Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Lehrstuhl für Rechnernetze und Telematik

WS 2007/2008

Pro-Seminararbeit

Network Security
Schriftliche Ausarbeitung

Ivo Malenica

6. Januar 2008

Die Vorgabe(Stallings, Kapitel 21): Die Bedrohungen für ein Rechnernetz lassen sich in zwei Kategorien aufteilen. Die erste Kategorie zeichnet sich dadurch aus, dass ein Angreifer versucht Information über eine Verbindung zwischen Teilnehmern zu erhalten. Dieser Typ wird als passive Bedrohung bezeichnet. Die zweite Kategorie beinhaltet Angriffsmuster, die gezielt Datenverkehr manipulieren oder ungültige Verbindungen erzeugen. Dieser Typ als aktive Bedrohung bezeichnet.

Inhaltsverzeichnis

1 Einleitung

Wenn wir einen Blick in die Welt werfen, so werden wir feststellen, dass IT Bestandteil unseres täglichen Lebens ist. Menschen kommunizieren über das Internet, verwalten ihre persönlichen Daten auf Computern oder tätigen ihre Einkäufe über das Web. In vielen Bereichen der Wirtschaft kommen vernetzte Systeme zum Einsatz. Daraus ergibt sich eine gewisse Abhänigkeit gegenüber diesen Systemen bzw. Netzwerken. Die logische Schlussfolgerung ist ein starkes Bedürfnis nach Schutz für diese Systeme. Dieses Paper versucht einige Themengebiete zu erläuten, wie Verschlüsselung, Überwachung und Aufbau eines Rechnernetzes. Diese sind für die Sicherheit des Rechnernetzes von Bedeutung. Das Thema ist im Gesamten ein sehr weites Feld. Es würde den Rahmen dieses Papers sprengen, wenn man das Thema voll ausschöpfen würde. Bücher, Artikel und Vorlesung zu diesem Themenkomplex sind im Literaturverzeichnis dieses Papers zu finden.

2 Passive/Aktive Attacken

2.1 Passive Attacken

Nach William Stallings [8] sind passive Attacken dadurch charakterisiert, dass man versucht, einen Datenaustausch abzuhören bzw. zu beobachten, ohne dabei aufzufallen. Da viele Informationen in einem Netzsegment z.B. das eigene Heim-LAN in Klartext übermittelt werden, stehen Angreifern für den Erhalt von Information Tür und Tor offen. Eine Lösung wäre, dass man seine Verbindungswege verschlüsselt (ipsec,tls). Jedoch sind die Gefahren damit nicht beseitigt. Ein Angreifer könnte dann noch immer den verschlüsselten Datenverkehr beobachten und versuchen, mit den aufgezeichneten Daten einen Rückschluss auf das verwendete Verschlüsselungsverfahren zu bekommen. Dieses Vorgehen fasst man auch unter dem Begriff traffic analysis(Datenverkehrs analyse) zusammen. Das Erkennen eines "Lauschangriffes"bzw. einer passiven Attacke ist im allgemeinen nicht einfach. Der Angreifer nimmt keine Veränderungen an den Daten vor. Deswegen gibt es weniger Indizien, die auf einen solchen Angriff schließen lassen. Somit ist auch das Vorbeugen gegen einen solchen Angriff von größerer Bedeutung als das Erkennen solcher Angriffe. Beispiele aus dem täglichen Leben sind z.B. das Abhören eines Netzwerkes, die über einen Hub verbunden sind. Der Hub ist der Bitubertragungsschicht des ISO/OSI Schichten Modells zugewiesen. Vom Prinzip her arbeitet er wie ein Repeater d.h. alle übertragenen Daten werden an alle angeschlossenen Teilnehmer gesendet.

2.2 Aktive Attacken

Diese sind dadurch charakterisiert, dass man Veränderungen an einer Übertragung von Daten vornimmt oder bewusst Datenströme erzeugt, die beim Empfänger ein ungewolltes Verhalten bewirken. Nach Stallings [8] wird dieser Typ von Angriffen nochmals in vier Unterkategorien eingeteilt.

- **Masquarde**: Das Verbergen der Identität. Angenommen, man hätte eine Firewall, die nur bestimmten Rechnern mit einer gewissen IP Zugang zu einem Netz oder Dienst gewährt. Ein Angreifer könnte nun hingehen und mit Absicht den IP Header so verändern, dass ihn der Dienst akzeptiert. Durch Verändern der MAC Adresse in einem Netzsegment kann man seine Identität verschleiern.[1].

- **Replay**: Das Wiederversenden aufgzeichneter Daten (passiv), um ein bestimmtes Verhalten zu erzeugen. Dieser Angriffstyp wird bei verschlüsselten WLAN Netzen (WEP, WPA) verwendet. Man versucht damit, Antwortverkehr zu erzeugen, um mehr verschlüsselte Datenpakete aufzuzeichnen. Sie dienen dazu, Rückschlüsse auf den verwendeten Schlüssel zu erhalten.

- **Modifcation of messages**: Abändern von Nachrichten, um ein bestimmtes Verhalten beim Empfänger zu erzeugen. Beabsichtigtes Fragmentieren von Datenpaketen in TCP/IP basierten Netzen, wäre ein Beispiel dafür. Ein Angreifer zerhackt sein Datenpaket in kleinere Datenpakete, wobei jedes nur eine Kopie des IP Headers erhält. Nur das erste Fragment enthält den TCP Header. Diese Technik findet unter anderem Anwendung beim Versuch, IDS-Systeme zu umgehen(weiter unten werden IDS noch besprochen). Ein bekanntes Tool zur Erzeugung von fragmentierten Paketen ist fragroute.[2]

- **Denial of Service (DOS)**: Dieser Angriffstyp baut auf Schwachstellen der Betriebsysteme oder Dienste auf, die auf einem Rechner laufen. Dabei versucht man, durch diese Schwachstellen den normalen Arbeitsprozess des Rechners zu stören bzw. diesen unter Umständen auch völlig stillzulegen. So gab es z.B. nach heisec.de in der Linux kernel verion 2.6.23 eine im 80211-WLAN-Code die Möglichkeiten, durch preparierte WLAN FRAMES einen Kernel PANIC zu provozieren.[3]

3 Verschlüsselungsverfahren

Dieses Paper behandelt in erster Line Strukturen, wie man Rechnernetze absichert gegen die im vorherigen Kapitel eingeführten Angriffstypen. Daher werde ich nicht bis ins Detail erläutern, wie die vorgestellten Verschlüsselungsverfahren funktionieren. Vielmehr werde ich sie nur kurz erklären.

3.1 Symmetrische Verschlüsselung

Bei symmetrischen Verschlüsselungen ist dem Empfänger und Sender ein gemeinsamer Schlüssel bekannt. Dieser wird verwendet, um mit einem Verschlüsselungsalgorithmus zu verschlüsseln und zu entschlüsseln.(siehe Abbildung). Dabei müssen nach Stallings [8] folgende Bedingungen gelten. Bei Kenntnis des Algorithmus und einigen Stücken verschlüsselter Nachrichten sollte man nicht auf den Klartext schließen können oder

[1] man ifconfig bei Unix Systemen und Unix nahen Systemen
[2] http://monkey.org/ dugsong/fragroute/
[3] http://www.heise.de/security/suche/ergebnis/?rm=result;q=dos;url=/security/news/meldung/98706/;words=DOS

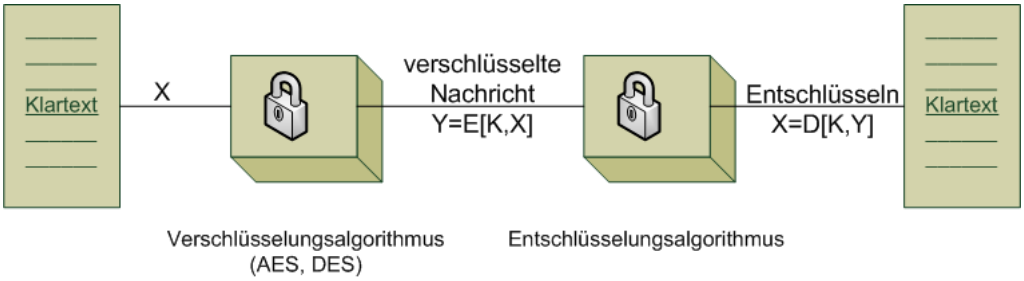

Abbildung 1: Symmetrische Verschlüsselung, aus [8] nachempfunden

auf den verwendeten Schlüssel. Der Schlüssel muss beim Sender und Empfänger sicher aufbewahrt sein. Bekannt für diese Klasse sind DES[4] und AES[5]. Aber auch in Protokollen wie ipsec und tls finden sich solche Verfahren wieder.

3.2 Assymmetrische Verschlüssungverfahren/Öffentliche Verschlüsselungsverfahren

Ein öffentliches Verschlüsselungsverfahren zeichnet sich dadurch aus, dass zum Verschlüsseln und Entschlüsseln ein Schlüsselpaar erzeugt wird, das aus einem privaten Schlüssel, den wir d nennen, und einen öffentlichen Schlüssel,den wir e nennen, besteht. Der öffentliche Schlüssel wird "öffentlich gemacht", also an die Teilnehmer verteilt. Mit einem Verschlüsselungsalgorithmus, den wir E_e nennen, wird dann eine Klartextnachricht m verschlüsselt: $E_e(m) = c$, wobei c die verschlüsselte Nachricht ist. Zum Entschlüsseln wird ein Algorithmus verwendet, den wir D_d nennen mit $D_d(c) = m$. Dabei werden an öffentliche Verschlüsselungsverfahren folgende Bedingungen gestellt. Es darf nicht möglich sein, dass man bei Kenntniss von c und E_e auf m schließen kann. Das bedeutet auch, dass wir d nicht aus e berechnen können. Man bezeichnet E_e in diesem Fall auch als eine Falltürfunktion bzw. Falltüralgorithmus oder auch als eine Einwegsfunktion. Das Entschlüsseln der Nachricht ist nur mit D_d möglich. Darin liegt auch der Unterschied zun den symmetrischen Verfahren. Diese zeichnen sich dadurch aus, dass e und d genau die gleichen Schlüssel sind.

Ein bekanntes Verfahren ist:

1. RSA: Benannt nach Ronald L. Rivest, Adi Shamir und Leonard Adleman.: Es gehört mit zu den bekanntesten Verfahren. Zur Erzeugung der Schlüssel werden große Primzahlen herangezogen und diese miteinander mutlipliziert. Anhand von einigen Berechnungen wird dann ein öffentlicher Schlüssel und ein privater Schlüssel erzeugt. Die Stärke dieses Verfahrens zeichnet sich dadurch aus, dass das Faktorisieren dieses Produkts sich schwieriger erweist als das Erzeugen von Primzahlen.[6].

[4]Data Encryption Standard
[5]Advanced Encryption Standard
[6]Vorlesung DAS, gehalten von Prof. Dr. Siebert SS2007

Öffentliche Verschlüsselungsverfahren spielen in vielen Bereichen, wie der digitalen Signatur, eine wichtige Rolle. Dabei signiert ein Sender mit seinem privaten Schlüssel eine Nachricht. Mit Hilfe des öffentlichen Schlüssels kann dann ein Empfänger die Identität des Senders überprüfen und erkennen, dass diese Nachricht von diesem Sender stammt.

3.3 Anwendung von Verschlüsselungsverfahren

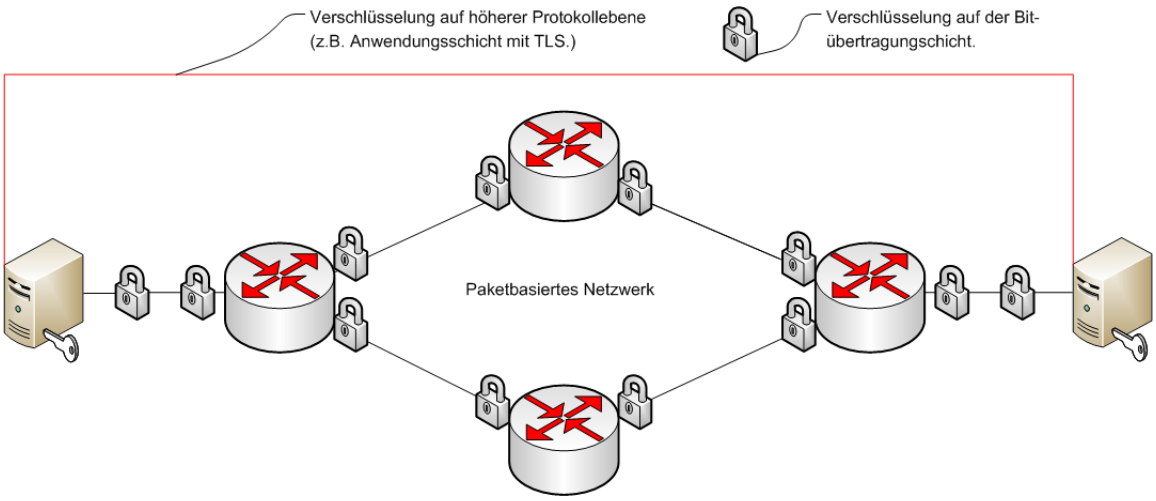

Abbildung 2: verschlüsseltes Netz aus [8] nachempfunden.

Eine Möglichkeit, einem Angreifer das Lauschen auf einer Leitung zu erschweren, ist nach Stallings [8] das Verschlüsseln von Leitungen zwischen den Teilnehmern eines paketaustauschenden Netzwerks(siehe Abbildung). Dabei bieten sich verschiedene Methoden an, die von der Skizze beschrieben werden. Bei der Leitungsverschlüsselung, dargestellt durch die Schlösser, wird auf der Bitübertragungschicht mit einem symmetrischen Verfahren wie AES verschlüsselt. Vor und hinter jeden Teilnehmer des Netzes befindet sich dann eine Schnittstelle zum Verschlüsseln/Entschlüsseln. Diese kann als Bridge realisiert werden oder direkt die Netzschnittstelle des Teilnehmers sein. Bei der Ende-zu-Ende Verschlüsselung wird das Verschlüsseln dem Sender und Empfänger überlassen.(Computer mit dem Schlüssel). Dabei werden jedoch nur die Daten verschlüsselt, die für das Routing im Netz nicht gebraucht werden, z.B. alles über Transportsschicht. Ein Protokoll, das dies tut, ist das TLS. Dazu aber später mehr. Auch hier sind symmetrische Verschlüsselungsverfahren, wie das AES, anwendbar. In Kombination, wie in der Illustration dargestellt, liefert dieses Verfahren eine hohe Sicherheit. Ein Angreifer kann nur mit hohem Aufwand die Leitungsverschlüsselung knacken. Gelingt ihm das, so kann er Informationen über die Netzstruktur bekommen, wobei die eigentlichen übermittelten Daten zwischen Sender und Empfänger wieder verschlüsselt

http://home.mathematik.uni-freiburg.de/siebert/Veranstaltungen/DAS.SS07/DAS.pdf

sind. Die Analyse des verschlüsselten Datenverkehrs bei Leitungsverschlüsselungen kann nach Stallings[8] zusätzlich noch durch sogenanntes Traffic Padding erschwert werden. Solange die Schnittstelle zum Verschlüsseln/Entschlüsseln nichts zu verschlüsseln hat, übermittelt sie wahllos per Zufall erzeugte Daten. Sobald reale Daten vorhanden sind, werden diese verschlüsselt und übertragen.

3.4 Ende zu Ende Verschlüsselung mit TLS/SSL

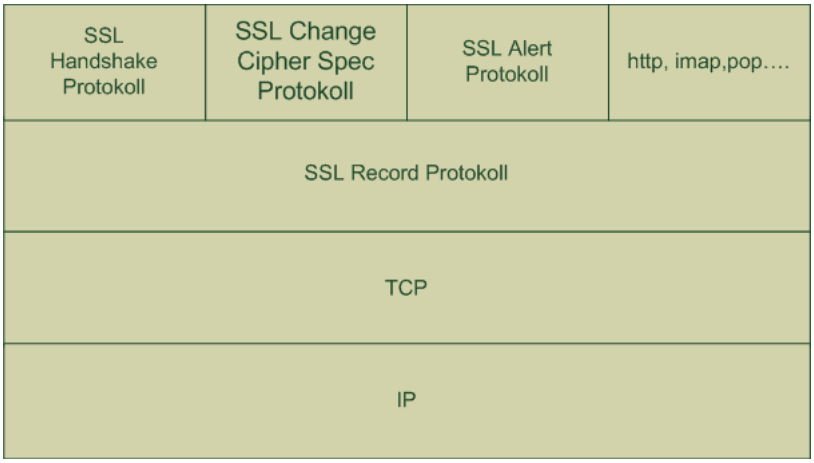

Abbildung 3: SSL/TLS,aus [8] nachempfunden

SSL (SECURE SOCKET LAYER)ist ein Protokoll zur verchlüsselten Datenübertragung in einem Rechnernetz. Im ISO/OSI Schichten Modell liegt es über der Transportschicht. Sein Nachfolger heißt TLS(Transports Security Layer, RFC 2246). Es wird von der The TLS Working Group der ietf seit 1996 verwaltet und entwickelt[3]. Es dient dazu, höheren Protokollen,die keine eigenen Methoden zur Sicherung der Übertragung besitzen, diese Methoden bereitzustellen(www,imap,pop,smtp....). Genau betrachtet spaltet sich TLS in verschiedene Schichten auf.

1. **SSL Record Protocol**: Es dient zur symmetrischen Verschlüsselung zwischen den Teilnehmern. Bevor dies jedoch möglich ist, handeln beide Teilnehmer einen Schlüssel mit dem Handshake Protokoll aus. Dieser Schlüssel ist einmalig für diese Verbindung.

2. **SSL Handshake Protokoll**: Es setzt auf dem SSL Record Protokoll auf. Bevor überhaupt irgend etwas übertragen wird, handelt dieses Protokoll den verwendeten Schlüssel aus. Es authentifiziert auch den anderen Teilnehmer. Dies ist nur eine optionale Einstellung. In Client/Server Netzwerken ist es eher üblich, dass sich nur der Server gegenüber dem Clienten authentifiziert

3. **Alert Protokoll**: Es dient zum Austausch von SSL spezifischen Alarmnachrichten. Je nachdem, welche Nachricht übermittelt wird, ändert sich das Verhalten von den

Teilnehmern. Dies kann das sofortige Beenden der Verbindung sein, bis hin zur Verweigerung neuer Verbindung.

4. **Cipher Spec Protokoll**: Dies ist ein einfaches Protokoll. Die übertragene Nachricht ist ein byte. Es dient dazu, die Verbindung aufrecht zu erhalten, damit die verwendeten Schlüssel gültig bleiben.

. Ein Beispiel für den Einsatz dieses Protokolls sind die mail-server der Informatik an der Universität Freiburg. Auf imap.informatik.uni-freiburg.de Port 993 läuft ein imap zum Abholen der E-Mails. Der Server authentifiziert sich hier nur gegen über den Client. Umsetzungen dies Protokolls sind openssl [7] und gnutls. [8] Ein anderes Protokoll, welches man zur Ende-zu-Ende Verschlüsselung verwenden kann, ist ipsec.

3.5 Schlüsselverteilung

Damit jeder Knoten bzw. jede Verschlüsselungs/Entschlüsselungs-Schnittstelle den Klartext verschlüsseln und entschlüssen kann, muss ihm der Schlüssel dazu bekannt sein. Damit stellt sich die Frage, wie man den Geräten die Schlüssel zukommen lässt. Nach Stallings [8] ergeben sich hierfür vier Möglichkeiten:

1. Ein Schlüssel kann von Knoten A gewählt werden und zu anderen Knoten B physikalisch übertragen werden(auch der Administrator zufuß).

2. Ein Schlüssel wird von einer dritten Instanz gewählt und an A sowie B ausgeliefert.

3. Insofern A und B noch mit einem gültigen Schlüssel arbeiten, können sie den neuen Schlüssel mit dem alten übertragen.

4. Falls A und B eine verschlüsselte Verbindung zu einem Teilnehmer C haben, so kann C den Schlüssel, den A und B verwenden wollen, verschlüsselt zukommen lassen.

Methode 1 und 2 setzt eine manuelle Auslieferung voraus, was bei einem größeren Rechnernetz zu hohem Aufwand führen kann. Methode 3 hat die Schwachstelle, dass bei einmaligem Knacken eines Keys, der neue übertragene Schlüssel mitgehört werden kann. Stallings schlägt für großere Rechnernetze folgendes Schema vor.[8] Rechner A und B haben eine verschlüsselte Verbindung zu einem Schlüsselverteiler (Teilnehmer C). Möchten zwei Teilnehmer miteinander komunizieren, können sie über C einen einmaligen Schlüssel aushandeln. Der Vorteil dieses Aufbaus ist die Verwaltung der Schlüssel. Desweiteren erhält man die Möglichkeit, mit verschiedenen Schlüsseln zu arbeiten, ohne einen großen Aufwand für die Konfiguration betreiben zu müssen. Ein Angreifer, der den Verkehr mitschneidet, kann so noch schlechter Muster im verschlüsselten Text finden, um Rückschlüsse ziehen zu können. Der Nachteil ist, dass C den zentralen Angriffspunkt darstellt. Übernimmt man diesen, hat man Zugriff auf alle verwendeten Schlüssel. Führt man auf diesen eine DOS Attacke aus, kann das auch zum Stillstand des Netzwerks führen.

[7] http://www.openssl.org/
[8] http://www.gnu.org/software/gnutls/

4 Einsatz von Firewalls

Der Einsatz einer Firewall trägt maßgeblich zur Sicherheit eines Netzwerkes bei. Dabei gibt es verschiedene Umsetzungen bzw. Methoden. Die folgenden Umsetzungen und Methoden orientieren sich an der Vorlesung System II von Prof Dr. Schindelhauer 2007.[7]

4.1 Einteilung

1. **Netzwerk Firewalls**: Diese trennen das externe Netz vom internen Netz oder von einer DMZ (demilitarisierte Zone). Auf diesen Punkt wird später eingegangen werden.

2. **Host Firewalls**: Diese überwachen den Traffic und Prozesse auf einem System und schützen den Rechner von außen und von innen.

4.2 Möglichkeiten konkreter Methoden

1. **Paketfilter**: Paketfilter werten eingehende Pakete anhand von Header Information aus. Mögliche Daten können z.B. Sender und Empfänger sein, sowie der angesprochene Dienst. Anhand dieser Information kann entschieden werden, ob ein Paket angenommen, weitergegeben oder verworfen wird. Bei Router finden diese Filtersysteme Einsatz, um Pakete mit gefälschten Adressen zu verwerfen. Bekannte Paketfilter sind PF (packet filter) von openbsd,[6] ipfw von Freebsd[4], der auch von MAC OS X verwendet wird und iptables [5] für linux. Paketfilter finden Anwendung in Netzwerk Firewalls und Host Firewalls. Paketfilter sind keine Contentfilter.

2. **Contentfilter**: Contentfilter analysieren die Daten eines Pakets. Sie finden Anwendung z.B. beim Entgegenwirken von SPAM und Viren bei Mails. Ein bekannter Vertreter ist amavis [1] für das Filtern von Email nach Spam und Viren.

3. **Proxy**: Sie dienen dazu, den Datenverkehr unteranderem für www zu bündeln. Eingehende wie ausgehende Pakete laufen über diesen Rechner. Dort kann der Traffic analysiert werden durch ein Content Filter System. Diese Rechner sollten besonders geschützt werden. (Siehe Bastion Host) Bekannte Proxys sind Squid und Privoxy.

4. **NAT RFC(3022): Network Address Translation**: Es dient dazu, einem ganzen Netzsegment eine Adresse zuzuweisen. Üblicherweise bieten Router diese Funktionalitat für IP Netze an, die keinen größeren Vorat an öffentlichen Adressen haben. Dieser Router besitzt meist eine öffentliche IP Adresse im Internet. Wenn nun Rechner vom internen Netz in das externe eine Verbindung aufbauen wollen, werden am Router die Header Einträge der ausgehenden Pakete angepasst. Dabei schreibt der Router seine öffentliche IP Adresse in den IP Header und den geöffneten Port für diese Verbindung in den TCP Header. Zur Zuordnung der Verbindung an den Host im internen Netz codiert er den Port und die Quell-adresse

so um, dass man anhand dieser Informationen entscheiden kann, welchen Rechnern er die Antwortpakete im internen Netz zuschicken muss. Dieses wird auch als Network Address Port Translation bezeichnet und wird meistens nur mit NAT benannt. Einerseits wird damit der Adressenknappheit, die in IPv4 Netzen besteht, entgegengewirkt. Andereseits können Rechner im internen Netz nicht direkt von Rechnern aus dem externen Netz angesprochen werden. Eine Verbindung lässt sich auch nicht auf einen Rechner im internen Netz zurückführen. Rechner aus dem externen Netz erhalten keine Information über die Struktur des internen Netzes. Der NAT(NAPT) Router erfüllt dabei die Rolle des Bastion Host.

5. **Bastion Host**: Dies sind spezielle Rechner, die besonderen Gefahren ausgesetzt sind. Sie stehen in unsicheren Netzen und bieten Dienste an wie www. Sie zeichnen sich dadurch aus, dass sie einen hohen Sicherheitsgrad besitzen. Dieser ist auch nötig, da Angreifer direkten Zugriff auf ihn haben.

5 Was tun,wenn der Einbruch geglückt ist?

In den vorherigen Kapiteln habe ich verschiedene Lösungansätze vorgestellt, die die Sicherheit in einem Rechnernetz erhöhen. Die Gefahr, dass ein Angreifer diese Schutzmaßnahmen überwindet, ist damit nicht ausgeschlossen. Damit werden Maßnahmen nötig, die bei einem erfolgreichen Einbruch in ein Rechnernetz den Schaden klein halten.

5.1 Screened Subnet

In der Vorlesung System II von Prof Dr. Schindelhauer 2007 [7] wurde das screened Subnet eingeführt.

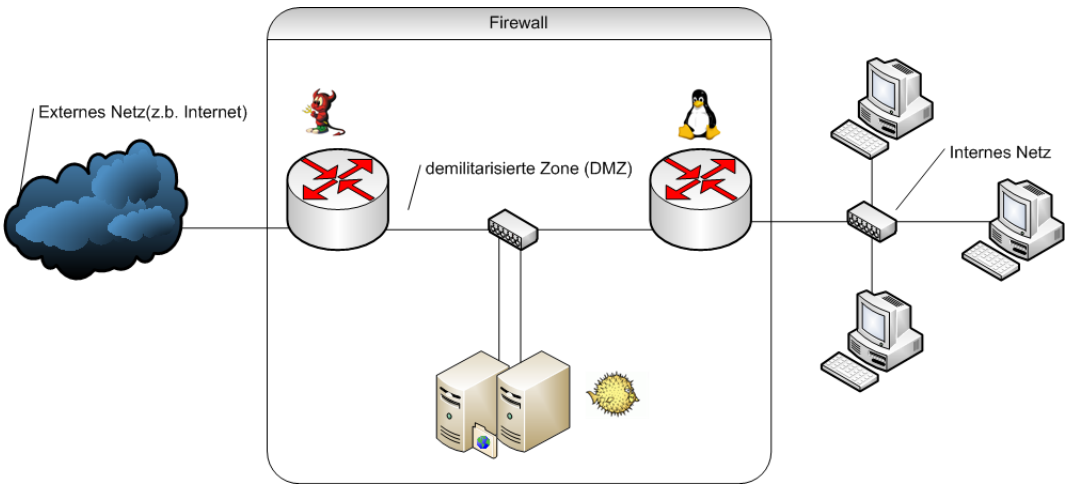

Abbildung 4: Screened Subnet,DMZ,entnommen aus [7] und angepasst.

Das Modell entpsricht auch dem zweistufigen Firewallkonzept des Grundschutzkatalogs der BSI[2]. Dabei teilt man sein Netz in folgende Bereiche auf: externes Netz, DMZ(demilitarisierte Zone) und das private Netz. Der erste Router trennt das externe Netz(Internet) von der DMZ. Der zweite Router trennt die DMZ von dem internen Netz. Dabei nimmt die DMZ eine besondere Rolle ein. Möchte man nach außen hin für das Internet Dienste(mail,www,ftp usw) bereitstellen bzw. Dienste für das interne Netz bereitstellen, stellt man diese Systeme (Bastion Host) in DMZ. Wird ein Rechner von einem Angreifer übernommen, sind passive und aktive Attacken auf die DMZ beschränkt. Damit sind auch die anderen Systeme in der DMZ angreifbar. Jedoch kann man auch dies einschränken durch geschicktes Filtern der Pakete. Eine mögliche Filterregel könnte lauten:

1. Lasse nur Rechner aus dem internen Netz eine Verbindung aufbauen in die DMZ.
2. Rechner aus der DMZ dürfen keine Verbindungen von Rechnern aus der DMZ akzeptieren.
3. Rechner aus der DMZ dürfen keine Verbindung zu Rechnern im internen Netz aufbauen.

Ein weitere wichtiger Punkt ist die Vielfältigkeit der verwendeten Komponeten und Systeme. Man sollte nicht alle Komponenten vom gleichen Hersteller verwenden. Die Möglichkeit würde sonst bestehen, dass man einen Fehler für mehrere Systeme verwenden könnte. In der Skizze oben, wird z.B für den Bastion eine openBSD verwendet. Für die Firewall, die das Internet von der DMZ trennt, wird ein Freebsd verwendet. Für die Firewall, die das interne Netz von der DMZ trennt, wird eine Linux verwendet. Ich möchte damit nicht behaupten, dass dies die einzig richtige Wahl ist. Jedoch zeichnen sich diese Systeme dadurch aus, dass sie allgmein sehr zuverlässig sind.[9]

5.2 Intrusion-Detection-Systeme

IDS bedeudet Intrusion-Detection-Systeme. Nach der BSI [2] bedeutet Intrusion-Detection das 'Überwachen von Computersystemen bzw. Rechnernetzen zur Erkennung von Angriffen und Missbrauch'. Sie soll als Prozess verstanden werden und bedarf technischer Hilfsmittel. Intrusion-Detection-Systeme bedeuten dann die 'Zusammenstellung von Werkzeugen, die den gesamten Intrusion-Detection-Prozess von der Ereigniserkennung über die Auswertung bis hin zur Eskalation und Dokumentation von Ereignissen unterstützen.' IDS sind Mustererkenner. Sie sammeln Daten,werten diese aus und versuchen, ein Angriffsmuster zu erkennen. Wird ein Muster erkannt, schlägt das System Alarm.Siehe Abb.[5]. Sie teilen sich in 2 Arten auf.

1. **Host basierte IDS**: Diese laufen auf den zu überwachenden Systemen. Zur Analyse von Daten werden Log-Dateien herangezogen, sowie Daten, die der Kernel liefert. Sie dienen mit dazu, Rechtsüberschreitungen von Usern zu dokumentieren

[9] http://uptime.netcraft.com/up/today/top.avg.html

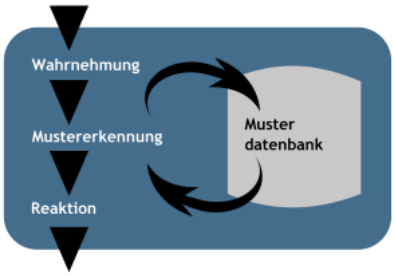

Abbildung 5: IDS-Mustererkennung,entnommen von www.wikipedia.org

oder den möglichen Einsatz eines Trojaners zu erkennen. Nach [9] lässt sich dieser Typ von IDS in folgende Kategorien einteilen.

a) **Protokolldatei-Monitore**: Diese ziehen zur Untersuchung Log Dateien des System heran. Mögliche Kandidaten befinden sich z.B. unter Unix und Unix-nahen Systemen in /var/log wie pflog(BSD) oder messages. Ein Programm, das zu dieser Gruppe gehört, könnte dann Tools verwenden wie cat, grep oder tcpdump, um diese logfiles auszufiltern. Findet es vorher festgelegte Muster, die auf einen Angriff hinweisen, schlägt das Programm Alarm. Das Ausfiltern der Information findet auf einer höheren Ebene statt. Angriffe, die auf tieferen Ebenen, wie ein Portscan, stattfinden, können übersehen werden. (Bemerkung: Man kann natürlich auch diesen Angriffstyp mit Protokolldatei-Monitoren erkennen. Dies setzt aber voraus, dass der Paketfilter Anfragen auf Ports mitprotokolliert und diese dann in einer Log ablegt. Mit einem Analyse-Programm können diese ausgwertet werden.) Swatch ist ein Programm, das zu dieser Gruppe gehört.

b) **Integritäts-Monitore**: Diese Programme überwachen eine zuvor definierte Menge von Dateien auf gewisse Eigenschaften. Diese Eigenschaften wären unteranderem Dateigröße, Dateiflags(rwx usw.), letzter Zugriff beim Schreiben/Lesen und Hash-Werte. Bei Manipulation einer Datei aus dem Definitionsbereich protokolliert das Programm alle Veränderungen mit. Tripewire, AIDE sind Tools, die unter diese Kategorie fallen. Man kann auch mit einfachen Unix-Tools dieser Aufgabe gerecht werden. Dabei baut man sich eine sogenannte baseline mit dem Befehl find und ls. In ihr listet man die zu überwachenden Dateien mit ihren wichtigsten Attributen auf. Dieses Vorgehen schreiben wir als Script. Dieses wiederrum führen wir dann in regelmäßigen Zeitabständen mit Cron aus und speichern die neuen Auflistungen unter einen anderen Namen ab. Nun hat man die Möglichkeit, mit dem Befehl diff Unterschiede zwischen der baseline und den vom Script erstellten Dateien zu finden.[10]

[10]GO!Linux Ausgabe 5 2000, Intrusion Detection unter Linux. Wer hackt den da?, von Oliver Müller

2. **Netzwerk basierte IDS**: Sie zeichnen den Netzverkehr eines Netzsegments auf und versuchen, an den gewonnenen Daten Angriffe zu erkennen. Oftmals laufen sie auf separaten Systemen,damit Rechner, die Dienste anbieten, nicht noch stärker belastet werden. Eine Bridge würde sich dafür anbieten. Sie sammelt die Pakete auf einem Inteface und gibt diese auf einem zweiten Interface wieder auf die Leitung. Für höhere Protokolle wie IP ist dieser Rechner(Sensor für das IDS) transparent. Nach [9] lässt sich dieser Typ von IDS in folgende Kategorien einteilen.

   a) **Signatur-basierte IDS**: Dieser Typ versucht Signaturen im Hex-Code eines Pakets zu erkennen. Dabei vergleicht es die gefilterten Zeichenketten mit einer Datenbank ab. Siehe Abb.[5]. Findet es ein passendes Muster, schlägt das System Alarm. Dabei sollte das IDS mit fragmentierten Paketen umgehen können. Das bedeutet, dass das IDS fähig sein muss, aufgeteile Pakete zu verfolgen und das zusammengesetze Paket dann wieder zu untersuchen. Diese IDS nennt man zustandsorientiert. Ansonsten wäre es möglich, dass ein Angreifer die Pakete so zerhackt, dass die Signaturen über diese verteilt sind. Ein Beispiel: Es ist ihnen bekannt, dass das Programm netcat in ihrem Netz verwendet wird. Sie wissen auch, das netcat sich als Backdoor zweckentfremden lässt (nc -l -p 2000 -d -e /bin/bash). [11] Nun könnte man mit einem Signatur-basierten IDS Packete nach Usernamen, wie root, paul, Mr.T, wonderwoman...., oder nach Befehlen, wie rm, cd, su...., durchsuchen. Falls einer der Teilnehmer in ihrem Netz diese Fähigkeit von netcat verwendet und auch noch root Rechte besitzt, könnten z.B. Zeichenketten, wie in Abbildung 6, vorkommen. In dem ersten aufgezeichneten Paket können wir erkennen, dass

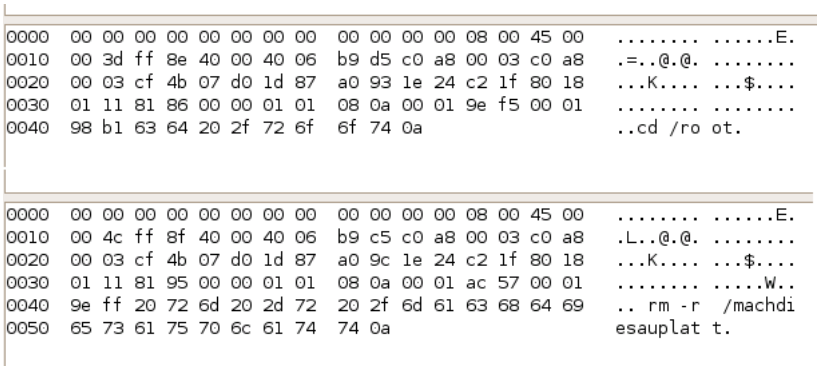

Abbildung 6: Mitschnitt des Datenverkehrs mit Wireshark

   der Befehl cd /root enthalten ist. In dem zweiten Befehl können wir erkennen, dass der Befehl rm -r /machdiesauplatt den Ordner machdiesauplatt und alle Unterobjekte löscht. Das IDS würde dann Alarm schlagen und den Administrator benachrichtigen. Ein bekannter Vertreter dieser Gattung ist Snort.

[11] Es gibt Distributionen, in denen die -e Funktion nicht vorhanden ist. Ein erneutes Kompilieren des Quellcodes schafft Abhilfe

b) **Anomalie Detektoren**: Diese IDS haben eine Definition des Normalzustandes eine Systems und einen Rechnernetzes als Grundlage. Mögliche Parameter können Netzlast, CPU Nutzung, Quelle, Ziel oder Leistungspitzen sein. Tritt eine Abweichung von dem Normalzustand auf, schlagen diese Systeme Alarm.

IDS haben ihre Grenzen. Sie können nicht garantieren, dass alle Aufälligkeiten in einem Netz erkannt und zudem richtig bewertet werden. Betrachtet man ein Netz, das ein sehr hohes Datenaufkommen hat, so ist das IDS gezwungen, gewisse Pakete bei der Untersuchung zu verwerfen. Angreifer könnten dies ausnutzen, um ihre Pakete in diesem riesigen Fluss an Information zu verstecken.Dies geschieht zum Beispiel bei Rootkits[10]. Im Allgmeinen betrachtet, liefern sie jedoch immer genug Informationen, um einen Überblick über das gesamte Netz zu erhalten. Für die erfolgreiche Abwehr eines Angriffs ist jedoch letztendlich der Administrator verantwortlich.

Literatur

[1] Amavis. http://www.amavis.org/.

[2] BSI:Bundesamt für Sicherheit in der Informationstechnik. http://www.bsi.de/.

[3] ietf. http://www.ietf.org/html.charters/tls-charter.html.

[4] IPFW. http://www.freebsd.org/doc/.

[5] Iptables. http://www.iptables.org/.

[6] Pf. http://www.openbsd.org/faq/pf/index.html.

[7] Prof Dr. Schindelhauer Kapitel 11, 2007.

[8] William Stallings. *Data and Computer Communications (8th Edition).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[9] Cyrus Peikari und Anton Chuvakin. *Kenne deinen Feind.* O Reilly, 2004.

[10] Greg Hoglund und James Butler. *Rootkits Windows Kernel unterwandern.* Addison-Wesley.

Abbildungsverzeichnis