kommen keine brauchbaren Daten beim Empfänger an.

Forward und Backward Secrecy sind bei diesem Verfahren gegeben, da ein Re-Keying-Mechanismus eingesetzt wird, welcher die relevanten Schlüssel austauscht, sobald ein Mitglied die Gruppe verlässt oder neu beitrifft. Dieser Mechanismus wird von den Autoren zusätzlich als effizient beschrieben. Containment wird gewährleistet, da die Entschlüsselung der Nachricht nur am Ende möglich ist. Nur der letzte Router besitzt die notwendigen Informationen, um die Nachricht für seine lokale Sub-Gruppe in Klartext umzuwandeln. Über Authentizität wird im vorliegenden Paper keine Aussage gemacht.

8.4.3 L-Layer Encryption Trees

Ähnlich wie das Proxy-Verfahren arbeiten auch die L-Layer Encryption Trees in [8]. Auch hier bekommt jeder Zwischenknoten wieder einen spezifischen Zwischenschlüssel und verändert die Verschlüsselung, so dass nur die Blätter am Ende des Baums die Daten entschlüsseln können. Allerdings ist der Algorithmus ein anderer. Jeder Zwischenknoten bekommt dabei einen Schlüssel zugewiesen und den Schlüssel des l-ten Elternknotens. Jedes Blatt bekommt alle Schlüssel der Elternknoten von Level L bis 1. Sollte ein Blatt nicht erst auf Ebene L im Baum, sondern schon früher auftauchen, so bekommt es die fehlenden Schlüssel von der Quelle zugewiesen.

Ein beispielhafter Baum ist in Abbildung 8.6 dargestellt. Das linke Blatt (in der Abbildung LEAF genannt) befindet sich auf Ebene 2 und erhält somit K_5 vom Elternknoten. Da es keine weiteren Eltern gibt, bekommt es die fehlenden drei Schlüssel K_4 , K_3 , und K_2 direkt von der Quelle. Anders verhält es sich mit dem rechten Blatt. es hat vier Eltern-Knoten und bekommt nur deren Schlüssel K_{13} , K_{12} , K_{10} und K_7 zugewiesen.

Die Quelle verschlüsselt die Daten mit L (entsprechend der Anzahl der Ebenen im Baum) Schlüsseln. Jeder Zwischenknoten entschlüsselt die Daten mit einem Schlüssel und verschlüsselt die Daten mit einem neuen Schlüssel (in der Grafik werden die oberen Schlüssel zum Ent- und die unteren zum Verschlüsseln benutzt). Die Blätter des Baums (die Empfänger der Daten) nutzen ebenfalls L Schlüssel, um die Daten wieder zu entschlüsseln. Als Verschlüsselungsalgorithmus kommen in der Regel Block-Chiffren zum Einsatz.

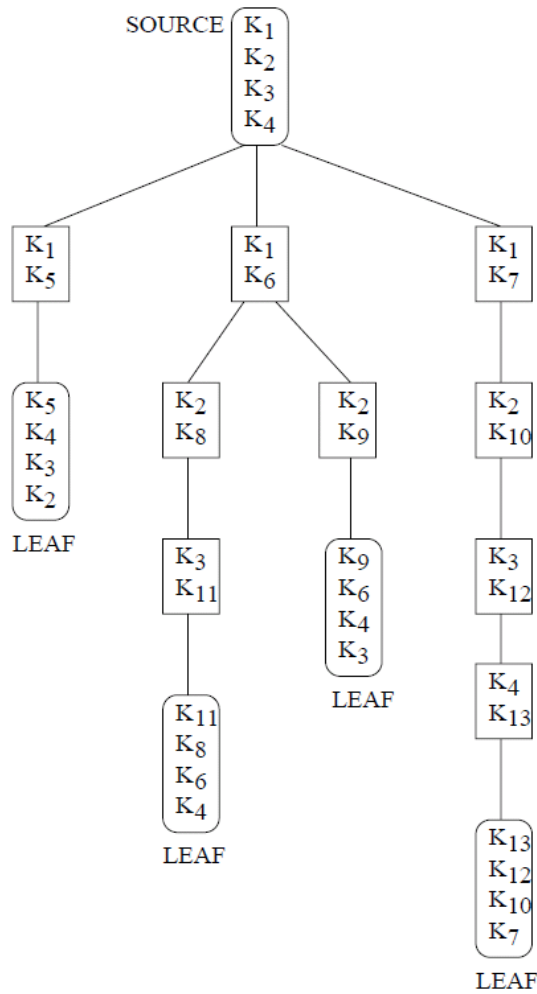


Abbildung 8.6: Beispielhafter 4-Layer Encryption Tree (aus: [8])

Allerdings sind auch andere Algorithmen möglich, die das Verschlüsseln mit mehreren Schlüsseln und anschließende Entschlüsseln über verschiedene Etappen zulassen. Durch die Mehrfach-Verschlüsselung am Anfang ist trotz des Entschlüsselns mit einem Schlüssel in jedem Knoten sichergestellt, dass die Original-Nachricht ausschließlich bei den Blättern des Baums als Klartext vorliegt.

Da die Blätter des Baumes nicht nur einzelne End-Rechner, sondern Teil-Netze, die lokal verwaltet werden, darstellen können, läuft das Hinzufügen eines Knotens wie folgt ab: Wenn das neue End-System bereits an ein Blatt physikalisch angeschlossen ist, muss nur der Verschlüsselungs-Schlüssel des Elternknotens geändert werden und die empfangenen Daten müssen fortan auch an das neue End-System geschickt werden. Wenn das End-System noch nicht in einem Blatt ist, dann muss der Baum expandiert werden. Analog läuft das Entfernen von Mitgliedern ab: Entweder muss nur der Schlüssel geändert werden und die Daten werden nicht mehr an das End-System geschickt oder der betreffende Ast des Baumes kann abgeschnitten werden.

Forward und Backward Secrecy sind durch den Re-Keying-Mechanismus gegeben. Dabei werden tendenziell wenig Re-Keying-Nachrichten verschickt und auch die Anzahl der

neuen Schlüssel ist gering. Containment ist ebenfalls gewährleistet, da nur die Gesamtheit aller Schlüssel entlang eines Pfades im Baum zu einer Entschlüsselung der Nachricht führt. Auch dieses Verfahren trifft keine Aussagen über Authentizität.

8.4.4 Verfahren bei Teil-Kompromittierung

[16] beschreibt ein Szenario ähnlich zu dem, welches in Abschnitt 8.3.1 genannt wurde. Allerdings verweist die Arbeit darauf, dass gerade im Pay-TV-Bereich häufig die so genannten Smartcards, also Karten, die die Schlüssel zur Entschlüsselung des Programms enthalten, kompromittiert werden und somit Decoder erstellt werden, die, ohne dass der Anbieter es genehmigt, die Programme entschlüsseln können. Das vorliegende Paper beschreibt eine Methodik, auch in diesem Fall die Sicherheit der Übertragung in möglichst großem Rahmen zu gewährleisten.

Grundsätzlich wird davon ausgegangen, dass jeder Nutzer eine Smartcard mit mehreren Schlüsseln besitzt. Wenn nun so genannte Pirate-Decoder, also solche, die mit nicht autorisierten Schlüsseln laufen, erkannt werden, gelten ab sofort diese dort genutzten Schlüssel als ungültig. Aus Effizienzgründen werden Schlüssel auf verschiedenen Karten mehrfach verwendet, so dass ab dem Zeitpunkt der Entdeckung von illegalen Decodern nicht mehr alle Schlüssel auf der Karte eines legalen Nutzers verwendbar sind. Ebenso gelten Schlüssel als ungültig, wenn der jeweilige Vertrag vom Kunden mit dem Anbieter ausläuft. Wenn die Anzahl der nutzbaren Schlüssel eine gewisse Grenze unterschreitet, muss die Smartcard ausgetauscht werden.

Teilt man die Zeitachse in gleiche Abschnitte, so wird davon ausgegangen, dass durchschnittlich eine gleiche Anzahl von Karten kompromittiert wird. Da dies aber nicht alle Nutzer betrifft, müssen nur diejenigen mit neuen Karten versorgt werden, welche einen hohen Prozentsatz von nicht mehr nutzbaren Schlüsseln auf ihren Karten haben. Der Rest kann auch im nächsten Zeitabschnitt die alten Schlüssel behalten.

Experimente der Autoren zeigen, dass es für den Pay-TV-Anbieter am günstigsten ist, möglichst viele Schlüssel vorzuhalten. Dies führt zu einer geringeren Austauschrate der Karten. Je mehr Schlüssel jedoch auf einer einzelnen Karte gespeichert werden, umso häufiger müssen die Karten ersetzt werden, da von kompromittierten Schlüsseln mehrere Karten gleichzeitig betroffen sind. Daher sollten möglichst wenig Schlüssel auf einer Karte gespeichert werden.⁵ Je mehr kompromittierte Karten man zulässt, bevor man die Karten austauscht, umso weniger Karten müssen ausgetauscht werden. Allerdings kann dies zu unsicheren Übertragungen führen. Hier ist der Pay-TV-Anbieter gefragt, ein möglichst gutes Mittelmaß zu finden.

Forward und Backward Secrecy sind hier nicht zu 100% gegeben. Dies liegt daran, dass auf den Smartcards mehrere Schlüssel gespeichert sind. Dadurch kann es vorkommen, dass ein Nutzer mit einer bereits abgelaufenen Smartcard noch für einen kurzen Zeitraum⁶

⁵Ideal wäre hier ein Schlüssel pro Karte, da dann ein kompromittierter Schlüssel keine weiteren Auswirkungen hätte. Jedoch führt dies zu einem großen Effizienzverlust, da deutlich mehr Schlüssel generiert und verwaltet werden müssen.

⁶Also bis sein Schlüssel für ungültig erklärt wird.

noch das Programm empfangen kann. Aus einem ähnlichen Grund ist auch Containment nicht gegeben, da man es, um Kosten zu sparen, billigend in Kauf nimmt, dass zeitweise nicht-autorisierte Benutzer das Programm empfangen können. Backward Secrecy spielt in diesem Szenario keine Rolle, da die Nachrichten bereits alle verschickt wurden. Nachrichten werden zum Re-Keying in diesem Szenario nicht verschickt, sondern gleich ganze Smartcards. Über die Performance in Bezug zu einzelnen Parametern wurde bereits oben gesprochen. Die Authentizität der Vertragspartner wird hier über externe Verfahren, wie sie unter Handelspartnern üblich sind, sichergestellt.

8.4.5 Überblick

Tabelle 8.1 gibt einen Überblick, inwieweit die vorgestellten Verfahren die Anforderungen aus Abschnitt 8.3.2 umsetzen.

8.5 Zusammenfassung und Ausblick

Diese Arbeit hat zunächst verschiedene Multicast-Verfahren vorgestellt. Dabei ist IP-Multicast das effizienteste der vorgestellten. Allerdings müssen, um das Verfahren nutzen zu können, Anpassungen im Kernnetz vorgenommen werden, die aufwändig und teuer sind. Overlay-Multicast ist für einzelne Dienstanbieter einfach zu implementieren, muss dafür Performance-Einbußen hinnehmen. Wie so oft stellt der Mittelweg, also Hybrid-Multicast eine gute Lösung dar, um die Vorteile aus beiden Welten miteinander zu kombinieren. Dabei hängt die Einsatzmöglichkeit von Hybrid-Multicast jedoch von den Netz-Voraussetzungen ab. Das heißt, die Empfänger müssen nahe an Multicast-Routern liegen, da sonst der Performance-Gewinn gegenüber reinem Overlay-Multicast zu gering ist.

Anschließend wurden einige Schlüsselverteilungsverfahren für Multicast-Verfahren untersucht. Dabei wurde deutlich, dass keines der Verfahren alle Anforderungen, welche in Abschnitt 8.3.2 erarbeitet wurden, erfüllt. Für Kontexte, in denen die Empfänger vorher bekannt sind, sich nicht ändern und von einer zentralen Stelle verwaltet werden, kann mit manueller Schlüsselverteilung gearbeitet werden. Dies ist z. B. in militärischen Kontexten der Fall. Abseits von diesem sehr eingeschränkten Nutzungsfeld stellen Schlüssel-Bäume das Mittel der Wahl dar. Dabei bieten die L-Layer-Encryption und das Proxy-Verfahren Möglichkeiten, die Schlüssel effizient zu verteilen. Daher sollten diese beiden Varianten bei einer anstehenden Implementierung in Betracht gezogen werden. Der beschriebene Smartcard-Austausch ist für sensible Daten nicht anwendbar, da man billigend in Kauf nimmt, dass Daten an ungewollte Empfänger übertragen werden. Jedoch stellt er eine wirtschaftlich interessante Alternative dar, wenn dieser Kritikpunkt keine großen Auswirkungen hat.

Weitere Arbeiten sollten eine genauere Untersuchung der praktischen Anwendbarkeit der vorgestellten bzw. favorisierten Lösungsansätze durchführen. Dabei ist vor allem bei der L-Layer-Encryption und beim Proxy-Verfahren zu untersuchen, ob die Zwischenknoten der Verschlüsselungsalgorithmus verlässlich anwenden und wie einfach es ist, diesen Schritt

zu stören - sei es, um an die Daten zu gelangen oder einfach nur die Daten unbrauchbar zu machen. Dieser Schritt stellt offensichtlich eine Schwachstelle in den beiden Ansätzen dar. Des Weiteren ist zu prüfen, wie viel Aufwand die Implementierung dieser Verfahren in den Zwischenknoten darstellt und mit welchen Übertragungsverfahren, sprich IP- oder Overlay-Multicast, sie sich kombinieren lassen. Dabei scheint die Verwendung von Overlay-Multicast fast zwingend zu sein, da beim IP-Multicast die Router das rechenintensive Ver- und Entschlüsseln übernehmen müssten, was ihre Leistungsfähigkeit deutlich beeinträchtigen würde.

Tabelle 8.1: Gegenüberstellung der Lösungsansätze

Anforderung	Manuell	Paarweise	Hierarchische Bäume	Proxy-Encryption	L-Layer Encryption	Trees	Smartcards
Secrecy	unnötig	ja	ja	ja	ja	ja	teilweise
Re-Keying	n.a.	aufwändig	mittelmäßig	effizient	effizient	effizient	n.a.
Containment	nein	nein	nein	ja	ja	ja	nein
Authentizität	unnötig	n.a.	n.a.	n.a.	n.a.	n.a.	ja

Literaturverzeichnis

- [1] GANJAM, A., ZHANG, H. *Internet Multicast Video Delivery*, Proceedings of the IEEE 2005, Vol. 93, No. 1, 2005, S. 159-170
- [2] NÜRNBERGER, S. *Overlay-Multicast*, Seminararbeit, TU Cottbus, WS 2005/2006, Januar 2006
- [3] DREO, G. *Rechnernetze - Kapitel 4: Internet-Protokolle für Multimedia-Anwendungen*, Vorlesungsfolien, Wintertrimester 2011, Universität der Bundeswehr München
- [4] MOEN, D. *Overview of Overlay Multicast Protocols*, George Mason University, C3I Center, Dezember 2004
- [5] *IPv4 Multicast Address Space Registry*, <http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xml>, zuletzt abgerufen am 22.11.2011
- [6] DEERING, S. *Multicast routing in internetworks and extended LANs* in Proc. ACM SIGCOMM, Aug. 1988
- [7] CHU, Y., RAO, S., ZHANG, H. *A case for end system multicast* in Proc. ACM Sigmetrics, Juni 2000
- [8] PANNETRAT, A., MOLVA, R. *Multiple Layer Encryption for Multicast Groups*, Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security, 2002
- [9] CHAN, KIN-CHING, CHAN, S.-H. GARY *Key Management Approaches to Offer Data Confidentiality for Secure Multicast*, Network, IEEE, vol.17, no.5, Seiten 30-39, Sept.-Okt. 2003
- [10] DONDETI, L., MUKHERJEE, S., SAMAL, A. *A Dual Encryption Protocol for Scalable Secure Multicasting*, Computers and Communications, 1999. Proceedings. IEEE International Symposium on, Seiten 2-8, 1999
- [11] PESSI, P. *Secure Multicast*, <http://www.tml.tkk.fi/Opinnot/Tik-110.501/1995/multicast.html> (Online-Paper), zuletzt abgerufen am 20.10.2011

- [12] CANETTI, R., GARAY, J., ITKIS, G., MICCIANCIO, D., NAOR, M., PINKAS, B. *Multicast Security: A Taxonomy and Some Efficient Constructions*, INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , vol.2, Seiten 708-716 vol.2, 21.-25. März 1999
- [13] WONG, C. K., GOUDA, M., LAM, S. S. *Secure Group Communications Using Key Graphs*, IEEE / ACM Transactions on Networking, Vol. 8, No. 1, Februar 2000, Seiten 16-30
- [14] MICCIANCIO, D., PANJWANI, S. *Multicast Encryption: How to maintain secrecy in large, dynamic groups?*, <http://cseweb.ucsd.edu/~spanjwan/multicast.html>, zuletzt abgerufen am 20.10.2011
- [15] JIA, H., CHEN, Y., MAO, X., DOU, R. *Efficient and Scalable Multicast Key Management Using Attribute Based Encryption*, Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on, Seiten 426-429, 17.-19. Dez. 2010
- [16] GARAY, J., STADDON, J., WOOL, A. *Long-Lived Broadcast Encryption*, in: M. Bellare (Ed.): CRYPTO 2000, LNCS 1880, Seiten 333-352, Springer-Verlag Berlin Heidelberg 2000
- [17] WEILER, N. *SEMSOMM - A Scalable Multiple Encryption Scheme for One-To-Many Multicast*, Enabling Technologies: Infrastructure for Collaborative Enterprises, 2001. WET ICE 2001. Proceedings. Tenth IEEE International Workshops on, Seiten 231-236, 2001
- [18] KRUIUS, P. *A Survey of Multicast Security Issues and Architectures*, NAVAL RESEARCH LAB WASHINGTON DC, 1994
- [19] CHIU, YUN-PENG, LEI, CHIN-LAUNG, HUANG, CHUN-YING *Secure Multicast Using Proxy Encryption*, in: S. Qing et al. (Eds.): ICICS 2005, LNCS 3783, Seiten 280-290, Springer-Verlag Berlin Heidelberg 2005
- [20] BILDER AUS WIKIPEDIA <http://upload.wikimedia.org/wikipedia/commons/7/75/Unicast.svg>, <http://upload.wikimedia.org/wikipedia/commons/d/dc/Broadcast.svg> und <http://upload.wikimedia.org/wikipedia/commons/3/30/Multicast.svg>, zuletzt abgerufen am 13.12.2011

Kapitel 9

Social VPN: Vergleich verschiedener VPN-P2P Ansätze

Marcel Bassüner

Diese Ausarbeitung beschäftigt sich mit dem Thema "Social VPN". Dabei geht es um VPN gesicherte P2P Netze, auf Grundlage von Freundschaftsbeziehungen aus sozialen Netzwerken. Dabei ist hervorzuheben, dass es sich um ein vorkonfigurierendes Konzept handeln soll welches einen gesicherten Datenaustausch ermöglicht. VPN und P2P Netze werden als Grundlagen kurz behandelt. Anschließend werden mögliche Ansätze zur Realisierung vorgestellt. Letztlich wird ein kurzer Überblick zu aktuellen Vertretern der Thematik gegeben.

Inhaltsverzeichnis

9.1	Einleitung	167
9.2	Grundlagen	168
9.2.1	Peer-to-Peer Verbindungen	168
9.2.2	VPN	171
9.3	VPN-P2P Ansätze	172
9.3.1	Zentralisiertes Modell	172
9.3.2	Semi-Zentrales Modell	173
9.3.3	Dezentrales Modell	173
9.4	Vertreter	175
9.4.1	Hamachi	175
9.4.2	Wippien	175
9.4.3	SocialVPN	176

9.1 Einleitung

Viele Menschen besitzen heutzutage einen Account in einem sozialen Netzwerk. Diese Netzwerke werden genutzt um neue Leute kennenzulernen oder Freundschaften über weite Entfernungen aufrecht zu erhalten. Zusätzlich bieten diese Plattformen die Möglichkeit zum Datenaustausch. So kann ein Nutzer zum Beispiel Texte bzw. Nachrichten, aber auch Fotos und Videos, mit seinen Freunden teilen.



Abbildung 9.1: Soziale Netzwerke[1]

Doch warum Fotos umständlich auf den Server eines sozialen Netzwerkes hochladen, wenn sie doch bereits alle sortiert in Ordnern auf der Festplatte liegen, ebenso wie Texte, Tabellen und andere Dokumente?



Abbildung 9.2: Geteilte Bilder auf Facebook

Diesen Gedanken greift das Thema "Social VPN" auf. Hierbei soll die Möglichkeit geschaffen werden, den Freunden auf einer sozialen Plattform Zugriff auf freigegebene Daten zu verschaffen, als ob sie sich im selben lokalen Netzwerk befinden würden. Zusätzlich werden natürlich Bedingungen an diese Verbindung gestellt. So muss der Zugriff gesichert und

verschlüsselt geschehen, damit keine unautorisierten Personen Zugang zu sensiblen Daten erhalten. Die Einrichtung des Systems soll so unkompliziert sein, wie die Verwendung einer sozialen Plattform selbst. Wenn diese Anforderungen gewährleistet sind, steht dem Datenaustausch nichts mehr im Weg.

9.2 Grundlagen

Im folgenden Kapitel werden die Grundlagen des Themas "Social VPN" behandelt. Dabei wird genauer beschrieben wie die einzelnen Mechanismen funktionieren und welche verschiedenen Möglichkeiten der Umsetzung es gibt. Dies soll in erster Linie dazu dienen einen gemeinsamen Wissenstand zu schaffen und das Interesse an der Problemstellung wecken.

9.2.1 Peer-to-Peer Verbindungen

Bei P2P-Verbindungen handelt es sich um eine selbst organisierende Struktur eines Rechnernetzes, welche im Gegensatz zum Client-Server Ansatz steht.[4]

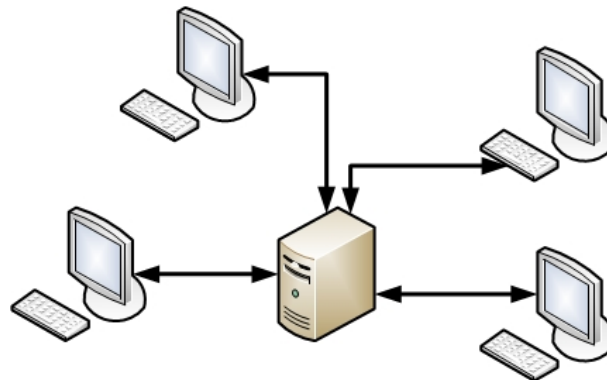


Abbildung 9.3: Server-Client Netzwerk

Eine hierarchische Ordnung wird nicht verwendet. "Peer" bedeutet aus dem englischen Übersetzt: "Gleichgestellter" oder "Ebenbürtiger". Dies definiert auch die Struktur des Netzes selbst. Hierbei ist jeder Teilnehmer gleichgestellt. Jeder stellt sowohl Server als auch Client dar, dass heißt es können sowohl Dienste angeboten als auch genutzt werden.

P2P-Netzwerke bilden ein Overlay-Netzwerk zum Internet. Es werden vorhandene Infrastrukturen genutzt. Jedoch verbinden sich die Peers unabhängig von einander und verwenden zum Teil eigene Adressbereiche. Hintergrund für die Verwendung von P2P Netzen ist zumeist die der Austausch von Daten und Ressourcen unter den Teilnehmern. In diesen Zusammenhang sind P2P Netze, wie BitTorrent, Napster oder Gnutella bekannt und weit verbreitet. Im Aufbau von diesen Netzen gibt es je nach Anwendungsgebiet einige Unterschiede. Entsprechend dieser Unterschiede existieren Klassifizierungen, welche im Folgenden beschrieben werden.[2]

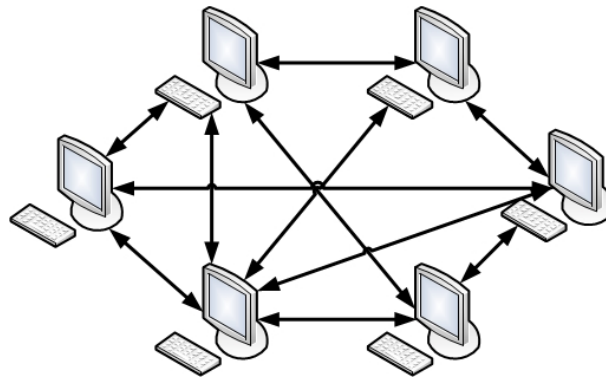


Abbildung 9.4: P2P-Netzwerk

Unstrukturierte P2P Netze

In den Anfängen des Datenaustausches über P2P-Netze wurden unstrukturierte Ansätze verwendet. Es gibt verschiedene Möglichkeiten der Umsetzung. Eine Variante verwendet einen zentralen Server, welcher Informationen über die verfügbaren Daten bzw. Ressourcen um Netz verwaltet. Hier können Abfragen bezüglich vorhandener Daten, von den Teilnehmern, gestellt werden. Mit den zurückgelieferten Speicherorten wird anschließend der Datenaustausch direkt zwischen den Peers durchgeführt (s. Abbildung 9.5). [3]

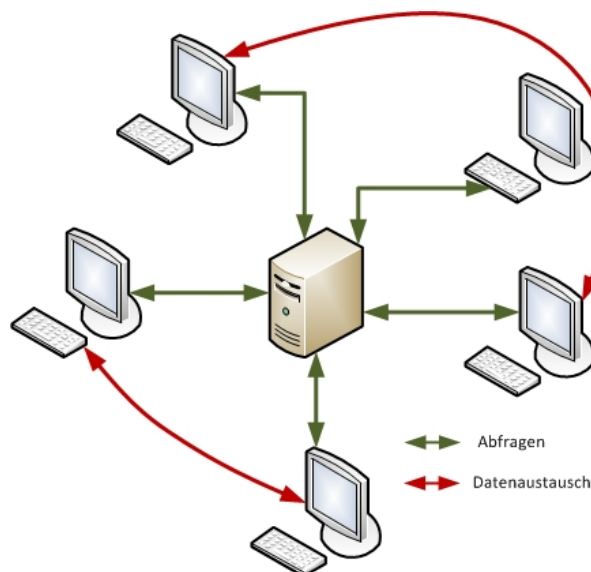


Abbildung 9.5: Zentralisiertes Netzwerk

Die zentrale Datenbank bildet hierbei einen Flaschenhals bezüglich Speicher, Verarbeitungsgeschwindigkeit und Bandbreite. [4] Zusätzlich existiert hier der "single point of failure".

Eine Alternative zu dieser Form der unstrukturierten Netze bilden die reinen P2P Netze (s. Abbildung 9.6). Bei dieser Organisation des Netzes gibt es nur gleichgestellte Teilnehmer, wie bereits beschrieben, kann Jeder Daten bzw. Ressourcen sowohl anbieten als auch nutzen. Da kein zentrales Register existiert werden Suchanfragen durch das Netz flutartig verbreitet. Dazu wird der "Look-Up" solange an alle erreichbaren Peers weitergeleitet, bis

die gesuchten Daten mindestens einmal gefunden wurden. Wie sofort erkennbar ist, wird bei dem angesprochenen "Fluten", pro Suche eine relativ hohe Bandbreite benötigt.[3]

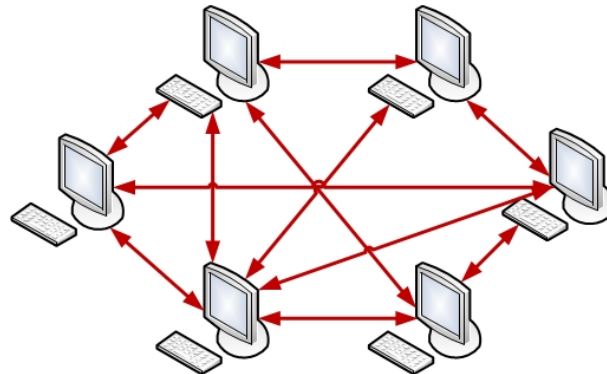


Abbildung 9.6: Reines Netzwerk

Eine dritte Variante der unstrukturierten Netze versucht nun die Vorteile der beiden zuvor genannten Formen zu kombinieren. Diese Hybriden (oder auch Super) P2P Netze bilden eine zusätzliche Overlay-Schicht. Dabei bilden die leistungstärksten Teilnehmer (bezieht sich auf alle Faktoren, welche zum Flaschenhals der Zentralisierten Methode führten) untereinander ein reines P2P Netz. Alle schwächeren Teilnehmer stellen nun eine Client-Server Verbindung, zu dem besten erreichbaren, sogenannten Super-Knoten, her. In Abbildung 9.7 ist der Aufbau eines solchen Netzes schematisch dargestellt.[4]

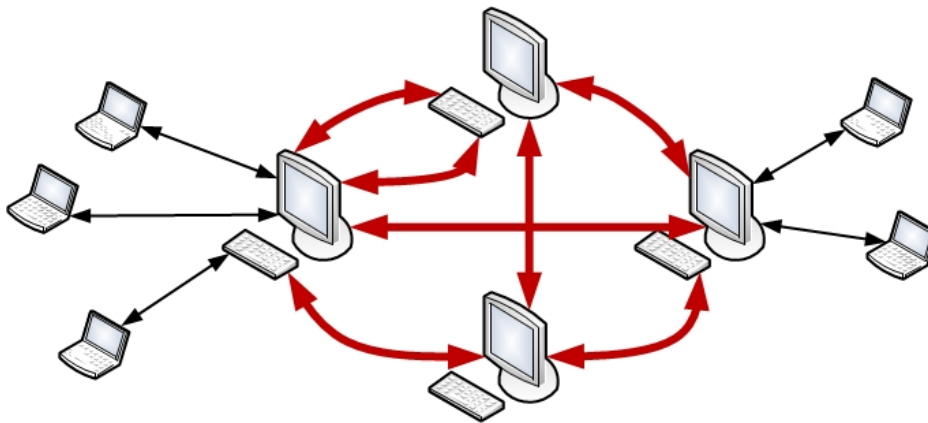


Abbildung 9.7: Hybrides Netzwerk

Historisch betrachtet ist diese Form der Netze keine gewollte Weiterentwicklung im eigentlichen Sinne gewesen. Vielmehr hat sich dieses System automatisch ergeben, da immer mehr Teilnehmer Daten bzw. Ressourcen konsumierten aber selbst kaum anbieten konnten oder wollten.[5]

Strukturierte P2P Netze

Nun bestand eine große Nachfrage selbst organisierende Netze zum Datenaustausch zu schaffen, welcher im Gegensatz zu einem unstrukturierten Ansatz, je nach Anforderung

skalieren können. Hierbei ist es notwendig die Position von Daten systematisch und verteilt zu speichern und die Verbindungen der Teilnehmer zu organisieren. Für diesen Zweck werden die sogenannten Distributed Hash Tables (DHTs) genutzt. DHTs funktionieren wie normale Hash-Tabellen und bilden ebenfalls Schlüssel-Werte Paare. Dabei wird versucht mit möglichst linearen Schlüsseln die Daten homogen auf alle Knoten zu verteilen. Für die Ausfallsicherheit gibt es zwischen den einzelnen Knoten eine gewisse Redundanz. Sollte es zum Verlust eines Teilnehmers kommen, wird eine möglichst geringe Neuordnung angestrebt. Diese Aufgabe übernehmen die den DHTs zugrundeliegenden konsistenten Hashfunktionen. Die Daten des ausgefallenen Knotens werden nun auf die nächsten Teilnehmer übertragen. Alle anderen Peers bleiben unverändert. Somit verändern sich Suchpfade zu bestimmten Ressourcen nur in den letzten Abschnitten.[3]

9.2.2 VPN

Durch die Verwendung von Firewalls oder anderen Mechanismen werden Teilnetzwerke vor dem Zugriff aus dem Internet geschützt. So werden zum Beispiel wichtige Firmenserver vor dem Zugriff durch unautorisierte Personen bewahrt. Dennoch muss es für Außendienst- oder andere Mitarbeiter möglich sein das Firmen-Netz zu erreichen (s. Abbildung 9.8). Eine Alternative zur Realisierung bietet Virtual Private Network (VPN). Diese sind geschlossene virtuelle Teilnetze, welche ein physisches Netz verwenden ohne von anderen darauf laufenden Verbindungen beeinflusst zu werden. Umgesetzt wird diese Funktion durch sogenanntes "Tunneln". Dabei werden die eigentlichen Netzwerkpakete am VPN Gateway verpackt, verschlüsselt und neu adressiert. So gelangen die Pakete durch das verwendete Netz zum anderen Endpunkt der VPN Verbindung. Dort angekommen werden die Pakete wieder in ihren ursprünglichen Zustand zurückversetzt und übertragen so die Daten des Absenders.[7]

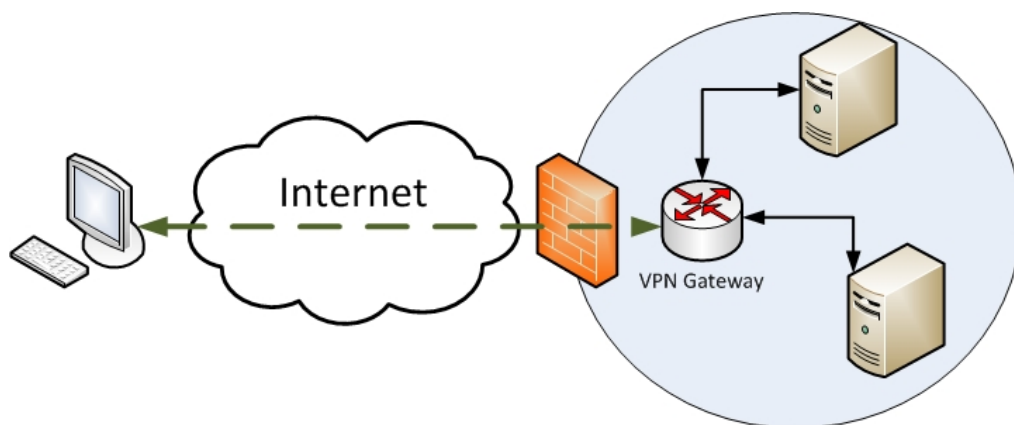


Abbildung 9.8: End-to-Site Verbindung

Die Verschlüsselung der Pakete hängt von dem zugrundeliegenden Protokoll ab. So können für VPN Verbindungen zum Beispiel IPSec, SSL/TLS oder auch PPTP verwendet werden. Abhängig vom Einsatzgebiet gibt es verschiedene Verbindungskonstellationen.

Bei der "End-to-End" Methode (Abbildung 9.9) wird die Kommunikation zweier Einzelgeräte durch ein anderes Netz getunnelt. Bei dem oben beschriebenen Szenario kommt eine



Abbildung 9.9: End-to-End Verbindung

"End-to-Site" Verbindung (Abbildung 9.8) zu Stande. Dabei verbindet sich ein Einzelgerät mit einem Intranet, wie es zum Beispiel in einer Firma vorkommt. Die dritte Variante ist eine "Site-to-Site" Verbindung (Abbildung 9.10), bei der die Kommunikation zwischen zwei kompletten Teilnetzen getunnelt wird.[6]

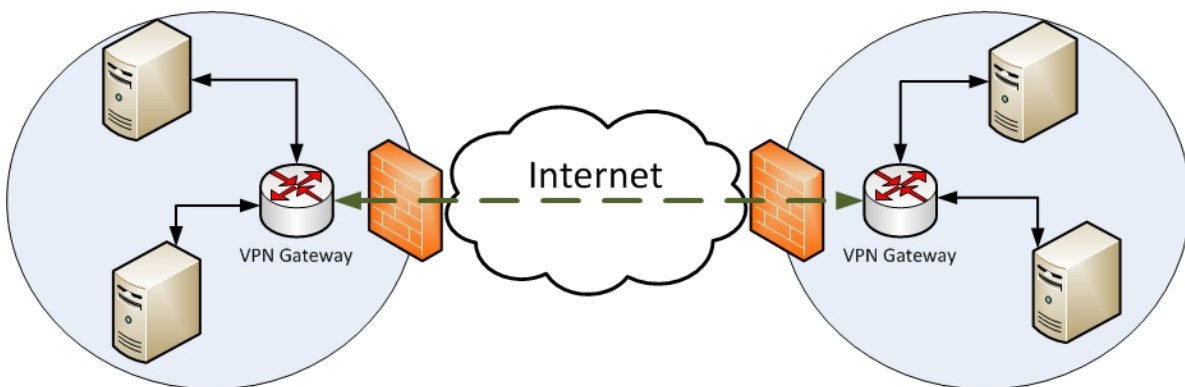


Abbildung 9.10: Site-to-Site Verbindung

9.3 VPN-P2P Ansätze

Wie im vorangegangenen Kapitel gezeigt wurde, eignen sich P2P Netzwerke hervorragend zum Austausch von Daten in einer Teilnehmer Runde. VPN Verbindungen bieten die Möglichkeit virtuelle Netze vor dem Zugriff unautorisierter Personen zu schützen. Im Zusammenhang von "Social VPN" fehlt jetzt nur noch die Einbringung der sozialen Komponente, die nach dem Prinzip des "Web of Trust", eine vertrauenswürdige Liste von Teilnehmern zu Verfügung stellt. Zwischen diesen Teilnehmern kann nun eine gesicherte Datenübertragung stattfinden. Auch hier gibt es verschiedene Möglichkeiten die fehlenden Komponenten einzubinden. Es müssen nicht nur Identitäten und Beziehungen der Peers bereitgestellt werden, sondern auch entsprechende Zertifikate zur Absicherung. Im folgenden werden die verschiedenen Alternativen zur Umsetzung vorgestellt und genauer beleuchtet.

9.3.1 Zentralisiertes Modell

Bei der Verwendung des "Zentralisierten Modells" stellt ein einzelner Server alle drei angesprochenen Elemente zur Verfügung. Dieser Server wird im folgenden "Social Backend"

genannt. Grundlage dieses Modells ist die Vertrauenswürdigkeit dieses Social Backends. Am Beispiel der bekannten Plattform: "Facebook", wird diese Variante nun erklärt.

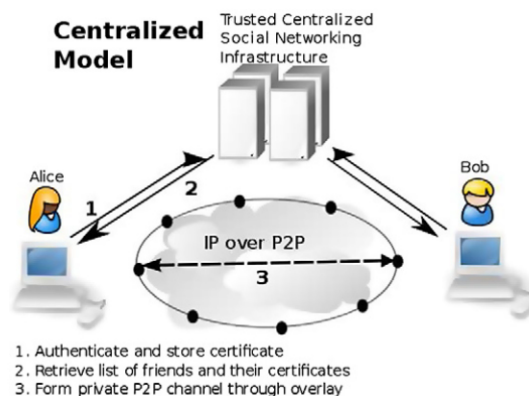


Abbildung 9.11: Zentralisiertes Modell[8]

Über die Freundesliste in Facebook wird eine Peerliste für mögliche Teilnehmer bereitgestellt. Ein Nutzer vertraut diesen Identitäten bereits, da sie sich in dieser Liste befinden. Zusätzlich stellt das verwendete Social Backend die Möglichkeit zur Verfügung Informationen über diese Teilnehmer abzurufen und über den Datastore auch weitere Daten zu speichern. So können an entsprechender Stelle die Zertifikate der Peers gespeichert und zur Authentifizierung genutzt werden. Alle nötigen Informationen zur Datenübertragung und Verschlüsselung konnten nun abgerufen werden. Es werden eindeutige Identifier der Teilnehmer bereitgestellt und über den Datastore können die notwendigen P2P Adressen für ein entsprechendes Overlay-Routing ausgelesen werden. Die Integrität dieser Daten wird durch die Vertrauenswürdigkeit des verwendeten SocialBackend sichergestellt. Diese Methode ist unter den vorgestellten die am einfachsten zu handhabende.[9]

9.3.2 Semi-Zentrales Modell

Der semi-zentrale Ansatz unterscheidet nur gering vom zentralisierten Modell. Identitäten und Vertrauensbeziehungen werden wiederum über das SocialBackend bereitgestellt. Für die Speicherung der Zertifikate wird bei dieser Methode ein verteilter Ansatz gewählt. Hierfür bieten sich die, bereits im Vorfeld erwähnten, DHTs an. Die öffentlichen Keys der Zertifikate werden also per DHTs gespeichert. Um jedoch deren Integrität weiterhin sicherzustellen wird im SocialBackend deren Fingerprint hinterlegt. Dieser benötigt wesentlich weniger Speicherplatz, wodurch die Anforderungen an das SocialBackend sinken. Durch einen geringeren Speicherbedarf erhöht sich auch die Skalierbarkeit des Gesamtsystems. Nachdem die Wahrhaftigkeit eines Zertifikates durch den Fingerprint sichergestellt wurde, kann das Tunneln des Datenaustausches beginnen.[8]

9.3.3 Dezentrales Modell

Das Dezentrale Modell bietet am meisten Freiheiten bei der Implementierung des Systems. Informatione bezüglich der Beziehungen und zu den einzelnen Teilnehmern können über

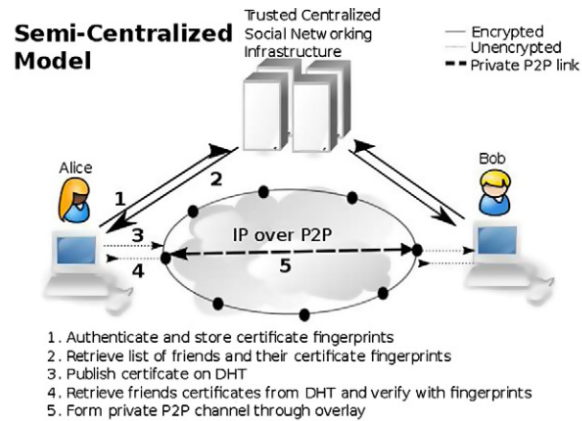


Abbildung 9.12: Semi-Zentrales Modell[8]

jeweils verschiedene Quellen bezogen werden. Die Zertifikate für eine sichere Datenübertragung werden wie im vorangegangenen Modell über DHTs gespeichert.

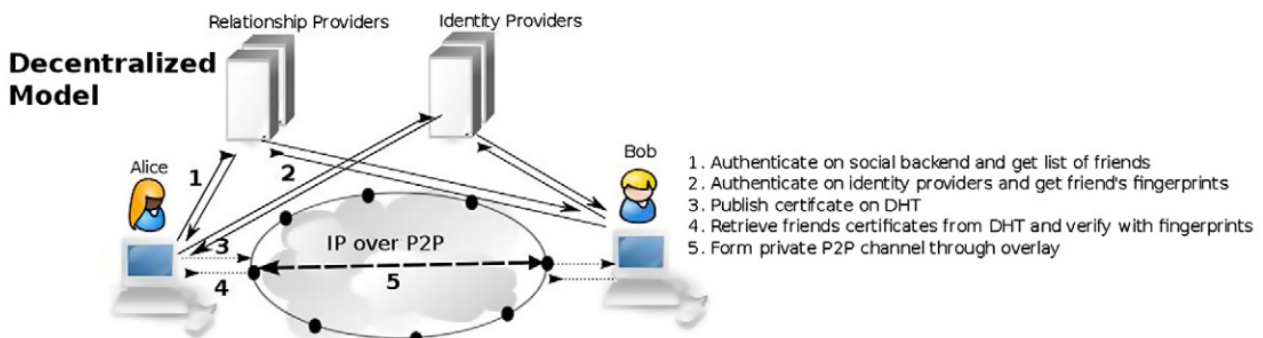


Abbildung 9.13: Dezentrales Modell[8]

Die verwendeten Einzelkomponenten werden bei dieser Methode zu einer sozialen Netzwerkstruktur kombiniert. Für die Bereitstellung von Beziehungen wird eine Liste von eindeutigen Identifiern benutzt. Dies können zum Beispiel Email-Adressen oder Nutzernamen eines Instant-messengers sein. Identitätsinformationen und der bereits im vorangegangenen Modell verwendete Fingerprint, können über einen anderen Provider bezogen werden. So kann der Fingerprint zum Beispiel im öffentlichen Profil von Google+ gespeichert werden. Zertifikate können nicht nur über DHTs verbreitet werden, sondern auch über PGP signierte Nachrichten. Hierbei wird die Authentizität der Schlüssel durch gegenseitiges Vertrauen sichergestellt. Es werden mehrere disjunkte Wege gesucht, welche die Integrität eines verwendeten Schlüssels bestätigen. Das System der PGP Authentifizierung basiert auf dem Prinzip des "Web of Trust". Das dezentrale Modell bietet die Möglichkeit mehrere verschiedene Provider für Peers zu integrieren.[8]

9.4 Vertreter

Im folgenden Kapitel werden einige Vertreter des Social VPN Ansätze vorgestellt. Dabei wird kurz auf ihre Funktionen und damit auf mögliche Einsatzbereiche eingegangen. Die verwendeten Sicherheitsmechanismen werden aufgezeigt.

9.4.1 Hamachi

Hamachi ist ein Produkt der Firma "LogMeIn". Es handelt sich hierbei um ein Closed-Source Projekt, welches ein zentralisiertes Modell implementiert. Der entsprechende Server wird von der Firma selbst zur Verfügung gestellt. Nach einer Registrierung kann man eigenen Netzwerke mit Freunden erstellen. Dabei stützt sich das System auf keine vorhandenen sozialen Plattformen ab, es müssen neue Beziehungen aufgebaut werden.

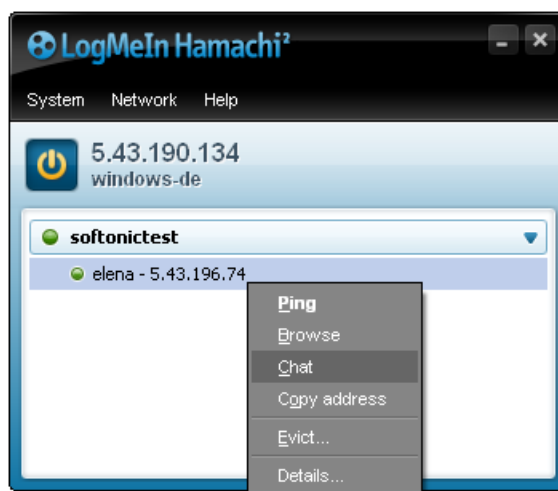


Abbildung 9.14: LogMeIn: Hamachi Client[11]

Durch den zentralisierten Ansatz bietet Hamachi zusätzlich die Möglichkeit den Datenverkehr umzuleiten. Wenn es Teilnehmern nicht möglich ist Daten direkt untereinander auszutauschen, können sie ebenso über den Server vermittelt werden. Diese Funktion wird bei Hamachi "Datenrelay" genannt. Aufgrund des proprietären Ansatzes können die verwendeten Sicherheitsmechanismen des Systems nicht eindeutig nachvollzogen werden. Jedoch tätigt LogMeIn die Aussage das ihre Sitzungen mit einer 256bit AES Verschlüsselung versehen sind.[10]

9.4.2 Wippien

Wippien ist im Gegensatz zu Hamachi eine freie Umsetzung des "Social VPN" Gedanken. Es verwendet ein abgewandeltes zentralisiertes Modell. Dabei ist es möglich die Beziehungen und Profilinformationen über jeden XMPP Server abzurufen. So können zum Beispiel ICQ- oder Facebook- Freundeslisten genutzt werden. Für den Verbindungsaufbau wird ein

sogenannter Mediationsserver verwendet. Dieser wird einerseits von der Firma "Wippien Software" bereitgestellt, kann andererseits jedoch auch selbst aufgesetzt werden. [12]



Abbildung 9.15: Wippien Client[13]

Auch in diesem freien Projekt wird zum Verbindungsaufbau ein proprietärer VPN Client, der Firma "WeOnlyDo! Software" verwendet. Dieser benutzt eine AES Sitzungsverschlüsselung von 128bit.[14]

9.4.3 SocialVPN

Das SocialVPN Projekt nimmt in der Liste der Vertreter eine Sonderrolle ein. Es ist die einzige Implementierung, welche auf wissenschaftlichen Ausarbeitungen basiert. Genau wie bei Wippien handelt es sich auch hierbei um eine Open Source Umsetzung.



Abbildung 9.16: SocialVPN Client[15]

Des weiteren wird, im Gegensatz zu den Anderen Vertretern ein dezentrales Modell genutzt. So ist es möglich verschiedene soziale Netzwerke einzubinden und deren Profilinformationen zu nutzen. Die Teilnehmerliste kann zusätzlich auch aus weiteren Quellen mit eindeutigen Benutzernamen gespeist werden. In Abbildung 9.16 ist zu sehen, wie bspw. Kontakte mit Email-Adressen und Jabber-Kontos hinzugefügt werden können. Der zur Verbindungssicherung benutzte VPN Client benutzt IPSec und ist zu einer AES Sitzungsverschlüsselung von 256bit in der Lage.[15]

Literaturverzeichnis

- [1] "SOCIAL NETWORKS: GESCHÄFTSMODELLE, NUTZERVERHALTEN UND ERFOLG": CREATE OR DIE. <http://createordie.de/cod/artikel/Social-Networks-Geschaeftsmodelle-Nutzerverhalten-und-Erfolg-1720.html>, Stand: 15.12.2011
- [2] "PEER-TO-PEER": WIKIPEDIA. <http://de.wikipedia.org/wiki/Peer-to-Peer>, Stand: 15.12.2011
- [3] "PEER-TO-PEER SYSTEMS AND APPLICATIONS": STEINMETZ, WEHRLE. *What Is This "Peer-to-Peer" About?*, Springer, Berlin / Heidelberg, 2005.
- [4] "PEER-TO-PEER-ARCHITEKTUREN": WILHELM EISENSCHMID. *Proseminar Virtuelle Präsenz*, Universität Ulm, 2005.
- [5] "HYBRIDE P2P-NETZE": JAN RITZENHOFF. Universität Duisburg-Essen, 2003
- [6] "VIRTUAL PRIVATE NETWORK": WIKIPEDIA. , Stand: 15.12.2011
- [7] "NETZWERK- UND DATENSICHERHEIT": MARTIN KAPPES. *Virtual Private Networks (VPN)*, S.193-217, Teubner, 2007.
- [8] "SOCIALVPN": JUSTE, WOLINSKY, BOYKIN, COVINGTON, FIGUEIREDO. *Enabling wide-area collaboration with integrated social and overlay networks*, Computer Networks, Volume 54, Issue 12, S.1926-1938, August 2010.
- [9] "WORKSHOP ON ENABLING TECHNOLOGIES: INFRASTRUCTURE FOR COLLABORATIVE ENTERPRISES": JUSTE, WOLINSKY, BOYKIN, FIGUEIREDO. *Integrating Overlay and Social Networks for Seamless P2P Networking*, WETICE '08. IEEE 17th, S.93-98, June 2008.
- [10] "OFFIZIELLE HAMACHI WEBSITE": LOGMEIN. <https://secure.logmein.com/products/hamachi/> , Stand: 15.12.2011
- [11] "HAMACHI": SOFTONIC. <http://hamachi.en.softonic.com/> , Stand: 15.12.2011
- [12] "OFFIZIELLE WIPPIEN WEBSITE": WIPPIEN SOFTWARE. <http://www.wippien.com/> , Stand: 15.12.2011
- [13] "WIPPIEN": WIKIPEDIA. <http://de.wikipedia.org/wiki/Wippien> , Stand: 15.12.2011
- [14] "WODVPN CLIENT": WEONLYDO! SOFTWARE. <http://www.weonlydo.com/VPN/p2p-vpn-component.asp> , Stand: 15.12.2011

- [15] "OFFIZIELLE SOCIALVPN WEBSITE": ACIS P2P RESEARCH GROUP.
<http://socialvpn.wordpress.com/> , University of Florida, Stand: 15.12.2011

Abkürzungsverzeichnis

BMI	Bundesministerium des Inneren.
C&C-Server	Command and Control Server.
CERT	Computer Emergency Response Team.
CISO	Chief Information Security Officer.
COBIT	Control Objectives for Information and related Technology.
DDoS	Distributed Denial of Service.
DHTs	Distributed Hash Tabela.
DNS	Domain Name System.
DRP	Disaster Recovery Plan.
FTP	File Transfer Protocol.
HTTP	Hypertext Transport Protocol.
IDS	Intrusion Detection System.
IEC	International Electrotechnical Commission.
IM	Instant Messaging.
IP	Internet Protocol.
IRC	Internet Relay Chat.
IRP	Incident Response Plan.
ISACA	Information Systems Audit and Control Association.
ISMS	IT-Sicherheitsmanagementsystem.
ISO	International Organization for Standardization.
IT	Informations-Technik.
IT-SIBE	IT-Sicherheitsbeauftragter.
MOTD	Message of the Day.
OCTAVE	Operationally Critical Threat, Asset, and Vulnerability Evaluation.

P2P	Peer-to-Peer.
PDCA	plan-do-check-act.
SLA	Servie Level Agreement.
SMB	Server Message Block.
SQL	Structured Query Language.
TCP	Transmission Control Protocol.
USV	Unterbrechungsfreie Stromversorgung.
VPN	Virtual Private Network.
XDCC	Xabi Direct Client-to-Client oder eXtended DCC.