

Projektdokumentation

<<**security|engineering**>>

Rafael Riester (mailto:mailonfly@gmx.net),
Marco Christian (mailto:marco.christian@gmx.net),
Egon Windhager (mailto:mail@burn3r.de),
Johannes Veit (mailto:leerstring@gmail.com),
Matthias Brettschneider (mailto:frosch03@frosch03.de)

4. Juli 2006

Inhaltsverzeichnis

1	Projektbeschreibung	6
1.1	Einführung	6
1.2	Projektorganisation	6
1.3	Vorgehensweise	6
1.3.1	Kickoff Meeting	6
1.3.2	Arbeitstreffen	7
1.3.3	Arbeitsbesprechungen	7
2	Computersicherheit	8
2.1	Einführung	8
2.1.1	Schutzziele	8
2.1.2	Security through obscurity	10
2.2	Passwörter	11
2.2.1	Kennwörter/Passwörter	12
2.2.2	Smartcards/Dongels	13
2.2.3	Biometrie	14
2.2.4	Fazit	16
2.3	Social Engineering	16
2.4	Kelogger	17
3	Netzwerksicherheit	18
3.1	Sniffen und Spoofen	18
3.1.1	Sniffen	18
3.1.2	Spoofen	20
3.2	(Distributed) Denial of Service	22
3.2.1	Ping of Death	23
3.2.2	Flooding	23
3.3	Wlan	25
3.3.1	Einführung	25
3.3.2	Leistungen	25
3.3.3	Betriebsmodi	26
3.3.4	Sicherheit	26
3.4	VPN	27
3.4.1	Einführung	27

3.4.2	Sicherheit	28
3.4.3	Transport	30
3.5	Firewalls	30
3.5.1	Hostfirewall	31
3.5.2	Netzwerkfirewall	31
3.6	DMZ	32
4	Websicherheit	33
4.1	XSS - Cross-Site Scripting	33
4.1.1	Einleitung	33
4.1.2	Vorgehensweise	33
4.1.3	Beispiele	34
4.1.4	Sonstiges	35
4.1.5	Fazit	35
4.2	HTTP-Response-Splitting	35
4.2.1	Einleitung	35
4.2.2	Vorgehensweise	36
4.2.3	Fazit	38
4.3	Google-Hacking	38
4.3.1	Der Google-Cache	41
4.3.2	Google als Proxy benutzen	41
4.3.3	Verzeichnislisten	42
4.3.4	Network Mapping	43
4.3.5	und vieles vieles mehr	44
4.4	PHP-Sicherheit	45
4.4.1	Fehler in PHP	45
4.4.2	Bestandteile eines 'sicheren' Servers	46
4.4.3	Installation	46
4.4.4	suExec	46
4.4.5	Safe Mode	47
4.4.6	Weitere PHP-Einstellungen	47
4.4.7	PHP-Hardening	48
5	Honeypots	51
5.1	Theorie	51
5.1.1	Die Täuschung	51
5.1.2	Die Idee eines Honigtopfes?	51
5.1.3	Interaktionsstufen der Honeypots	51
5.1.4	Welche Software gibt es?	52
5.1.5	Überblick über verwandte Techniken	53
5.2	Honeypots in der Praxis	53
5.2.1	Die Netzwerksimulation	54
5.2.2	Die Netzwerkkonfiguration	55
5.2.3	Konfiguration der simulierten Computer	56

5.2.4	Probleme bei der Konfiguration	57
5.2.5	Fazit	58
6	Computerforensik	59
6.1	Hinführung	59
6.2	Ein paar Begriffe vorweg	59
6.2.1	Die Forensik	59
6.2.2	Kunstwort Computerforensik	59
6.2.3	Computerkriminalität allgemein	60
6.2.4	Ziele einer Forensischen Analyse	60
6.3	Der Forensik-Arbeitsplatz	60
6.3.1	Vorstellung des Arbeitsplatzes	60
6.3.2	Hardware	61
6.3.3	Die Forensik-Tools	62
6.3.4	Shell Skripte zur standardisierten Erstellung von Datenträgerimages	63
6.4	Vorbereitende Arbeiten	67
6.5	Anschließen eines Datenträgers	68
6.6	Mounten einer Partition	70
6.7	Erzeugen eines Images einer Partition	71
6.8	Mounten des Images zur späteren Bearbeitung	71
6.9	Geführte Erstellung eines forensischen Duplikates	72
7	Ausblick	74
7.1	Sicherheits-Kompendium	74
7.2	Honeypot	74
7.3	Forensik	74

1 Projektbeschreibung

1.1 Einführung

Die illegale Nutzung von Rechnern und Daten nimmt in einer raschen Geschwindigkeit zu. Häufig sind Rechner oder ganze Netze Angriffen ausgesetzt. Um dem entgegen zu können, ist es notwendig, sich mit gängigen Angriffsmustern auseinander zu setzen. So kann man Rechner oder Netze gegen typische Angriffe schützen. Mit den neusten Techniken der Computer-Forensik, oder mit Monitoringtools wie Honeypots ist es möglich Angriffe zu verfolgen.

1.2 Projektorganisation

Zunächst haben wir uns Gedanken zur Projektorganisation gemacht. Hierbei haben wir folgende Punkte festgelegt:

- Projektart: Forschungsprojekt
- Projektleitung: Matthias Brettschneider
- Qualitäts- und Zielkontrolle: Jedes Arbeitspaket wurde mit Zielen und Messkriterien versehen
- Zeit- und Terminplanung: Erfolgt über den Zeitplan im Wiki
- Dokumentation: Erfolgte ebenfalls mithilfe des Wiki's
- Abstimmregeln: mündliche, einfache Mehrheit
- Protokollart: schriftlich mithilfe des Wiki's
- Protokollant: Johannes Veit

1.3 Vorgehensweise

1.3.1 Kickoff Meeting

Zunächst haben wir uns zu einem Kickoff Meeting zusammengefunden. Hierbei wurden das grobe Projektumfeld abgesteckt. Dazu wurde ein Brainstorming zum Thema Computersicherheit und Computerforensik durchgeführt. Anhand der gewonnenen Daten wurden

Teilbereiche definiert. Diese Teilbereiche haben wir dann aufgrund des gegebenen Themas und unseren Interessen eingegrenzt. Aus diesen Themen wurde zusätzlich zu den gegebenen Teilbereichen Computerforensik und Sicherheitskompendium noch zusätzlich das Thema Honeypot ausgewählt.

Die nun gefundenen drei Themen wurden dann weiter in Arbeitspakete aufgeteilt. Ausserdem haben wir den Zeitaufwand jedes Arbeitspaketes geschätzt, es dann einer Person zugeordnet und in einem Zeitplan zusammengestellt.

1.3.2 Arbeitstreffen

Jedes Arbeitstreffen hatte eine festgelegte Struktur. So hatte jeder Projektmitarbeiter je einen Mittwoch-Nachmittag zur Verfügung um seine Arbeitspakete zu bearbeiten. Hierfür haben wir uns immer gegen 14:00 Uhr getroffen und die anstehenden Arbeiten durchgesprochen. Danach begann jeder individuell an seinen Aufgaben zu arbeiten. Gegen 19:00 Uhr haben wir uns dann erneut zusammengesetzt und die Fortschritte im Projekt festgehalten.

1.3.3 Arbeitsbesprechungen

Es wurden wöchentliche Arbeitsbesprechungen durchgeführt. Hierzu haben wir uns mit unserem Projektbetreuer Prof. Veihelmann getroffen. Wir haben ihn über den Fortschritt im Projekt aufgeklärt. Ausserdem hatten wir die Möglichkeit Fragen zu stellen oder auch Probleme in einer allgemeinen Sichtweise zu klären. Je nach Situation dauerten die Treffen zwischen einer Viertelstunde und einer Stunde.

2 Computersicherheit

2.1 Einführung

Als das Internet entstand, hätte sich wohl niemand vorstellen können, was wenige Jahrzehnte später daraus wird. Moderne Kommunikation wäre heute ohne das Internet nicht mehr denkbar. So bietet uns das Netz die Möglichkeit elektronische Post zu verschicken, direkt miteinander zu chatten oder auch miteinander zu telefonieren. Auch Einkäufe laufen heute über das Internet ab.

Die genannten Tätigkeiten sind jedoch selten Tätigkeiten, die jeder mitbekommen soll. So ziehen sich viele Leute zurück, wenn sie einen Anruf auf ihrem Handy bekommen. Auch sieht man Menschen, die ihre Post an Litfaßsäule hängen doch eher selten. Es ist das Recht eines jeden, gewisse Sachen privat zu halten.

2.1.1 Schutzziele

Jede neue Technologie bringt auch neue Gefahren und Risiken mit sich. Um diesen Gefahren vorzubeugen sollte man sich vorher Gedanken zur Sicherheit und zum Schutz der jeweiligen Anwendung machen. Genau dazu gibt es sogenannte Schutzziele, welche 1998 von der ISO spezifiziert wurden (siehe ISO/IEC SC27 N2162 Common Criteria for Information Technology Security Evaluation - Part 2). Als Ziele die schützenswert sind werden hier genannt:

- **Vertraulichkeit** sichert die Geheimhaltung von Daten während der Übertragung. Niemand außer den Kommunikationspartnern kann den Inhalt der Kommunikation erkennen.
- **Verfügbarkeit** sichert die Nutzbarkeit von Ressourcen und Diensten, wenn ein Teilnehmer sie benutzen will.
- **Integrität** sichert, dass Modifikationen der kommunizierten Inhalte (Namen des Senders eingeschlossen) durch den Empfang erkannt werden.
- **Verdecktheit** versteckt die Übertragung von vertraulichen Daten. Niemand außer den Kommunikationspartnern kann die Existenz einer vertraulichen Kommunikation erkennen.
- **Anonymität** sichert, dass ein Nutzer Ressourcen und Dienste nutzen kann, ohne seine Identität zu offenbaren. Selbst der Kommunikationspartner erfährt nicht die Identität des Nutzers.

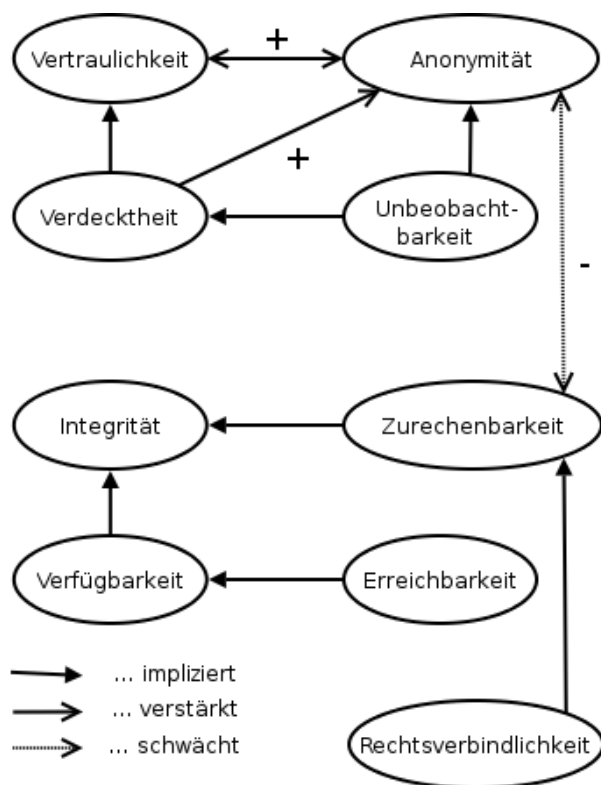


Abbildung 2.1: Wechselwirkung zwischen Schutzzielen vgl. PSWW_00

- **Pseudonymität** sichert, dass ein Nutzer eine Ressource oder einen Dienst benutzen kann, ohne seine Identität preiszugeben, ihm aber trotzdem diese Nutzung zurechenbar ist.
- **Unbeobachtbarkeit** sichert, dass ein Nutzer Ressourcen und Dienste nutzen kann, ohne dass andere beobachten können, dass die Ressource oder der Dienst genutzt wird. Dritte können weder das Senden noch den Erhalt von Nachrichten beobachten.
- **Zurechenbarkeit** sichert, dass Sendern bzw. Empfängern von Informationen das Senden bzw. der Empfang der Information gegenüber Dritten bewiesen werden kann.
- **Erreichbarkeit** sichert, dass zu einer Ressource (Nutzer oder Maschine) Kontakt aufgenommen werden kann.
- **Verbindlichkeit** sichert, dass ein Nutzer belangt werden kann, um seine Zusagen innerhalb einer angemessenen Zeit zu erfüllen.

1

¹vgl. PSWW_00

Schaut man sich diese Ziele an, so erkennt man, dass nicht alle Ziele erreicht werden können oder anders gesagt, dass ein Schutzziel ein anderes schwächt. So ist dies z.B. mit Anonymität und Zurechenbarkeit. Auch gehen einige Schutzziele aus anderen hervor (Bsp.: Unbeobachtbarkeit \Rightarrow Verdecktheit).

Um sinnvoll über Sicherheit reden zu können ist es wichtig zu wissen, was geschützt werden soll. Sind die Schutzziele einmal festgelegt, ist es möglich über geeignete Maßnahmen nachzudenken, die Schutzziele auch umzusetzen.

Die Unbeobachtbarkeit ist wohl das Schutzziel, welches von jedem Nutzer angestrebt werden sollte. Unbeobachtbarkeit impliziert Verdecktheit und Anonymität. Die Verdecktheit wiederum impliziert Vertraulichkeit. So könnte man sich anonym durch das Internet bewegen und sein Recht auf Datenselbstbestimmung ausüben.

Leider ist das Erreichen dieses Schutzziels nicht so einfach wie es möglicherweise scheint. Bewegt man sich von Website zu Website so hinterlässt man eine breite Datenspur. An jeder Ecke werden Cookies verteilt, um einen beim nächsten Besuch der Seite wiederzuerkennen. Die Internetprovider sind mittlerweile daran gebunden, die Verbindungsdaten 6 Monate vorrätig zu speichern. Das EU-Recht sieht sogar eine vorrätige Speicherung von bis zu 24 Monaten vor. Die Deutsche Bundesregierung geht mit der Umsetzung jedoch viel weiter:

*So sollen nicht nur bei erheblichen Straftaten die Daten den Sicherheitsbehörden zur Verfügung gestellt werden, vielmehr soll dies auch bei mittels Telekommunikation begangenen Straftaten jedweder Art der Fall sein. Diese kleine Formulierungsänderung weitet die Zugriffsmöglichkeiten auf die durch die VDS entstandenen Datenberge erheblich aus. So kämen sowohl Beleidigung, Verleumdung oder üble Nachrede als auch (ironischerweise) die Verletzung des Post- und Fernmeldegeheimnis als Begründung für einen Zugriff auf die entstandenen Daten in Frage.*²

Aber nicht nur rechtliche Feinheiten erschweren die Umsetzung unseres Schutzziels. Auch der Einsatz von veralteten Verschlüsselungsverfahren wie z.B. MD5 oder der Einsatz von ungepatchter Software machen Angriffe auf die Schutzziele möglich. Ein anderes Problem ist ein falsches Verständnis von Sicherheit. So wird Sicherheit oft als etwas gesehen, das man mit einer entsprechenden Software erreichen kann. Dem User muss man hier fairer Weise einräumen, dass viele Unternehmen genau das mit ihrer Werbung suggerieren. Allerdings ist Sicherheit niemals etwas, das man nur mit Software erreichen kann. Sicherheit muss als Prozess verstanden werden, in dem Software eine nicht unerhebliche Rolle spielt. Aber auch das Verhalten des Users ist mindestens genauso wichtig.

2.1.2 Security through obscurity

Ein weiterer Punkt ist die Offenheit von Verfahren. So wird ein Cryptoalgorithmus niemals dadurch sicherer, dass niemand weiß wie er funktioniert. Unter bestimmten Umständen kann die Geheimhaltung von Verfahren zu Verzögerungen führen, es macht den Algorithmus jedoch niemals besser.

²vgl. TP_0206

Die Enigma wurde im zweiten Weltkrieg zur Verschlüsselung von Funksprüchen verwendet. Diese wurde schon 1918 erfunden und wurde bis zum Ende der 20er Jahre frei zum Verkauf angeboten. Dies bedeutet, dass das Verschlüsselungsverfahren den Alliierten bekannt war und sie trotzdem nicht mitbekamen was gefunkt wurde. Das Problem war auch hier der Benutzer. So wäre die Enigma wohl nicht geknackt worden, wenn man nicht davon hätte ausgehen können, dass in jedem Funkspruch bestimmte Wörter vorgekommen wären. Dieses Wissen, kombiniert mit ein paar anderen Problemen, führte am Ende dazu, dass die Enigma geknackt wurde.

Das Beispiel zeigt, dass Sicherheit nicht aus Geheimhaltung von Verfahren gewonnen werden kann. Vielmehr wird Sicherheit aus verantwortungsvollen Umgang mit der Technik und bekannten, guten Verfahren gewonnen.

Dieses Kompendium soll Schwachstellen in Verfahren aufzeigen und so auf die daraus entstehenden Probleme hinweisen. Es werden allgemeine Probleme wie schwache Passwörter oder gutmütige Menschen aufgezeigt. Als nächstes werden technische Probleme niedriger und höherer Stufen betrachtet. Des Weiteren zeigen wir auch forensische Maßnahmen auf.

2.2 Passwörter

Fährt man heute einen Computer hoch, so ist der erste Schritt meist das Anmelden am System. Danach loggt man sich bei seinem Emailbetreiber ein um die neusten Mails zu lesen. Vielleicht meldet man sich noch in Wikis oder Foren an und jedes mal benötigt man seinen Benutzernamen und sein Kennwort.

Benutzernamen und Kennwörter sind die Mittel mit denen sich ein Benutzer gegenüber einem System authentifizieren kann. Der Benutzername ist meist nicht geheim, das zugehörige Passwort sollte dies aber sein. Das System erfragt nun vom Benutzer etwas, das nur er wissen kann, z.B. das Kennwort, um dessen Identität festzustellen. Es gibt auch andere Verfahren um die Identität eines Gegenübers zu prüfen. Im wesentlichen gibt es drei Dinge die ein System erfragen kann:

- Wissen: ein Kennwort, eine Passphrase
- Besitz: eine Smartcard, ein USB-Dongel, ein Schlüssel
- Merkmal: meist etwas Biometrisches wie eine Fingerabdruck, eine Netzhaut

3

Diese drei Methoden sind sehr unterschiedlich in der Funktionsweise und auch in ihrer Genauigkeit. So kann Wissen verloren gehen, oder geteilt werden. Besitz kann auch verloren gehen oder an Unberechtigte geraten. Ein Biometrisches Merkmal führt nicht zu einer eindeutigen Identifizierung, sondern gibt nur eine Wahrscheinlichkeit zurück. Daher können sogenannte "false positivs" oder "false rejects" entstehen. Im Folgenden wird allerdings nur auf die Verwendung von Passwörtern vertieft eingegangen, da die

³Wikipedia-Link

Sicherheit bei Hardwarelösungen und biometrischen Verfahren stark von der eingesetzten Hardware abhängig ist.

2.2.1 Kennwörter/Passwörter

Passwörter wurden zuerst im Militärbereich zur Unterscheidung von Freund und Feind eingesetzt. Heute werden Passwörter in vielen Bereichen des alltäglichen Lebens genutzt. Die PIN des Girokontos und des Handys sind die wohl bekanntesten Formen. Hier werden numerische Zeichenfolgen als Zugangsschutz verwendet, die z.B. beim Girokonto nicht frei gewählt werden können. Dies sind Unterformen der Passwörter, welche im Gegensatz zu PINs alphanumerisch sind.

Die Verwendung von Passwörtern stellt allerdings nur eine Grundsicherheit dar. Die Wirksamkeit eines Passworts hängt stark von dessen Zusammensetzung ab. So sollte man bei der Wahl eines Passworts auf gewisse Dinge achten. Die 3 essentiellen Punkte sind:

- Groß-/Kleinschreibung
- Zahlen
- Sonderzeichen

Dadurch hat man schon einen gewissen Grad der Sicherheit erreicht, da das "Erraten" solcher Passwörter durch die hohe Kombinationsmöglichkeit an Zeichen sehr aufwendig wird. Wobei zu beachten ist, keine

- Namen, darunter fallen neben Vor-/Nachnamen auch Eigennamen und Produkt-namen usw.
- Wörter, die in Wörterbüchern zu finden sind
- Persönliche Informationen, wie Geburtstag, Namen von Angehörigen, Wohnort

zu verwenden. Wenn man dies alles beachtet hat man schon ein ziemlich sicheres Passwort. Das Ideale ist allerdings ein zufällig erstelltes Passwort, welches die o.g. Kriterien erfüllt. Um sich ein Passwort leichter merken zu können kann man auch die Anfangsbuchstaben eines Satzes nehmen. Natürlich sollten hier auch die o.g. Kriterien beachtet werden. Es gibt auch Programme die einem die Verwaltung von Passwörtern vereinfacht. Vor der Benutzung solcher Programme sollte man sich aber über die Funktionsweise und Sicherheit dieser informieren. Selbst ein jetzt schon relativ sicheres Passwort, kann die Sicherheit noch einschränken wenn man die folgenden Punkte nicht beachtet.

- häufiges Ändern des Passwortes
- Passwörter an einem sicheren Ort aufbewahren, oder am Besten merken und nir-gends aufschreiben

- niemandem seine Passwörter mitteilen
- für **jeden** Zugang ein eigenes Passwort verwenden
- Browser Cookies löschen, speziell auf Fremdrechner
- bei der Eingabe des Passwortes aufpassen, dass dies von Niemandem beobachtet wird
- je länger das Passwort ist, desto sicherer. Hier werden die Grenzen allerdings von der Anwendung gesetzt.

Wurden alle Kriterien beachtet so hat man von seiner Seite aus alles getan die Sicherheit des Passwortes zu maximieren. Allerdings wird man nie ein absolut sicheres Passwort haben, allein der Aufwand dieses Passwort zu knacken kann erhöht werden. In den meisten Fällen ist dieser Aufwand so hoch dass das Knacken eines Passworts Jahre dauern kann und dies somit für den Angreifer uninteressant wird, oder schlicht nicht durchführbar ist.

indestlänge	maximal benötigte Zeit
(bei angenommener 1 Million Tastaturanschlägen pro Sekunde)	
3 Zeichen	ca. 0,2 Sekunden
5 Zeichen	ca. 14 Minuten
8 Zeichen	ca. 53252 Stunden
10 Zeichen	ca. 1179469 Wochen
12 Zeichen	ca. 84168853 Jahre
15 Zeichen	ca. 19104730610573 Jahre

Listing 1: Zeitaufwand um ein Passwort zu knacken

2.2.2 Smartcards/Dongels

Hier werden die Anmeldedaten wie Benutzerkennung und Passwort verschlüsselt auf einem Medium gespeichert. Die Kriterien eines sicheren Passworts sind hier die selben wie im oberen Abschnitt erwähnt, jedoch wird hier die Sicherheit dadurch erhöht, dass die Daten über ein verschlüsseltes System übertragen werden. Somit entfallen Angriffsmöglichkeiten wie Keylogger, allerdings muss das Kartenterminal eine eigene Tastatur bereitstellen.

Die Kommunikation zwischen Smartcards verläuft immer verschlüsselt. Gängige Verfahren sind DES, Triple-DES oder das public key RSA Verfahren. Dies erlaubt Schlüssel bis zu einer Länge von 1024 Bit. Jedoch ist diese Verfahren auch nicht sicher und selbst solche Schlüssel sind durch "Brute-Force-Attacken" schon geknackt worden. Dies ist allerdings nicht die einzige Schwachstelle. Dadurch, dass die Daten in einem EPROM gespeichert werden ist es Möglich diese durch, z.B. Überspannung, zu beeinflussen. Aus diesem Grund werden auf einigen Smartcards Sensoren eingesetzt die auf solche äußeren Einflüsse reagieren. Die Erkennung erweist sich allerdings als nicht ganz einfach, was zur

Folge hat, dass vergleichbare Verfahren nur sehr selten eingesetzt werden. Eine weitere Möglichkeit die Sicherheit auszuhebeln, ist durch starke Wärmeeinwirkung oder die Aussetzung unter UV Strahlung. Dies ist wie das Zerlegen und Reverse-Engineering allerdings eine zerstörerische Methode. Es gibt auch noch die Möglichkeit den Schlüssel mittels DPA (Differential Power Analysis) zu extrahieren. Diese statistische Attacke ist oft erfolgreich und wird auch zusammen mit SPA (Simple Power Analysis) kombiniert.

Möglichen Schutz gegen diese Angriffe bieten unter Anderem die folgenden Techniken:

- Fertigungsprozess: Durch Reduzierung der Größe und der Leistungsaufnahme von Smartcards wird es schwer DPA/SPA Angriffe zu starten, da man die Ursache der Fluktuationen nicht mehr genau zuordnen kann.
- Unvorhersehbares Verhalten: Der Chip erzeugt hier zufällig Interrupts und verändert somit den Ablauf der Software, was zu unvorhersehbaren Mustern der Leistungsaufnahme führt.
- Design: Modulares Design welches es erlaubt, schnell und günstig auf neue Angriffe zu reagieren.
- Firmware und Sicherheitsmechanismen: Firmwarefunktionen und Sicherheitsmechanismen sind in der Lage, mögliche Angriffe zu entdecken und auf verschiedene Art darauf zu reagieren. So kann man auf falsche Opcodes, Speicheradressen u.Ä. Zum Beispiel mit Löschen des RAM, Interrupts oder Resets reagieren.

2.2.3 Biometrie

Biometrie bedeutet sinngemäß Lebensvermessung. Sie beschäftigt sich mit der Vermessung quantitativer Merkmale von Lebewesen. Dazu werden statistische Methoden auf Datenmengen angewendet, welche z.B. durch Netzhautscans, Fingerabdruckscans oder aus Fotos gewonnen werden. Man kann nun entweder eine Person aus einem gewissen Personenkreis authentifizieren, oder eine Person aus einem undefinierten Personenkreis identifizieren. Dabei handelt es sich dann entweder um Verifikation oder Identifikation. Diese Verfahren können nun entweder mittels eines, oder auch mittels der Kombination mehrerer biometrischer Merkmale umgesetzt werden.

Der Ablauf zur Erkennung eines Musters gestaltet sich folgendermaßen. Zuerst wird ein Referenzmuster abgelegt, soll nun eine Person authentifiziert werden, wird ein aktuelles Probemuster aufgenommen. Dies kann z.B. über einen Fingerabdrucksensor, Netzhautscanner oder einer einfachen Digitalkamera erfolgen. Dieses Muster wird nun mit dem Referenzmuster verglichen und die Wahrscheinlichkeit einer Übereinstimmung ausgerechnet. Da dies kein 100% zuverlässiges System (nicht deterministisch) ist, wird die Sicherheit biometrischer Kontrollsysteme nach "false positives" und "false rejects" bewertet. Das heißt wie oft hat ein System befugte Personen abgewiesen und wie oft unbefugte zugelassen.

Ein weiterer Faktor, der bei biometrischen Kontrollsystemen berücksichtigt werden muss, ist die Erkennungszeit. Die hohen Datenmengen und komplizierten Algorithmen

setzen komplexe technische Einrichtungen voraus. Deswegen hat sich die Biometrie erst in den letzten Jahren richtig weiterentwickelt und gilt als vielversprechendes Verfahren. Schwachstellen gibt es hier vor allem im Bereich der Algorithmen und bei den Erfassungstechniken.

Fehlerhafte Algorithmen sind für einen Angreifer nur schwer auszunutzen, da er dessen Funktionsweise und die daraus folgenden Schwächen kennen muss. Außerdem muss er den vom anzugreifenden System verwendeten Algorithmus kennen. Die Hardware ist zur Zeit der häufigste Angriffspunkt. So kann man Fingerabdrucksensoren mittels Puder und Klebeband, oder durch abgetrennte Gliedmaßen überlisten. Abhilfe schafft in diesem Fall die zusätzliche Messung der Körpertemperatur. Generell ist zu sagen, dass erst die Kombination verschiedener Merkmale ein biometrisches System anwendbar macht. Die Kombination muss natürlich im Einzelfall genau abgewogen werden, um erstens eine hohe Sicherheit zu erreichen, aber auch um die Erkennungszeit in einem akzeptablen Rahmen zu halten.

Neben den bereits erwähnten Merkmalen können noch weitere zur biometrischen Auswertung herangezogen werden. Nachfolgend eine Auflistung gebräuchlicher Merkmalen.

⁴

- Körpergröße
- Iris- oder Retina-Merkmale (Regenbogenhaut, Augenhintergrund)
- Fingerabdruck (Fingerlinienbild)
- Gesichtserkennung
- Handgefäßstruktur / Venenerkennung
- Handgeometrie, Handlinienstruktur
- die Stimme und das Sprachverhalten (Spracherkennung)
- die Handschrift
- das Tippverhalten auf Tastaturen (engl. keystroke dynamics)
- Lippen, Stimmprofil
- Verhalten des Menschen, Füße (Gang, engl. automatic gait recognition)
- der Geruch
- DNA (mobiler DNA-Test, entwickelt von Mitsubishi, genetischer Fingerabdruck)

⁴vgl. <http://de.wikipedia.org/wiki/Biometrie>

2.2.4 Fazit

Die Entscheidung welche Zugangskontrolle gewählt werden soll, muss immer im Einzelfall entschieden werden. Wichtige Faktoren sind hier:

- Wert der Daten
- Sicherheitsstufe
- technischer Aufwand
- Kosten
- Einsatzgebiet
- Datenschutzfragen

Generell gilt aber die Zugangskontrolle immer aktuell zu halten und auf neue Angriffsmöglichkeiten zu reagieren und Sicherheitslöcher schnell zu stopfen. Dazu muss man immer auf dem aktuellen Stand bleiben und sich ständig informieren. Sicherheit ist keine einmalige Installation, sondern ein dynamischer Prozess. Die Zukunft der Sicherheitssysteme liegt, durch die immer größeren Fortschritte in der Quantenforschung, vermutlich in diesem Bereich. Die ersten Versuche speziell im Bereich der Quantenkryptographie zeigen recht positive Tendenzen und lassen auf einen hohen Grad der Sicherheit hoffen.

2.3 Social Engineering

Social Engineering beschreibt eine Angriffsmethode, bei der nicht der Computer oder die Software direkt angegriffen wird. Vielmehr macht man sich den guten Glauben der meisten Menschen zu nutze. So bekommt man in aller Regel von Usern die Logindaten gesagt, wenn man sich nur als EDV-Mitarbeiter ausgibt und freundlich fragt.

Privatdetektive, Reporter oder andere "Angreifer" machen sich diese Technik schon lange zunutze. Natürlich hängt das Gelingen solcher Angriffe stark vom jeweiligen Einzelfall ab. Häufig ist es nötig, schon etwas an "Insiderwissen" mit zu bringen. So hat jedes Unternehmen bestimmte Bezeichnungen für Abteilungen, Produkte oder auch Leute. Mit Hilfe dieses Wissens ist es dann möglich sich wie ein Mitarbeiter auszudrücken und sensitive Daten zu erlangen.

Ein Beispiel, der Angreifer *Eve* ruft beim Mitarbeiter *Bob* an und gibt sich als Mitarbeiter der EDV aus. Er erzählt ihm jetzt von der Mailserver-Reorganisation und dass er jetzt die nervtötende Aufgabe bekommen hat alle Mailkonten neu einzurichten. Wenn er jetzt *Bob* um sein Passwort bittet, bekommt er es in den meisten Fällen. Sehr oft ist das Passwort dann auch der Schlüssel zu allen anderen Bereichen wie Userlogin mit dem man dann wieder Zugriff auf die Firmenlaufwerke und vielleicht noch Kalenderdaten hat. In einigen Situationen kann es sein, dass *Bob* sein Passwort nicht unbedingt hergeben möchte. Hier reicht es oft anzubieten, dass er sein Passwort für kurze Zeit auf z.B. *test* ändern kann. *Eve* hat jetzt zwar nur diesen einen Zugriff, aber auch hier bekommt er

schon viel zu viele Informationen, die möglicherweise für weitere Angriffe hilfreich sein können.

Die wenigsten User werden überhaupt nichts rausrücken. Allerdings wäre genau das die richtige Vorgehensweise und man kann nur jedem Empfehlen auf solche Anfragen nicht zu reagieren, oder zumindest eine gehörige Portion Misstrauen mitzubringen.

Social Engineering Angriffe werden, wie auch viele andere Angriffe, nur selten bekannt. Vielen Unternehmen fürchten einen Imageschade durch das Bekanntwerden von derartigen Angriffen.

Ein öffentlich bekannter Fall ist der von Kevin Mitnick. Kevin Mitnick war in den 90ern aktiv. Er drang auch mittels Social Engineering in die Netzwerke von Fujitsu, Motorola, Nokia und Sun Microsystems ein. 1995 wurde er als meistgesuchter Mann der Vereinigten Staaten verhaftet. Seit 2000 ist er wieder auf freiem Fuß und arbeitet seit dem als Sicherheitsberater. Außerdem hat er mittlerweile zwei Bücher veröffentlicht.

- "The Art of Deception"
- "The Art Of Intrusion"

2.4 Kelogger

Keylogger protokollieren jeden Tastaturanschlag und ermöglichen es, so an vertrauliche Daten, wie Passwörter, PINs, Loginnamen, besuchte Webseiten, Briefe usw. zu gelangen. Keylogger können entweder als Software oder als Hardware realisiert werden.

Software Keylogger fangen Tastatureingaben ab, bevor diese vom Betriebssystem verarbeitet werden. Die gewonnenen Daten werden entweder versteckt auf dem lokalen Rechner gespeichert, oder direkt über das Internet an einen beliebigen Rechner gesendet. Diese Art von Keyloggern kann als Wurm oder Virus auf den Rechner des Opfers gelangen und ist eine sehr komfortable Methode für den Angreifer, eine hohe Zahl an Rechnern zu erreichen. Voraussetzung ist, dass diese ungenügend geschützt sind. Natürlich ist auch eine Installation vor Ort möglich, dies ist dann meist spezielle Logging Software, oder ein KeyLog Treiber.

Im Gegensatz dazu benötigt man für einen Hardware Keylogger Zugang zum zu überwachenden Rechner. Diese sind in kürzester Zeit angebracht. Sie werden jedoch, trotz ihrer geringen Größe (ca. 2 – 3cm), einfacher entdeckt, da sie zwischen Tastatur und Rechner gesteckt werden. Allerdings hinterlassen sie auch keine Spuren auf den auszustühenden Rechner und können durch keine Sicherheitssoftware aufgespürt werden. Anders als Software-Keylogger kann hier nur eine begrenzte Menge an Informationen gespeichert werden. Sie bieten jedoch mit ca. 500.000 Tastaturanschlägen einen für die meisten Zwecke mehr als genügenden Speicherplatz. Diese Daten müssen nun an einem anderen Computer ausgelesen werden, was wieder einen physischen Zugang zum Rechner erfordert.

Zur Auswertung dieser gewonnenen Daten gibt es eine Vielzahl an Filtersoftware, mit der sich sehr komfortabel nach den gewünschten Informationen suchen lässt.

3 Netzwerksicherheit

3.1 Sniffen und Spoofen

3.1.1 Sniffen

Sniffen kommt aus dem englischen und bedeutet soviel wie "schnüffeln".

Bei dieser Form der Netzanalyse wird der Netzwerkverkehr mitgelesen (geschnüffelt), um so Problemen jeglicher Art auf die Spur zu kommen. Ursprünglich hatte die Firma "Network General" eine solche Software unter dem Namen "Sniffer" vertrieben. Der Begriff wird heutzutage allerdings häufiger als Oberbegriff für solche Soft- oder Hardware genutzt. Auch "protocol"- oder "network analyzer" sind hierfür gängige Begriffe.

Non-Promiscuous mode

Normalerweise kann man mit einem Sniffer nur den eigenen Netzwerkverkehr sniffen. Dies kann schon ausreichend sein, wenn man eigenen Problemen auf die Spur kommen möchte. Es ist möglich, sicherheitskritische Anwendungen auf dem eigenen Rechner zu erkennen da man sieht, welche Daten über das Netz geschickt werden und ob diese verschlüsselt sind. Auch bei der Entwicklung von Netzwerkssoftware kann ein Sniffer hilfreich sein, da man sehen kann wie Verbindungen aufgebaut werden, welche Daten tatsächlich auf dem Netzwerke ankommen, etc..

Promiscuous mode

Viel häufiger möchte man allerdings nicht nur seine eigenen Verbindungen sniffen, sondern ganze Netzwerke. Dieser Modus wird benötigt bei:

- unbekanntem Netzwerkverkehr aufspüren
- Auslastung des Netzwerkes sehen und/oder zuordnen
- Netzwerkstatistiken erstellen
- Angriffe erkennen
- Angreifer beobachten
- Netzwerkanwendungen debuggen
- Daten ausspähen

Um jetzt aber nicht nur seinen eigenen Verbindungen zu sniffen, ist es nötig die Netzwerkkarte in einen Modus zu versetzen, in welchem sie nicht nur die für sie bestimmten Pakete annimmt, sondern auch alle nicht für sie bestimmten. Diesen Modus nennt man "promiscuous mode". Es ist möglich alle Pakete zu empfangen, die über das Netzwerkkabel ankommen.

Sniffen mit Hubs Es kommt jetzt noch auf die Netzwerk-Topologie an um zu wissen, welche Pakete den Sniffer überhaupt erreichen können. In einem mit Hubs verbundenem Netzwerk ist der gesamte Netzwerkverkehr sichtbar. Der Grund hierfür ist, dass ein Hub die Pakete die an einem Port ankommen einfach an alle Ports weiterleitet. Somit "sieht" ein Rechner auch alle nicht für ihn bestimmten Pakete.

Sniffen mit Switches Üblicher sind heute allerdings geschaltete Netzwerke. Ein Switch ist im Gegensatz zum Hub "intelligenter", da er weiß welche Computer an welchen Port angeschlossen sind. Mit diesem Wissen kann der Switch erkennen wohin ein Paket soll und es dann nur an diesen Port weiterleiten. Der Sniffer "sieht" jetzt wieder nur die Pakete die auch tatsächlich für ihn bestimmt sind.

Remote analyzer

Heute ist es möglich an einem Switch ein "Monitorport" einzurichten. Dabei wird dem Switch gesagt, dass er sämtlichen Netzwerkverkehr auf jeden Fall auch an den Monitorport weiterleitet. An diesem einen Port ist es nun wieder möglich zu sniffen. Bei großen Netzen kann es unter Umständen schwierig werden, in jedem Netzsegment von "Hand" zu sniffen. Aus diesem Grund greift man auf "remote analyzer" zurück. Hierbei wird in jedem Segment ein Sniffer an einen Monitorport platziert. Dieser Sniffer fungiert als Server. Er arbeitet wie ein normaler Sniffer, sendet die Daten aber dann an den eigentlichen Sniffer weiter. So ist es möglich, auch in fremden Netzsegmenten zu sniffen.

Software

Die Liste bekannter Sniffer von Wikipedia umfasst folgende Produkte:

- AiroPeek
- dSniff
- Ethereal (seit Juni 2006 Wireshark)
- EtherPeek
- Ettercap
- Sniffer
- OmniPeek

- PRTG
- tcpdump

Noch vor 15 Jahren gab es ausschließlich kommerzielle Produkte zum Sniffen. Seit ende der 90er hat sich dies jedoch verändert. So ist Ethereal, wie tcpdump und Ethercap auch, ein frei erhältliches Tool. Gerade Ethereal steht den kommerziellen Tools heute um nichts mehr nach.

3.1.2 Spoofen

Spoofen kommt ebenfalls aus dem englischen und bedeutet soviel wie "manipulieren". Beim Spoofing wird generel etwas manipuliert. Es gibt verschiedene Arten des Spoofings:

- ARP-Spoofing
- IP-Spoofing
- DNS-Spoofing
- Mail-Spoofing
- URL-Spoofing

Alle hier aufgeführten Spoofing-Methoden haben einen sehr unterschiedlichen Hintergrund. Auch der Nutzen ist von der einen zur anderen Methode ein völlig unterschiedlicher. So wird Mail-Spoofing z.B. eingesetzt um SPAM verschicken zu können. Im Zusammenhang mit einem Sniffer ist allerdings das ARP-Spoofing am interessantesten. Daher wird hier nur auf das ARP-Spoofing eingegangen.

Anwendungsgebiete

Es gibt für ARP-Spoofing mehrere Anwendungsgebiete. Diese reichen von legitimen Anwendungen wie PacketFence bis hin zu kritischen Man in the Middle (MitM) Attacken.

PacketFence ist ein Produkt welches alle unregistrierten Rechner per ARP-Spoofing auf eine Anmeldeseite umlenkt. Nur Computer die im Netzwerk bekannt sind, können dieses dann auch nutzen. Über die Tauglichkeit dieser Methode lässt sich streiten. PacketFence ist aber ein Beispiel dafür, dass Spoofing-Techniken nicht per default kritisch sein müssen.

Möchte man in einem geschwittem Netz sniffen, so ist dies auch über ein ARP-Spoofing möglich. Hierbei wird der Netzwerkverkehr derart umgebogen, dass bestimmte (oder auch alle) Verbindungen, über den Rechner des Sniffers geleitet werden. Es ist dann wieder möglich, mit einem Sniffer den Netzwerkverkehr aufzuzeichnen und hinterher auszuwerten. Es handelt sich hierbei im Prinzip schon um einen MitM Angriff. Jedoch wird diese Technik desweilen auch von Systemadministratoren angewendet, um in ihrem Netzwerk sniffen zu können.

ARP Cache Poisoning

Die Technik, mit der ARP-Spoofing funktioniert, nennt man ARP-Cache Poisoning. Der Name beschreibt schon treffend, was hier tatsächlich gemacht wird. Aber fangen wir vorne an.

ARP, das Adress Resolution protokoll, ist für die Umsetzung von MAC-Adressen zu IP-Adressen zuständig. Angenommen wir haben folgende Netzwerkkonfiguration:

Alice	MAC: aa:aa:aa:aa:aa:aa	IP: 10.0.0.1
Bob	MAC: bb:bb:bb:bb:bb:bb	IP: 10.0.0.2
Eve	MAC: ee:ee:ee:ee:ee:ee	IP: 10.0.0.3

Möchte Alice nun mit Bob kommunizieren und hat bereits dessen IP, muss im nächsten Schritt die MAC-Adresse von Bob ermittelt werden. Dazu stellt Alice eine ARP-Anfrage an die Broadcast-Adresse, in welcher nach der MAC-Adresse zu der IP 10.0.0.2 gefragt wird. Bob wird nun mit einer ARP-Antwort an Alice antworten und mitteilen, dass die IP 10.0.0.2 die MAC-Adresse bb:bb:bb:bb:bb:bb hat. Jetzt speichert sich Alice diese Zuordnung noch in einer lokalen Tabelle, damit nicht für jede weiter Kommunikation die Auflösung immer neu gemacht werden muss.

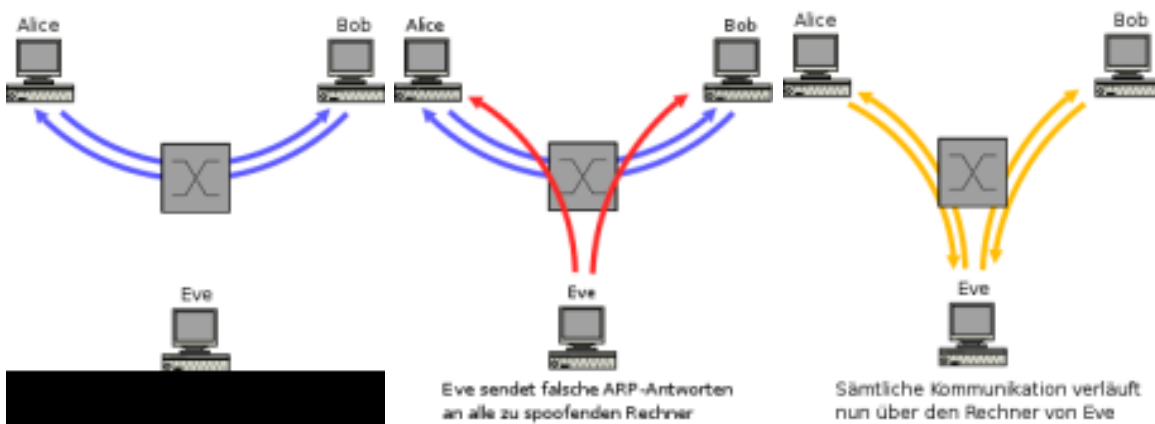


Abbildung 3.1: Ein Spoofing-Angriff

Das ARP-Protokoll ist sehr einfach aufgebaut und geht von einer vertrauenswürdigen Umgebung aus. Eine ARP-Antwort führt immer dazu, dass der Computer daraufhin seine lokale ARP-Tabelle ändert. Dies geschieht auch dann, wenn gar keine ARP-Anfrage abgeschickt wurde.

Möchte Eve jetzt die Kommunikation zwischen Alice und Bob belauschen/sniffen, so ist dies mit der Hilfe von zwei ARP-Antworten möglich.

```
Sender MAC-Adresse: ee:ee:ee:ee:ee:ee   Ziel MAC-Adresse: aa:aa:aa:aa:aa:aa
Sender  IP-Adresse: 10.0.0.2             Ziel  IP-Adresse: 10.0.0.1
```

Listing 2: ARP-Antwort 1

```
Sender MAC-Adresse: ee:ee:ee:ee:ee:ee   Ziel MAC-Adresse: bb:bb:bb:bb:bb:bb\\
Sender  IP-Adresse: 10.0.0.1             Ziel  IP-Adresse: 10.0.0.2
```

Listing 3: ARP-Antwort 2

Nachdem Alice und Bob die gefälschten Pakete erhalten haben, tragen sie die neue Zuordnung in Ihre lokal ARP-Tabelle ein. Jetzt werden alle Pakete von Alice für Bob an Eve geschickt. Hat Eve seinen Rechner so konfiguriert, dass die Pakete auch noch weitergeleitet werden, dann merken weder Alice noch Bob, dass ihre Verbindung nicht mehr vertrauenswürdig ist.

Es ist allgemein nicht einfach ARP-Spoofing zu erkennen. Allerdings gibt es mittlerweile Implementierungen, welche sich genau dies zum Ziel gemacht haben. Hierbei wird nicht mehr einfach jede ARP-Antwort in die ARP-Tabelle eingetragen. Vielmehr wird überprüft, ob eine ARP-Anfrage gesendet wurde und nur falls dies der Fall ist, wird die dazugehörige ARP-Antwort in die ARP-Tabelle geschrieben. Man könnte auch auf ARP ganz verzichten und die Zuordnung von MAC zu IP fest zu 'verdrahten'. Allerdings ist dies in der Realität kaum durchführbar.

ARP-Spoofing Tools

- Ettercap - Ein Sniffer, der ARP-Spoofing beherrscht
- arprelay - Die ARP-Spoofing Implementierung von Felix von Leitner
- Cain & Able - Ein ARP-Spoofing unter Windows funktioniert

3.2 (Distributed) Denial of Service

"Denial of Service", das bedeutet soviel wie "außer Betrieb" und beschreibt eine Angriffsmethode in der Computersicherheit die zur Folge hat, dass danach etwas "außer Betrieb" ist. Genauer, unter einem "Denial of Service" Angriff versteht man einen Angriff, welcher dazu führt, dass ein Dienst, ein Server oder auch ein ganzer Rechner seine eigentliche Aufgabe nicht mehr erfüllen kann.

Es gibt auch Angriffe, bei denen nicht der Rechner selber das Opfer ist, sondern der Nutzer. Eine Mailbombe führt so dazu, dass die Mailbox eines Users so mit sinnlosen Mails überflutet wird, dass er seinen "normalen" Emails nicht mehr herausfiltern kann. Allerdings kommen wir heute damit wieder besser zurecht, da man ja täglich Opfer von unzähligen SPAM-Mails ist. SPAM könnte daher als distributed denial of Service Attacke gesehen werden. Aber fangen wir vorne an.

DoS-Angriffe können viele Ursachen haben. So können Server z.B. durch extrem vielen Anfragen dazu gebracht werden, dass sie überlastet werden und ihren Dienst verweigert.

Weiter können Fehler im Programmcode aber auch Fehler im Softwarekonzept ausgenutzt werden um selbige Programme zur Aufgabe zu zwingen.

3.2.1 Ping of Death

Vor einiger Zeit, als Windows noch kein Betriebssystem war, gab es einen DoS-Angriff, der viele damalige Systeme zum sofortigen Absturz brachte. Der Ping of Death war ein IP-Pakete, welches sich einen Programmfehler im IP-Stack der damaligen Systeme zunutze machte. Ein IP-Packt darf laut seinem Header nur 2^{16} Bit Nutzinformationen also 65.536 Bytes mit sich bringen. Einige Implementationen der IP-Stacks prüften die Größe eingehender Pakete nämlich nicht gegen ≤ 65535 sondern gegen < 65535 . Dieses eine fehlende Zeichen war ein so gravierender Fehler, dass ein IP-Packe mit 65.536 Bytes Nutzinformation, was ja ein vollkommen legitimes Paket ist, die Systeme zum Absturz brachte. Mit Windows 95/NT 4.0 und unter Linux ab dem Kernel 2.0, gab es diese Probleme nicht mehr.

3.2.2 Flooding

TCP SYN-Flooding

Das TCP SYN-Flooding beschreibt einen weiteren, sehr einfachen DoS-Angriff. Um diesen Angriff verstehen zu können muss man ein wenig über den Verbindungsaufbau des TCP-Protokolls bescheid wissen. TCP nutzt zum aufbauen von Verbindungen den sogenannten 3-Wege-Handshake.

- Zuerst schickt der Client dem Server eine Verbindungsanfrage (SYN-Flag gesetzt).
- Danach antwortet der Server auf diese Verbindungsanfrage (SYN- und ACK-Flag gesetzt).
- Zuletzt bestätigt der Client dies nochmal mit einem Pakete (ACK-Flag gesetzt).

Schickt nun ein böartiger Client dem Server nur das erste Paket, mit dem gesetzten SYN-Flag, so muss der Server trotzdem eine Weile abwarten, bis er den Verbindungsaufbau als gescheiterten Versuch verwirft. Das letzte Paket des Clients könnte sich ja ein wenig verspäten. Das heißt aber, dass der Server diese "halb offene"-Verbindung im Speicher haben muss. Kein Server hat unendlich viel Speicher, woraus folgt, dass er nur eine bestimmte Anzahl von Verbindungen gleichzeitig bearbeiten kann.

Der böartige Client könnte jetzt den Server mit solchen sogenannten SYN-Paketen überfluten (flooding), so dass der Server keine anderen Verbindungen mehr annehmen kann. Dies führt, sofern keine Abwehrmaßnahmen getroffen wurden, dazu, dass der Server für andere Clients nicht mehr erreichbar ist. Im schlimmsten Fall könnte der Server auch zum Absturz gebracht werden.

Es sollte noch gesagt werden, dass diese Art des DoS-Angriffs für den Angreifer sehr einfach ist. Für den Verteidiger ist es doch mit einigem Aufwand verbunden, einen solchen Angriff abzuwehren.

UDP Flood

Eine weitere Angriffsmöglichkeit bietet das UDP-Flooding. Hierbei werden dem Server sehr viele UDP-Pakete an zufällige Ports geschickt. Dies hat zur Folge, dass der Server für jedes ankommende Paket prüfen muss, ob es einen Dienst hinter diesem Port gibt. Da es sich aber um zufällig generierte Portadressen handelt, wird dies in der Regel nicht der Fall sein. Zuletzt muss er dann noch ein "ICMP Destination Unreachable"-Paket verschicken. Je mehr UDP-Pakete man dem Server zukommen lässt, um so höher ist auch der Aufwand für den Server. Die Aufwandssteigerung ist jedoch nicht proportional, so muss der Angreifer nur ein UDP-Paket generieren. Der Server muss jedoch für jedes UDP-Paket die oben erklärten Prüfungen mit der Fehlerrückmeldung machen. Unter Umständen kann man durch diesen Angriff erreichen, dass der Server für andere Dienste nicht mehr zur Verfügung steht.

Der Angreifer hat außerdem auch noch eine sehr einfache Methode unerkannt zu bleiben. Eine Eigenschaft des User Datagramm Protocols ist, dass es ein verbindungslosen Protokoll ist. Der nicht vorhandene Verbindungsaufbau hat zur Folge, dass der Server jedes UDP-Paket als möglicherweise gültiges Datenpaket behandeln muss. Der Angreifer erzielt daher den gleichen Effekt beim Server, wenn er das Paket mit einer falschen Absenderadresse abschickt. Der Angreifer verhindert somit das feststellen des tatsächlichen Urhebers. Das Opfer kann auch nicht nur Traffic eines bestimmten Absenders blocken, um sich zu schützen.

ICMP Flood

Einer der einfachsten Angriffe ist das ICMP Flooding. Bei diesem Angriff muss der Angreifer eine bessere Netzanbindung haben als sein Opfer. Außerdem muss das Opfer auch auf ein ICMP Echo Request Paket mit einem ICMP Echo Replay Paket antworten. Jetzt ist es dem Angreifer möglich einfach so viele Pings an das Opfer zu schicken, dass die gesamte Bandbreite des Opfers für die Bearbeitung der Ping-Paket aufgebracht werden muss. Vor allem bei besonders unterschiedlichen Bandbreite-Verhältnissen funktioniert diese Angriffsmethode.

Der Angriff kann auch noch verschärft werden, indem man in einem ICMP-Paket nicht nur 64Bytes Nutzdaten mitschickt, sondern gleich 65kB. Das Opfer muss so die Mehrdaten auch in der Replay mit zurückschicken. Die Leitung geht somit um einiges früher in die Knie.

Wieder ist es dem Angreifer möglich, seine eigentliche Identität geheim zu halten, da er wieder die Absenderadresse spoofen kann. Auch kann das Opfer damit nicht einfach den Traffic einer Adresse blocken. Der einzig wirksame Schutz ist, sämtliche Ping-Requests durch einen Packetfilter heraus zu filtern.

Smurf Attack

Der Smurf-Attack hat seinen Namen von der Software, welche diesen Angriff ausgenutzt hat. Dieser Angriff funktioniert heutzutage nicht mehr. Viele DDoS-Angriffe funktionieren allerdings nach ähnlichen Prinzipien.

Bei einem Smurf-Attack schickt der Angreifer Ping-Pakete an die Broadcast-Adresse eines Netzwerks. Die Rechner in dem Netzwerk antworten ihrerseits mit einem Replay-Paket an den Absender des Pings. Der Angreifer kann jetzt natürlich wieder die Absenderadresse spoofen, so dass sämtliche Antworten den Rechner treffen, der tatsächlich die gespoofte IP hat. Das Opfer wird also mit einer Flut von ICMP Replay-Paketen konfrontiert, die er garnicht ausgelöst hat. Der Aufwand für den Angreifer ist gerade bei diesem Angriff sehr gering, die Datenflut die das Opfer trifft ist aber um ein vielfaches größer.

3.3 Wlan

WLAN steh für Wireless LAN und bezeichnet ein drahtloses lokales Computernetzwerk nach dem IEEE 802.11¹ Standard. Sie finden überall dort Einsatz wo eine feste Verkabelung nur schwer oder gar nicht möglich ist. Der einfache Aufbau sowie die hohe Mobilität sind weitere Kriterien, die für ihren Einsatz sprechen.

3.3.1 Einführung

Speziell durch die rasante Verbreitung von DSL erhöht sich die Zahl der WLAN Hotspots in jüngster Zeit sehr schnell. Waren früher WLAN Hotspots nur in größeren Firmen und Organisationen zu finden, so sind sie mittlerweile für jeden Normalbürger, oft kostenlos, erhältlich. Durch diese hohe Zahl und die Überwiegende Unkenntnis der Nutzer ergeben sich hohe Sicherheitsrisiken. Das folgende Kapitel ist eine Übersicht über die Funktionsweise und die Schwachstellen dieser Technologie.

3.3.2 Leistungen

Übertragungsraten

Der zum Mai 2006 am weitesten verbreitete Standard ist 802.11g mit einer Geschwindigkeit von 54MBit/s. Durch proprietäre Lösungen sind jedoch Übertragungsraten möglich die selbst FastEthernet mit 600 Mbps übertreffen. Nachfolgend eine Übersicht über zur Zeit gültige WLAN Standards.

Standard	Datenrate
IEEE 802.11	2 Mbps maximal
IEEE 802.11a	54 Mbps maximal
IEEE 802.11b	11 Mbps maximal
IEEE 802.11g	54 Mbps maximal
IEEE 802.11h	54 Mbps maximal
IEEE 802.11n	600 Mbps maximal

Listing 4: WLAN-Datenraten

¹http://de.wikipedia.org/wiki/IEEE_802.11

Reichweite

Handelsübliche 802.11 WLAN Adapter schaffen Reichweiten von 30 bis 100 Metern im freien. Mit externen Antennen lässt sich die Reichweite um ca. dem Faktor 3 steigern. Die Reichweiten innerhalb von Gebäuden sind sehr stark von der Bauweise abhängig. Durch spezielle Richtfunkantennen lassen sich selbst Reichweiten von einigen 100 Kilometern erreichen². Dies liegt aber meistens außerhalb des gesetzlichen Rahmens, und setzt Sichtverbindung voraus.

3.3.3 Betriebsmodi

WLANs kennen 2 Betriebsmodi, Infrastruktur Modus und Ad-hoc Modus.

Im Infrastruktur Modus wird ein Netzwerkknoten zur Koordinierung der restlichen Knoten verwendet. Dieser wird allgemein als Access Point bezeichnet. Über eine Ethernet Schnittstelle kann auf einfache Art und Weise eine Verbindung zu kabelgebundenen Netzen hergestellt werden. Der Access Point arbeitet in diesem Fall als eine Art "Vermittler" zwischen dem kabelgebundenen Netz und dem Funknetz. Dies ist die am weitesten verbreitete Form eines WLANs.

Der Ad-hoc Modus kennt nur gleichwertige Netzwerkknoten und lässt sich daher einfach und bequem aufbauen. Pakete werden in diesem Netzwerk nicht weitergereicht, d.h. Knoten die Daten austauschen wollen müssen sich in Reichweite befinden. Die Verbindung zu einem kabelgebundenen Netz muss hier jeder Knoten selbst herstellen. Ad-hoc Netze werden meist nur kurzzeitig zum schnellen Austausch von Daten unter einer kleinen Anzahl von Knoten verwendet.

3.3.4 Sicherheit

Der Verschlüsselungsstandard im WLAN Bereich war lange Zeit WEP. WEP steht für Wired Equivalent Privacy und basiert auf dem RC4-Algorithmus. Üblich sind Verschlüsselungen mit 40 bis 104 Bit. Da sich dieses Verfahren als ziemlich unsicher herausgestellt hat wurden über die Zeit Erweiterungen und neue Verschlüsselungsverfahren entwickelt. Da wären z.B. WEP-Plus, WPA, WPA2, Kerberos u.a. Im Rahmen dieses Kapitels wird auf die 2 verbreitetsten Verfahren, WEP und WPA eingegangen.

WEP

Der Zweck des Wired Equivalent Privacy Verfahrens ist, die Gewährleistung der Vertraulichkeit und Integrität der Daten.

Es basiert auf einem geheimen Schlüssel, welcher zwischen dem mobilen Gerät und dem Accesspoint ausgetauscht wird. Damit wird der Datenstrom verschlüsselt und anschließend eine Prüfsumme gebildet. Zur Verschlüsselung verwendet WEP den RC4 Algorithmus um aus einem kurzen Schlüssel einen "unendlichen" pseudo-zufälligen Schlüssel zu erzeugen. Dieser erzeugte Schlüssel wird mit dem Datenstrom XOR verknüpft und ergibt so das verschlüsselte Datenpaket. Der Client besitzt den gleichen Schlüssel und kann

²<http://www.golem.de/0604/44975.html>

somit durch eine weitere XOR Verknüpfung den ursprünglichen unverschlüsselten Datenstrom wiederherstellen. Werden 2 Pakete abgefangen, die mit dem gleichen Schlüssel verschlüsselt wurden, so ist der Schlüssel über eine XOR Verknüpfung der beiden Pakete herausfindbar. Dazu wird ein statistischer Angriff gestartet, dessen Erfolg mit der Anzahl der Pakete steigt. Ist dann ein Paket im Klartext vorhanden, ist die Entschlüsselung der restlichen Pakete kein Problem mehr.

Damit nicht jedes Paket mit dem gleichen Schlüssel verschlüsselt wird setzt WEP auf einen Initialisierungs Vector. Dieser besteht aus 24 Bit und wird mit jedem Paket im Klartext übertragen. Das bedeutet, dass nach spätestens 2^{24} Paketen die Schlüssel sich wiederholen. Auch da der IEEE 802.11 Standard nicht zwingend vorschreibt den Schlüssel zu verändern ist es theoretisch möglich jedes Paket mit dem selben Schlüssel zu verschlüsseln, ohne dem Standard zu verletzen.

Die Integrität der Daten wird mittels CRC-32 sichergestellt. Da dies aber ein linearer Algorithmus ist, kann man durch Ändern eines Bits ein Bitmuster errechnen, das geändert werden muss, um wieder eine gültige Prüfsumme zu erhalten. Somit kann man dem Netzwerk scheinbar gültige Pakete unterschieben.

WPA

Wi-Fi Protected Access (WPA) basiert im Wesentlichen auf WEP, erweitert dieses jedoch um dynamische Schlüssel und erweiterte Authentifizierungsmethoden.

WPA unterstützt nicht nur einen 24bittigen Initialisierungsvektor, sondern auch Per Paket Mixing, Rekeying und einen Message Integrity Check. Dieser verhindert schon einige bekannte Angriffe indem Datenpakete nummeriert werden und diese Daten im verschlüsselten Teil der Nachricht übertragen werden. Pakete die nicht zu dieser Nummer passen, werden vom Client verworfen.

Authentifizierung geschieht bei WPA auf zwei unterschiedliche Arten, PSK oder EAP. PSK wird häufig im privaten Bereich eingesetzt. Aber auch in sehr kleinen Geschäftsumgebungen findet PSK seinen Einsatz. Dieses Verfahren basiert auf einem Schlüssel, der allen Teilnehmern im LAN bekannt sein muss. Die Schwachstellen liegen hier beim Passwort, welches über Brute Force oder Wörterbuch Attacken erraten werden kann und somit der Schlüssel generiert werden kann

EAP wird meist in größeren Firmennetzwerken eingesetzt, da hier häufig zusätzliche Authentifizierungsserver zur Verfügung stehen. Dies wird von EAP unterstützt und erhöht die Sicherheit des Netzwerks. EAP unterstützt außerdem mehrere Authentifizierungsverfahren die auch in Kombination genutzt werden können. Es gibt zu Zeit etwa 40 verschiedene Verfahren. Nicht alle Verfahren sind RFC-Konform.

3.4 VPN

3.4.1 Einführung

Ein VPN (Virtuelles Privates Netzwerk) ist ein Netzwerk, welches private Daten über ein öffentliches Netzwerk transportiert. Die Clients müssen dazu nicht untereinander

verbunden sein, da dies durch den VPN Server erreicht wird. Für den Einzelnen sieht es so aus, als würden sich alle Clients im selben Netz befinden. Üblicherweise sind VPNs verschlüsselt, was jedoch nicht zwingend erforderlich ist. Normalerweise werden sie in Unternehmen eingesetzt, um Mitarbeitern den Zugriff von außen zu ermöglichen. Es besteht die Möglichkeit zwei Netze zu verbinden, oder bspw. zwei Filialen über das Internet zusammenzuführen. Die dritte Möglichkeit ist zwei Clients über VPN zu verbinden, sie wird jedoch kaum genutzt.

3.4.2 Sicherheit

Es gibt verschiedene Möglichkeiten ein VPN abzusichern. Dazu gehören Public Keys, Zertifikate und Passwörter.

Die am weitesten verbreitete Implementierung ist IPSec. Es wurde 1998 mit dem Ziel entwickelt Vertraulichkeit, Authentizität und Integrität über IP Netzwerke zu gewährleisten. Es arbeitet auf Schicht 3 des OSI Schichtenmodells. Die wichtigsten IPSec-Protokolle sind:

- Authentication Header Protokoll (AH)
- Encapsulated Security Payload Protokoll (ESP)
- Internet Key Exchange Protokoll (IKE)

Die genauen Anforderungen sind in folgenden RFCs spezifiziert:

- RFC 2401
- RFC 2402
- RFC 2407
- RFC 2408
- RFC 2409
- RFC 4301
- RFC 4302
- RFC 4303
- RFC 4306

Der Schlüsselaustausch kann auf zwei Arten erfolgen. Zum einen kann man den Schlüssel an beiden Enden fest konfigurieren. Dies ist jedoch unsicher und sollte vermieden werden. Zum anderen kann man IKE (Internet Key Exchange) nutzen. Dieses Protokoll dient dem Austausch von Schlüsseln über ein unsicheres Netzwerk. Es definiert wie Schlüssel ausgetauscht werden und Parameter definiert werden. IKE arbeitet in 2 Phasen.

- Aushandlung einer Security Association

- Erzeugung einer Security Association
- 1. Identifikation
- 2. Schlüsselalgorithmus
- 3. Source IP
- 4. Destination IP
- 5. Zeiträume in denen eine neue Authentifizierung erfolgen muss
- 6. Gültigkeit(Zeit) des Schlüssels

Phase 1 Hier gibt es erneut 2 Möglichkeiten, den Main Mode und den Aggressive Mode.

Der Main Mode funktioniert folgendermaßen. Er läuft in der ersten Phase der Verschlüsselungsvereinbarung ab wobei die 2 Endpunkte direkt beteiligt sind. Sie Verhandeln über die Security Association, was wiederum in 5 Schritten geschieht.

1. Es werden Vorschläge mit Authentifizierungs- und Verschlüsselungsalgorithmen zum Client gesendet
2. Der Client wählt, aus den von ihm unterstützten Verfahren, das sicherste aus
3. Der Server sendet den öffentlichen Teil des DHS³ und einen Zufallswert an den Client
4. Der Client sendet seinen öffentlichen Teil des DHS und ebenfalls einen Zufallswert zurück
5. Die Verbindung wird Authentifiziert.

Da Client und Server die öffentlichen Teile für den Diffie-Hellman-Schlüsselaustausch kennen, wird dieses Verfahren genutzt, um den geheimen Schlüssel zu berechnen. Dieser wird dann für die Verschlüsselung nach dem vereinbarten Schlüsselverfahren für die folgenden Schritte verwendet, sowie zur Erzeugung eines Schlüssels für die Authentifizierung. Hierfür müssen sich beide als zugriffsberechtigt ausweisen. Dabei kommen zwei unterschiedliche Verfahren zum Einsatz: die Authentifizierung mittels PSK und die zertifikatsbasiert Authentifizierung. Die Authentifikationsmethoden unterscheiden sich zwar, jedoch ist das grundsätzliche Vorgehen immer das gleiche: Es wird ein Hashwert über das mit dem Diffie-Hellman-Schlüsselaustausch erzeugte Geheimnis, die Identität, den ausgehandelten Kryptoverfahren sowie den bisher versandten Nachrichten gebildet, verschlüsselt und versendet. Der Schlüssel, der hier für die Verschlüsselung genutzt wird, ist jedoch nicht der aus dem Diffie-Hellman-Schlüsselaustausch, sondern ein Hashwert über ihn sowie die versandten Nachrichten.

Der Aggressive Mode fasst die oberen 5 Schritte in 3 Schritten zusammen. Zusätzlich wird der 5 Schritt nicht verschlüsselt, sondern die Übertragung erfolgt im Klartext.

³Diffie-Hellmann-Schlüsselaustausch

Phase 2 Quick Mode

Der Quick Mode verläuft in der zweiten Phase des IKE. Die gesamte Kommunikation in dieser Phase erfolgt verschlüsselt. Wie in der ersten Phase wird ein Vorschlag gesendet, welcher zusammen mit einem Hashwert einem Zufallswert übertragen wird. Später werden die Schlüssel neu berechnet, und es fließen keinerlei Informationen aus den zuvor generierten SA ein. Dies garantiert, dass niemand aus den zuvor erzeugten Schlüsseln auf die neuen schließen kann. Dies wird durch einen weiteren DHS erreicht.

3.4.3 Transport

Der Transport der Daten erfolgt entweder im Transportmodus oder im Tunnelmodus. Genau genommen ist der Transportmodus ein Teil des Tunnelmodus.

Im Transportmodus wird der IPsec-Header zwischen IP-Header und Nutzdaten eingefügt. Der IP Header bleibt unverändert. Nach dem Empfang des IPsec-Paketes werden die ursprünglichen Daten ausgepackt und an die höherliegende Schicht weitergegeben. Der Transportmodus wird z.B. für Client-Client oder Client-Router Verbindungen verwendet.

Im Tunnelmodus wird das komplette Paket verschlüsselt. Es wird mit einem IPsec-Header und einem neuen IP-Header versehen, wodurch das ursprüngliche Paket gekapselt wird und die Sicherheitsdienste von IPsec auf das komplette Paket angewendet werden. Der neue IP-Header dient dazu, die Tunnelenden zu adressieren, während die Adressen der eigentlichen Kommunikationsendpunkte im inneren IP-Header stehen. Der ursprüngliche IP-Header wird erst wieder verwendet, wenn das andere Tunnelende die Kapselung entfernt dem eigentlichen Empfänger zustellt. Der Tunnelmodus wird vor allen zur Verbindung von Netzwerken verwendet.

3.5 Firewalls

Eine Firewall ist ein Soft-/Hardwaresystem, welches den Zugriff zwischen verschiedenen Rechnernetzen einschränkt. Der häufigste Einsatz einer Firewall ist es, das lokale Netz vom Internet zu trennen. Prinzipiell kann man aber den Verkehr zwischen allen Netzwerkkarten über Firewalls kontrollieren. Dies bedeutet das eine Firewall das Senden und Empfangen von Daten aus und in den geschützten Bereich kontrolliert.

Je nach Einsatz kann man verschiedene Arten von Firewalls unterscheiden:

- Hostfirewall
- Netzwerkfirewall

Häufig wird auch von Software oder Hardwarefirewalls gesprochen, allerdings ist diese Definition nicht sinnvoll, da zu jeder Firewall Soft- und Hardware gehört. Als Softwarefirewall werden sogenannte Personal Firewalls bezeichnet, welche auf dem lokalen Rechner ausgeführt werden. Firewalls die auf Routern ausgeführt werden, bezeichnet man als Hardwarefirewalls. Diese Firewalls sind natürlich auch von Software, insbesondere

von einem Betriebssystem abhängig. Hier gelten proprietäre Firewall-Betriebssysteme als sicher, da sie speziell für diesen Zweck entworfen wurden und die möglichen Lücken eines unterliegenden Standard-Betriebssystems nicht aufweisen. Allerdings bietet dies auch keinen 100% Schutz, da diese Systeme auch Lücken aufweisen können.

3.5.1 Hostfirewall

Wird die Firewallsoftware auf einem Unix-/Windows-/Sun-Rechner ausgeführt, so spricht man von einer Hostfirewall. Diese arbeitet auf den Schichten 2 bis 7 des OSI Schichtenmodells und besteht üblicherweise aus verschiedenen Komponenten. Die wichtigsten werden nachfolgend beschrieben. Eine Untergruppe dieser Hostfirewalls sind Personal Firewalls. Diese laufen üblicherweise auf einer Workstation, die primär zu anderen Zwecken genutzt wird. Diese eignen sich dann nur dazu, den Verkehr zum/vom lokalen Rechner aus zu kontrollieren.

1. Applicationlayer Firewall: Diese Komponente arbeitet auf Schicht 7 des OSI Schichtenmodells und kontrolliert bspw. den Inhalt von HTML Seiten auf Viren vor den Auslieferung.
2. Proxy: In vielen Firewalls arbeiten transparente Proxys, deren Funktionen allerdings eingeschränkt sind. So können keine aktiven Eingriffe in den Datenstrom erfolgen sondern es ist meistens nur Blacklisting erlaubt.
3. Content Filter: Sie untersuchen den Inhalt von Datenpaketen und erlauben es so z.B. JavaScript oder ActiveX Inhalte aus HTML Seiten zu filtern, oder Spam Mails zu markieren bzw. zu löschen. Sie können auch verhindern, dass vertrauliches Material aus dem Netz nach außen gelangt. Dies ist ein sehr komplexes System dessen Realisierung nicht immer möglich bzw. Aufgabe einer Firewall ist. Zum Beispiel ist das Erkennen von Spam-E-mails Aufgabe eines Spam Filters, welcher alle Pakete einer Email erst zusammensetzt und diese dann auf Viren überprüft.
4. Paket Filter: Dies ist eine Vereinfachung eines Content Filters und untersucht Daten nur auf Paketebene. Diese Filter arbeiten mit Regeln, welche festlegen was mit bestimmten Arten von Paketen geschehen soll. So können Pakete verworfen, zurückgeschickt und weitergeleitet werden. Man kann so Ports sperren, oder bestimmte IP-Adressen blockieren, aber auch bestimmte Protokollarten. Die Regeln können auch kombiniert werden.

3.5.2 Netzwerkfirewall

Eine Netzwerkfirewall ist eine Kombination aus Soft- und Hardware welche genau auf diesen Einsatz abgestimmt wurde. Diese besitzt mehrere LAN Anschlüsse um die verschiedenen Netze physisch zu trennen. Es gibt dabei mindestens einen externen Anschluss (WAN) und einen lokalen Anschluss (LAN). Damit wird gewährleistet dass nur der Verkehr zwischen den Netzen stattfindet, den die Software als gültig angesehen hat.

Üblicherweise gibt es aber auch noch einen oder mehrere Anschlüsse für DMZ⁴. In diesen DMZ stehen Server die Dienste anbieten, welche aus dem WAN erreichbar sein sollen.

3.6 DMZ

DMZ steht für Demilitarized Zone und bezeichnet ein Netzwerksegment, welches durch Filterung von einem lokalen Netzwerk abgeschirmt ist. Ziel der DMZ ist es, Verkehr der durch die Bereitstellung von Diensten für externe Netze entsteht vom lokalen Netzwerk zu trennen und somit dieses vor Angriffen zu schützen. Dazu wird der Verkehr meistens durch eine Firewall gefiltert und Anfragen für bestimmte Dienste nur an die konfigurierten Server geschickt. Ist eine DMZ auch physisch, über einen eigenen LAN Anschluss von den übrigen Netzen getrennt, so nennt man das auch protected DMZ. Oft werden für verschiedene Dienste auch verschiedene DMZ eingerichtet damit bei Kompromittierungen nicht alle Rechner betroffen sind, sondern nur das jeweilige Netzwerksegment.

Viele einfache Router bieten die Möglichkeit eine DMZ einzurichten, jedoch handelt es sich hier meistens nur um einen Exposed Host. Das heißt alle Anfragen von außen werden an diesen einen Rechner geleitet. Dies kann bei schlechter Umsetzung einen DoS Angriff ermöglichen. Darum sollten solche Lösungen nicht alle von außen ankommenden Anfragen weiterleiten, sondern nur vorkonfigurierte Dienste. Vorteil dieser Lösung ist es, dass der Router auch als Proxy eingesetzt werden kann und somit der Verkehr protokollier- und filterbar ist.

⁴Demilitarized Zone

4 Websicherheit

4.1 XSS - Cross-Site Scripting

4.1.1 Einleitung

Cross-Site Scripting (XSS) wird durch Schwachstellen in dynamisch erstellten Webseiten ermöglicht, bei denen beispielsweise Inhalte ohne ordnungsgemäße Validierung aufgrund von böswilligen Benutzereingaben erstellt werden. So lassen sich z. B. Inhalte von Seiten manipulieren, um in Formulare eingegebene Daten auszulesen. Die vom Angreifer zur Manipulation der Web-Anwendung benötigten Daten werden üblicherweise in Form eines präparierten Hyperlinks übermittelt.

Die Bezeichnung *Cross-Site* bezieht sich auf die webseitenübergreifende Angriffsart (Hinweis: Cross-Site-Scripting wird oftmals mit CSS abgekürzt, darf jedoch nicht mit der Technologie der Cascading Style Sheets (CSS) verwechselt werden.)

4.1.2 Vorgehensweise

Der Anwender klickt beispielsweise auf einen vom Angreifer manipulierten Hyperlink der JavaScript Code enthält. Auf dem Server auf den der Link verweist läuft ein falsch konfiguriertes CGI Script das bei einem durch JavaScript präpariertem Datenpaket ermöglicht das Java-Script auszuführen. Oft wird das falsch konfigurierte CGI Script durch Parameter im Hyperlink dazu gebracht diesen Code auszuführen.

Wenn das Opfer den beispielsweise per Email oder durch ein Forum enthaltenen Link öffnet wird das in dem Link injizierte JavaScript ausgeführt. Das ermöglicht dem Angreifer alle Arten von Scripte auf dem Rechner des Opfers auszuführen. Darunter beispielsweise Code das bei dem User Cookies oder Anmeldedaten ausliest.

Solche Attacken gehören zu den häufigsten Angriffe überhaupt. Diese Angriffe stoßen meistens aus einem für das Opfer vertrauenswürdigen Kontext heraus Aktionen an, welche dann diesen Kontext verlassen und in einer anderen Umgebung ausgeführt werden. Mit einem vertrauenswürdigen Kontext ist eine für den Anwender vertraute Webseite gemeint, auf der vom Angreifer Schadcode hinterlegt wurde. Dieser Schadcode erscheint für den Anwender legitim da er auf einer vertrauenswürdigen Seite auftaucht und darauf ausgeführt wird.

Dadurch das der vom Angreifer hinterlegte Code auf der Client Seite ausgeführt wird ist dies für den Server Administrator nicht ohne weiteres ersichtlich. Je weiter die Entwicklung fortschreitet um so mehr Komfort wird geboten. Beispielsweise der Password-Safe eines Browsers, der dem Anwender ermöglicht Login Daten zu speichern die automatisch beim Aufruf einer Anmeldeseite eingetragen werden. Somit muss sich der Benutzer

die Anmeldedaten nicht merken und bei jedem Aufruf der Seite manuell einzutragen. Diese Anmeldedaten werden nicht nur im Klartext im Password-Safe des Browsers gespeichert sondern auch im Klartext auf der Anmeldeseite automatisch eingetragen. Somit kann der Angreifer durch ein mit JavaScript präparierten Link den Benutzer auf die jeweilige Login Seite locken, um die Benutzerdaten auszuspähen.

4.1.3 Beispiele

Für XSS gibt es viele verschiedene Angriffsarten wobei das Prinzip immer gleich bleibt. Bei einem clientseitigen Cross-Site Scripting wird versucht Code auf dem Client Rechner auszuführen, bei einem serverseitigen Cross-Site Scripting wird versucht, Code auf dem Server auszuführen.

Nicht nur auf größeren Portalen, Einkaufshops oder sonstige größere Webseiten gibt es XSS Lücken. Schon ein einfaches PHP Formular kann zur Gefahr werden.

```
<?php
if(isset($_GET['vorname'])){
    echo '<h1>Hallo, ' . $_GET['vorname'] . '</h1>';
} else{
    echo '<from method="GET" action=''' .
strip_tags($_SERVER['PHP_SELF']) -'>;
    echo 'Bitte Vornamen eingeben: ';
echo '<input type="text" name="vorname">';
echo '<input type="submit"></form>';
}
?>
```

Listing 5: einfaches Formular

Dieses Script spricht den Besucher persönlich anhand der GET Variable *vorname* an. Ist die Variable nicht gesetzt, fragt das Skript den Namen ab um ihn anzuzeigen. Die GET Variable unterliegt keinerlei Überprüfung und man kann beliebige Werte übergeben. Man könnte so ein JavaScript in der Variable unterbringen das dann im Client ausgeführt wird. Beispielsweise:

```
<script>alert("XSS");</script>
```

Listing 6: Script-Tag wird statt der Variable gesetzt

Es würde ein leeres Java Script PopUp erscheinen welches aber keinen Inhalt anzeigen würde, was daran liegt, dass in PHP alle Anführungszeichen in Backslashes gewandelt werden. Dies kann durch folgendes ausgetrickst werden:

```
<script>x=/XSS/; alert(x.source)</script>
```

Listing 7: Ersetzungen von PHP umgehen

Viele Eingabefelder werden durch den String "javascript" überprüft. Doch auch das ist zu überbrücken. Denn der String könnte komplett in UTF-8 kodiert werden:

```
&#x6A;&#x61;&#x76; . . .
```

Listing 8: UTF-8 Codierung

Der String wird durch den Internet-Explorer wieder zu dem ursprünglichen JavaScript dekodiert. Durch solche Tricks lassen sich manche XSS Filter überlisten.

4.1.4 Sonstiges

Neben XSS das durch aktive Inhalte wie beispielsweise JavaScripte auf der Client Seite Aktionen ausführt, steht eine SQL-Injection dem Prinzip nahe. Nur wird hierbei Schadcode in ein Datenbanksystem eingeschleust das dann bei einem Datenbankzugriff ausgeführt wird, was beispielsweise eine Datenbankabfrage auf einer in PHP geschriebenen Weboberfläche sein kann.

4.1.5 Fazit

Eine XSS-Attacke ermöglicht dem Angreifer ein raffiniertes Täuschungsmanöver welches großen Schaden anrichten kann. Dieses kann dann von dem Opfer nur noch schwer als solches erkannt werden. Um sich vor so einer Attacke zu schützen müssen alle Parameter, die vom User kontrolliert werden, als unsicher betrachtet werden und vor der Verwendung geprüft werden. Viele dieser Angriffe benutzen JavaScript um Links zu verseuchen. Der Benutzer kann sich vor diesen Angriffen schützen, indem er JavaScript im Browser deaktiviert.

4.2 HTTP-Response-Splitting

4.2.1 Einleitung

Ähnlich wie bei dem zuvor behandelten Cross-Site-Scripting wird eine URL, besser eine Anfrage an den Server, manipuliert. Es wird aber kein Scriptcode injiziert. Der Server nimmt Benutzerparameter aus der Anfrage und baut sie in eine HTTP-Antwort ein die an den Client gesendet wird. Meistens handelt es sich bei den Benutzerparameter um unzulässige Zeichen, insbesondere CR- und LF-Zeichen (Carriage Return, Line Feed).

URL kodiert Schreibweise:

```
\r -> %0d  
\n -> %0a
```

Listing 9: blub

Bei diesem Angriff wird die fehlerhafte Filterung der Webservern, Web-Cache, Browser-Cache-Poisoning ausgenutzt. Es sind lediglich immer drei Parteien mit eingebunden, der

Webserver, der Angreifer und das Ziel das mit dem Webserver im Auftrag des Angreifers kommuniziert.

Das eigentliche Ziel ist Webseiten mit gefälschten Anfragen zu verunstalten. Es wird hierbei nicht direkt auf den Webserver Einfluss genommen sondern zielt eher auf die davor geschalteten Systeme. Was ein Proxy-Server oder ein Cache-Server sein kann.

4.2.2 Vorgehensweise

Ein vom Angreifer durch unzulässige Zeichen manipulierter HTTP-Request löst beim Webserver zwei HTTP-Responses aus. Denn durch die unzulässigen Zeichen in dem Request splittet der Webserver die Anfrage in zwei Antworten. Der Angreifer sendet danach eine weitere unwichtige Anfrage. Der Cache vor dem Webserver ordnet diese Anfrage der zweiten Antwort zu. Somit kann der Cache vor dem Webserver mit gefälschten Anfragen vergiftet und verunstaltet werden. Das kann zur Folge haben, dass Webseiten mit vertraulichen Informationen weitergeleitet werden. Den der zwischen Benutzer und Webserver geschalteten Proxy-Server oder Cache-Server liefert zu einer korrekten Anfrage eine gefälschte Antwort.

Es ist lediglich eine Schwachstelle in einer Applikation vorausgesetzt und ist unabhängig vom Browser oder Webserver. Diese Schwachstellen sind Mängel in der Überprüfung auf Sonderzeichen, welche den Webserver dazu veranlasst gewisse Anfragen die durch die Sonderzeichen manipuliert wurden zu splitten. Diese Sonderzeichen sind Parameter dieser Sorte:

```
\r Carriage Return - Zeilenumbruch
\n Line Feed       - Zeilenumbruch
```

Listing 10: Sonderzeichen

Im Grunde werden PHP-Skripte benötigt welche die Daten vom Angreifer in den HTTP-Header einfügen. Beispielsweise `header()`. Beispiel:

```
<%
  response.sendRedirect("/foo.jsp?lang="+
  request.getParameter("lang"));
%>
```

Listing 11: Funktion header()

Bei diesem Beispiel wird Benutzerparameter *lang* an die Seite *foo.jsp* weitergeleitet. Um jetzt eine HTTP-Response-Splitting Attacke durchzuführen muss die Anfrage folgendermaßen manipuliert werden.

```
%
  /redir_lang.jsp?lang=foobar%0d%0aContent-
  Length:%200%0d%0a%0d%0aHTTP/1.1%20200%200K%0d%0aContent-
  Type:%20text/html%0d%0aContent-
  Length:%2019%0d%0a%0d%0a<html>h4xx0r</html>
```

Listing 12: HTTP Anfrage mit Sonderzeichen

Die Antwort darauf sieht folgendermaßen aus:

```
HTTP/1.1 302 Moved Temporarily
Date: Wed, 24 Dec 2003 15:26:41 GMT
Location: http://10.1.1.1/by_lang.jsp?lang=foobar
Content-Length: 0
```

Listing 13: Antwort 1

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 19
```

Listing 14: Antwort 2

```
<html>h4xx0r</html>
.
.
.
```

Listing 15: Unbekannter Rest

Man sieht hier schön zwei angekommenen Antworten, zum ersten ein *302 Status Code* und zum zweiten eine Antwort mit einem *200 Status Code* deren Content HTML Inhalt ist. Der Rest ist unbedeutend.

Es wurden jetzt zwei Antworten auf die manipulierte Anfrage gesendet, die erste Antwort wird der Anfrage zugeordnet. Um eine HTTP-Response-Splitting Attacke durchzuführen, muss der Angreifer eine weitere Anfrage senden. Diese wird dann der im Cache übrig gebliebene zweite Antwort, zugeordnet.

Anfrage 1 - verursacht 2 Antworten

```
%
/redirect_lang.jsp?lang=foobar%0d%0aContent-
Length:%20%0d%0a%0d%0aHTTP/1.1%20200%200K%0d%0aContent-
Type:%20text/html%0d%0aContent-
Length:%2019%0d%0a%0d%0a<html>h4xx0r</html>
```

Listing 16: HTTP-Response-Splitting Anfrage

Anfrage 2 - wird im Cache des Proxy-Servers der zweiten Antwort zu geordnet

```
/index.html
```

Listing 17: zweite Anfrage

Die beiden Antworten darauf sehen wie folgt aus

```
HTTP/1.1 302 Moved Temporarily
Date: Wed, 24 Dec 2003 15:26:41 GMT
Location: http://10.1.1.1/by_lang.jsp?lang=foobar
Content-Length: 0
```

Listing 18: Antwort 1

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 19
<html>h4xxor</html>
```

Listing 19: Antwort 2

Der Cache vom Proxy-Server merkt sich diese Anfrage-Folgen. Ab dem Moment erhält jeder Benutzer die vom Angreifer gestellte Seite. Durch diese Attacke können Angreifer Webseiten umleiten oder sogar an böswillige Applikationen. Somit kann der Angreifer die vorhin besprochene Cross-Site-Scripting Attacke ausführen.

4.2.3 Fazit

Hier wird schnell klar was für ein Schaden eine solche Attacke auslösen kann. Bei dieser Attacke wird nicht im geringsten eine Änderung oder Manipulationen auf dem Webserver vorgenommen. Das wichtigste für den Betreiber eines Webservers ist darauf zu achten das Sonderzeichen wie obige herausgefiltert werden.

4.3 Google-Hacking

Aufbau der Google-URLs

Untersuchen wir einmal, welche Parameter der Suchmaschine mitgegeben werden.

Syntax:

```
http://www.google.de/search?variable1=value&variable2=value
```

In dem folgenden Beispiel wurde eine Suche mit dem Suchtext "indisch Kochen" gestartet:

```
http://www.google.de/search?hl=de&q=t%C3%BCrkisch+backen
```

Zerlegen wir einmal die URL:

```
http://www.google.de/search
```

Dies ist der offizielle Standard von Google, mit dem eine Suche eingeleitet wird.

?

Das Fragezeichen nach 'search' zeigt an, dass Suchbegriffe, also Parameter, mitgegeben wurden.

hl=de

Die Zuweisung des Wertes 'de' an die Variable 'hl' gibt an, in welcher Sprache die Ergebnisse

\&

Bool'scher Operator UND, zur Verknüpfung mit folgenden Suchbegriffen. Alternativ auch '+'

q=

Definiert den Anfang des eigentlich eingegebenen Suchtextes.

t\%C3\%BCrkisch+backen

Eingegebener Text "türkisch backen", wobei das 'ü' mit '%C3%BC' hex-kodiert ist.

Die goldenen Regeln der Googleuche nach Johnny Long

- Google unterscheidet nicht zwischen Groß- und Kleinschreibung. Ob Sie jetzt nach 'kochen', 'Kochen' oder 'kOcHeN' suchen, Sie erhalten immer die gleiche Ergebnisliste. Ein Ausnahmefall ist hier der Bool'sche Operator 'or'. Wenn Sie Diesen anwenden möchten, muss groß -also 'OR'- geschrieben werden. Dies gilt nicht für seinen Verwandten 'and'.
- Google geht mit Wildcards anders um als der typische Programmierer. Viele User betrachten Wildcards als symbolische Darstellung eines einzelnen Zeichens oder einer Zeichenkette. Für Google stellt der Asterisk '*' jedoch ein einzelnes Wort dar. Und durch das setzen der Sternchen an den Anfang bzw. ans Ende eines Suchwortes erhalten Sie die gleiche Ergebnisliste, die Sie auch ohne die Sternchen bekommen hätten.
- Google verkürzt bzw. erweitert Suchbegriffe automatisch. Wenn sie beispielsweise nach dem Wort 'pet dietary' suchen, werden Sie in Ihrer Ergebnisliste auch Einträge mit dem Suchbegriff 'diet', da 'dietary' den Wortstamm 'diet' hat.
- "Google behält sich das Recht vor Sie zu ignorieren." (Zitat von Johnny Long). Google ignoriert bestimmte sehr häufig vorkommende Worte, wie beispielsweise 'wer', 'wo', 'und', 'in', 'so', 'da' oder 'wie'. Dieser Wörter werden als Stoppbegriffe bezeichnet.
- Zwingen Sie die Google-Suchmaschine dazu, auch nach Stoppbegriffen zu suchen, indem Sie ein Pluszeichen '+' vor das Wort stellen, zum Beispiel '+und'.
- Wenn Sie einen Satz suchen, stellen Sie diesen in Anführungszeichen "". Damit haben Sie nur Ergebnisse, die den kompletten Text enthalten, also die Suchbegriffe in der selben Reihenfolge.

Googles Operatoren

Neben den einfachen Suchbegriffen, kennt Google auch spezielle Begriffe oder so genannte erweiterte Operatoren. Richtig eingesetzt helfen Ihnen diese Operatoren Informationen zu finden, ohne dass Sie viel Zeit für das Durchforsten von unzähligen Suchergebnissen verloren geht.

Hier eine Erklärung der erweiterten Google-Operatoren:

- 'allintext' durchsucht den BODY-Bereich einer HTML-Seite. Dieser Operator ist der Default-Operator, der verwendet wird, wenn eine normale Suche gestartet wird.
- 'intitle' durchsucht den TITLE-Bereich der HTML-Seite. Folgende Suche würde Seiten auflisten, bei denen 'kochen' enthalten ist und in der Titel-Leiste 'index' steht. Zu beachten ist, dass hier 'kochen' nicht im Titel steht. Hier muss dann der Operator 'allintitle' verwendet werden.

```
intitle:index kochen
```

'inurl' und 'allinurl'

Sucht nach einer Seite, bei der 'admin' in der URL steht und das Wort 'kochen' wird normal behandelt.

```
inurl:admin kochen
```

Falls die 'kochen' auch in der URL suchen wollen, können sie den Operator 'allinurl' verwenden

```
allinurl:admin kochen
```

oder

```
inurl:admin inurl:kochen
```

'site'

Mit dem site-Operator können Sie eine bestimmte Domäne durchsuchen.

```
site:heise.de sicherheit
```

'filetype'

Suchen nach Dokumenten mit einer bestimmten Datenextension.

```
filetype:pdf kochen
```

Suchen nach Seiten, die auf eine angegebene Seite verlinken. Hier wird als Parameter eine URL erwartet.

```
link:tagesschau.de
```

'cache'

Google speichert Schnappschüsse der durchforsteten Seiten. Wenn Sie nun nicht auf die Originalseite, sondern direkt auf die gecachte Seite springen wollen, verwenden Sie den cache-Operator.

```
cache:chefkoch.de
```

'numrange'

Der numrange-Operator erwartet zwei Parameter. Eine niedrige und eine hohe Zahl, mit einem Strich getrennt. So kann nach einem Zahlenbereich gesucht werden.

```
numrange:5-10
```

Vom numrange-Operator gibt es auch eine Kurzschreibweise, bei der die zwei Zahlen mit zwei Punkten getrennt werden.

```
5..10
```

Laut Johnny Long ist der numrange-Operator der am meisten missbrauchteste Operator beim Erstellen von Google-Queries.

4.3.1 Der Google-Cache

Wenn Google eine Seite oder ein Dokument einmal eingelesen hat, kommen Sie fast immer an eine Kopie. Die Seiten werden im sogenannten Google-Cache gespeichert. Dieses Feature ist meist direkt vor unserer Nase, wird aber selten wahrgenommen. Geben Sie in Google eine beliebige Query ein und betrachten Sie die Ergebnisliste einmal genauer. Bei fast jedem Treffer wird unterhalb der Seitenbeschreibung ein Link mit der Aufschrift 'Im Cache' angezeigt. Klicken Sie auf diesen Link und Sie gelangen zu einer Kopie der Originalseite. In diesem Cache stehen allerdings auch Seiten, die es längst nicht mehr gibt, was dann auch ein Nachteil des Caches ist. Jedoch ist der größte Nachteil des Caches, dass Angreifer Ihre gesamte Webseite 'durchwühlen' können, ohne auch nur ein einziges Datenpaket an Ihren Server senden zu müssen. Somit kann auch kein Logfile erstellt werden und der Angriff wird nie erkannt oder geahndet werden.

4.3.2 Google als Proxy benutzen

Google bietet eine Funktionalität eines automatisierten Übersetzers von Seiten. Im selben Bereich der Ergebnisliste wie der Link 'Im Cache' ist, erscheint -wenn eine automatische Übersetzung möglich ist- ein weiterer Link mit der Aufschrift 'Diese Seite übersetzen'. Wenn Sie diesem Link folgen erscheint die gewünschte Seite in einer übersetzten Form. Ähnlich wie beim Cache kann hier auch die Identität verschleiert werden, da Sie durch die Verwendung des Google-Translators Google als Proxy-Server verwenden. Auch hier wird kein einziges Paket an die angeschaute und übersetzte Seite gesendet. Wenn Sie eine Seitenänderung (durch das surfen auf der Seite) auslösen, fungiert Google als Kommunikator zwischen Ihnen und der Zielseite.

4.3.3 Verzeichnislisten

Eine Verzeichnislisten ist eine Webseite, welche die Dateien und Verzeichnisse eines Webservers aufführt. Die Navigation erfolgt durch das Anklicken der Verzeichnis- oder Dateilinks.

Verzeichnislisten finden

Die meisten Verzeichnislisten fangen mit 'Index Of' im Titel an. Mit folgender Query können Sie zahlreiche Verzeichnislisten via Google finden.

```
intitle:index.of.
```

Beachten Sie, dass der Punkt (.) für ein einzelnes Zeichen steht. Natürlich liefert die Query auch viele Falschpositive Ergebnisse, beispielsweise 'Index of Native American Ressources on the Internet'. Um eine genauere Ergebnisliste zu erzielen könnten wir auch zusätzlich nach dem höchstwahrscheinlich enthaltenen 'Parent Directory' suchen

```
intitle:index.of. 'Parent Directory'
```

Bestimmte Verzeichnislisten finden

Um etwas sicherheitsrelevanten Daten aufzuspüren, kann auch anstatt nach 'Parent Directory' direkt nach 'Admin' gesucht werden, um möglicherweise Benutzernamen, Passwörter oder ähnliches auf zu spüren. Folgende Queries liefern nahezu die selben Ergebnisse:

```
intitle:index.of. 'Admin'
```

oder

```
intitle:index.of. inurl:Admin
```

oder

```
intitle:index.of.admin
```

Indem Sie diese Vorgehensweise beispielsweise mit weiteren Operator kombinieren, können Sie die Liste sehr stark präzisieren und nach gezielten Dateien suchen:

```
intitle:index.of ws\_ftp.log
```

Serverversion ermitteln

Für weitere Angriffe kann es äußerst hilfreich sein, die Programmversion eines Servers zu kennen. Normalerweise kann bei der Konfiguration eines Servers festgelegt werden, ob unter der Verzeichnisliste auch die Versionsnummer des Webservers angezeigt wird. Es ist dringendst zu empfehlen, diese Anzeige zu unterbinden. Folgende Query sucht nach Servern, die dieses nicht unterbinden und die Versionsnummer anzeigen:

```
intitle:index.of ''server at''
```

Sie können natürlich auch nach einer bestimmten Versionsnummer suchen:

```
intitle:index.of ''Apache 1.3.27 server at''
```

4.3.4 Network Mapping

Neben dem reinen Aufspüren von Zielen weiß jeder kompetente Angreifer, dass die leichtesten Ziele diejenigen sind, die vergessen wurden und nicht im Fokus der IT-Sicherheitsteams liegen. Diese Kenntnis ist wichtig, weil immer mehr Netzwerke nicht durch Angriffe auf die Schwachstellen der gut geschützten und scharf überwachten Systeme kompromittiert werden, sondern durch das Erforschen von verlorenen, vergessenen Systemen, die aus dem Blickfeld der überarbeiteten Administratoren geraten sind. Hier gehen wir mehr in die Google-gestützte Netzwerkerkennungsmethodik ein.

Site-Crawling

Wenn wir beispielsweise die Query 'site:microsoft.com' ausführen und uns die ersten 5 Ergebnisse ansehen, stellen wir fest, dass sich die DNS-Namen alle ähneln (meist microsoft.com und msdn.microsoft.com). Wir können die Suche eingrenzen und weitere Domänen finden, indem wir eine negative Bedingung für 'microsoft.com' hinzufügen. Beispielsweise:

```
site:microsoft.com -site:www.microsoft.com
```

oder

```
site:www.microsoft.com -site:microsoft.com
```

Diese Suche liefert eine wesentlich höhere Vielfalt. In den ersten vier Ergebnissen gleich vier neue Domänen. Nun können wir die gefundenen Domänen ebenfalls als negative Bedingung in die Query hinzufügen. Diese Technik ist sehr aufwändig, aber sehr effektiv in Bezug auf die Ermittlung von Netzwerken. Wesentlich effektiver ist hier die Automatisierung mit Hilfe der Google-API. Die Google-API ist übrigens die rechtlich einzige Möglichkeit Google-Abfragen zu automatisieren.

Link-Mapping

Neben dem Sammeln von Domänen und Subdomänen ist es oft wichtig, weniger offensichtliche Beziehungen zwischen Websites zu verstehen. In manchen Fällen bietet eine Schwachstelle bei einer nicht gut gesicherten, aber vertrauten Partnersite dem Angreifer eine Möglichkeit, "den Panzer des von schweren Geschützen gesicherten Haupt-Ziels zu knacken", so Johnny Long. Zum Tragen kommt hier die Vertrauensstellung zwischen Sites. Wenn ihr Ziel einen Link zu einer anderen Seite enthält, existiert möglicherweise solch eine Vertrauensstellung. Eingehende Links haben hier so gut wie keinen Stellenwert, da jeder User einen Link zu jeder beliebigen Seite einrichten kann. Wenn jedoch

zwei Seiten gegenseitig verlinkt sind, deutet das auch eine sehr enge Beziehung hin. Diese Art von Beziehung steht ganz oben auf der Skala der Interessensstufen.

Erwähnt werden muss an dieser Stelle das äußerst intelligente Perl-Programm BiLE (Bi-directional Link Extractor) von sensepost. Dieses verwendet die Google-API um Link-Listen automatisiert zu erstellen. Die Dokumentation dieses Programms nennt hier verschiedene Punkte zur Gewichtung von Links:

- Ein Link von einer Seite hat mehr Gewicht als ein Link zu einer Seite
- Ein Link von einer Seite mit vielen Links hat weniger Gewicht als ein Link von einer Seite mit wenigen Links
- Ein Link zu einer Seite mit vielen Links hat weniger Gewicht als ein Link zu einer Seite, die wenig verlinkt wurde
- Die Seite, die als Eingabeparameter eingegeben wurde, erhält nicht automatisch die höchste Gewichtung - daran erkennt man, dass die angegebene Seite [vielleicht] nicht die zentrale Seite der Organisation ist.

Nachdem BiLE nun eine Liste von Links erstellt hat kommt nun das Programm BiLE-Weigh zum Einsatz, welches die Liste auswertet und jeder Seite eine punkte-mässige Gewichtung vergibt. Je mehr Punkte eine Seite erhält, desto relevanter ist sie für das Ziel. An dieser Stelle muss unbedingt auf einen Phrack-Artikel von Michal Zelewski unter www.phrack.org/show.php?p=57&a=10 verwiesen werden. Dieser Artikel mit dem Namen 'Rise of the Robots' beschreibt ein Szenarion, in dem Würmer diese Technologie nutzen, um Angriffe zu planen. (Bericht von Imperva unter www.imperva.com/docs/Application_Worms.pdf)

Group-Tracing

Unter Group-Tracing versteht man das Durchwühlen von Google-Groups mit dem author-Operator. Google-Groups werden oft zur Veröffentlichung von technischen Fragen genutzt. Über den NNTP-Header kann dann zum Beispiel zurückverfolgt werden, von welchem NNTP-Server aus diverse Veröffentlichungen versendet wurden. Beispielsweise:

```
author:@microsoft.com
```

So können Mitarbeiternamen herausgefunden werden und möglicherweise auch Mitarbeiter einer gleichen Abteilung zugeordnet werden, falls Sie zum Beispiel in den selben Groups Artikel veröffentlichen. Diese Informationen können auch für Social Engineering-Angriffe nützlich sein.

4.3.5 und vieles vieles mehr

Die Möglichkeiten von Google sind einfach erschreckend! Passwörter, Logfiles, Buddylisten persönliche Finanzdaten ja selbst Kreditkartennummer und Sozialversicherungsnummern können Beispiel für Registryeinträge mit Defaultpasswörtern:

```
filetype:reg reg intext:''defaultusername'' intext:''defaultpassword''
```

Allerdings würde alles aufzuführen den Rahmen dieser Arbeit sprengen und es wird nicht näher auf die Queries eingegangen. Falls Sie dennoch selbst überzeugen wollen und tiefer in die Thematik Google-Hacking eintauchen wollen empfehle ich ihnen das Buch 'Google-Hacking' von Johnny Long. Dieses Buch diente auch dieser Arbeit als Grundlage. Beachten Sie jedoch, dass das Buch 2005 veröffentlicht wurde und daher (glücklicherweise) nicht mehr alles funktioniert. Auch empfehlenswert ist die Homepage von Johnny Long <http://johnny.ihackstuff.com/>. Riskieren Sie doch mal einen Blick.

4.4 PHP-Sicherheit

PHP ist eine Script-Sprache die in Webanwendungen sehr weit verbreitet ist. Da PHP wie jede andere Programmiersprache auch von Menschen entwickelt wurde, ist sie somit natürlich auch nicht gegen Sicherheitslücken geveit. Dieses Kapitel widmet sich einiges Bugs in PHP und sicherheitsrelevanten Modulen für PHP, sowie Konfigurationen, die verschiedenen Angriffen vorbeugen.

4.4.1 Fehler in PHP

Die hier besprochenen PHP-internen Fehler beziehen sich auf die Version 4 und 5. Man muss jedoch beachten, das fast wöchentlich Fehler entdeckt werden. Die folgenden Fehler sind nur beispielhaft.

File-Upload-Bug

In den Versionen 4.0.2 bis 4.0.7RC2 war eine Lücke enthalten, die den Upload von Schadcode via HTTP erlaubte. Somit gelang dem Angreifer einen Zugriff auf das Zielsystem, meist mit den Benutzerrechten des Webservers. Der Exploit ist unter dem Namen '7350fun' bekannt.

CGI-Lücke in PHP 4.3.0

Diese PHP-Version hatte einen Fehler, der die Konfigurationsoption "–enable-force-cgi-redirect" unwirksam machte. Somit konnte jeder Angreifer, der Zugang zu den CGI-Scripten hatte, dieses für die Ausführung beliebiger eigener Kommandos missbrauchen.

Unsichere (De-)Serialisierung

Durch einen Bug in den Funktionen `serialize()` und `deserialize()` wurde bei einem Aufruf mehr Speicher freigegeben als erwünscht. Angreifer konnten so eigenen Schadcode zum Beispiel direkt über ein Cookie ausführen.

Gefährliches Speicherlimit

PHP-Versionen bis 4.3.7, die mit der Option "–enable-memory-limit" kompiliert wurden, konnten durch einen Fehler im Code, der das Speicherlimit setzt, ausgetrickst werden. Auch hier war die Ausführung von Schadcode möglich.

4.4.2 Bestandteile eines 'sicheren' Servers

Die Absicherung eines PHP-Servers umfasst die folgenden drei Teilsysteme:

- Webserver
- PHP-Installation
- Datenbankserver

Generell kann eines über das Gesamtsystem gesagt werden: Es besteht ein Kompromiss aus den folgenden drei konträren Extremen:

- Sicherheit
- Feature
- Geschwindigkeit

Möchten Sie beispielsweise möglichst viele Features erhalten, dabei aber keine Abstriche in der Geschwindigkeit machen, werden sie zwangsläufig Abstriche in der Sicherheit machen müssen.

4.4.3 Installation

Grundsätzlich haben Sie zwei Möglichkeiten bei der Installation von PHP:

- Die Installation als Apache-Modul. Bringt bestmögliche Integration in den Webserver und bestmögliche Features, jedoch auch Sicherheitsprobleme mit sich.
- Installation als CGI. So können Sie PHP-Skripte besser von einander trennen, riskieren allerdings Geschwindigkeitseinbußen und verlieren einige Features.

4.4.4 suExec

Der zentrale Vorteil einer Installation von PHP als CGI liegt darin, dass Sie hier die Sicherheitsmechanismen von Apaches suExec nützen können. Dieses Programm dient als Wrapper, um CGI-Skripte unter einer anderen UID und GID als der des Webserver ausführen zu können. Praktisch bedeutet dies, dass der Administrator für jeden virtuellen Host im Webserver einen eigenen Nutzer und eine Gruppe festlegen kann, in die der Webserver mit einem Aufruf der Betriebssystemfunktion `setuid()` vor der Skriptausführung wechselt. Nur diese dem Benutzer gehörenden Dateien darf das CGI-Skript

dann manipulieren, womit die gängigsten PHP-Sicherheitsprobleme weitgehend gelöst werden können. Insbesondere mit der Kombination von "open_basedir" können Sie ihren Webserver so nahezu wasserdicht machen. ¹

4.4.5 Safe Mode

Der Safe-Mode (nicht zu verwechseln mit dem "abgesicherten Modus" des Betriebssystems Windows) ist per `php.ini` oder Virtualhost-Konfiguration aktivierbar. Dieser Sicherheitsmodus führt für sämtliche PHP-Skripte eine Zugehörigkeitsprüfung durch: Versucht das gerade ausgeführte Skript auf Dateien zuzugreifen, die einem anderen Benutzer gehören, verweigert der Safe Mode den Zugriff. Folgendes Beispiel soll die Vorgehensweise näher erläutern:

- Ein PHP-Skript wurde vom Prozess mit der UID (Unix User ID) 1001, Gruppe `www-data` angelegt.
- Dieser Programm versucht eine Textdatei zu inkludieren, oder zu öffnen (`'/etc/passwd'`), die dem Benutzer `root` (UID=0) gehört.
- PHP überprüft die UID und die Gruppe des Skriptes (`101/www-data`) und die durch das Skript zu öffnende Datei (im Falle `/etc/passwd` `0/0`). Unterscheiden sich die UIDs, wird dem Skript die Verwendung untersagt.

Nicht jede Funktion beachtet den Safe Mode! Unsichere Extensions können ihn aushebeln!!

4.4.6 Weitere PHP-Einstellungen

An dieser Stellen sollen einige PHP-Konfigurationsoptionen genannt werden. Auch hier muss erwähnt werden, dass nicht alle sicherheitsrelevanten Optionen aufgeführt werden können, dies würde den Rahmen der Arbeit sprengen.

open_basedir

Die Konfigurationsdirektive "open_basedir" stellt in vielen Fällen einen wirksameren Schutz als der Safe Mode dar, obgleich auch diese Maßnahme von ausreichend motivierten Angreifern ausgehebelt werden kann. PHP-Skripte dürfen nur Dateien lesen und schreiben, die in dem oder in den offenen Grundverzeichnissen liegen, alle anderen Verzeichnisse sind tabu. Allerdings agiert diese Funktion nicht auf Betriebssystemebene und kann somit umgangen werden. Im Gegensatz zu Safemode stellt es in der Regel keine Einschränkung der Features für den Kunden dar.

¹vgl. <http://httpd.apache.org/docs/1.3/suexec.html>

disable_functions

Alle vom Administrator als gefährlich oder als unerwünscht erachteten Funktionen können mit der Direktive "disable_functions" deaktiviert werden. Diese Konfigurationsvariable erwartet eine kommentarseperierte Liste von Funktionen. Jedoch kann eine getrennte Konfiguration für jeden Virtualhost nicht möglich. Die Liste der deaktivierten Funktionen könnte folgendermaßen aussehen:

```
disable_functions = pcntl_exec, system, shell_exec, mysql_connect, posix_set
```

disable_classes

Ähnlich wie bei "disable_functions" können Sie mit der Direktive "disable_classes" die Verwendung bestimmter Klassen untersagen. Da erst in PHP 5 "objektorientierung" eingeführt wurde, ist diese Direktive auch erst an dieser Version relevant. Die Liste der deaktivierten Klassen könnte, genau wie bei "disable_functions" folgendermaßen aussehen:

```
disable_classes = mysqlConnectionInter, simplexml
```

max_execution_time

Dieser Parameter bestimmt, wie lange ein PHP-Skript ausgeführt werden darf, d.h. wie lange die PHP- oder Apacheinstanzen für andere Aufgaben blockiert werden. Der Standardwert ist auf 60 Sekunden gesetzt.

max_input_time

Diese Option bestimmt, wie lange die Verarbeitung einer Eingabe maximal dauern darf.

memory_limit

Hiermit kann verhindert werden, dass Skripte den gesamten vorhandenen Speicher beanspruchen. Vorsicht! Ein von "memory_limit" gesetztes Speicherlimit kann von Anwendungen geändert werden. Wenn Sie das Speicherlimit festsetzen wollen, so dass es auch nicht durch Skripte oder Kunden geändert werden kann, müssen Sie ein so genanntes Hardening-Patch einspielen, welches im Anschluss vorgestellt wird.

4.4.7 PHP-Hardening

Dieses Kapitel stellt das Patch PHP-Hardening vor, welches weitere Sicherheitsfunktionen in PHP ermöglicht. Grund für die Einführung von PHP-Hardening war die Tatsache, dass PHP nicht nur unter nachlässigen Programmierern litt, sondern bereits Fehler in der Zend-Engine (Die Zend-Engine wird von PHP als Parser und Compiler benutzt) barg. PHP-Hardening zielt somit nicht auf den Schutz vor Programmiererfehler, sondern zum Schutz des PHP-Kerns. Der PHP-Hardening-Patch ist experimenteller Natur, daher Vorsicht auf Produktivsystemen!

Schutz vor Buffer Overflows im Hardening-Patch

Das Mittel zum Zweck sind sogenannte "Canaries". Hier werden vor jeder Rücksprungadresse feste Werte als Warnbalken eingefügt. Werden diese überschrieben, hat ein Bufferoverflow stattgefunden. Der Name "Canaries" ist übrigens angelehnt an das Verfahren, Canarienvögel bei Bergwerkarbeiten als lebende Gasmelder mitzuführen.

Schutz vor Format-String-Schwachstellen

Funktionen wie printf, sprintf oder vprintf usw. erwarten bei Ausgabe einer Variablen einen Format-String (%s für Strings, %i für Ganzzahlen etc). Der Platzhalter %n sorgt dafür, dass beliebige Zeichen an genauso beliebige Speicherpositionen geschrieben werden können. Dadurch ist nicht nur die Ausgabe sensibler Inhalte im Speicher, sondern sogar mit etwas Aufwand auch die Ausführung eigenen Schadcodes möglich. PHP verwendet in C eigene Varianten von sprintf(), die den Platzhalter %n ebenso implementieren, obgleich er von PHP überhaupt nicht benötigt wird. PHP-Hardened entfernt das %n und leitet zusätzlich sämtliche Aufrufe fremder sprintf()-Implementierungen auf die von PHP-Hardened "reparierte" Funktion umgeleitet.

Schutz gegen Remote-Includes und Nullbytes

Oft leiden PHP-Anwendungen darunter, dass durch nicht ausreichend gefilterte Benutzereingaben dem Skript eine Include-Datei untergeschoben werden kann. PHP-Hardening bewirkt, dass alle Aufrufe des Sprachkonstrukts include() deren Übergabeparameter einer URL ähneln geblockt werden.

Funktions- und Evaluations-Beschränkungen

Verwalter eines Server möchten in der Regel einige Funktionen per Default deaktivieren (mit der Konfigurationsdirektiven disable_functions), sich aber weiterhin die Möglichkeit offen halten, sie bei Bedarf wieder einschalten zu können. Ein Weg, dies zu bewerkstelligen, ist die Umstellung auf CGI-Skript, bei dem für jeden Kunden eine eigene php.ini verwendet werden kann. Möchten Sie jedoch mod_php weiterverwenden. So lösen die im aktuellen Hardening-Patch eingebauten Positiv- und Negativlisten dieses Problem. Sie können je eine White- und eine Blacklist definieren. Die Whitelist enthält Funktionsnamen, die ausgeführt werden dürfen, die Blacklist enthält Funktionsnamen, die nicht ausgeführt werden dürfen.

Schutz gegen Response-Splitting

Das Hardening-Patch dass bei einem einem Aufruf der PHP-Funktion header() Angabe mehrerer Header angegeben werden können, womit die Attacke nicht mehr möglich ist.

Variablenschutz

Viele Lücken kamen zustande, weil über aufwändige Deregistrierungsfunktionen die Auswirkungen von "Register Globals" rückgängig gemacht werden sollten. Hierbei wurde übersehen, dass der Angreifer so auch die wichtigen superglobalen Arrays (\$_COOKIE etc.) modifizieren konnte.

SQL Intrusion Detection

Alle Fehlgeschlagenen SQL-Abfragen der Extensions (mySQL, fdsn, SQLite etc.) werden auf Wunsch hin in eine Log-Datei geschrieben. Das Skript kann auch nach einer fehlgeschlagenen Query automatisch abgebrochen werden, um SQL-Injections zweiter Ordnung keine Chance zu geben.

Loggin

Damit Sie ausreichende Informationen über Angriffe erhalten, führt das HardeningPatch genau Buch über geblockte Attacken.

Schreibtgeschütztes Speicherlimit

Das Ändern des Speicherlimits (gesetzt durch `memory_limit`) mit der Funktion `ini_set()` oder durch `.htaccess` kann mit PHP-Hardening untersagt werden. So vermeiden zum Beispiel Serverbetreiber, dass speicherhungrigen Skripten mehr Speicher zugeteilt werden kann, als vorgesehen.

und vieles mehr

Das PHP-Hardening-Patch bietet noch eine Vielfalt von weiteren Sicherheitsvorkehrungen, die allerdings nicht alle in dieser Arbeit besprochen werden können. Für weitere Informationen empfehle ich die Website des Projektes <http://www.hardened-php.net/> oder das Forum rund um dieses Thema unter <http://forum.hardened-php.net/>. Ansonsten empfehle ich das Buch mit dem Titel PHP-Sicherheit von Christopher Kunz und Peter Prochaska, welches hauptsächlich als Vorlage für diesen Zusammenschrieb diente. Das Buch erschien beim dpunkt-Verlag.

5 Honeypots

5.1 Theorie

5.1.1 Die Täuschung

Die gesamte Kriegsführung stützt sich auf Täuschung. Daher müssen wir, wenn wir fähig sind anzugreifen, unfähig erscheinen; Wenn wir unsre Kräfte nützen, müssen wir untätig erscheinen; Wenn wir nahe sind, müssen wir dem Feind weismachen, dass wir weit weg sind; Wenn wir weit weg sind, müssen wir ihm weismachen, dass wir nahe sind; Halte den Köder her, um den Feind wegzulocken.

Sun Tzu, ca. 400 v. Chr.

Ob jetzt Naturvölker mit Getrommel Stärke und Überlegenheit simulieren wollen oder ein Chamäleon seine Farbe der Umgebung anpasst, haben alle diese Techniken einen gemeinsamen Zweck: seine Gegner in die irre zu führen um sie zu besiegen oder nicht von ihnen besiegt zu werden.

5.1.2 Die Idee eines Honigtopfes?

Ein Honeypot, auf deutsch Honigtopf, dient als eine Art Falle bei eventuellen Angriffen auf ein relevantes Netzwerk. Der Angreifer wird somit von kritischen Netzknoten abgelenkt und hin zum Honeypot gelockt, da dieser z.B. eine bessere Angriffsfläche darbietet. Der Angriff wird dann protokolliert und kann für Bugfixing- oder Forensikzwecke verwendet werden. Wichtig ist, dass der Honeypot nur den einzigen Zweck hat, nämlich ein attraktives Ziel für Angreifer darzustellen. Er darf keine 'echten' Nutzdaten halten. Ein weiteres Hauptcharakteristika ist es auch, dass den normalen Netznutzern die Existenz des Honeypots nicht bekannt ist, womit garantiert werden soll, dass selbst ein potentieller Angreifer aus dem engeren Kreise nicht weiß, dass ein Solcher installiert wurde.

5.1.3 Interaktionsstufen der Honeypots

Low-Interaction Honeypot:

- Ein Programm, welches Dienste emuliert. Es ist hervorragend geeignet um automatisierte Angriffe zu protokollieren. Kann jedoch mit ausgeprägtem Know How vom Angreifer erkannt werden.

High-Interaction Honeypot:

- Hier liegt der Fokus nicht auf automatisierten Angriffen, sondern darauf manuell ausgeführte Angriffe zu beobachten und protokollieren, um so neue Angriffsmethoden rechtzeitig erkennen und angemessene Sicherheitsvorkehrungen treffen zu können. Sinnvoll ist es, dass es sich bei einem High-Interaction Honeypot um ein Ziel handelt, welchem ein hoher Wert nachgesagt wird.

Im Gegensatz zum LI-Honeypot wird hier nicht auf dem selben Computer protokolliert, sondern die anfallenden Daten werden an einen separaten, loggenden Server gesendet. Dies geschieht über eine Ziel ist es, dass die Protokolle sendende Software nicht erkannt oder gar geändert wird. Der Angreifer soll nicht wissen, dass er überwacht wird.

5.1.4 Welche Software gibt es?

Low-Interaction Honeypots:

- honeyd

honeyd unter der GPL veröffentlicht, ist es in der Lage, gesamte Netzwerkstrukturen zu emulieren, mit einer Instanz der Software kann man viele verschiedene virtuelle Computer in einem Netzwerk simulieren, die alle unterschiedliche Dienste anbieten.

- mwcollect

mwcollect ist ein freier Honeypot unter der GPL für posix-kompatible Betriebssysteme mit der Zielsetzung, automatisierte Attacken von Würmern nicht nur zu erkennen und protokollieren, sondern die Verbreitungsmechanismen der Würmer zu nutzen, um eine Kopie des Wurms zu erhalten. Dazu werden als verwundbar bekannte Dienste nur soweit wie benötigt emuliert, ausgehend von verfügbaren Angriffsmustern.

- Nepenthes

Nepenthes, ebenfalls unter der GPL veröffentlicht, ist Nepenthes wie *mwcollect* ein Honeypot für posix-kompatible Betriebssysteme mit dem Fokus, Würmer zu sammeln.

- MultiPot

multipot ist ein Honeypot für Windows, er emuliert wie *Nepenthes* und *mwcollect* Schwachstellen unter Windows um Würmer zu sammeln.

High-Interaction Honeypots:

Meist das frei verfügbare *Sebek*, die vom Kernspace aus alle Programme des Userspace überwacht, und die anfallenden Daten vom Kernspace aus an einen loggenden Server sendet.

5.1.5 Überblick über verwandte Techniken

Honeytoken

Prinzip ist ein Honeytoken auch ein Honey pot. Jedoch handelt es sich bei einem Honeytoken nicht um einen Computer oder ein Programm. Auch dient es nicht zur Abwehr von Angriffen nach außen hin, wie das beim Honey pot in der Regel der Fall ist, sondern es bietet Schutz vor Angriffen aus dem eigenen Netzwerk. Ein Honeytoken ist z.B. ein nutzloser Datensatz in einer sicherheitsrelevanten Datenbank, der -wie der Honey pot auch- als Köder dient. Ein Honeytoken könnte aber auch eine Kreditkartennummer, ein Excel-Sheet, ein Word-Dokument oder gar ein gefälschtes Benutzerkonto sein. Sobald ein Zugriff auf diese Daten erfolgt, kann davon ausgegangen werden, dass auch ein unautorisierter Zugriff auf andere Nutzdaten erfolgt. Lance Spitzner, einer der führenden "Erfinder" des Honey pot-Konzeptes führt hier das klassische Beispiel für den Einsatz von Honeytokens auf, die sogenannte "John F. Kennedy"-Krankengeschichte: In Krankenhäusern herrscht strenger Datenschutz und um dieses zu gewährleisten wird häufig ein Patient mit dem Namen John F. Kennedy angelegt. Da dieser nicht existiert, darf auch unter keinen Umständen ein Zugriff auf dessen Patientenakte erfolgen.

Honey net

Unter einem Honey net versteht man ein ganzes Netzwerk von high-interactive Honey pots. Sie erhöhen den Täuschungseffekt und sollen dem Angreifer ein möglichst reales Bild vom Netzwerk geben. Hier werden weder irgendwelche Dienste emuliert, noch die einzelnen Systeme speziell abgesichert. Sicherheitslücken sind dabei häufig vorhanden und gewollt.

Tarpits

Tarpits, zu deutsch *teergruben*, dienen dazu die Verarbeitungsgeschwindigkeit von zum Beispiel Würmern zu verringern. Das verfahren ist auch unter dem Namen *labrea* bekannt. *labrea* (the tarpit) ist im IT-Bereich eine teergrube, d.h. ein Programm, mit dem man virtuelle Netzwerke vortäuschen und so z.b. Internetwürmer festhalten und/oder netzwerkscans blockieren kann. ebenso gibt es aber auch teergruben, die offene Proxy server emulieren, und falls jemand versucht Spam über diesen dienst zu verschicken, den Sender dadurch ausbremst, dass es die Daten nur sehr langsam überträgt.

5.2 Honey pots in der Praxis

Es gibt zahlreiche Dienste die den Aufgaben eines Honey pots gerecht werden. Darunter auch honeyd. honeyd ist in der Lage gesamte Netzwerkstrukturen zu emulieren und virtuelle Computer zu simulieren. Darüber raus kann honeyd auf jedem der simulierten Computer verschiedene Dienste emulieren.

Das Programm führt ein separates Statistik-Tool zur visuellen Aufbereitung der von verschiedenen Honeyd-Instanzen eingehenden Daten. Jede Bewegung und jeder Kontakt

mit dem virtuellen Computer wird protokolliert und kann detailliert ausgewertet werden. Es lassen sich im laufenden Betrieb Betriebssystemstatistiken, Portverteilungen in Echtzeit abrufen. honeyd ist ein Open Source Projekt und ist unter der GPL gestellt. Es läuft als Userspace-Daemon auf den meisten Unixen, inzwischen wurde er auch für Windows portiert. honeyd benutzt libcap, libevent und libdumbnet um für jeden virtuellen Honeyd einen eigenen TCP/IP Stack zu simulieren.

Eigentliche Vorteile

Der Anteil und Nutzen an Server und am Internet ist in letzter Zeit rasch gestiegen. Leider kehrt auch aus der Seite der Angreifer keine Ruhe ein, wenn nicht noch immer mehr Angreifer mit Attacken sämtliche Server kompromittieren. Oft werden immer wieder die gleichen Server angegriffen.

Oftmals bringt eine forensische Analyse wenig Ergebnisse. In diesem Fall könnte der Administrator auf einem oft kompromittierten Server einen Honeyd installieren. Oft reicht auch eine abgespeckte Version die beispielsweise unter Linux im Usermode läuft. Im Endeffekt kein extra Server, sondern gewisse Einheiten in ein vorhandenen Server einzubinden. Somit wird bei weiteren Attacken alles protokolliert und der Angreifer auf frische Spur ertappt.

Gefahren

Ein Honeyd ist kein Produkt, das man im Laden kauft, per Plug 'n Play anschließt und dann in die Ecke stellt. Ein Honeyd sollte am Besten von mindestens 2 Leuten 24/7 im Auge behalten. Man sollte sich vor allem auch langsam an die Thematik herantasten, wenn man vor hat ein solchen Honeyd zu verwenden. Denn je mehr Dienste mit Skripten und Ähnlichem simuliert werden, desto mehr Sicherheitslücken kann das mit sich bringen.

5.2.1 Die Netzwerksimulation

Auf den einzelnen emulierten Computer können unterschiedliche Dienste ausgeführt werden, die beim Ansprechen von Außen spezielle Aufgaben und Prozeduren durchführen. Für die Simulation der Dienste verwendet honeyd externe Skripte (Perl, Python, Unix-Shell). Es ist ein Daemon, der Netzwerk-Sockets abhört und bei Anfrage auf einem bestimmten Port ein voreingestelltes Programm (Skript) startet. Somit kann man ein speziell konzipierten Computer simulieren der gewünschte Dienste ausführt, eine perfekte Falle. Es gibt Skripte die komplette IIS/Apache Webserver simulieren, was wir aufgrund der kurzen Zeit, die wir uns gesetzt haben leider nicht in die Praxis umsetzen konnten.

Die Skripts protokollieren die Vorgänge in dem sie in Log-Dateien schreiben bzw. *stderr*. Zur Auswertung der Logfiles gibt es, außer den üblichen, für alle Logfiles nutzbaren Hilfsmitteln, spezielle Tools wie *Honeyview* oder *Honeydsum.pl*.

Jede Aktivität kann bemerkt und gemessen werden. Man sieht wie und wo der Angreifer eingedrungen ist.

5.2.2 Die Netzwerkkonfiguration

Den praktischen Einsatz des Honeypots honeyd wurde in der Fachhochschule für Kommunikation- & Softwaretechnik Albstadt-Sigmaringen getestet. Es stand uns ein eigenes Netz mit einem Switch zur Verfügung, welcher in dem Hochschulnetzwerk eingebunden war.

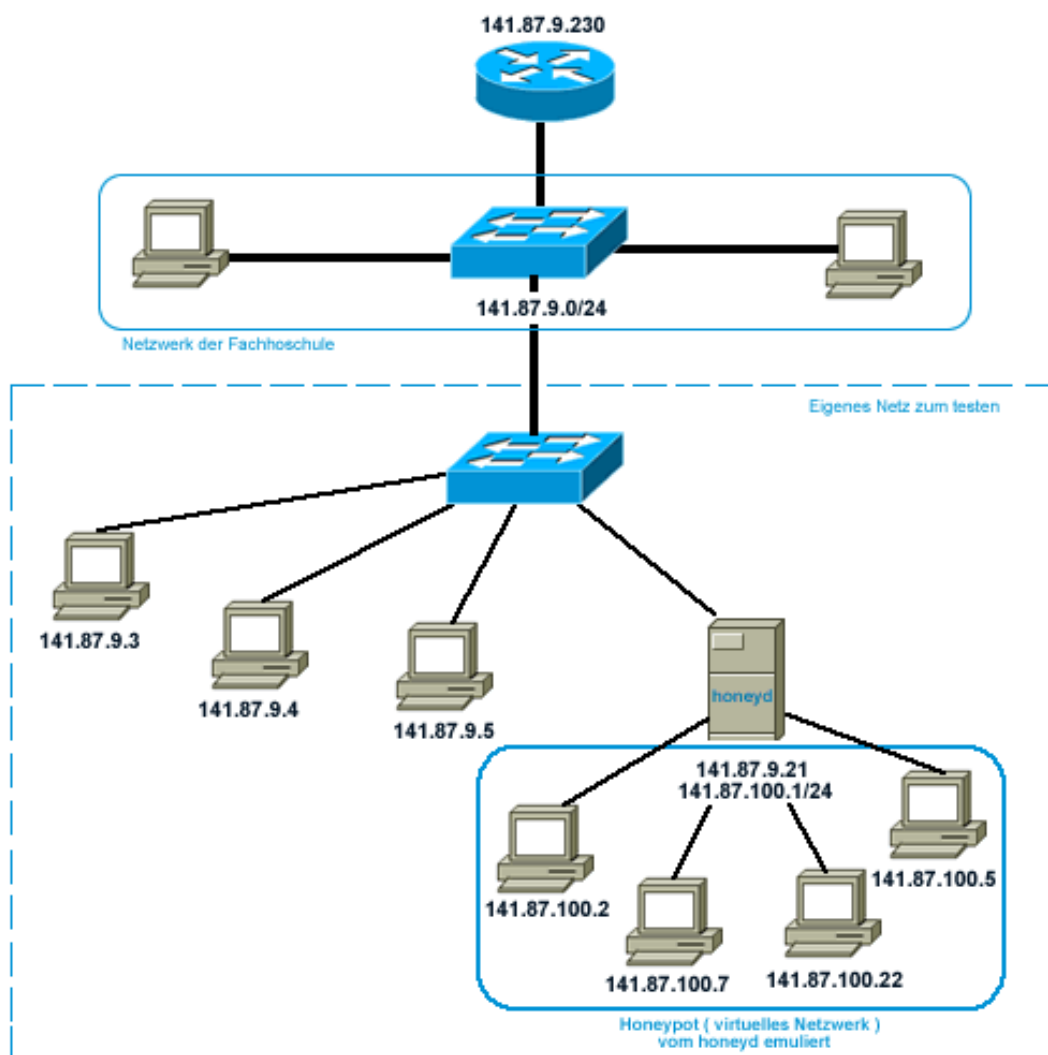


Abbildung 5.1: FH-Netzaufbau

Honeyd wird über eine Konfigurationsdatei gesteuert, in der die zu emulierenden Betriebssysteme und Dienste sowie deren Verhalten festgelegt werden. Die dabei definierten virtuellen Rechner werden Templates genannt. Generell befindet sich die Konfigurationsdatei von *honeyd.conf* im Verzeichnisse */etc/honeypots/honeyd.conf*.

Um in der Vorkonfiguration dem honeyd virtuelle Netze hinzu zuzufügen, muss mit einem Eintrag die eigene IP des honeyd Servers eingetragen werden. Zu dem eine Route auf das virtuelle Netz das simuliert wird. In unserem Fall hat der *honeyd*-Server die IP 141.87.9.21. Das simulierte Netz soll in folgender IP Range liegen: 171.87.100.1/16:

```
Route entry 141.87.9.21

route 141.87.9.21 add net 141.87.100.0/24 141.87.100.1 latency 7ms bandwidth 10Mb
route 141.87.100.1 link 141.87.100.0/24
```

Listing 20: Konfiguration virtueller Netze

5.2.3 Konfiguration der simulierten Computer

Der honeyd kann je nach Konfiguration mehrere verschiedene Computer emulieren auf den unterschiedlich konfigurierte Dienste laufen. Hier ein Beispiel aus der honeyd.conf:

```
create WinXPSP1
set WinXPSP1 personality "Microsoft Windows XP Professional SP1"
set WinXPSP1 uptime 1728650
set WinXPSP1 maxfds 35

add WinXPSP1 tcp port 80 "sh /usr/share/honeyd/scripts/win32/web.sh"
add WinXPSP1 tcp port 22 "/usr/share/honeyd/scripts/test.sh $ipsrc $dport"
add WinXPSP1 tcp port 23 proxy $ipsrc:23
add WinXPSP1 udp port 53 proxy 141.211.92.141:53
set WinXPSP1 default tcp action reset

create WinNT40
set WinNT40 personality "Microsoft Windows NT 4.0 SP3"
set WinNT40 uptime 2865
set WinNT40 maxfds 35

add WinNT40 tcp port 80 "sh /usr/share/honeyd/scripts/win32/web.sh"
add WinNT40 tcp port 22 "/usr/share/honeyd/scripts/test.sh $ipsrc $dport"
add WinNT40 tcp port 23 proxy $ipsrc:23
add WinNT40 udp port 53 proxy 141.211.92.141:53
set WinNT40 default tcp action reset
```

Listing 21: Eine honeyd.conf

Es wurde ein Windows XP inkl. SP1 und eine WinNT40 Maschine konfiguriert, die von *honeyd* emuliert werden soll. Man sieht schon an der Konfiguration die flexiblen Möglichkeiten die einem gegeben sind um ein Computer zu emulieren. Nach dem einleitenden Name, mit dem sich der Rechner im Netzwerk ausgibt, folgen Einstellungen wie die Zeit die er schon online ist und sonstige Eigenkonfigurationen. *Honeyd* emuliert

den TCP/IP-Stack gemäß der definierten Personality, in diesem Fall "Microsoft Windows XP Professional SP1". Dabei benutzt Honeyd die gleiche Fingerprint-Datenbank wie nmap, um die Reaktion des emulierten Betriebssystems auf einen Fingerprinting-Versuch vorzutäuschen.

Interessant sind hier die Dienste, die bei Kontakt von außen anspringen. Der Dienst hört auf ein bestimmten Port und hält sich startbereit. Versucht jemand über den Port 80 mit dem Rechner zu kommunizieren, weil er eine Lücke im Webserver erwartet. Wird ein bestimmter Dienst (hier das in Perl geschrieben *web.sh* Skript) gestartet. Der Angreifer meint eine Lücke gefunden zu haben, wobei es ein spezielles Skript ist, mit dem sich der Angreifer in einer Falle befindet

Hier ein Beispiel aus den Logfiles, bei einem Angriff auf den Server über Port 80, mit einem Skript welches einen Apache Web-Server emulieren soll:

```
honeyd[8032]: Connection request: tcp (141.87.9.39:47761 - 141.87.100.9:80)
honeyd[8032]: Connection established: tcp (141.87.9.39:47761 - 141.87.100.9:80) \
    <-> sh /usr/share/honeyd/scripts/unix/linux/suse8.0/apache.sh
honeyd[8032]: Connection request: tcp (141.87.9.39:47762 - 141.87.100.9:80)
```

Listing 22: blub

Hier ist schön zu sehen das der Angreifer (141.87.9.39) es auf unseren Webserver abgesehen hat. In Wirklichkeit läuft kein Webserver, dem Angreifer wird jedoch einer vorgetäuscht. Unser *honeyd* hört auf Port 80 und sobald eine Anfrage kommt wird auf dem virtuellen Suse 8.0 Rechner das Skript *apache.sh* gestartet. Der Angreifer bekommt nicht mit, dass es den Rechner überhaupt nicht gibt und dass dieser Rechner 141.87.100.9 von *honeyd* simuliert wird. Trotz dem geht er jetzt davon aus einen Webserver vor sich zu haben. Die Aktivitäten des Angreifers können nicht nur protokolliert, sondern auch durch das Skript gesteuert werden.

5.2.4 Probleme bei der Konfiguration

Problem: Der *honeyd*, der alle an ihn gesendeten Pakete in sein virtuelles Netz weiterleiten sollte, sendet unwiderrufflich ein "*Destination unreachable*".

Erklärung: Der *honeyd* routet zwar alle Pakete weiter die der honeypot (virtuelles Netz) beantwortet. Jedoch schickt er ein ICMP Type 3 zurück an den Sender. Der Grund liegt darin, dass das Betriebssystem bevor es das Paket in das virtuelle Netz weiterleitet ebenfalls antwortet.

Lösung: Es muss in den iptables vermerkt werden das alle ausgehenden ICMP Type 3 Pakete verworfen werden

```
iptables -A OUTPUT -p icmp -s 0/0 --icmp-type 3 -j DROP
```

Somit sind die emulierten Computer für die Außenwelt sichtbar und "physisch" vorhanden. Auf ein Ping würde man, wie erwartet, eine Antwort bekommen.

5.2.5 Fazit

Honeyd ist ein sehr mächtiges und flexibles Werkzeug welches virtuelle Netze, Computer und Dienste simuliert. Angriffe im Internet können sofort protokolliert und ermittelt werden. Jedoch ist Vorsicht bei der Nutzung geboten. Bei Verwendung im Internet sollte man sein Handwerk verstehen und genau wissen was man tut.

6 Computerforensik

6.1 Hinführung

Einbrüche in fremde Rechnersysteme, das verbreiten von Viren und Trojanern oder illegales Kopieren von urheberrechtlich geschützten Werken, alle diese Dinge haben eines gemeinsam: Es sind Straftaten und ihre Zahl nimmt täglich zu.

Hierzu ein kleines Beispiel: Unser Rechner, sei es zuhause oder im Büro, wird von einem Trojaner heimgesucht. Dieser Trojaner öffnet seinem Schöpfer, nennen wir ihn mal Mr.X, eine Hintertür. Mit Hilfe dieser Hintertür ist es möglich in das System einzudringen und es zum Beispiel "Fernzusteuern" ohne daß wir etwas davon mitbekommen. Die Hintertür nutzt Mr.X um mit unserem Rechner weitere Straftaten zu begehen. Ein unangenehmer Gedanke, welcher noch unangenehmer wird, wenn die Polizei unseren Rechner beschlagnahmt und wir beschuldigt werden, diese Straftaten begangen zu haben.

Was hat das alles mit dem Thema dieses Kapitels, der Computerforensik, zu tun? Die Computerforensik setzt genau hier an. Sie beschäftigt sich mit dem "danach", mit der Frage "Was ist passiert?". Im obigen Beispiel gilt es nachzuweisen, was Mr. X auf und mit unserem System gemacht hat. Hierzu müssen sämtliche Daten und die zum System gehörenden Datenträger untersucht und die Ergebnisse ausgewertet werden. Damit hierbei keine Beweismaterialien zerstört oder verändert werden, ist es notwendig, sämtliche Daten zu duplizieren. Wie diese sogenannte "forensische Duplikation" durchgeführt wird, welche Ausrüstung und Spezialgeräte benötigt werden soll im Verlaufe des Kapitels erläutert werden.

6.2 Ein paar Begriffe vorweg

6.2.1 Die Forensik

Der Begriff Forensik stammt ursprünglich aus dem lateinischen und bedeutet so viel wie "Marktplatz" oder "Forum", da vormals Gerichtsverfahren, Untersuchungen, Urteilsverkündungen und der Strafvollzug öffentlich auf dem Marktplatz durchgeführt wurden. Der Begriff Forensik bezeichnet damit alles, was mit gerichtlichen oder kriminologischen Aktivitäten zu tun hat.

6.2.2 Kunstwort Computerforensik

Computer-Forensik ist ein Kunstwort aus Computer und Forensik und hat in den letzten Jahren an Bedeutung gewonnen. Hierbei geht es um Aktivitäten im Bereich der

digitalen Datenverarbeitung, welche dazu dienen Straftaten aufzudecken und Beweise zu sichern. Diese Beweise sind in unserem Fall Daten von Datenträgern wie Festplatten, Speicherkarten, etc. welche zur Beweissicherung ausgewertet werden sollen.

6.2.3 Computerkriminalität allgemein

Das Themengebiet Computerkriminalität umfasst alles, was mit der Ausführung von Straftaten unter Nutzung von Computern und anderen Kommunikationstechnologien zu tun hat. Hierbei gibt es unterschiedlichste Szenarien:

- Der Drogendealer, welcher per Internet und E-mail seine Geschäfte abwickelt und auf seinem PDA akribisch Buch darüber führt
- Der Raubkopierer, welcher via Internet seine "Werke" verbreitet
- Der Cracker, welcher sich in fremde Rechner "einbricht" und geheime Firmendaten stiehlt
- Mr.XY, der durch Phishing Zugangsdaten von Bankkonten Anderer ausspäht
- Die Skript-kiddies, die ein Internetforum lahm legen

Alle brechen sie Gesetze. Bei der Computer Forensik geht es darum, den Tathergang zu rekonstruieren und zu ermitteln wie der oder die Täter vorgegangen sind.

6.2.4 Ziele einer Forensischen Analyse

Die Ziele der Forensischen Analyse sind:

- Sicherstellen von Beweismaterial für weitere juristische Aktionen
- Vermeiden jeglicher juristischer Anfechtbarkeit dieses Sicherstellungsvorgangs
- Exakte Nachvollziehbarkeit und akribische Dokumentation jeglicher Aktivität der Datensicherung

6.3 Der Forensik-Arbeitsplatz

6.3.1 Vorstellung des Arbeitsplatzes

Es sei die folgende Situation gegeben: Ein Forensik-Arbeitsplatz soll eingerichtet werden um Daten von Datenträgern zu untersuchen und auszuwerten. Im Verlauf des Kapitels wird erleutert wie der Arbeitsplatz aufgebaut ist und was beim Erstellen von forensischen Duplikaten beachtet werden muß.

Dabei betrachten wir folgendes:

- Was muß beim Umgang mit Beweismaterial beachtet werden ?

- Wie kann sichergestellt werden, daß das Beweismaterial nicht verändert oder vernichtet wird ?
- Welche Möglichkeiten gibt es, während der Untersuchungen nachzuweisen, daß das Duplikat zu jedem Zeitpunkt mit den Original Daten übereinstimmt?
- Welche Maßnahmen müssen getroffen werden um das Beweismaterial und den Arbeitsplatz vor unbefugter Benutzung zu schützen ?



Abbildung 6.1: Der Arbeitsplatz

6.3.2 Hardware

Der Forensik-Arbeitsplatz besteht grundlegend aus einem PC mit folgenden technische Details:

- Intel Pentium D Prozessor 2x 2,66 GHz Taktfrequenz
- LG DVD-Brenner DVDR \pm 16x
- 80 GB SATA2 - Festplatte für das Betriebssystem
- 320 GB SATA2 - Festplatte für die Speicherung der duplizierten Daten (Festplattenimages)
- PCI IDE Controller 2 Kanal UDMA133

- Externe Spannungsversorgung
- Externe IDE UDMA-133 Festplattenanschlüsse
- Betriebssystem: GNU/Linux (Kubuntu 6.06)

Spezial Ausrüstung:

- DriveLock Kit mit Serial ATA Single Male Connector von LSK
- Mini Card Reader mit Schreibschutz USB 2.0
- Adapter 2.5 Zoll Notebook-Festplatten

Weiterhin gehört eine CD mit der benötigten Forensik-Software zum Lieferumfang.

6.3.3 Die Forensik-Tools

Der IDE Schreibschutzadapter wird zwischen Datenträger und Controller des Rechners geschaltet, um jeglichen Schreibzugriff zu verhindern. Der Schreibschutzadapter muß unbedingt verwendet werden! Denn sollte nur ein einzelnes Bit auf dem Datenträger verändert werden, ist das Beweismaterial irreparabel zerstört

Der SATAT/SATA2 Schreibschutzadapter wird zwischen Datenträger und Controller des Rechners geschaltet, um jeglichen Schreibzugriff zu verhindern. Der Schreibschutzadapter muß unbedingt verwendet werden! Denn sollte nur ein einzelnes Bit auf dem Datenträger verändert werden, ist das Beweismaterial irreparabel zerstört.

Der Speicherkartenleser kommt immer dann zum Einsatz, wenn es um das Auslesen von Speicherkarten geht. Es können folgende Karten-Formate ausgelesen werden:

- CF-I, CF-II, Smart Media, Memory Stick,
- Memory Stick, PRO, Memory Stick DUO,
- Memory Stick PRO DUO, Micro Drive,
- Multimedia Card, Secure Digital Card,
- MINI Secure Digital Card, XDcard



Die Software-Tools

Das sleuthkit 2.04 ist eine Sammlung von Unix-Tools, die einem Ermittler die nicht invasive Analyse von NTFS-, FAT-, FFS-, EXT2- und EXT3 Partitionen ermöglicht. Viele kleine Kommandozeilentools machen die Anpassung an die eigene Ermittlungsstrategie möglich. Die einzelnen Komponenten lassen sich einfach in eigenen Skripte einbinden und nutzen.

Der autopsy 2.07 Forensic Browser ist ein HTML-Interface, das die Tools aus dem Sleuth Kit mit Standard-Unix-Werkzeugen (strings, md5sum, grep, ect.) wirkungsvoll verbindet.

Merkmale von autopsy:

- Case Management
- Dateianalyse
- Analyse von Dateiinhalten
- Prüfsummen-Datenbank
- Dateityp-Analyse
- Timeline-Analyse
- Suche nach Schlüsselwörtern anhand regulärer Ausdrücke
- Metadaten und Datenblock-Analyse
- Berichtswesen

dcfldd 1.3.4 ist im Grunde genommen ein erweitertes dd¹. dcfldd wird in unserem Fall dazu verwendet, eine Partition Bitweise zu kopieren und gleichzeitig die zugehörige MD5 Prüfsumme zu erstellen.

Der Forensikfox 1.5.3 ist ein modifizierter Mozilla Firefox. Er wurde so konfiguriert, daß ein leichtes Arbeiten mit autopsy und sleuthkit gewährleistet ist.

6.3.4 Shell Skripte zur standardisierten Erstellung von Datenträgerimages

Die hier vorgestellten Shell-Skripte wurden erstellt um dem Forensiker die Arbeit zu erleichtern und um sicherzustellen, daß die Erstellung von forensischen Duplikaten von Datenträgern in immer der selben, genau nachvollziehbaren Weise abläuft. Die Skripte befinden sich im Verzeichnis /usr/local/scripts. Die Benutzerrechte wurden so gewählt, daß

¹dd - konvertiert und kopiert Dateien

die Skripte nur vom Benutzer "root" geändert werden können. So ist gewährleistet, daß nur autorisierte Personen Änderungen vornehmen können. Dies ist wichtig, da sich bei Veränderungen auch der Weg der Erstellung eines Images ändert. Änderungen müssen deshalb Dokumentiert werden. Der User Forensik hat Lese,- und Ausführungsrechte.

```
root@forensik-pc:/usr/local/scripts# ll
insgesamt 16
-rwxr-xr-x 1 root root 472 2006-06-19 14:57 forensikduplikat
-rwxr-xr-x 1 root root 746 2006-06-19 14:57 makeimage
-rwxr-xr-x 1 root root 468 2006-06-19 14:57 mountdevice
-rwxr-xr-x 1 root root 253 2006-06-19 15:20 mountimage
root@forensik-pc:/usr/local/scripts#
```

Listing 23: blub

Es besteht für jedes Skript ein Link im Verzeichnis /usr/bin, damit eine Ausführung aus jedem beliebigen Verzeichnis heraus möglich ist.

Anmerkung: Die einzelnen Skripte werden in der Reihenfolge aufgeführt, in der sie beim Erstellen des Images verwendet werden.

Skript 1: mountdevice

```
1 #!/bin/sh
2 #Shell Skript zum mounten einer Festplattenpartition
3 #
4 #Bildschirm leeren
5 clear
6
7 #Ausgabe der vorhandenen Partitionen
8 sudo fdisk -l | grep hd
9
10 echo ""
11 echo "Welche Partition soll eingehängt werden? [example: 'hdc1']"
12 read part
13
14 echo""
15 echo "Bitte geben sie den Typ des Dateisystems an [vfat|ntfs|ext2|ext3]"
16 read type
17
18 #Einbinden der Partition
19 echo "mount -v -r -t $type /dev/$part /mnt/$part"
20 mount -vr -t $type /dev/$part /mnt/$part
21
22 echo " "
```

Listing 24: blub

Das Skript `mountdevice` dient dazu, eine Partition auf einem Datenträger in das System einzuhängen.

Funktionsweise:

- Mit dem aufruf `mountimage` wird das Skript aufgerufen.
- Als erstes wird der Nutzer gefragt, welche Partition eingehängt werden soll. Eine zuvor mittels `fdisk` ausgegebene Liste mit vorhandenen Laufwerken soll ihm die Auswahl erleichtern.
- Nachdem die zu einzuhängende Partition definiert wurde, der Typ des Dateisystems angegeben werden. Dies kann sein : `vfat—ntfs—ext2—ext3` Der Dateisystemtyp ist der ausgegebenen Laufwerksliste zu entnehmen.
- Als letzter Schritt erfolgt das Einhängen der Partition mittels `"mount -vr -t $type /dev/$part /mnt/$part"`, wobei `$part` den Partitionsnamen und `$type` den Typ des Dateisystems enthalten.

Skript 2: `makeimage`

```
#!/bin/bash
2 clear
3
4 DATE='date'
5 DIR="$DATE-$1.img"
6 IMGDIR='/mnt/sdb1/Images'
7
8 if [ $# -ne 1 ]; then
9     echo "usage: $0 DEVICE"
10    echo "example: $0 hda1"
11    exit 1
12 fi
13
14 mkdir "$IMGDIR/$DIR"
15
16 echo "Erstelle image und dazugehörige Prüfsumme "
17
18 dcfldd if=/dev/$1 of="$IMGDIR/$DIR/$DATE-$1.img" hash=md5 hashlog="$IMGDIR/$DIR/$DATE
19
20 echo "Image wurde in \"$IMGDIR/$DIR\" unter \"$DATE-$1.img\" erstellt"
21 echo "Prüfsumme wurde in $IMGDIR unter $DATE-$1.img.md5 erstellt"
22 echo""
23
24 echo "Erstelle MD5-Prüfsummen der Quellpartition"
25 md5sum -b /dev/$1 > "$IMGDIR/$DIR/$DATE-$1.orig.md5"
```

```
26 echo "Prüfsumme wurde in \"\$IMGDIR/\$DIR\" unter \"\$DATE-\$1.orig.md5\" erstellt
27 echo ""
28 cd \$IMGDIR
29 echo ""
30 ls -l
31 echo ""
```

Listing 25: blub

Das Skript `makeimage` dient zur standardisierten Erstellung einer Partitionsimages. Das Skript wird mit `makeimage [Partitionsname]` aufgerufen. Der Befehl `dcfldd if=/dev/\$1 of="\$IMGDIR/\$DIR/\$DATE-\$1.img" hash=md5 hashlog="\$IMGDIR/\$DIR/\$DATE-\$1.img.md5" hashconv=after hashformat="#hash#"` ruft das Programm `dcfldd` auf, welches ein Image der als Aufrufparameter angegebenen Partition erstellt. Dieses wird im Verzeichnis `/media/sdb1` in einem eigenen Verzeichnis abgespeichert. Der Name des Verzeichnisses entspricht dem Namen des Images. Der Name des erzeugten Images wiederum besteht aus dem aktuellen Datum, der aktuellen Uhrzeit und dem Namen der Partition. Ebenfalls erstellt wir jeweils eine Prüfsumme von Originalpartition sowie des Images um prüfen zu können ob die Duplikation erfolgreich verlaufen ist.

Skript 3: mountimage

```
#!/bin/bash
2
3 IMGDIR='/mnt/sdb1/Images'
4 MNTDIR='/mnt/image'
5
6 if [ $# -ne 1 ]; then
7     echo "usage: \$0 IMAGEFILE"
8     echo "  IMAGEFILE = Dateiname aus dem images Ordner ohne Pfadangabe"
9     exit 1
10 fi
11
12 sudo mount "\$IMGDIR/\$1/\$1" \$MNTDIR -o loop=/dev/loop/0
```

Listing 26: blub

Das Skript `mountimage` dient dazu, ein erzeugtes Image in das Dateisystem einzuhängen. Vor dem Aufrufen des Images sollte in das Verzeichnis gewechselt werden, in welchem sich das Image befindet. Mit dem Befehl `mountimage *.img` wird das Image eingehängt. Anschließend kann auf die Daten des Images wie auf einen normalen Datenträger zugegriffen werden. Eingehängt wird das Image mittels des Befehls `"sudo mount "\$IMGDIR/\$1/\$1" \$MNTDIR -o loop=/dev/loop/0"` im Verzeichnis `/mnt/image`.

Skript 4: forensikduplikat

```
1 #!/bin/bash
```

```
2
3 clear
4 echo "Forensik Analysye"
5 echo "-----"
6 echo""
7 echo "Ablauf:"
8 echo "-----"
9 echo "1.Einhängen der zu untersuchenden Partition"
10 echo "2.Erstellen einer forensischen Dublikation"
11 echo "3.Einhängen des Images"
12 echo""
13 echo "Weiter mit beliebiger Taste!"
14 read n
15
16 clear
17 echo ""
18
19 mountdevice
20
21 echo ""
22 echo "Erstelle forensisches Duplikat"
23 echo "Bitte den Namen der zu duplizierenden Partition angeben: (z.Bsp: hda1)"
24 read PART
25
26 makeimage $PART
```

Listing 27: blub

Das Skript forensikduplikat dient der Erstellung eines forensischen Duplikates einer Partition. Hierzu wird der Benutzer automatisiert durch den Erstellungsvorgang geleitet.

- Im ersten Schritt wird mittels des Skriptes mountdevice eine Partition in das Dateisystem eingehängt.
- Im zweiten Schritt wird abgefragt, von welcher Partition ein Duplikat erstellt werden soll.
- Im dritten Schritt wird das Image der in Schritt 2 angegebenen Partition erstellt.

6.4 Vorbereitende Arbeiten

Bevor ein forensisches Duplikat eines Datenträgers erstellt werden kann, müssen noch ein paar Dinge vorbereitet werden. Folgende Punkte sind unbedingt zu beachten:

- Vor dem Anschließen eines Speichermediums² ist zu gewährleisten, daß der Rechner ausgeschaltet ist.
- Vor dem Abschließen eines Speichermediums muß mindestens 1/2 Minute gewartet werden, damit bewegte Bauteile zum Stillstand kommen können. Bei Nichtbeachten dieser Anweisung ist eine Zerstörung des Beweismaterials nicht auszuschließen!
- Störquellen wie Telefone oder andere Geräte, welche starke elektromagnetische Strahlung erzeugen, sind vom Arbeitsplatz fern zu halten. Ansonsten besteht die Möglichkeit einer Verfälschung des Duplikates.
- Es ist ein Protokoll anzufertigen, in dem alle Arbeitsschritte dokumentiert werden.
- Alle nicht unmittelbar zu den forensischen Ermittlungen gehörenden Materialien (Datenträger ect.) müssen vom Arbeitsplatz entfernt werden um Verwechslungen auszuschließen.
- Der Arbeitsplatz ist unbedingt vor dem Zugang unbefugter Personen zu schützen.

6.5 Anschließen eines Datenträgers

WICHTIG: Vor dem Anschluss eines Datenträgers muß gewährleistet sein, daß der Rechner ausgeschaltet und vom Stromnetz getrennt ist. Ansonsten ist eine Zerstörung des Beweismaterials nicht auszuschließen. Ausgenommen hiervon ist der Card-Reader.

Schritt 1: Den passenden Hardware-Schreibschutz-Adapter auswählen.

- Der IDE/ATA Anschluss ist relativ breit und besitzt in der Regel 40 Pins.
- Der SATA / SATA2 Anschluss ist L-förmig.
- Speicherkarten gibt es in den verschiedensten Bauformen



Abbildung 6.2: Die verschiedenen Anschlüsse im Überblick

²ausgenommen sind Speicherkartenleser



Abbildung 6.3: Die verschiedenen Adaptertypen im Überblick

Schritt 2: Den Hardwareadapter mit den Anschlüssen des Rechners verbinden.

IDE Laufwerke:

- Prüfen ob der Rechner ausgeschaltet und vom Stromnetz getrennt ist!
- Zuerst verbinden Sie den 4 poligen Stromanschluss mit dem Adapter
- Danach verbinden Sie den 40 poligen IDE - Stecker mit dem Adapter
- Nun verbinden Sie das Laufwerk mit dem Stromanschluss
- Zuletzt verbinden Sie das Laufwerk mit dem 40 poligen IDE Anschluss des Adapters

SATA / SATA2 Laufwerke:

- Prüfen ob der Rechner ausgeschaltet und vom Stromnetz getrennt ist!
- Den 4 poligen Stromanschluss mit dem Adapter verbinden.
- Den Schutzstecker des 40 poligen Anschlusses am Adapter abnehmen und sicher aufbewahren.
- Den 40 poligen IDE - Stecker des Rechners an den Adapter anschliessen. Vorsicht da sich die Pins leicht verbiegen!
- Das rote Laufwerkskabel mit dem Adapter verbinden (4 poligen Stromstecker und flachen SATA-Stecker anschliessen).

Speicherkarten:

- Schliessen Sie den kleinen Stecker des USB-Kabels an den Card-Readers an.
- Schliessen Sie den großen Stecker des USB-Kabels an den USB-Anschluss des Rechners an.
- Führen Sie die Speicherkarte in den passenden Slot des Card-Readers ein.

Schritt 3: Prüfen Sie den korrekten Sitz aller Anschlüsse. Hier nochmal alle 3 Laufwerkstypen im Überblick:

Schritt 4: Nachdem der Datenträger korrekt angeschlossen ist, kann die Stromversorgung wiederhergestellt und der Rechner eingeschaltet werden.

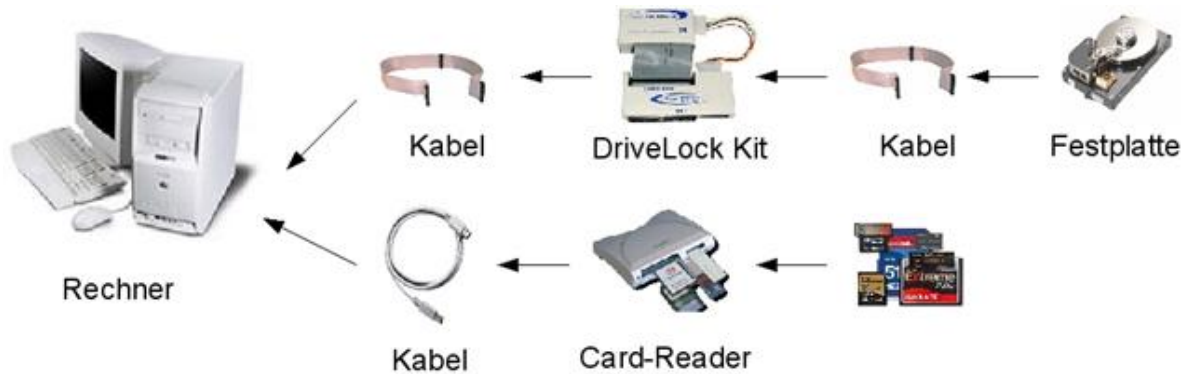


Abbildung 6.4: Gesamtschema

6.6 Mounten einer Partition

Das Mounten oder Einhängen einer Partition in das Dateisystem wird unter Verwendung des Skriptes "mountimage" durchgeführt.

Vorgehensweise: Sie sollten sich im Verzeichnis /mnt/sdb1/Images befinden.

- Aufrufen des Skriptes mit "mountimage".
- Es folgt die Frage nach der einzuhängenden Partition

```
Disk /dev/hda: 40.0 GB, 40020664320 Bytes
/dev/hda1      1          13      104391    b  W95 FAT32
/dev/hda2     14          77      514080    b  W95 FAT32
/dev/hda3     78         204     1020127+  7  HPFS/NTFS
/dev/hda4    205        4865    37439482+ 7  HPFS/NTFS
Welche Partition soll eingehängt werden? [example: 'hdc1']
```

Listing 28: blub

Soll z.B die Partition /dev/hda2 eingehängt werden, wird "hda2" als Partitionsname angegeben.

Es folgt die Frage nach dem Dateisystemtyp. Dieser kann der ausgegebenen Tabelle entnommen werden.

Es gibt folgende Typen:

```
W95 FAT32 --> vfat (FAT 32 Partition)
HPFS/NTFS --> ntfs (NTFS Windows 2000 / XP Partition)
EXT2/EXT3 --> ext2/3 (Linux Partition)
```

Listing 29: blub

Die Partition wird nun eingehängt. Bei Erfolg sollte

```
mount -v -r -t vfat /dev/hda2 /mnt/hda2
/dev/hda2 on /mnt/hda2 type vfat (ro)
```

Listing 30: blub

ausgegeben werden.

6.7 Erzeugen eines Images einer Partition

Das Erstellen eines Images einer Partition wird unter Verwendung des Skriptes "makeimage" durchgeführt.

Vorgehensweise: Sie sollten sich im Verzeichnis /mnt/sdb1/Images befinden.

Aufrufen des Skriptes mit "makeimage [Name der Partition]."

Beispiel:

```
makeimage hda1
```

Listing 31: blub

Der Name der Partition ist hier der Name der zuvor eingehängten Partition.

1. Das Image wird nun erstellt. Auf der Kommandozeile wird der Fortschritt des Vorgangs angezeigt.

ACHTUNG: Je nach Größe der Partition kann der Vorgang eine längere Zeit in Anspruch nehmen.

2. Im Anschluss an das Erstellen der Partition wird die Prüfsumme von Image und Quellpartition erstellt.

ACHTUNG: Je nach Größe der Partition kann der Vorgang eine längere Zeit in Anspruch nehmen.

6.8 Mounten des Images zur späteren Bearbeitung

Vorgehensweise: Sie sollten sich im Verzeichnis /mnt/sdb1/Images/[Name des Images] befinden. Der Name des erzeugten Images sowie der Name des Verzeichnisses in welchem sich das Image befindet besteht aus dem aktuellen Datum, der aktuellen Uhrzeit und dem Namen der Partition.

Beispiel:

```
Mi\ 21\ Jun\ 17\ :19\ :30\ CEST\ 2006-hda2.img/
```

Listing 32: blub

- Wechseln sie in das Verzeichnis mit dem einzuhängenden Image.
- Rufen sie das Skript mountimage auf mit mountimage [Name des Images]

Beispiel:

```
mountimage Mi\ 21\ Jun\ 17\ :19\ :30\ CEST\ 2006-hda2.img
```

oder

```
mountimage *.img
```

Listing 33: blub

Das Image sollte nun eingehängt sein.

6.9 Geführte Erstellung eines forensischen Duplikates

Die Erstellung eines forensischen Duplikates kann auch mittels des Skriptes forensikduplikat erstellt werden. Hier wird der Benutzer durch den Vorgang geführt. Somit kann nichts vergessen werden.

- Das Skript mit forensikduplikat aufrufen.
- Es folgt die Frage nach der einzuhängenden Partition

```
Disk /dev/hda: 40.0 GB, 40020664320 Bytes
/dev/hda1          1          13      104391    b W95 FAT32
/dev/hda2          14          77      514080    b W95 FAT32
/dev/hda3          78         204     1020127+  7 HPFS/NTFS
/dev/hda4         205        4865    37439482+ 7 HPFS/NTFS
```

Welche Partition soll eingehängt werden? [example: 'hdc1']

Listing 34: blub

Soll z.B die Partition /dev/hda2 eingehängt werden, wird "hda2" als Partitionsname angegeben.

Es folgt die Frage nach dem Dateisystemtyp. Dieser kann der ausgegebenen Tabelle entnommen werden.

Es gibt folgende Typen:

W95 FAT32 --> vfat (FAT 32 Partition)

HPFS/NTFS --> ntfs (NTFS Windows 2000 / XP Partition)

EXT2/EXT3 --> ext2/3 (Linux Partition)

Listing 35: blub

Die Partition wird nun eingehängt. Bei Erfolg sollte

```
mount -v -r -t vfat /dev/hda2 /mnt/hda2  
/dev/hda2 on /mnt/hda2 type vfat (ro)
```

Listing 36: blub

ausgegeben werden.

- Der Name der Partition die gerade eingehängt wurde wird abgefragt. z.B. hda2
- Das Image wird nun erstellt. Auf der Kommandozeile wird der Fortschritt des Vorgangs angezeigt.

ACHTUNG: Je nach Größe der Partition kann der Vorgang eine längere Zeit in Anspruch nehmen.

Im Anschluss an das Erstellen der Partition wird die Prüfsumme von Image und Quellpartition erstellt.

ACHTUNG: Je nach Größe der Partition kann der Vorgang eine längere Zeit in Anspruch nehmen.

Das Image sollte nun erstellt sein. Weiter geht es mit dem Einhängen des Images.

Vorgehensweise: Sie sollten sich im Verzeichnis /mnt/sdb1/Images/[Name des Images] befinden. Der Name des erzeugten Images sowie der Name der Verzeichnisses in welchem sich das Image befindet besteht aus dem aktuellen Datum, der aktuellen Uhrzeit und dem Namen der Partition.

Beispiel:

```
Mi\ 21\ Jun\ 17\ :19\ :30\ CEST\ 2006-hda2.img/
```

Listing 37: blub

- Wechseln sie in das Verzeichnis mit dem einzuhängenden Image.
- Rufen sie das Skript mountimage auf mit mountimage [Name des Images]

Beispiel:

```
mountimage Mi\ 21\ Jun\ 17\ :19\ :30\ CEST\ 2006-hda2.img
```

oder

```
mountimage *.img
```

Listing 38: blub

Das Image sollte nun eingehängt sein.

7 Ausblick

7.1 Sicherheits-Kompendium

Sicherheit ist kein Zustand den man erreichen kann, sondern ein sich stetig ändernder Prozess. Erschwerend kommt hinzu, dass gefundene Sicherheitslücken schnell auch in Angriffe umgesetzt werden. Aus diesem Grund ist es wichtig ein Sicherheitsbewusstsein zu entwickeln. Aufgrund der ständigen Weiterentwicklung von Angriffen ist es quasi nicht möglich ein Sicherheitskompendium zu schreiben welches zu jeder Zeit aktuell ist. Themen für ein weiteres Projekt sind daher reichlich vorhanden. So könnte man z.B. über Viren und Würmer schreiben, über lokale Angriffe auf Windows oder Linux, über Rootkits, etc..

7.2 Honeypot

Das Thema Honeypot ist noch verhältnismäßig neu und es gibt nicht sehr viele Dokumentationen oder Tutorials rund um dieses Thema. Jedoch lohnt die Mühe, Honeypot ist ein unglaublich spannendes Thema. Falls eine Projektgruppe Interesse hat, das Thema fortzusetzen, würde wir jedoch empfehlen, dieses Thema als einzelnes Projekt zu verfolgen. Für weitere Informationen Sie auf <http://www.projecthoneypot.org/> schauen. Diese Seite verweist auch auf die wichtigsten Seiten bezüglich Honeypots.

7.3 Forensik

Wie geht es weiter? Die Computerforensik ist ein spannendes Themengebiet bei dem es auf alle Fälle lohnenswert ist, sich weiterhin damit zu beschäftigen. Ein erster Einstieg ist durch die Einrichtung des Forensik-Arbeitsplatzes gelungen. Es ist die Möglichkeit geschaffen worden, Datenträger zu duplizieren um mit dem ertellten Duplikat zu arbeiten und dieses zu analysieren. Hierauf kann nun weiter aufgebaut werden und es gibt noch so manches zu tun. Im direkten Anschluss sollte nun das nötige know-how erarbeitet werden, wie die Analyse der gewonnenen Daten durchgeführt wird. Hierzu muß eine detaillierte Anleitung erstellt werden, wie mit autopsy und sleuthkit eine Analyse durchgeführt werden kann. Weiter sollten die Skripte verbessert und weiterentwickelt werden.