Technische Universität Berlin

DAI-Labor
TU Berlin

Nachrichtenaustausch unter Angriff

Diplomarbeit

am Fachgebiet Agententechnologien in betrieblichen Anwendungen und der Telekommunikation
Prof. Dr.-Ing. habil. Sahin Albayrak
Institut für Wirtschaftsinformatik und Quantitative Methoden
Fakultät IV - Elektrotechnik und Informatik
Technische Universität Berlin

vorgelegt von
Thomas Bosch
(Matrikelnr.: 197609)

Betreuer: Dipl.-Inform. Olaf Kroll-Peters
Dipl.-Inform. Rainer Bye

17. Dezember 2006

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

In den vergangenen Jahrzehnten hat das Thema Netzwerksicherheit (Network Security) immer mehr an Bedeutung gewonnen. Mit wachsenden IT-Strukturen wachsen auch die Möglichkeiten diese Strukturen anzugreifen. In den kommenden Jahren wird die Sicherheit in Netzwerken immer mehr zu einem zentralen Forschungsthema werden.

1.1 Motivation

Ausgehend von den stetig wachsenden IT-Strukturen gibt es auch immer mehr Menschen, die diese nutzen. Diese Entwicklung wird folgend mit statistischen Erhebungen belegt. Es wird gezeigt, wie sich dieses Wachstum auswirkt und welche Möglichkeiten dadurch für potentielle Angreifer geschaffen werden.

1.1.1 Was ist

Die Zahl der Internetnutzer in Deutschland ist, wie in Abbildung 1.1 zu sehen ist, in den Jahren von 1997 bis 2004 von $4,1$ auf $35,7$ Millionen gestiegen. Allein in diesen sieben Jahren hat sich die Nutzerzahl fast verneunfacht. Somit stiegen auch die Möglichkeiten, Angriffe auf Computernetzwerke auszuführen. Mit steigender Anzahl an Nutzern werden auch mehr Ressourcen benötigt, um diese Flut an Daten bewältigen zu können. Das sog. *Backbone* des Internets, d.h. die Verbindung von Routern und Gateways, über die der gesamte Verkehr fließt, musste demnach ebenfalls ausgebaut werden. Mehr Rechner bedeuten dann wieder mehr Angriffspunkte für potentielle Angreifer.

Bis 2008 werden es laut Infratest[34] sogar $75,6$ Millionen Menschen in Deutschland sein, die das Internet nutzen. Hierbei ist zu beachten, dass verschiedene Institute auch verschiedene Herangehensweisen zur Erhebung der Schätzungen haben und je nach Definition des Begriffs „Nutzung“ auch verschiedene Werte entstehen. So definiert das Institut

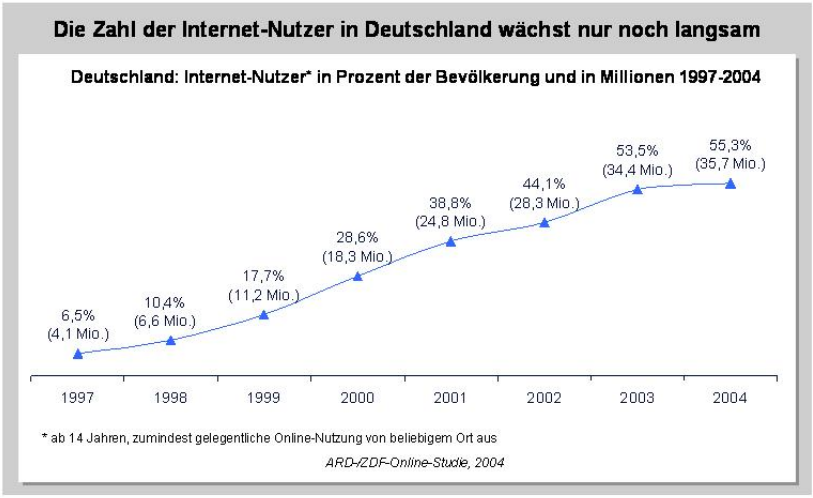

Abbildung 1.1: Anzahl der Internetnutzer in Deutschland - Quelle: Infratest[32]

EITO (European Information Technology Observatory) den Begriff Nutzer als „Personen, die das Internet mindestens einmal im Monat nutzen, unabhängig vom Alter und Nutzungsort“[34]. In Abbildung 1.2 ist zu sehen, dass die Zahlen unterschiedlich sind, abhängig vom Institut.

Wie in Abbildung 1.3 zu sehen ist, sehen die weltweiten Schätzungen im Vergleich zu den Schätzungen, die die deutsche Bevölkerung wiedergeben, nicht anders aus. Dort ist ein Wachstum des Internets von 1999 bis 2008 von 286 Millionen auf 1,28 Milliarden Nutzer zu sehen, was innerhalb von 4 Jahren eine Verdopplung bedeutet.

Neben privaten Nutzern sind es aber auch vor allem gewerbliche und staatliche Einrichtungen, die vermehrt die Ressourcen des Internets nutzen. So gibt es beispielsweise Internetplatformen der *Agentur für Arbeit*[1] oder gewerbliche Nutzung, wie die Handelsplatform *ebay*[2]. Viele Arbeitsplätze und ein stetig steigender Anteil am Bruttoinlandsprodukt sind direkt oder indirekt mit dem Internet verbunden. Die gesamtwirtschaftliche Bedeutung der IKT (Informations- und Kommunikationstechnologien) hat sich von 4,7% im Jahr 1995 auf 6,8% im Jahr 2004 erhöht[7]. Der genannte Prozentsatz bezieht sich auf den Anteil der IKT-Waren und IKT-Dienstleistungen am deutschen Bruttoinlandsprodukt. Dies ist nur die gewerbliche Form der Nutzung von IT-Strukturen. Auch öffentliche Einrichtungen werden immer abhängiger von den Ressourcen des Internets.

[1] http://www.agenturfuerarbeit.de
[2] http://www.ebay.de

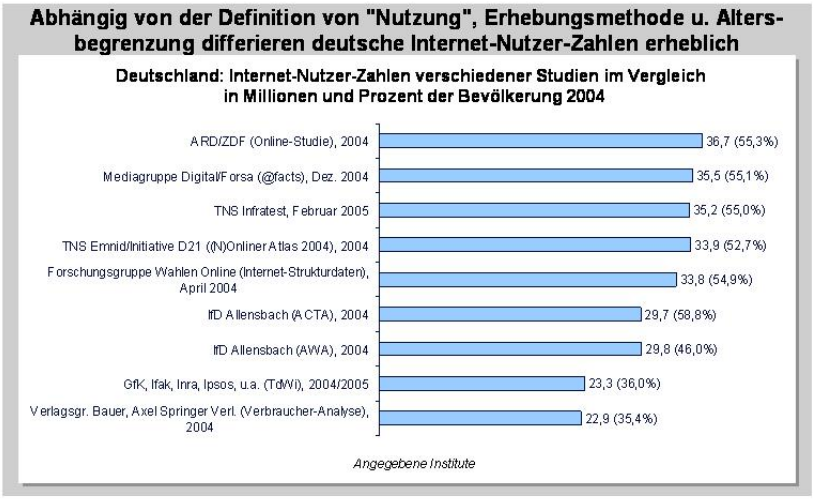

Abbildung 1.2: Verschiedene Institute im Vergleich - Quelle: Infratest[33]

Weiterhin gibt es Strukturen, auch *Kritische Infrastrukturen* genannt, die für die deutsche Gesellschaft von sehr großer Bedeutung sind.

> „Organisationen und Einrichtungen mit wichtiger Bedeutung für das staatliche Gemeinwesen, bei deren Ausfall oder Beeinträchtigung nachhaltig wirkende Versorgungsengpässe, erhebliche Störungen der öffentlichen Sicherheit oder andere dramatische Folgen eintreten würden."
>
> (BSI, 2004)

Hierbei handelt es sich um ein Zitat[21] des BSI (Bundesministerium für Sicherheit in der Informationstechnik). Kritische Infrastrukturen sind in Deutschland beispielsweise Transport und Verkehr, Energie, Finanzwesen, Versorgung oder Justiz. Alle diese Sektoren haben in Deutschland eine wichtige Bedeutung und deren Beeinträchtigung wäre für die Gesellschaft schwer zu verkraften. Durch verschiedenartige Angriffe könnten solche Beeinträchtigungen entstehen.

1.1.2 Was sein könnte

Fehlerhafte Software, Schadprogramme (z.B. Trojaner), Angriffe von innenhalb und außenhalb des eigenen Netzwerks, aber auch physische Einwirkungen, wie Blitzeinschläge oder terroristische Anschläge, können die Funktionalität von Netzwerken behindern. Kritische Infrastrukturen kommunizieren auch über Netzwerke, durch Bildung eines Intranets. Ein Intranet ist ein autonomes Netzwerk, welches unter der Administration einer

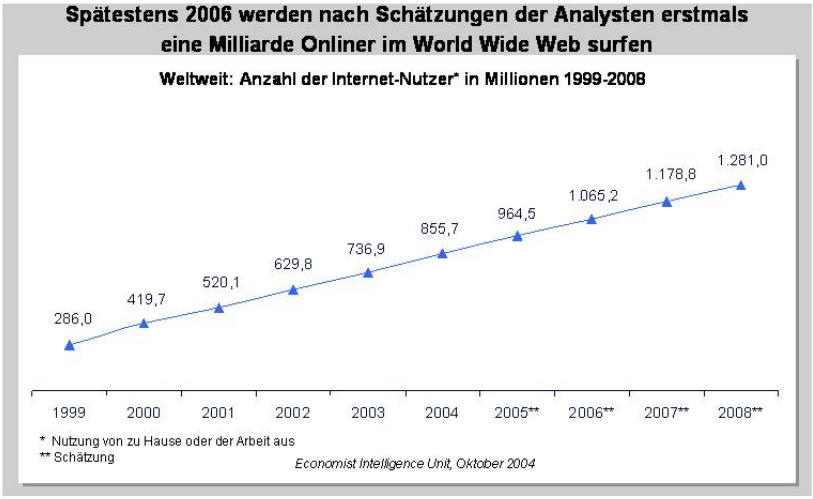

Abbildung 1.3: Internationales Internetwachstum - Quelle: Infratest[31]

einzelnen Instanz steht. So können beispielsweise verschiedene Kontrollsysteme, wie z.B. Sensordaten, innerhalb des Intranets von beliebiger Stelle abgefragt werden. Diese internen Netzwerke sind auch, genau wie das Internet, durch unterschiedliche Faktoren bedroht. So können auch hier fehlerhafte Software oder Trojaner den Nachrichtenaustausch beeinträchtigen. Die Beeinträchtigung eines der Sektoren von Kritischen Infrastrukturen, z.B. des Transportsektors, hätte alleine für den Wirtschaftsstandort Deutschland verheerende Folgen.

Ein gezielter Angriff auf das Verkehrsleitsystem der Deutschen Bahn könnte etwa den Ausfall von lokalen Betriebsstrecken bedeuten. Möglicherweise könnte es sogar zu einem bundesweiten Ausfall kommen. Dieses Beispiel zeigt, wie relevant die Sicherung solcher Strukturen ist. Es muss jederzeit gewährleistet sein, dass der Nachrichtenaustausch innerhalb der Netzwerke funktioniert. Daher sind Sicherstellungen der Verfügbarkeit, Integrität und Vertraulichkeit der Kritischen Infrastrukturen Hauptschutzziel des BSI.

Mit steigender Nutzung von Netzwerktechnologien in der Gesellschaft steigt auch die Gefahr, dass diese Technologien angegriffen werden. Der Nachrichtenaustausch in Netzwerken muss demnach auch für den Fall garantiert sein, dass dieses Netzwerk angegriffen wird. Um das Verhalten eines Netzwerks oder einzelner Netzwerkkomponenten zu untersuchen, werden Netzwerksimulatoren entwickelt. Diese Simulatoren können der Überprüfung und Bewertung neu entwickelter Verteidigungsstrategien dienen.

1.2 Ziel der Arbeit

Diese Arbeit behandelt unterschiedliche Bedrohungen der Verfügbarkeit, Integrität und Vertraulichkeit von Netzwerken. Zu Beginn wird dies in einer theoretischen Betrachtung vertieft. Dieser theoretische Hintergrund dient dann dazu, nachzuvollziehen, wie sich solche in der Theorie bekannten Bedrohungen in der Praxis auswirken. Solche Bedrohungen beeinträchtigen den Nachrichtenaustausch innerhalb von Netzwerken.

Ziel ist es unter anderem, die Bedeutsamkeit und Problematik des Nachrichtenaustauschs innerhalb von Netzwerken darzustellen. Es soll erörtert werden, wie der Nachrichtenaustausch durch einen Angriff beeinflusst wird. Weiterhin soll in einer Simulation die Bedrohung und der Angriff eines Netzwerks implementiert und auf diesen reagiert werden. Dazu wird eine Gegenmaßnahme integriert, die dann den Austausch von Nachrichten während eines stattfindenen Angriffs weiterhin ermöglicht.

1.3 Aufbau der Arbeit

Um die genannten Ziele zu erreichen, wird in Kapitel 2 zunächst der aktuelle Stand der Wissenschaft geschildert.

In Kapitel 3 werden dann ein paar Grundlagen zum Thema Netzwerke erläutert, anhand derer dann Bedrohungen und Schäden durch Angriffe auf IT-Strukturen betrachtet werden. Weiterhin wird dort untersucht, inwieweit solche Angriffe zu verhindern sind. Welche Gegenmaßnahmen existieren, um einen Angriff zumindest einzudämmen?

Zudem gibt es in Kapitel 3 noch einen historischen Rückblick, bei dem auf schon stattgefundene Angriffe eingegangen wird. Diese werden analysiert und es wird aufgezeigt, welche Maßnahmen bisher gegen die Verteidigung solcher Angriffe ergriffen wurden. Was lassen sich außerdem aus diesen Angriffen für Erkenntnisse mitnehmen? Wurde aus den Erfahrungen gelernt? Für den Fall, dass alle Maßnahmen der Abwehr scheitern, bleibt zuletzt noch die Möglichkeit, andere Wege für den Nachrichtenaustausch zu finden. Wie können solche Notfallpläne aussehen? Welche anderen Ressourcen lassen sich hierzu nutzen? Es werden einige alternative Wege des Nachrichtenaustauschs genannt, wobei positive und negative Eigenschaften dieser Alternativen aufgezeigt werden.

Kapitel 4 beschreibt die Herangehensweise und Implementierung eines ausgewählten Angriffs in einem, im DAI-Labor entwickelten Netzwerksimulator namens **NeSSi** (Network Security Simulator). Weiterhin wird die Simulation eines Angriffs bewertet. Was passiert nun wirklich während des Angriffs? Was bewirken die Gegenmaßnahmen, die nach dem Angriff eingeleitet werden, oder schon präventiv eingeleitet wurden? Dieser praktische Teil soll den theoretischen Teil unterstützen und die darin enthaltenen Ansätze zur Lösung des Problems „Nachrichtenaustausch unter Angriff“ untersuchen.

Zuletzt wird in Kapitel 5 ein Fazit gezogen. Was lässt sich den Ergebnissen der Simulation entnehmen? Hier werden die Ergebnisse abschließend bewertet und in einen Ausblick integriert.

Bevor theoretische und praktische Aspekte in den Vordergrund treten, wird im folgenden Kapitel zunächst auf den aktuellen Stand der Wissenschaft eingegangen und deren Erkenntnisse zu diesem Thema dargelegt.

Kapitel 2

State of the Art

Im Bereich der Verhinderung bzw. Abschwächung von Angriffen auf große Netzwerke wurde schon viel geforscht. Die Forschung ist bisher jedoch noch auf der Suche nach einer optimalen Methode. Bisher wurde noch keine Methode gefunden, die beispielsweise DDoS wirksam bekämpft.

In den letzten Jahren wurde das Thema *Angriffe auf Netzwerke und deren Bekämpfung* immer intensiver in der Wissenschaft erforscht. Bereits im Jahr 1990 wurde von Heady et al.[22] die Erkennung von Anomalien in Netzwerken behandelt. So wurden in den folgenden Jahren viele Systeme entwickelt, die sich mit dieser Anomalieerkennung beschäftigen, wie beispielsweise die LOF-Methode (*Local Outlier Factor*) von Breuning et al.[5]. Mit der Entwicklung weiterer Methoden stieg die Notwendigkeit des Vergleiches, so dass von Lazarevic et al.[38] verschiedene Methoden zur Anaomalieerkennung getestet und miteinander verglichen wurden. Die erwähnte LOF-Methode wurde dabei als die zuverlässigste bewertet.

Neben der Anomalieerkennung sind weitere Methoden zum Schutz von Netzwerken entworfen worden. Verschiedenste Filter wurden entwickelt um *Distributed Denial of Service* (DDoS: siehe Abschnitt 3.2.1) einzudämmen. So haben Keromytis et al.[36] eine Methodik zur Verminderung von DDoS entwickelt, welche nur authentifizierten Datenpaketen erlaubt, ein bestimmtes Ziel zu erreichen. So ein Overlay-basiertes Filtern wurde auch von Weiteren[54, 4] untersucht. Weiterhin wurden Filtermethoden vorgeschlagen, die das *Spoofen* (Fälschen) von IP-Adressen verhindern sollen[18], oder durch so genanntes *traceback*, d.h. durch eine schrittweise Rückverfolgung, die Quellen solcher Attacken herauszufiltern[47, 48]. Eine dritte Methode, das *pushback*[42, 35], ermöglicht das Platzieren von Netzwerkfiltern in Netzwerken, die der Quelle der Attacke näher sind.

Die Forschung steht erst am Anfang. Zwar wurden schon einige Methoden entwickelt, aber keine dieser Methoden hat sich bisher als guter Standard durchsetzen können. Es ist in zukünftiger Forschungsarbeiten von großer Bedeutung weitere Gegenmaßnahmen zur Abwehr von Angriffen zu entwickeln.

Die Entwicklung von Gegenmaßnahmen gegen Angriffe auf Netzwerke, bzw. einzelne Teilnehmer von Netzwerken, benötigt zunächst theoretische Kenntnisse. Dazu gehören Wissen über Aufbau von Netzwerken. Zudem auch die Kenntnisse über die Methoden von Angriffen. Diese Basis wird im folgenden Kapitel erörtert.

Kapitel 3

Theoretischer Hintergrund

In diesem Kapitel wird der Hintergrund eines *Nachrichtenaustauschs unter Angriff* theoretisch erläutert. Dazu werden zunächst Grundlagen geschaffen, auf denen dann in den weiteren Kapitel aufgebaut wird. So werden Bedrohungen von Netzwerken genannt und in einem historischen Rückblick exemplarisch dargelegt. Mit diesen Kenntnissen und Grundlagen lassen sich dann mögliche Verteidigungstrategien entwickeln.

3.1 Grundlagen

Zuerst einmal wird der Begriff„Netzwerk“ etwas näher gebracht. Ein Netzwerk beschreibt im allgemeinen eine Verknüpfung mehrerer Computer untereinander, so dass diese miteinander kommunizieren können. Das Internet ist das größte Netzwerk das wir kennen. Es beschreibt die Verknüpfung tausender Netzwerke weltweit.

Um nun miteinander kommunizieren zu können und die Komplexität einer solchen Kommunikation von Millionen von Computern im Rahmen zu halten, bedarf es einer Verteilung verschiedener Aufgaben auf einzelne Teilbereiche, **Layer** genannt. Ich benutzte im weiteren Verlauf die deutsche Übersetzung, Ebene.

Jede Ebene ist nun eine Ansammlung von Protokollen, die nun bestimmte Funktionalitäten bieten. So übernimmt eine Ebene beispielsweise die physikalische Übertragung, eine andere gewährleistet das Finden des Empfängers und eine weitere sorgt für die verlustfreie Übertragung.

3.1.1 Das ISO/OSI-Modell

Ein Modell, welches nun die Aufgaben einer jeden Ebene definiert, ist das **ISO/OSI-Referenzmodell**. Es wurde 1983 als ISO-Standard (International Standards Organization) eingetragen[51]. Das ISO/OSI-Modell definiert sieben Ebenen, wie sie in Ab-

bildung 3.1 zu sehen sind. Die Ebenen wurden so geschaffen, damit folgende Punkte gewährleistet sind[51]:

- Jede Ebene sollte ein eigene Abstraktionsebene bilden.
- Jede Ebene sollte eine wohl definierte Aufgabe haben.
- Die Aufgabe jeder Ebene sollte sich nach den internationalen Protokollstandards richten.
- Der Datenfluss zwischen den Ebenen sollte möglichst gering gehalten werden.
- Die Anzahl der Ebenen sollte so gewählt sein, dass Aufgaben sich nicht überschneiden, aber die Übersichtlichkeit gewahrt wird.

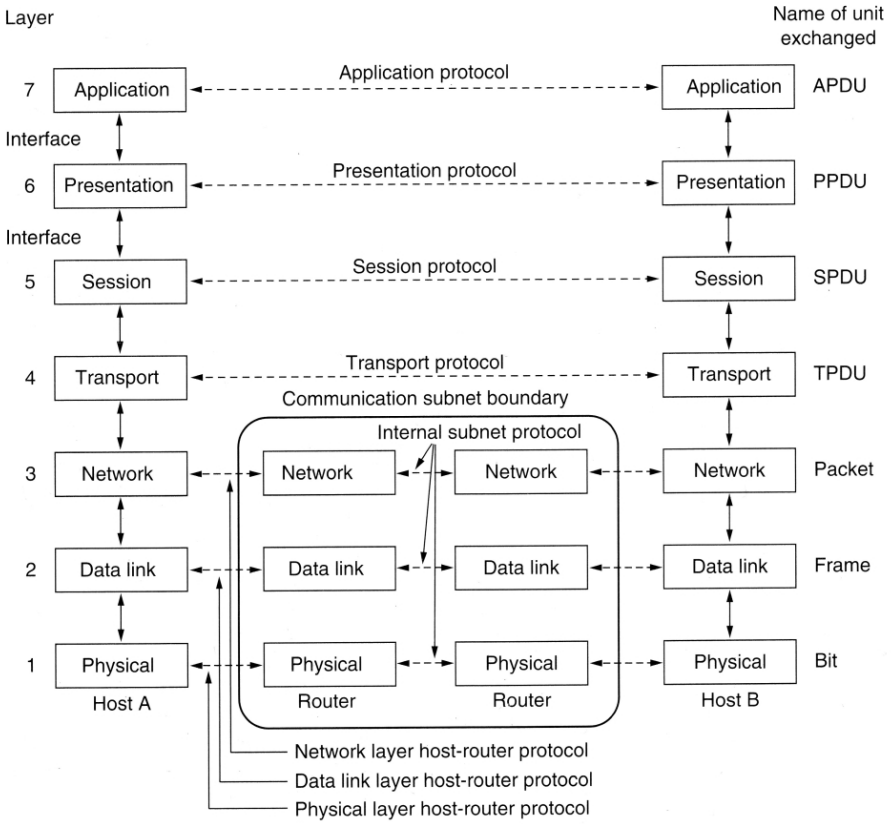

Abbildung 3.1: Das ISO/OSI-Ebenenmodell - Quelle: Tanenbaum[51]

3.1.2 TCP/IP-Modell

Neben dem ISO/OSI-Modell wurde 1974 auch das **TCP/IP-Referenzmodell** definiert[8]. In diesem Modell gibt es nur vier Ebenen. Verglichen mit dem ISO/OSI-Modell wur-

den beim TCP/IP-Modell die ersten beiden Ebenen, deren Aufgaben eher physikalischer Natur sind, zusammengelegt. Wie in Abbildung 3.2 erkennbar ist, wurden weiterhin die Ebenen fünf und sechs komplett weg gelassen.

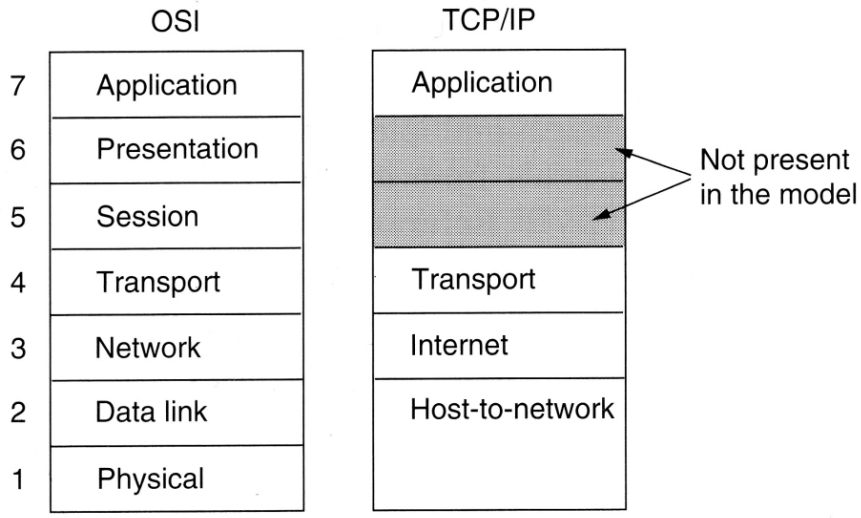

Abbildung 3.2: Das TCP/IP-Ebenenmodell im Vergleich - Quelle: Tanenbaum[51]

Der weitere Verlauf der Arbeit wird sich nur mit den Ebenen *Applikation*, *Transport*, *Internet* bzw. *Netzwerk* und *Host-to-Network* beschäftigen. Im Folgenden werden nun diese vier Ebenen näher erläutert und ein paar Beispiele verdeutlichen die Funktionsweise der einzelnen Ebenen.

3.1.3 Host-to-Network Ebene

Die Host-to-Network Ebene bildet die Basis der Datenübertragung. Sie handhabt die physikalische Übertragungsweise. Eine Art wäre die Übertragung über Lichtwellenleiter, eine andere die elektrische Übertragung über die üblichen Kupferkabel. Neben dieser untersten Basisfunktionalität bietet sie aber auch noch die Funktionalität der im ISO/OSI-Modells Data-Link-Layer genannten Ebene. Diese bietet Funktionen, wie die Behebung von Fehlern. Fehler können etwa durch physikalische Einflüsse, wie Leitungsrauschen entstehen.

3.1.4 Netzwerkebene

Die Netzwerkebene ist für den korrekten Versand von einzelnen Paketen durch ein Netzwerk verantwortlich. Das zugrunde liegende Protokoll ist das **IP-Protokoll**. Mit Hilfe dieses Protokolls und verschiedener Routingalgorithmen wird auf der Netzwerkebene sichergestellt, dass jedes erhaltene Paket korrekt weitergeleitet wird.

3.1.4.1 IP-Protokoll

Das IP-Protokoll wird auch umgangssprachlich das „Internet-Protokoll“ genannt. Es ist das Protokoll, auf welches alle anderen Protokolle, wie TCP, UDP oder ICMP aufsetzen. Es beinhaltet keine Verbindungsmethoden, jedoch eine Paket-Struktur.

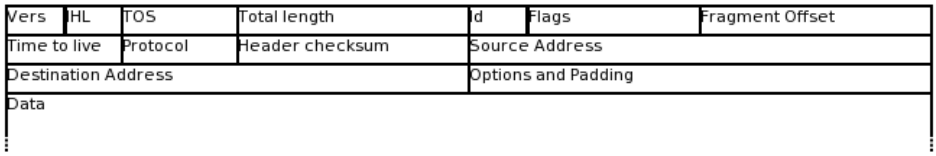

Abbildung 3.3: Aufbau eines IP-Pakets

Jeder Computer im Internet hat eine Adresse mit der er eindeutig verknüpft ist, die *IP-Adresse*. Diese IP-Adresse ist ein 4 Byte langes Feld, das zur besseren Lesbarkeit in vier, durch Punkte getrennte Dezimalzahlen von 0 bis 255 notiert wird. Also beispielsweise 243.155.5.56. Anhand dieser Adressierung ist es möglich, dass ein Nachrichtenaustausch, von einer Seite der Erde auf die andere funktioniert. Dieser Nachrichtenaustausch erfolgt über Nachrichtenpakete, die verschiedenste Informationen, wie Absender, Empfänger und natürlich auch Daten enthalten.

Wie in Abbildung 3.3 zu sehen ist, beinhaltet ein solches IP-Paket noch viel mehr Informationen auf die hier aber nicht eingegangen wird. Im Zusammenhang dieser Arbeit sind zunächst nur die Adressen, also *Destination Address* und *Source Address* von Bedeutung.

3.1.4.2 Routing-Protokolle

Routing-Protokolle ermöglichen den dynamischen Aufbau von Routing-Tabellen in jedem einzelnen Router (siehe Abschnitt 3.1.7.1). Die von den Protokollen genutzten Algorithmen lassen sich in zwei Klassen aufteilen.

Link-State-Protokoll Bei diesem Algorithmus wird allen anderen Routern mitgeteilt, welche Nachbarn jeder einzelne Router hat. Somit kann jeder Router für sich aus den Informationen der anderen eine Topologie aufbauen und seine Routingtabelle danach erstellen.

Distance-Vector-Protokoll Hier teilt jeder Router seinen Nachbarn mit, welche Router ihm bekannt sind und mit wie vielen Schritten diese erreicht werden können. Unter diese Art von Protokollen fallen auch die mit einem verbesserten Algorithmus versehenen Pfadvektor-Protokolle, wie das *BGP* (Border Gateway Protocol).

In diese Klassen fallen nun verschiedene Protokolle, die die Router nutzen, um ihre Pakete möglichst schnell an sein Ziel weiterzuleiten.

3.1.4.3 ICMP

Dieses Protokoll wird für Kontrollzwecke genutzt. ICMP (*Internet Control Management Protocol*) bietet Funktionalitäten, die für das Management von Verbindungen im Netzwerk nötig sind. So sendet ein Router dem Sender eines Paketes ein ICMP-Paket vom Typ „Destination Unreachable“ und dem Code „Host Unreachable“[29], wenn der Empfänger, welcher im Header des IP-Paketes enthalten ist, nicht in seinem Netzwerk existiert.

Ein weiteres Beispiel für ICMP ist das Paket vom Typ „Redirect“[29]. Dieses Paket wird von Routern benutzt, um Paketsendern aus dem eigenen Netzwerk mitzuteilen, dass dieser seine Pakete über einen anderen bestimmten Router schicken soll, da dies direkter zum Ziel führt.

3.1.5 Transportebene

Die Transportebene setzt direkt über der Netzwerkebene an. Die hier benutzten Protokolle bieten verschiedene Funktionalitäten an. Jedes hat spezielle Vorteile und Nachteile. Je nach Anwendungsbereich werden verschiedene Funktionen benötigt, die durch die folgenden Protokolle abgedeckt werden.

3.1.5.1 TCP

TCP ist ein verbindungsorientiertes Transportprotokoll. Verbindungsorientiert bedeutet, dass bei einer Kommunikation über dieses Protokoll eine Verbindung zwischen den beiden Endpunkten aufgebaut wird. TCP benutzt für den Aufbau einer solchen Verbindung ein Dreiwege-Handshake-Protokoll. Der Verbindungsaufbau wird demnach in drei Schritten vollzogen. In einem ersten Schritt meldet der Sender sich mit einem SYN-Paket

(SYN=Syncronize) an. Dieses Paket wird vom Empfänger mit einem SYN/ACK-Pakete (ACK=Acknowledge) bestätigt. Diese Paket bestätigt der Sender wiederum mit einem eigenen ACK-Paket. Nach dem Versand dieses Pakets gilt der Verbindungaufbau als abgeschlossen.

Neben dem Verbindungsstatus wird von TCP auch Zuverlässigkeit gewährleistet. Dies bedeutet, dass Pakete, welche über TCP gesendet werden auf jeden Fall beim Empfänger ankommen. Jedes erhaltene Paket muss vom Sender bestätigt werden. Zu diesem Zweck enthält das TCP-Paket eine Sequenznummer, welche das Paket eindeutig beschreibt.

Eine weiter Funktionalität von TCP ist Flusskontrolle. Flusskontolle bietet den Vorteil, dass eventuell ausgelastete Datenleitungen weniger stark belastet werden. Durch Verringerung der Datenrate wird eine gewisse Fairness gegenüber weiteren Datenverbindungen garantiert. Ebenso wird umgekehrt bei weniger Datenverkehr die Leitung voll ausgelastet.

3.1.5.2 UDP

Im Gegensatz zu TCP ist UDP verbindungslos. Das bedeutet, dass zum Start einer Kommunikation über UDP z.B. kein Handshake-Protokoll benötigt wird. UDP bietet keine Zuverlässigkeit, das heißt, Pakete, die auf dem Weg zum Empfänger verloren gehen, werden nicht wiederholt gesendet. Dadurch erreicht man zwar einen, im Gegensatz zu TCP verzögerungsärmeren Datenaustausch, muss dabei aber in Kauf nehmen, dass bei einer verlustreichen Verbindung Pakete nicht beim Empfänger ankommen.

UDP wird dort verwendet, wo ein solcher Datenverlust nicht schwerwiegend ist. Andererseits auch dort, wo die Applikationsebene entweder ebenfalls durch Wiederholung des Sendevorgangs selbst den Datenverlust bewältigt oder durch interne Algorithmen, je nach Anwendung anderweitig handhabt.

Bei VoIP (*Voice over IP*), also Telefonie über das Internet, sind Datenverluste nicht wesentlich. So wird auf den Verlust von Datenpaketen durch verschiedene Mechanismen reagiert. Hier wird z.B. durch Wiederholung des letzten eingegangenen Pakets, oder durch Interpolation zwischen dem letzten, vor dem Verlust eingegangenen Paket, und dem nächsten, nach dem Verlust eingegangen Paket, unterschiedlich gut mit Paketverlusten umgegangen[37].

Protokoll	**Vorteile**	**Nachteile**
TCP	zuverlässig, fair	viel Verzögerung möglich
UDP	kaum Verzögerung	unzuverlässig, unfair

Tabelle 3.1: TCP und UDP im Vergleich

Zusammenfassend lässt sich, gemäß Tabelle 3.1 sagen, dass die Wahl eines geeigneten Transportprotokolls immer von den gewünschten Eigenschaften abhängt.

3.1.6 Applikationsebene

Auf der Transportebene baut die Applikationsebene auf. Dieser Ebene ist die in der Abstraktion höchste Ebene. In ihr befinden sich nun die Protokolle, die vom Anwender direkt genutzt werden. Folgend werden einige Beispiele genannt.

3.1.6.1 HTTP

Das Protokoll, welches von vielen Anwendern üblicherweise genutzt wird ist das HTTP-Protokoll. HTTP (*Hypertext Transfer Protocol*, definiert in RFC 2616[26]) wird mehrheitlich zur Darstellung von Internetseiten benutzt. Web-Server, also Rechner im Internet, die Internetseiten speichern und anbieten, geben ihre Daten über das HTTP-Protokoll an die anfordernden *Browser* weiter. Der Browser ist ein Programm des Anwenders, welches die ankommenden Daten interpretiert und darstellt. HTTP kann weiterhin auch zum Dateitransfer genutzt werden, es ist aber nicht für eine solche Anwendung entwickelt worden. Diese Dateitransfers werden zumeist über ein speziell für diese Tätigkeit entwickeltes Protokoll namens FTP durchgeführt.

3.1.6.2 FTP

Das *File Transfer Protocol* (RFC 959[30]) ist speziell für den Transport von Dateien entwickelt worden. Mit Hilfe des FTP-Protokolls kann der Anwender sich auf einem FTP-Server einloggen und von dort Dateien herunter- bzw. heraufladen. Neben dem Transportdatenstrom gibt es noch einen Kontrolldatenstrom, der dazu da ist, Befehle an den FTP-Server zusenden und auch Kontrolldaten vom Server zu erhalten. Zwar lassen sich auch über HTTP-Server Dateien zum Herunterladen zur Verfügung stellen, die Handhabung und Effizienz des Navigierens innerhalb der Verzeichnisstruktur der Daten auf dem HTTP-Server ist jedoch nicht so gut wie über FTP. So lässt sich beispielsweise mittels eines Befehls „chdir pfad/zum/verzeichnis“ das aktuelle Verzeichnis wechseln. Für einen solchen Wechsel über HTTP wären meist drei Mausklicks notwendig.

3.1.6.3 SSH

Über FTP und HTTP werden die Daten ohne Sicherheitsmechanismen übertragen. Eine Verschlüsselung ist in diesen Protokollen nicht vorgesehen. Für solche sicheren, passwortgeschützten Verbindungen wurde SSH (*Secure Shell* RFC 4251[28]) entwickelt. Für die

Sicherheit des SSH-Protokolls sorgen viele kryptographische Algorithmen, die Authentisierung und Verschlüsselung gewährleisten. SSH ist entwickelt worden, um ein gesichertes Befehlsterminal auf einem entfernten Rechner zu öffnen. Aber insbesondere die Weiterentwicklung SSH2 erlaubt neben der gesicherten Terminalverbindung auch die Adaption an andere Programme. So setzt das Programm SCP auf SSH auf und ermöglicht einen Dateitransfer über das SSH-Protokoll, also einen verschlüsselten Transfer. Auch SFTP bietet eine gesicherte Alternative zu FTP, welche SSH benutzt.

3.1.6.4 DNS

Die einzelnen Rechner des Internets sind, wie in dem vorigem Abschnitt zu sehen ist, über IP-Adressen in nummerischer Form zuzuordnen. Da diese nummerische Adressgebung für die Router sehr einfach, für den Menschen aber schwer zu handhaben ist, wurde das DNS-Protokoll entwickelt[24, 25]. Es ermöglicht eine eindeutige Zuordnung von alphanumerischen Adressen, also beispielsweise `www.iv.tu-berlin.de` zu IP-Adressen (hier: `130.149.16.12`). Jeder IP-Adresse kann also mindestens einer so genannte *Domain-Adresse* zugeordnet werden. DNS heißt *Domain Name System*.

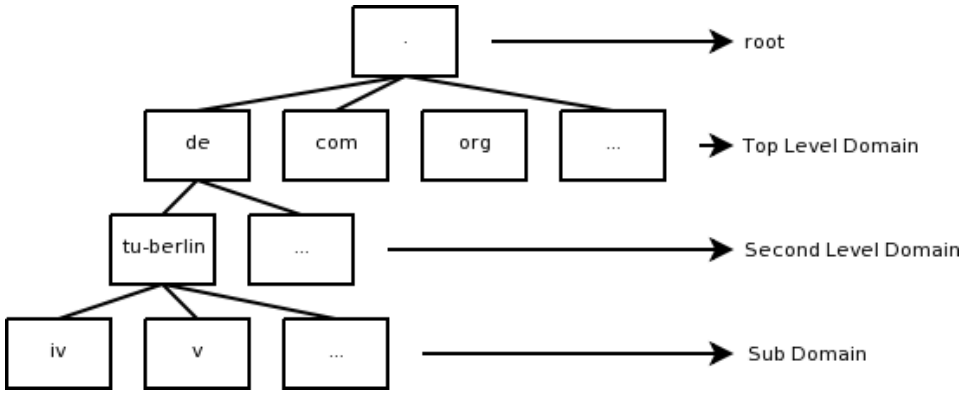

Abbildung 3.4: Baumartige Struktur der DNS-Hierarchie

Diese Namensauflösung, das heißt die Zuordnung eines Domain-Namens zu einer IP-Adresse, wird im Internet von den DNS-Servern durchgeführt. Die erste Anfrage eines Benutzers zu der IP-Adresse einer Domain geht zuerst einmal eine Anfrage an den DNS-Server des Providers. Sollte dieser keinen Eintrag zur gesuchten Domain haben, sendet er eine Anfrage an einen der 13 DNS-Root-Server.

Wie in Abbildung 3.4 zu sehen ist, ist dieser DNS-Root-Server der oberste Knoten eines Baumes - hier mit einem Punkt bezeichnet. Von dort aus wird nun rekursiv eine Anfrage gestartet. Der Domain-Name wird von rechts nach links aufgelöst, d.h. zuerst wird bei meinem Beispiel die `de`-Endung verfolgt, danach der Teil `tu-berlin` usw..

Bei jedem Schritt wird ein neuer DNS-Server nach der IP-Adresse gefragt. Sollte bei diesem Server kein Eintrag zu der Anfrage vorhanden sein, wird diese Anfrage an den nächsten zuständigen DNS-Server weitergeleitet. Demnach leitet der DNS-Root-Server die Anfrage an den DNS-Servers der Domain `tu-berlin.de` weiter. Dieser hat möglicherweise die IP-Adresse des gesuchten Web-Servers nicht eingetragen, sendet also nun die Anfrage weiter an den DNS-Server der Subdomain `iv.tu-berlin.de`. Spätestens hier sollte ein Eintrag für den Web-Server (`www.iv.tu-berlin.de`) dieser Domain vorhanden sein. Diese ermittelte IP-Adresse wird nun an den DNS-Server des Providers zurück gesendet, der diese wiederum dem anfragenden Benutzer mitteilt.

Die vermittelte Adresse kann der DNS-Server des Providers nun speichern. Dieses so genannte *Caching* erlaubt eine schnellere Beantwortung der DNS-Anfragen. Sollte nun wieder eine Anfrage an den DNS-Server gehen, welche die Domain `www.iv.tu-berlin.de` aufgelöst haben möchte, so kann dies schnell über die nun gespeicherte IP-Adresse geschehen. Diese Cache-Einträge werden jedoch nach einer bestimmten Zeit, der TTL (Time-To-Live), wieder gelöscht, da es oftmals vorkommt, dass sich IP-Adressen ändern und so dann evtl. eine falsche Antwort kommen würde.

Die zweite Methode ist die reversible Namensauflösung (*Reverse Lookup*). Hierbei wird der umgekehrte Weg beschritten, nämlich die Zuordnung des Domainnamens zu einer bestimmten IP-Adresse. Hierzu wird auf jedem DNS-Server eine *Reverse Lookup Tabelle* eingerichtet. Diese Tabelle enthält Einträge, nach denen jeder IP-Adresse im Subnetz des verwaltenden DNS-Servers ein Domainname zugeordnet werden kann.

Jede Ebene der Referenzmodelle birgt Schwachstellen. Jedes System eines Netzwerks nutzt die Funktionen dieser Ebenen. Somit besteht die Möglichkeit, dass jedes System auch Teil eines Angriffs wird.

3.1.7 Angriffsziele

Ein Angriff hat auf unterschiedlichen Systemen auch unterschiedliche Auswirkungen, sowohl für das System selbst, als auch für das Netzwerk, indem es integriert ist. Es muss überlegt werden, welche Ziele ein Angreifer im Netzwerk haben kann. Welches Ziel steuert er an? Warum möchte er gerade dieses Ziel angreifen? Diese Arbeit differenziert Angriffe auf Router, Server und auf Endbenutzer.

3.1.7.1 Router

Router sind Schnittstellen zwischen verschiedenen Netzwerken, deren Aufgabe darin besteht, den Nachrichtenaustausch zwischen den Netzwerken zu gewährleisten. Die *Destination Address*, also Zieladresse von IP-Paketen, werden von Routern benutzt. Sie

regeln das Weitersenden von eingehenden IP-Paketen aufgrund einer Routingtabelle. Eine solche Routingtabelle enthält zu jeder Ziel-IP-Adresse eines Nachrichtenpakets eine zugehörige Schnittstelle.

Destination Network	Netmask	Router	Metric
192.168.0.0	255.255.255.0	192.168.0.1	1
245.133.2.0	255.255.255.0	245.133.2.1	1
135.22.0.0	255.255.0.0	135.22.0.1	2
135.22.0.0	255.255.0.0	135.22.1.1	3

Tabelle 3.2: Beispiel einer Routingtabelle

Wie in Tabelle 3.2 zu sehen ist, geschieht die Zuordnung anhand einer Maskierung der Zieladresse. Das bedeutet, dass die Ziel-IP-Adresse mit der Netzwerkmaske (Netmask) logisch mit UND verknüpft wird und das Ergebnis mit der linken Spalte (Destination Network) verglichen wird. Stimmt nun eine der Adressen mit dem Ergebnis der UND-Verknüpfung überein, so wird das Paket an den nächsten Router, der in der entsprechenden Zeile notiert ist weiter geschickt. Sollte es keine Übereinstimmungen geben wird das Paket verworfen.

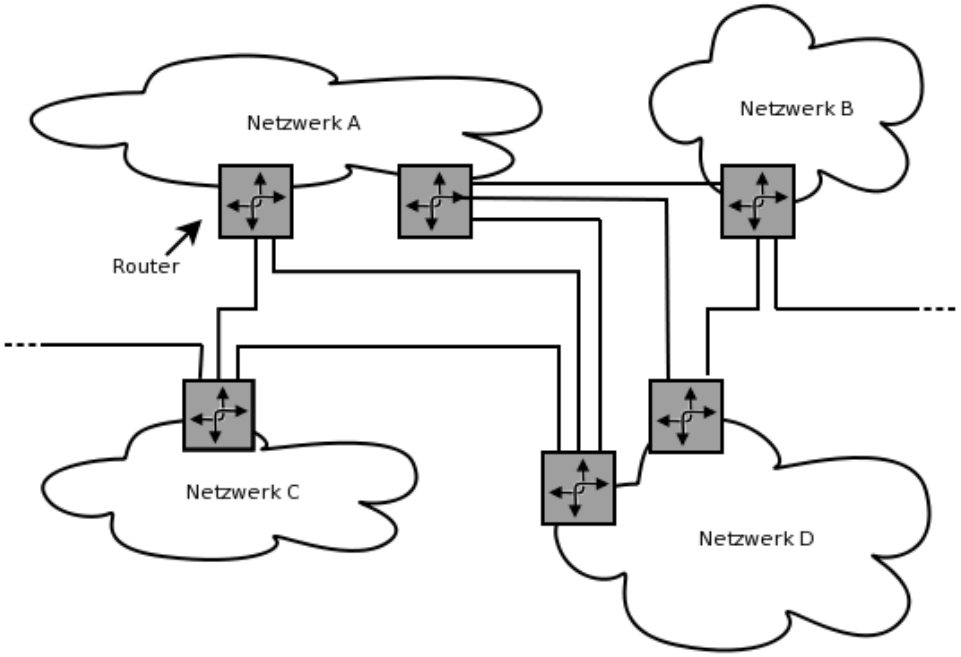

Abbildung 3.5: Grobe Skizzierung des Internets

Möglich ist auch, dass es zu einer Adresse mehrere Wege gibt. Dann entscheidet der Wert der letzten Spalte, die sog. Metrik über den nächsten Router. Die Metrik beschreibt die Anzahl der Router, die auf dem Weg zum Ziel noch zu passieren sind. Je kleiner also dieser Wert ist, desto schneller wird das Paket vermeindlicherweise am Ziel ankommen.

Wie in Abb. 3.5 zu sehen ist, kann das Internet grob gesehen als Zusammenschluss von vielen Netzen durch Router gesehen werden.

3.1.7.2 Server

Server bieten Dienste an, die von vielen oder gar allen Nutzern im Internet abgerufen werden können. Diese Dienste sind z.B. HTTP, FTP oder Mail. Ein Webserver beispielsweise bietet Webseiten, die über einen Browser betrachtet werden können. Ein Mailserver bietet dagegen Dienste an, über die sich eMails verschicken und empfangen lassen.

3.1.7.3 Endbenutzer

Der Begriff „Client“ bezeichnet den Nutzer der verschiedenen, von Servern angebotenen Dienste, also z.B. denjenigen, der den Browser bedient und damit Webinhalte der Webserver für sich nutzt. Der Endbenutzer ist der Nutzer des Internets, der z.B. als Client bei einem Server Dienste, wie HTTP nutzt. Dieser sitzt zumeist an einem eigenen PC.

3.1.8 Der Angriff

Die verschiedenen Ziele der Angreifer sind somit definiert. Nun stellt sich die Frage, was passiert, wenn das eine oder das andere ausgewählte Ziel geschädigt wird und was ein potentieller Angreifer mit einer solchen Schädigung erreichen möchte. Darauf wird nun im Einzelnen eingegangen.

1. **Router**: Ein Router ist, wie in Abb: 3.5 zu sehen ist, ein empfindliches Bindeglied zwischen den Netzwerken. Fällt einer aus, ist es eventuell noch nicht auffällig. Sobald aber mehrere Router ihren Dienst versagen, kann dies schnell zu größeren Problemen führen. Wenn eine bestimmte Route über keinen der Router mehr zu erreichen ist, fällt der gesamte Datenverkehr für diese Strecke aus. Jeder erdenkliche Dienst kann dann nicht mehr genutzt werden, da die Pakete, egal welcher Art, unterwegs auf jeden Fall verloren gehen.

2. **Server**: Bei Servern ist die Situation nicht ganz so kritisch, aber je nach Auftrag des Servers nicht minder wichtig. Der Ausfall eines oder mehrerer Server beeinträchtigt nicht den gesamten Verkehr im Internet. Es wird nur die Kommunikation mit diesem speziellen Server beeinträchtigt.

   So können z.B. Störungen von Suchmaschinen-Servern dazu führen, dass die eine oder andere Suchmaschine nicht mehr zu erreichen ist. Dies könnte beispielsweise eine kriminelle Handlung eines Konkurrenten sein, der seine eigene Seite dadurch bevorteilen möchte. Ein anderes Beispiel wäre ein Bank-Server, der in die Hände eines Kriminellen gerät, welcher dann Zugriff auf die Konten haben könnte.

3. **Endbenutzer**: Der Endbenutzer ist zwar das kleinste, aber auch das am wenigsten abgesicherte Ziel eines Angriffs. Die großen Systeme der Router und Server sind professionell überwacht, während der kleine Anwender als Endbenutzer zumeist wenig Kenntnisse von Netzwerken und deren potentiellen Gefahren hat. Die Rechner der Endbenutzer werden zumeist mit Würmern und Viren angegriffen.

   Die Angriffe auf Rechner der Endbenutzer sind zumeist noch die harmlosesten, da zum einen die Störung eines einzelnen Rechners im Internet keine Auswirkungen hat. Bei Endbenutzern sind es eher Schadprogramme, die z.B. die Daten des Benutzers löschen oder seine Kontodaten ausspionieren.

Oftmals werden solche Angriffe auf Server und Endbenutzer dazu genutzt um auf Schwachstellen aufmerksam zu machen. So genannte *Hacker* versuchen mit den ihnen zur Verfügung stehenden Mitteln in die Rechner einzudringen und oftmals harmlose Spuren zu hinterlassen, die eher lästig als gefährlich sind.

Jedoch ist jegliche Art von Angriff als „unerwünscht“ einzustufen, da jeder die Möglichkeit aufzeigt, wo und auf welche Art und Weise auch gefährliche Angriffe vollzogen werden könnten. Es gibt Angriffe von außen, also solche, bei denen der Angreifer nicht im eigenen Netzwerk sitzt. Zusätzlich gibt es noch Angriffe von innen, die prinzipiell gefährlicher sein können, da die meisten Abwehrmechanismen gegen externe Angriffe wirken. Intern lässt sich jedoch jegliche Manipulation leichter nach verfolgen, „Gefahrenherde“ sind schneller erkannt und können bekämpft werden, da der Administrator auf jeden Router, Server und Endrechner Zugriff hat. Dieser Zugriff ermöglicht die Auswertung verschiedenster Log-Dateien und somit das Nach verfolgen aller im Netzwerk gesendeten Pakete.

Zusätzlich lassen sich Angriffe physischer oder virtueller Art unterscheiden. Verschiedenste Bedrohungen für Netzwerke, seien es interne Netzwerke oder das Internet, werden im folgenden Abschnitt nun erläutert.

3.2 Bedrohungen

Der folgende Abschnitt beschreibt Angriffe auf Router, Server und Endbenutzer und zeigt, was bei welchen Systemen zu Problemen führen kann und welche Auswirkungen dies haben könnte.

3.2.1 (Distributed) Denial of Service

Denial of Service (abgekürzt: DoS) ist eine Variante eines Angriffs, die im Grunde jedes Computersystem lahm legen kann. Somit können damit Server, Router und Endbenutzer gleichermaßen angegriffen werden. DoS bedeutet übersetzt etwa „außer Betrieb setzen“. Grob gesagt wird dieses Versagen durch eine Flut von Paketen erzeugt, die an die jeweilige Adresse des anzugreifenden Systems geschickt werden. Diese Flut bewirkt dann eine Belastung des Systems, welche im schlimmsten Fall zu dessen Absturz führt. Im weniger brisanten Fall wird nur eine bestimmte Komponente des Systems lahm gelegt. DoS kann aber nicht nur durch Überlast geschehen, sondern auch auf Implementierungsfehlern basieren. Das heißt, dass durch Fehler im Programmcode, beispielsweise der Serveranwendung, gezielt gegen diese vorgegangen werden kann, was dann z.B. den Absturz der jeweiligen Anwendung zur Folge hat.

Man kann das Ziel eines solchen Angriffs grob in vier Punkte unterteilen:

1. Absturz eines Systems
2. Verhindern des Datenverkehrs zwischen zwei Systemen
3. Belastung eines Netzwerks zur Verlangsamung der Geschwindigkeit und Verschlechterung der Produktivität
4. Blockierung eines Systems. Gefährlicher als ein Absturz, da kein Neustart erfolgt

Die Überflutung ist natürlich um so wirkungsvoller, je mehr Pakete bei dem „Opfer“ ankommen. Hier gewinnt der Zusatz *Distributed*, also *verteilt*, an Bedeutung. Durch die Involvierung vieler, zumeist nichtsahnender Internetnutzer werden die Flut-Pakete von vielen PCs gleichzeitig zum Opfer geschickt. Dies lässt sich z.B. durch Trojaner (siehe Abschnitt 3.2.2) bewirken, durch die die Kontrolle eines fremden PCs möglich ist. Durch diese Fremdkontrolle gelingt es dem Angreifer unbemerkt viele Pakete ans Ziel zu bringen. Unbemerkt in dem Sinn, dass der Absender nicht Angreifer ist, sondern Dritte diesen Angriff unwissend durchführen.

Die Anzahl der DoS-Angriffe hat vor allem seit Anfang 2005 stark zugenommen. So hat sich die Anzahl der durchschnittlichen Angriffe pro Tag von ca. 200 auf ca. 1000 erhöht.

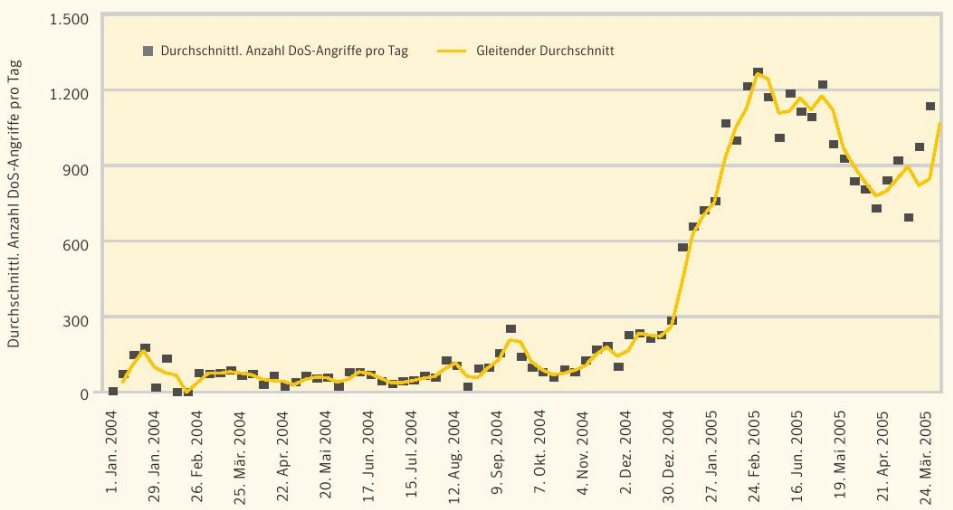

Abbildung 3.6: Anzahl der DoS-Angriffe - Quelle: Symantec Band VIII[12]

Durch diese DoS-Methode können auch Router angegriffen und deren Belastungsgrenze bis aufs Äußerste ausgereizt werden. Ohne Schutzmechanismen wäre ein Router schnell mit so vielen Paketen überfordert, bzw. würde sehr viele wichtige Pakete, die ihn passieren wegen Pufferüberlaufs verwerfen. Das würde das Internet im Ganzen blockieren, da nun die Sender immer und immer wieder versuchen würden ihre TCP-Pakete an den Empfänger zu senden, jedoch beim Router scheitern. Wenn nun auch noch mehrere Router betroffen sind, besteht nicht mehr die Möglichkeit, einen anderen Weg zu suchen. Ein solcher Angriff auf Router ist sicher der effektivste aber zugleich auch schwierigste, da die Router der heutigen Generation sehr spezialisierte Systeme sind, die diese DoS-Attacken erkennen und bekämpfen können.

Im folgenden werden nun ausschnittsweise verschiedene Varianten der DoS-Attacken erläutert.

3.2.1.1 SYN-Flooding/Land

Diese Methode nutzt den Verbindungsstatus einer TCP-Verbindung aus. Beim Aufbau einer solchen TCP-Verbindung wird, wie in Abschnitt 3.1.5.1 erwähnt, ein Handshake-Mechanismus verwendet. Dieser kann nun gezielt zum Angriff genutzt werden.

Die *Land-Attacke* realisiert dies, indem sie das SYN-Paket des Senders mit einer falschen Absenderadresse versieht. Das führt dazu, dass der Empfänger sein SYN/ACK-Paket an die falsche Adresse sendet und keine Antwort erhält. Daraus folgt wiederum, dass der Empfänger immer wieder versucht sein SYN/ACK-Paket zu senden, jedoch keine Antwort erhält. Dies geschieht so lange bis ein Timeout diese Versuche abbricht. Der

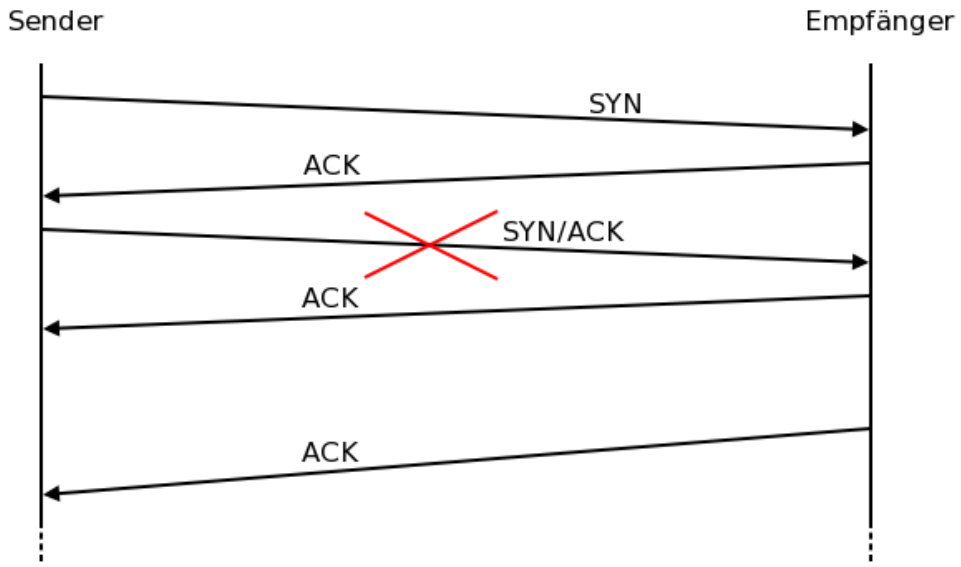

Abbildung 3.7: Handshake vor der TCP-Verbindung

Sender, also der Angreifer, wird aber nicht nur ein SYN-Paket schicken, sondern gleich eine ganze Reihe von solchen Paketen.

3.2.1.2 Ping of Death

Diese Variante ist eine der einfachsten Möglichkeiten eines DoS-Angriffs. Hierbei wird die Fragmentierung von TCP ausgenutzt. Die Maximale Paketgröße eines TCP-Paketes ist 65.535 Byte. Je nach Einstellung der Internetverbindung werden TCP Pakete fragmentiert, das heißt in mehrere kleinere Pakete unterteilt und beim Empfänger wieder zusammengesetzt.

Ein Offset-Wert bestimmt hierbei, wann welches Paket anfängt. Durch Manipulation dieses Wertes des letzten fragmentierten Paketes wird erreicht, dass das gesamte Paket die maximale Größe überschreitet. Wenn die zusammengesetzte Paketgröße die Größe von 65.535 Byte übersteigt gibt es einen Puffer-Überlauf und im ungünstigsten Fall einen Absturz des empfangenden Rechners. Diese Lücke wurde jedoch mittlerweile bei nahezu allen gängigen Betriebssystemen geschlossen.

Ein ICMP-Echo-Requests wäre die einfachste Variante eines solchen manipulierten Pakets. Mit einem *ping*-Befehl können solche ICMP-Nachrichten versendet werden. Allerdings muss hierzu das reguläre *ping* abgeändert werden, da in der offiziellen Version die fehlerhafte Konstruktion eines solchen Paketes nicht gestattet ist.

3.2.1.3 Smurf

Bei einem Smurf-Angriff werden vom Angreifer sehr viele ICMP-Pakete an die Broadcast-Adresse eines Netzwerk geschickt, so dass jeder Rechner dieses Netzwerks diese Pakete erhält.

Der Angreifer tarnt seine Pakete nicht mit einer unbekannten Adresse sondern gibt als Absenderadresse die Adresse des Opfers an. Somit werden nun alle Antwortpakete dieses Broadcast-Netzwerks an das Opfer geschickt. Sollten also beispielsweise 1000 Rechner in diesem Netzwerk antworten und der Angreifer hat 1000 Pakete geschickt erhält das Opfer $1000 * 1000 = 1.000.000$ Antwort-Pakete. Dadurch wird die Kapazität des Opfers voll ausgeschöpft, so dass dieser keine weiteren Datentransfers durchführen kann.

3.2.1.4 Teardrop

Wie beim *Ping of Death* nutzt auch diese Methode die Fragmentierung von großen TCP-Paketen aus. Jedoch wird hierbei keine Überlänge der Pakete generiert, sondern die Fragmente überlappen sich gegenseitig, so dass das Betriebssystem des angegriffenen Rechners mehr Ressources zum Zusammensetzen der Pakete belegt als es verkraften kann. Die Folge ist der Absturz des Systems.

Um einen DDoS-Angriffe starten zu können, müssen viele Rechner unter die Kontrolle des Angreifers gebracht werden. Um diese Kontrolle zu erreichen, werden Programme entwickelt. Diese Programme werden über Würmer oder Trojaner verbreitet. Die Existenz eines solchen Trojaners auf einem PC kann den Zugriff auf diesen ermöglichen.

3.2.2 Trojaner

Trojaner sind Programme, die zumeist über Dateianhänge in eMails verbreitet werden. Sie haben meist „harmlos“ klingende Namen. Diese Programme enthalten jedoch schädliche Komponenten. Der Begriff *Trojaner* wurde aus der Mythologie des *Tronjanischen Pferdes* übernommen.

3.2.3 Bots

Mit Trojanern ist es möglich, den infizierten PC zu einem Bot werden zu lassen (Bots: Abkürzung für das englische Wort für Roboter „robots“). Bots sind demnach infizierte PCs, die mit Hilfe von Trojanern ferngesteuert werden können. Sie können beispielsweise verwendet werden um DDoS-Attacken (vgl. Abschnitt 3.2.1) auszuführen. Sie sind in der Lage viele Pakete zu generieren und abzuschicken.

Trojaner können auf Servern und Endbenutzersystemen ausgeführt werden und diese fernsteuern. Zumeist werden hier aber die Systeme der Endbenutzer benutzt, da diese noch immer am wenigsten abgesichert sind und sich daher bestens für eine solche Infizierung und anschließende Fernsteuerung nutzen lassen.

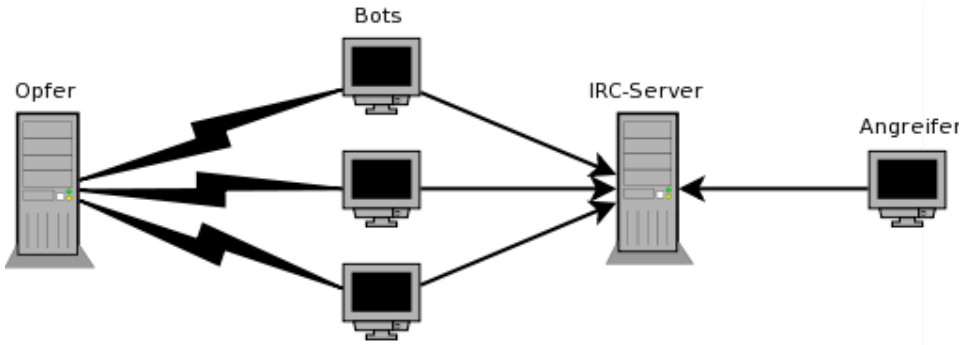

Abbildung 3.8: Darstellung eines typischen Botnetzwerks

Die Fernsteuerung läuft in der Regel über ein IRC-Netzwerk. IRC (*Internet Relay Chat*) ist ein Protokoll zur textuellen Kommunikation, dem so genannten *Chat*. Ein IRC-Server besitzt mehrere Chaträume, sog. *Channels*, denen IRC-Clients beitreten können. Innerhalb eines Chatraums können beliebige Textnachrichten ausgetauscht werden. Diese Prinzip machen sich nun Angreifer zunutze. Jeder Bot tritt einem bestimmten, vom Angreifer gewählten Chatraum bei. Somit besitzt der Angreifer die Möglichkeit, seinen Bots mittels Textnachrichten Befehle zu erteilten, die beispielsweise einen SYN-Flood Angriff einleiten.

Abbildung 3.9: Anzahl der Bots - Quelle: Symantec Band VIII[12]

Die Anzahl der Bots ist in dem Zeitraum von Anfang 2004 bis Anfang 2005 auf das 7-fache gestiegen (siehe Abbildung 3.9). Die Anzahl der Bots und DoS-Angriffe hängen zusammen, denn solche Bots eignen sich gut dafür, verteilte DoS durchzuführen. Das heißt, ein so genanntes Bot-Netzwerk ist aufzubauen, welches dann gemeinsam mehrere PC dazu missbraucht, ein ausgesuchtes anderes System im Internet, sei es einen Server, einen Router, oder einen anderer Endbenutzer, zu attackieren.

3.2.4 Würmer

Würmer sind eigenständige Programme, die über Lücken im Betriebssystem auf einen Rechner gelangen können. Diese Lücken entstehen durch offene Ports, d.h. durch Schnittstellen im Betriebssystem, die für bestimmte Dienst genutzt werden. Auch Würmer können so programmiert sein, dass sie dem Entwickler Fernkontrolle ermöglichen.

3.2.5 Viren

Viren sind kleine Programmstücke. Sie können sowohl in einer Datei oder im Speicher des betroffenen Rechners befinden. Viren sind nicht eigenständig, benötigen immer ein Medium, welches sie „infizieren“.

Viren sind in der Lage sich selbst zu verbreiten, indem sie sich beispielsweise über das Adressbuch des „befallenen“ PCs an viele weitere PCs per eMail verschicken.

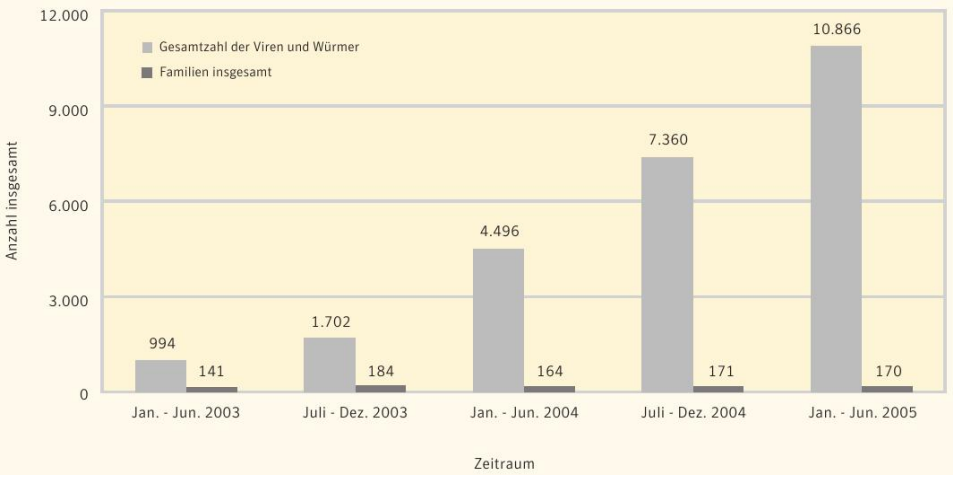

Abbildung 3.10: Statistik zum Virenzuwachs - Quelle: Symantec Band VIII[12]

Die Zahl der Viren hat sich, wie in Abbildung 3.10 zu sehen ist, innerhalb von 2 Jahren, von Januar 2003 bis Januar 2005 verzehnfacht.

Hierbei hat sich die Anzahl der Virenfamilien aber nicht merklich verändert. Viren werden aufgrund ihrer Struktur und Arbeitsweise in verschiedene Klassen, so genannte Virenfamilien aufgeteilt. Diese Anzahl der Virenfamilien schwankt um einen Wert von ca. 160. Auch innerhalb dieser Virenfamilien wird der Virencode oft leicht abgeändert, so dass Virenscanner den Ursprungsvirus nicht mehr erkennen. So müssen Virenscanner immer wieder aktualisiert werden und die ständigen Variationen in Ihre Datenbanken aufnehmen.

Viren, Würmer und Trojaner könnne befallene Rechner schädigen, indem sie beispielsweise bestimmte Verzeichnisse löschen. Somit kann ein irreparabler Schaden angerichtet werden. Manchmal werden Nutzer nur durch solche Programme belästigt, wie beispielsweise durch das Herunterfahren des Systems. Diverse Sicherheitsmechanismen, wie Dateirechte, Firewalls (siehe Abschnitt 3.4.1.1) oder Antivirenprogramme (Abschnitt 3.4.2.2), gewähren Schutz. Jedoch gibt es immer wieder Lücken, die solche Schutzmaßnahmen unwirksam machen.

3.2.6 Bedrohungen mittels des DNS-Protokolls

Das DNS-Protokoll (siehe Abschnitt 3.1.6.4) ist für die Vereinfachung der Navigation wegen der Vielzahl der Server weltweit ein sehr wichtiges Protokoll. Sollte dieses ausfallen oder anderweitig durch einen Angriff gestört sein, so lassen sich, je nach Ausmaß des Angriffs, viele Server nicht mehr über ihre Domain-Namen erreichen. Die Server sind zwar intakt und auch zu erreichen, jedoch nur direkt über ihre IP-Adresse. So ließe sich `www.iv.tu-berlin.de` nur noch über `130.149.16.12` erreichen.

Die Bedeutung des DNS-Protokolls führt auch zu indirekten Angriffsmöglichkeiten. Durch Methoden, die die Funktionalität des DNS-Dienstes ausnutzen, lassen sich beispielsweise falsche Einträge in DNS-Servern platzieren. Diese Einträge können zur Folge haben, dass Domainnamen nicht so aufgelöst werden, wie es korrekt wäre, sondern so, wie der Angreifer es „wünscht“.

3.2.6.1 Cache Poisoning

Beim *Cache Poisoning* wird ein anzugreifender DNS-Server so manipuliert, dass er auf eine Anfrage an einen DNS-Root-Server eine falsche Antwort erhält. Das DNS-Protokoll nutzt als alleinige Authentisierung eine 16-Bit große ID, die *Transaction-ID*. Sollte es dem Angreifer gelingen diese ID zu erraten, dann kann er das Antwortpaket fälschen.

Die ID ist aber nicht alleine nötig, um falsche Pakete zu senden. Der UDP-Port des DNS-Servers ist ebenso wichtig. Theoretisch wäre es möglich, dass jede einzelne Anfrage über einen anderen Port gesendet wird. Dann müsstes der UDP-Port auch erraten werden,

was dem Angreifer die Arbeit erschweren würde. Dies wird jedoch in den aktuellen BIND Versionen nicht praktiziert (BIND ist der DNS-Server, welcher am häufigsten verwendet wird - Berkley Internet Name Domain).

So ist es möglich, dass der Angreifer mittels eines eigenen DNS-Servers manipulieren kann. Hat er Kontrolle über einen DNS-Server, so kann er an den zu attackierenden DNS-Server eine Anfrage schicken, die die Auflösung des Domainnamens eines im Netz des Angreifers befindlichen Hosts enthält. Der angefragte DNS-Server wird dann wiederum eine Anfrage an den DNS-Server des Angreifer schicken, da der angefragte Host auf diesem Server eingetragen ist. Dieser Anfrage kann der Angreifer nun den von BIND benutzen Source-Port entnehmen und ihn für seine gefälschten Antworten benutzen.

Das letzte nötige Detail ist die Source-IP-Adresse, d.h. die IP-Adresse des zuständigen Nameservers. Da diese bekannt ist, ist es kein Problem mehr Antwort-Pakete zu erstellen, welche die korrekte Source-IP-Adresse, den korrekten Source-Port und eine generierte ID, sowie die gefälschte Antwort des Nameservers enthalten.

Eine besonders effiziente Art des *Spoofings*, d.h. des Versendens von Paketen mit falschem Absender und manipuliertem Inhalt, ist die *Geburtstags-Attacke* (Birthday-Attack), benannt nach dem Geburtstagsparadoxon:

> Die Wahrscheinlichkeit, dass aus einer Gruppe von 23 Leuten mindestens zwei am gleichen Tag Geburtstag haben, ist grösser als $0,5$.

Dieses mathematische Phänomen nutzt nun der Angreifer. Hierbei wird die Fragestellung umformuliert. Bei der Geburtstags-Attacke stellt sich nun die Frage, wieviele Anfragen und Antworten gesendet werden müssen, damit mit einer Wahrscheinlichkeit von über $0,5$ mindestens eine Anfrage und eine Antwort dieselbe *Transaction-ID* besitzen? Die Antwort auf diese Frage liefert folgende Formel:

$$P = 1 - (1 - \frac{1}{t})^{\frac{n*(n-1)}{2}}$$

Laut dieser Formel[50] lässt sich die Zahl der Pakete bestimmen. Es sind $n = 302$ Pakete, die nötig sind, um bei $t = 65535$ möglichen IDs eine Wahrscheinlichkeit von $P > 0,5$ zu erhalten.

Um nun seine Attacke mit möglichst großer Wahrscheinlichkeit zu starten, sendet der Angreifer nun 302 Anfragen an den anzugreifenden DNS-Server. Gleichzeitig werden ebensoviele manipulierte Antworten gesendet und der eigentlich zuständige Nameserver mit einer DoS-Attacke blockiert, so dass er nicht so schnell die korrekte IP-Adresse liefern kann. Jede dieser manipulierten Antwort-Pakete enthält eine andere generierte ID. Somit erreicht der Angreifer sein Ziel mit einer Wahrscheinlichkeit von etwa 50%. Sollte nämlich nun die ID der gefälschten Antwort mit der, der erwarteten Antwort übereinstimmen, so wird die Antwort für „wahr“ empfunden und ein Eintrag im DNS-Cache vorgenommen,

der die falsche IP-Adresse enthält. Dieser falsche Eintrag ist nun so lange gültig, bis der Timer abläuft, der in dem manipulierten Antwortpaket als TTL-Feld (Time-To-Live) gegeben war.

Bei einfachem DNS-Spoofing, würde die gleiche Paketanzahl mit einer Wahrscheinlichkeit von $\frac{302}{65535} = 0,0046$ zum einem erfolgreichen Angriff führen. Bei einfachem Spoofing werden auf eine einzelne Anfrage mehrere Antworten mit unterschiedlichen IDs gesendet. Wie aber zu sehen ist, ist diese Methode weit ineffizienter als die Geburtstags-Attacke.

Wenn nun ein Client eine Anfrage an den DNS-Server sendet und nach der IP-Adresse der soeben manipulierten Domain fragt, so wird er eine falsche Antwort bekommen und auf einen anderen Server geleitet, als gewünscht war.

3.2.6.2 DNS Amplification

Bei dieser Variante ist nicht der DNS-Server das eigentliche Ziel, sondern wird als Angreifer missbraucht. Es wird ausgenutzt, dass bei DNS auf kurze Anfragepakete (60 Byte) lange Antwortpakete folgen. Diese können bis zu 4000 Byte groß sein[52], so dass der so genannte Verstärkungsfaktor mit $\frac{4000}{60} \approx 67$ ziemlich hoch liegt. Pro gesendeter Datenmenge des Angreifers wird demnach die 67-fache Datenmenge produziert. Somit ist es möglich, durch gezielte Anfragen einen DNS-Server dazu zu veranlassen, Antwortpakete an einen anzugreifenden Dritten zu senden und ihn damit zu überfordern. Zu diesem Zweck wird in denen den Anfragepaketen die IP-Adresse gespooft.

3.2.6.3 Phase Space Analysis Spoofing

Die Geburtstags-Attacke kann mit einer Analyse der generierten Transaction-IDs noch verstärkt werden. Da die IDs mit Pseudo Zufallsgeneratoren erzeugt werden, besteht die Möglichkeit für den Angreifer, die Schwächen dieser Generatoren zu nutzen. Bei der Analyse einer Reihe von generierten IDs ist zu sehen, dass etwa bei BIND 8.4.3 ein großer Bereich von Zahlen gar nicht generiert wird[50]. So kann durch Ausschluss dieser Zahlen die Wahrscheinlichkeit des korrekten Ratens einer ID nochmals erhöht werden.

Neuere BIND-Versionen und andere DNS-Server, wie *djbdns*[1], benutzen bessere Generatoren für ihre Zufallszahlen[50]. Diese Zufallsgeneratoren schützen jedoch nicht vor Angriffen auf das DNS-Protokoll, sie verringern aber die Wahrscheinlichkeit eines erfolgreichen ID-Ratens.

Die falsche Auflösung von Domainnamen ist eine Bedrohung über das DNS-Protokoll. Diese Protokoll befindet sich in der Applikationsebene. Auf der Netzwerkebene besteht die Möglichkeit über das Routing, Pakete falsch zu vermitteln.

[1] http://cr.yp.to/djbdns.html

3.2.7 Bedrohungen über Routing Protokolle

Bei Angriffen auf das Routing können Router so getäuscht werden, dass sie falsche Routinginformationen in ihre Routing-Tabelle aufnehmen. Diese falschen Routinginformationen können Angreifer nutzen. Beispielsweise könnte erreicht werden, dass alle Pakete, die eigentlich zu Router XY gelangen sollten, beim Angreifer ankommen. So kann er spionieren und sogar fälschen, d.h. die Daten verändert an ihr eigentliches Ziel weiterleiten. Diese Methode wird im allgemeinen als **Man in the Middle**-Methode bezeichnet.

3.2.7.1 ARP

Die einfachste Methode bietet ARP. Es ist zwar nicht direkt ein Routing-Protokoll, hilft aber dabei, dass Pakete ihren Empfänger erreichen. Das *Address Resolution Protocol* ist dazu da, IP-Adressen MAC-Adressen zuzuweisen. Jede IP-Adresse hat eine eindeutige MAC-Adresse, welche für die Adressierung im Link-Layer wichtig ist. Mit ARP kann der Angreifer einfach eine Anfrage eines Senders nach der MAC-Adresse einer bestimmten IP mit seiner eigenen MAC-Adresse beantworten. Die IP bleibt zwar die gleiche, aber auf dem darunter liegenden Link-Layer (z.B. Ethernet) ist die Adresse falsch. So bekommt der Angreifer, statt des eigentlichen Empfängers, alle Pakete gesendet.

3.2.7.2 RIP

Das erste weit verbreitet eingesetzte Routing-Protokoll war RIP. Es arbeitet so, dass jeder der Router seine ihm bekannten Routen und deren Metrik, also die Anzahl der Hops bis zu einem bestimmten Ziel, an alle Router seiner Domain schickt. Es handelt sich bei RIP um ein *Distance-Vector-Protocol*. Die Domain ist im Falle von RIP v1 eine Broadcast-Domain, im Falle von RIP v2 eine Multicast-Domain. So kann einen Angreifer einfach die Route zu seinem Rechner mit einer sehr kleinen Metrik versehen und schon werden alle Pakete an ihn geschickt. Dieses Protokoll findet allerdings kaum noch Anwendung.

3.2.7.3 EIGRP

Heute wird eher das EIGRP benutzt (Enhanced Interior Gateway Routing Protocol). Wie der Name schon preisgibt, handelt es sich um ein internes Protokoll, welches für die Verständigung der Router innerhalb eines autonomen Systems benutzt wird. Ein autonomes System bezeichnet ein Teilnetzwerk, welcher unter nur einer administrativen Verwaltung steht. Es ist ein Cisco^{TM}-eigenes Protokoll, eine Erweiterung von IGRP und fällt auch in die Klasse der *Distance-Vector-Protocols*. Die Routingentscheidung wird hier

nicht mehr nur durch die Anzahl der Hops, sondern auch über Parameter wie Paketlaufzeit, Bandbreite, Verfügbarkeit und Auslastung von Verbindungen bestimmt[39]. Genau wie RIP v2 arbeitet EIGRP über Multicast, d.h. auch bei EIGRP kann der Angreifer mithören und auch selbst konstruierte Routinginformationen an einen Router senden. Dieser sorgt dann dafür, dass auch alle anderen Router diese neuen Routinginformationen erhalten und für ihre Routingentscheindungen verwenden.

Wie RIP v2 bietet auch EIGRP einen MD5-Hash-Algorithmus zur Wahrung der Integrität von Paketen (siehe Abschnitt 3.4.1.2). Es verhindert jedoch nicht, dass früher kopierte Update-Pakete zu einem späteren Zeitpunkt nochmal vom Angreifer gesendet werden können und das Routing beeinträchtigen, falls dieses Update nicht mehr gültig ist.

3.2.7.4 Protokollunabhängig

Auch das Einspeisen von Routinginformationen zu nicht zugewiesenen Adressblöcken stellt ein Problem dar. So sind von der IANA (*Internet Assigned Numbers Authority*) Bereiche von IP-Adressen als „nicht vergeben“ deklariert. Solche Adressen werden von DoS-Angreifern gerne als Source-IP-Adresse in den *gespooften* Paketen verwendet. Wenn das Angriffsziel nun von einer DoS-Attacke getroffen wird und Antworten an diese „nicht vergebene“ Adresse zurück schickt, so sind die Router meist so konfiguriert, dass sie Pakete mit einer solchen Ziel-Adresse verwerfen. Sollte aber nun in der Routingtabelle ein Eintrag zu einer solchen, eigentlich nicht vergebenen Adresse existieren, so würden die Pakete nicht verworfen. Diese Antwortpakete könnten nun an einen Dritten geleitet werden und somit eine weitere DoS-Attacke auslösen.

Ein weiteres Problem ist auch die „Monokultur“ im Bereich der Routerkonstrukteure. Cisco™ besitzt einen Markanteil von ca. 75%[9] im Bereich Internet Backbone Routing. Durch ein einzelnes Problem bei der Software der Router von Cisco™ ist es für Angreifer möglich, gleich große Teile des Internets zu gefährden. Beispiele für Probleme auf Cisco Routern finden sich auf *Security Focus*[2]. So ließ sich mit einfachen Mitteln Zugang zu Cisco Router erreichen[20] oder über geschickte Ausnutzung des SNMP-Protokolls (*Simple Network Managment Protocol*) Datenverkehr über gewünschte eigene Routen umleiten[19].

Insgesamt bilden Angriffe auf das Routing eine Gefahr auf unterster Ebene. Durch solche Angriffe lassen sich ganze Firmennetzwerke in ihrem Betrieb beeinträchtigen. Ebenso groß ist die Gefahr von Spionage und Fälschung. So können über manipulierte Routen Datenpakete an Dritte geschickt werden, kopiert und auch verändert werden, wenn Schutzmechanismen, wie z.B. MD5-Verschlüsselung nicht genutzt werden.

[2] http://www.securityfocus.com

Unterhalb der Netzwerkebene, in der die Funktionalität des Routings enthalten ist, existiert noch die Host-to-Network Ebene. Auch auf dieser Ebene können Netzwerkstrukturen bedroht werden.

3.2.8 Bedrohungen der Physikalischen Infrastruktur

Neben den virtuellen Angriffen, d.h. den Angriffen über entfernte Rechner auf die wichtigen Protokolle, gibt es natürlich auch Angriffe physikalischer Natur. Sei es durch mutwillige Zerstörung, durch Kriege oder Anschläge, oder durch natürliche Einflüsse, wie Erdbeben oder ähnliches.

Das Internet als Ganzes ist vom Grundkonzept her sehr robust. Ausfälle einzelner Knoten lassen sich durch die Flexibilität des Netzes schnell ausgleichen. So werden Routerausfälle von den Nachbarroutern schnell erkannt, die diese Information dann über die Routingprotokolle bekannt geben. Wichtig sind jedoch Schnittstellen und Leitungen, deren lokale Nähe zueinander zu einem Problem werden können. So enden beispielsweise weite Teile der transatlantischen Datenverbindungen in den USA im Bereich New Jersey und Rhode Island. Alleine vier Kabel landen in Manasquan (New Jersey). Weitere in New York, Brookhaven, Bellport oder Green Hill[11].

Eine Katastrophe in diesem Bereich der Vereinigten Staaten, die die Datenkabel treffen würde, könnte die Datenverbindungen zwischen Europa und den USA sehr stark beeinträchtigen. Es existieren zwar auch Leitungen von Amerika über Asien nach Europa, diese Kapazitäten sind jedoch nur unzureichend[14]. Ein Ausbau dieser Verbindungen zwischen Asien und Europa könnte auch für Entlastung der Verbindung USA - Europa sorgen.

Ein weiterer Punkt sind die Peering-Points und Internethubs, die ebenfalls zentrale Knoten im Internet darstellen. Über diese Knoten tauschen die großen Provider ihre Daten aus, so dass sehr viel Datenverkehr über diese Systeme geht. Folgend werden ein paar zentrale Peering-Points genannt.

DE-CIX Deutscher Commercial Internet Exchange. An diesen Netzknoten sind zur Zeit 180 Autonome System angeschlossen[15], knapp 41 Gbps durchschnittlicher Datenverkehr zeugen von hohem Durchsatz, denn fast alle Provider sind mit diesem Knoten verbunden.

LINX London Internet Exchange. Er ist Europas größter Peering-Point mit ca. 100 Gbps[40] durchschnittlichem Datenaufkommen.

NYIIX New York International Internet Exchange, mit etwa 12 Gbps Durchsatz[46].

AMS-IX Amsterdamer Internet Exchange - 41,7 Gbps durchschnittlicher Durchsatz[17].

EQUINIX 81 Gbps durchschnittlicher Durchsatz, kombinierter Verkehr von 5 großen Knoten innerhalb der USA[16].

Dies ist nur eine begrenzte Auswahl. Es gibt zwar noch weitere Peering-Points mehr, diese haben jedoch viel weniger Daten zu verarbeiten und zu vermitteln. Der längerfristige Ausfall des *DE-CIX* wäre sicherlich gut zu spüren. Durch einen solchen Ausfall würden die Verbindungen im Internet nur noch sehr langsam senden oder ganz abbrechen. Wenn dann noch ein weiterer alternativer Peering-Point in Deutschland ausfallen würde, könnte es zu größeren Problemen kommen. Mit zusätzlichen Peeringpoints als Alternativen zum *DE-CIX* wäre dessen Ausfall eventuell auszugleichen[23].

Es ist zu sehen, dass es viele Möglichkeiten gibt, Netzwerke anzugreifen. Es existieren viele Bedrohungen, die jeweils einer der Ebenen im Ebenenmodell zuzuordnen sind. Manche der Bedrohungen sind rein theoretischer Natur und haben noch nicht zu großen Problemen geführt. Jedoch soll diese Arbeit aufklären was möglich ist und die Bedrohungen benennen. Zu diesen zählen einige, die bereits zu größeren Ausfällen auf Servern geführt haben. Ein paar von diesen Ausfällen werden im kommenden Absatz beschrieben.

3.3 Angriffshistorie

Diese Sektion beschreibt exemplarisch Angriffe auf HTTP-Server. Staatliche, wie auch kommerzielle Seiten sind in den vergangen Jahren immer wieder solchen Angriffen ausgesetzt gewesen. Diese Daten können dazu benutzt werden, für die Zukunft aus diesen Angriffen zu lernen.

26. Mai 1999, FBI Der erste DoS-Angriff, der öffentlich diskutiert wurde war ein Angriff auf die Webseite des FBI[55]. Die US-Institution hatte nach einigen Angriffen auf US-Regierungscomputer eine Suche nach Personen der Hacker Gruppe „Global Hell“ begonnen. Daraufhin wurde die Webseite des FBI von den Hackern ins Visier genommen. Mehrere Stunden wurde die Seite außer Gefecht gesetzt, so dass das FBI erst einmal keine andere Wahl hatte, als die Seite vom Netz zu nehmen, um größeren Schaden abzuwenden.

7. Februar 2000, Yahoo™ Yahoo, mittlerweile ein große börsennotierte Gesellschaft, wurde Anfang 2000 mit einer sehr großen DoS-Attacke lahm gelegt. Nach Angaben von GlobalCenter™, dem Webhoster von Yahoo, wurde zur Angriffszeit ein Datenaufkommen von 1 GBit pro Sekunde gemessen[44]. Diese Ausmaß konnte der Server, trotz implementierter Filtermechanismen, die eine gewisse maximale Datenmenge garantieren sollten, nicht verkraften.

8. Februar 2000, Amazon™, Buy.com™, CNN™, and eBay™ Nachdem einen Tag zuvor Yahoo Ziel eines DoS-Angriffs war, wurden an diesem Tag gleich mehrere große Internetfirmen attackiert[10]. Und genau, wie tagszuvor, war die

riesige Menge an Datenaufkommen welche von hunderten oder gar tausenden Bots generiert wurde, von den Servern nicht zu bewältigen. So waren die Betreiber der Seiten gezwungen, ihre Inhalte für einige Stunden vom Netz zu nehmen. Auch am Tag danach wurden noch Webseiten wie ZDNet™und E*Trade™von einer Attacke betroffen[45]. Insgesamt wurde im Februar des Jahres 2000 das Thema *DoS-Attacken* erstmals weltweit wahrgenommen. Viele kleine weitere Angriffe fanden noch statt, und Experten waren gefragt, mit welchen Maßnahmen man sich gegen solche Angriffe wehren könne.

21. Mai 2001 CERT Für die Dauer von zwei Tagen wurde eine der wichtigsten Firmen für Internetsicherheitsthemen aufgrund einer DDoS-Attacke vom Netz genommen. Und diesmal ist zu sehen, dass selbst Internetfirmen wie CERT (Computer Emergency Response Team), die sich um die Sicherheit im Netz sorgen, nicht sicher vor DoS-Angriffen sind.

1. Februar 2004, SCO™ Mit einem sich weit verbreitenden Wurm hat sich Ende Februar 2004 ein DDoS Angriff angekündigt, der die Softwareschmiede SCO treffen sollte. Schon Wochen vorher war bekannt, dass am 01.02.2004 ein großer Angriff auf die Webseite der Firma SCO wirken würde. Mit einem Wurm, der sich selbstständig weiterverbreitete[13], wurden Trojaner installiert. Bei diesen war schon im voraus bekannt, dass sie für den Angriff auf SCO konzipiert wurden. Nach einer Analyse von Sophos™, eines Internetsicherheitsunternehmens, war im Januar 2004 dieser Wurm mit dem Namen *MyDoom*[6] mit einem Anteil von 25, 1%[49] an der weltweiten Virenverseuchung beteiligt. Jedoch konnte trotz dieses Wissens eine solche Attacke auf die Webseite nicht verhindert werden. Mehrere Tage wurde die Seite mit einer Datenflut überhäuft. Die einzige Gegenmaßnahme, die getroffen wurde, war den DNS-Eintrag zu löschen, so dass der Server nur noch direkt über die IP-Adresse erreichbar war[53]

04. Oktober 2004, holländische Regierung Nicht nur kommerzielle Betreiber von Webseiten, sondern auch staatliche Seiten werden vermehrt Ziel von DoS-Attacken. Im Oktober 2004 wurden mehrere Server der holländischen Regierung angegriffen. Nach Angaben von Regierungssprechern[43] waren jedoch nur öffentliche Informationsseiten betroffen und keine sicherheitsrelevanten Seiten. Dieser Angriff erfolgte vermutlich nach Protesten gegen die Arbeit der Regierung und stand im Zusammenhang mit diesen Geschehnissen. Im Juni 2006 passierte ähnliches mit einem Regierungsserver der schwedischen Regierung. Auch dieser Webserver der Regierung mit informativem Inhalt wurde für einige Stunden außer Gefecht gesetzt[41], nachdem bereits einen Tag zuvor die Webseite der schwedischen Polizei von einer DoS-Attacke betroffen war.

Wie an diesen Beispielen zu sehen ist, hat es in der jüngsten Vergangenheit zahlreiche Angriffe auf Computersysteme mit gesellschaftlicher Relevanz gegeben. Diese hier gewählte Zusammenstellung ist nur ein kleiner Ausschnitt aus der Historie der Angrif-

fe. Um weitere Angriffe zu verhindern müssen Maßnahmen getroffen werden, die den Angreifern entgegen wirken.

3.4 Gegenmaßnahmen

Um Angriffen, wie z.B. Trojanereinschleusung oder DoS etwas entgegen zu wirken, müssen Maßnahmen getroffen werden. Solche Gegenmaßnahmen können sowohl direkt als auch indirekt wirken. Einige der klassischen Gegenmaßnahmen werden in den folgenden Abschnitten einmal kurz erläutert.

3.4.1 Präventive Maßnahmen

Präventiv sind jene Maßnahmen, die schon im voraus die Folgen eines Angriffs abschwächen sollen. So kann durch derartige Maßnahmen verhindert werden, dass Angriffe überhaupt stattfinden, wodurch ein Einschreiten nicht vonnöten ist.

3.4.1.1 Firewalls

Eine Firewall ist im Idealfall ein zusätzlicher, autonomer Rechner, der zwischen zwei Netzwerken installiert wird. So erreicht man, dass der gesamte Datenverkehr den diese Netzwerke austauschen, diese Firewall passieren muss. Solche Netzwerke bestehen im allgemeinen aus einem lokalen Netzwerk (LAN) und dem Internet. Diese Firewall ist nun dazu da, nach bestimmten Regeln, Pakete zu filtern. Somit kann das interne Netzwerk über die Firewall geschützt werden. Durch die Filterung ist es beispielsweise möglich, Rechnern aus dem Internet das Verbinden über ein „unerlaubtes“ Protokoll mit einem internen Rechner zu verhindern.

Über eine Firewall kann also nach jeder in den Paketen oberhalb der IP-Ebene enthaltenen Informationen (Protokolltyp, Quellport, Zieladresse etc.) gefiltert werden. So lassen sich beispielsweise offene Ports von unsicher eingestuften Programmen vermeiden. Diese Ports sind zwar noch immer offen aber mit geeigneten Filterregeln nicht mehr von außen zu erreichen. Es lässt sich dadurch oberhalb der IP-Ebene der Zugang zum eigenen Netzwerk kontrollieren und beschränken, und somit Angreifern die Arbeit zumindest zu erschweren.

Über eine Firewall können auch DoS-Attacken bekämpft werden. Sollte eine DoS-Attacke stattfinden, ist es Aufgabe des Administrators, eine Signatur in der Flut der ankommenden Pakete zu erkennen. Diese erkannten Merkmale können dann der Firewall mitgeteilt

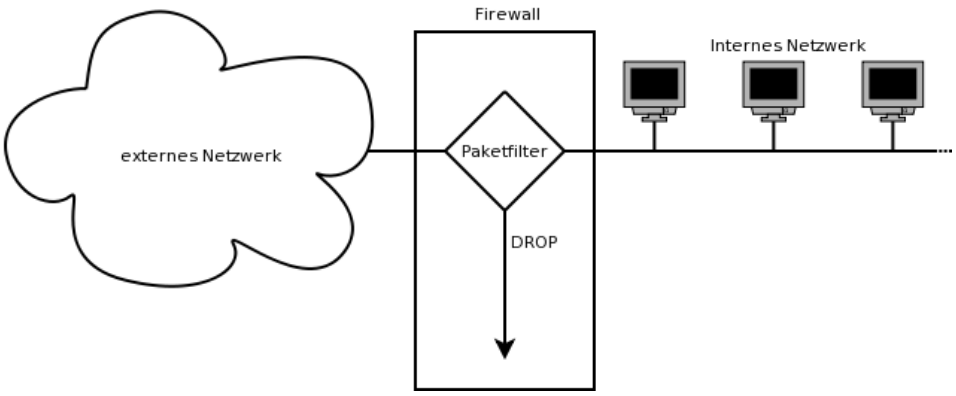

Abbildung 3.11: Grundprinzip einer Firewall

werden, welche dann dafür sorgt, dass Pakete mit diesen bekannten Merkmalen verworfen werden. So wird erreicht, dass die Paketflut zumindest nicht das interne Netzwerk erreicht. Die Firewall als solche kann nun aber trotz aller Vorsorge außer Gefecht gesetzt werden.

3.4.1.2 Verschlüsselung

Mit der Verschlüsselung von Daten lässt sich zwar nicht verhindern, dass ein System angegriffen und beispielsweise durch eine DoS-Attacke lahm gelegt werden kann. Jedoch kann hierdurch die Integrität von Daten gewährleistet werden.

Wie bei den Angriffen über Routingprotokolle schon erwähnt, können Angreifer durch das Einschleusen falscher Routinginformationen das Routing so beeinflussen, dass viele Pakete das System passieren, welches sie unter Kontrolle haben. So ist es dann möglich, Daten mitzulesen. Durch Verschlüsselung wird ein solches mitlesen unmöglich oder zumindest erschwert.

So könnten auch Routinginformationen ausgetauscht werden, ohne dass ein mitlesender Dritter Einsicht in die Topologie des Netzwerks nehmen könnte.

3.4.1.3 Eingangsfilterung

Die auch als *Ingress Filtering*[27] bekannte Methode ist eine von Netzbetreibern, speziell ISPs (Internet Service Provider), verwandte Methode. Sie verhindert, dass IP-Spoofing, also das Fälschen der Absender IP-Adresse über ihr Netzwerk, möglich ist. Mittels spezieller Paketfilter werden nur solche Pakete vom Kunden weitergeleitet, die als Absenderadresse auch wirklich die ihm zugeteilte Adresse enthält oder sich zumindest in dem providereigenen Subnetz befindet. Dadurch wird erreicht, dass das eigene Netz für Angreifer

weniger attraktiv ist, da ihre Bots innerhalb dieses Netzwerks keine Quell-IP-Adressen fälschen können. Somit sind sie dann später auch leichter aufzuspüren.

Präventive Maßnahmen sind notwendig, jedoch nicht das einzige Mittel, welches zur Sicherung von Netzwerken eingesetzt werden sollte. Jede präventive Maßnahme hat Schwächen die ausgenutzt werden können. Die präventiven Maßnahmen müssen durch Mechanismen ergänzt werden, die auf Angriffe reagieren.

3.4.2 Reaktive Maßnahmen

Die Erweiterung der Präventionsmaßnahmen sollten auf aktuelle Angriffe reagieren und deren Folgen mit geeigneten Maßnahmen abmildern oder ganz verhindern.

3.4.2.1 Intrusion Detection Systems

Intrusion-Detection-Systeme bilden eine Einheit aus Anomalie- und Mustererkennung. Anhand bekannter Angriffe werden Signaturen erstellt, so dass mittels dieser Muster Datenströme analysiert und eingestuft werden können. Diese Methode funktioniert jedoch bisher nicht bekannten Angriffsmustern. Bei der Anomalieerkennung werden Datenströme mit einem „Normalmodell“ verglichen. Werden Abweichungen oberhalb einer gewissen Toleranzschwelle erkannt, so können solche Datenströme als Angriffe eingestuft werden. Die Schwierigkeiten hierbei sind eine geeignete Wahl der Toleranzschwelle und die Definition des „Normalmodells“. Eine solches Intrusion-Detection-System kann leider auch missbraucht werden. So können Angreifer eine solche Anomalieerkennung durch gezielte Manipulation auch dazu bringen, legitimen Datenverkehr als Angriff zu interpretieren.

3.4.2.2 Antivirenprogramme

Neben der Firewall, die die Pakete kontrolliert und deren Zulässigkeit prüft, sind Antivirenprogramme dazu da, Viren, die den Weg an der Firewall vorbei geschafft haben zu erkennen und zu vernichten oder zumindest unter Quarantäne zu stellen.

Antivirenprogramme auf Endbenutzersystemen kontrollieren ständig das Dateisystem und reagieren sofort bei Veränderungen. Das heißt, sollte eine Datei abgelegt werden die für das Antivirenprogramm als gefährlich eingestuft wird, alarmiert es den Nutzer.

Antivirenprogramme können aber auch, z.B. auf Mailservern installiert werden. So können eventuell in Mails enthaltene Viren schon abgefangen werden, bevor sie das Endsystem erreichen.

Das Hauptproblem solcher Antivirenprogramme ist ihre Aktualität. Es können immer nur Viren erkannt werden, deren Struktur auch bekannt ist. Da aber täglich, oft sogar stündlich, neue Viren entstehen besteht, das Problem darin, diese neuen Viren ebenso zu erfassen, wie „alte Bekannte“. Daher sind solche Antivirenprogramme immer mit Updatefunktionen ausgestattet, die z.B. eine täglich Aktualisierung ermöglichen.

Antivirenprogramme entdecken auch Trojaner, die sich in Systeme einnisten. Diese können dann von Angreifern genutzt werden um DDoS-Attacken auf bestimmte Ziele zu starten. Da nach dem Start einer solchen Attacke die Verhinderung eines Ausfalls sehr schwer fällt, ist also die Entdeckung und Beseitigung solcher Trojaner das primäre Ziel zur Verhinderung von DDoS-Attacken. Aufgrund dieser Tatsache könnten Antivirenprogramme auch als *präventiv* gesehen werden.

Diese Gegenmaßnahmen, aber auch andere, sollen dazu beitragen, dass ein Angriff auf Netzwerke oder einzelne Systeme gar nicht erst stattfinden oder zumindest erschwert werden. Trotzdem besteht die Gefahr, dass diese Maßnahmen nicht wirken und Ausfälle auftreten können.

3.5 Ausfallpläne

Bei allen Angriffen auf die Computersysteme der Welt, stellt sich doch nun die Frage, was passiert, wenn nun größere, weitläufigere DoS-Attacken gestartet würden. Hier setzt nun meine Arbeit an. Im praktischen Teil meiner Ausarbeitung wird exemplarisch eine Angriffsvariante in einen Simulator integriert und analysiert, welche Auswirkungen in dem angegriffenen Netzwerk zu erkennen sind.

Prinzipiell ist der Angriff auf ein Netzwerk oder einen einzelnen Teilnehmer eines Netzwerkes leicht realisierbar. Sind genügend Bots im Internet „platziert“, kann durch Aktivierung dieser Bots ein DDoS-Angriff gestartet werden.

Daher könnte es im Extremfall eines großen Netzwerkabsturzes nötig sein, alternative Wege für eine Kommunikation zu nutzen. Da die Kritischen Infrastrukturen mehr und mehr von einem funktionierenden Nachrichtenaustausch abhängen, werden solchen Alternativen auch zunehmend größere Beachtung geschenkt.

Eine dieser alternativen Nachrichtenwege wäre der *Short Message Service* (SMS), welcher von Handynutzern zur Kommunikation mittels kleiner Textnachrichten genutzt wird.

3.5.1 SMS

Der *Short Message Service* ist ein Nachrichtensystem des GSM Mobilfunks, welches zum Austausch kleiner Textnachrichten mit einer Länge von 140 (lateinischen) Zeichen

entwickelt wurde. Um den Nachrichtenversand von einem PC über SMS zu ermöglichen, wurden so genannte SMS-Gateways entwickelt. Mit diesen ist es möglich, eine SMS (Kurzform für die Nachricht über den Short Message Service) via PC zu versenden. Somit könnte also der eMail-Versand über SMS alternativ geschehen. Dazu sieht die GSM-Spezifikation von SMS (3GPP TS 03.40[2]) diese Möglichkeit in Punkt 3.8 vor. So kann also die beidseitige Umsetzung von SMS zu eMail über ein kompatibles Gateway realisiert werden.

Zudem muss der Empfänger auch hinter einem SMS-Gateway sitzen, der die SMS empfängt und sie an den Empfänger-PC weiterleitet. Dort muss dann die Rückwandlung von SMS zu eMail erledigt werden.

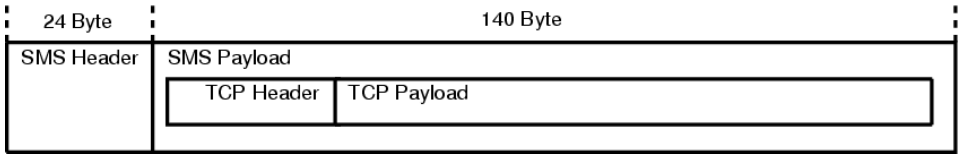

Abbildung 3.12: Struktur eines SMS SUBMIT Paketes mit SSH Daten

Bei komplexerem Nachrichtenaustausch, z.B. einer verschlüsselten SSH-Sitzung, könnte eine Verbindung über binäre Kurznachrichten, wie sie ebenfalls in der SMS-Spezifikation unter 9.1 zu finden ist, genutzt werden. Diese Möglichkeit wird heute auch schon genutzt, um beispielsweise Klingeltöne auf Handys zu übertragen. Eine Kurznachricht beinhaltet eine Payload von 140 Byte. Da SSH-Pakete eine variable Länge haben, müssten diese entsprechend der SMS-Norm angepasst werden. In Abbildung 3.12 ist die grobe Struktur eines solchen Pakets zu sehen. SMS-SUBMIT-Pakete sind die Datenpakete, die von den *Mobile Stations* an die *Service Center*, d.h. von den Handys an die Netzbetreiber, gesendet werden.

Der Short Message Service ist nicht für große Datenmengen ausgelegt. Mit HSCSD (*High Speed Circuit Switched Data*), einer Erweiterung des GSM-Standards (3GPP TS 02.34[1]) , ist eine Datenrate von maximal 57,6 kbit/s möglich. Dies reicht für eine einfache Verbindung, um beispielsweise Textdaten zu versenden, aus.

3.5.2 UMTS

Wenn eine schnellere Datenverbindung vonnöten ist, könnte auch UMTS (*Universal Mobile Telecommunication System*), die neueste mobile Datenübertragungstechnik, verwendet werden. UMTS bietet eine Übertragungsrate von bis zu 2 Mbps[3]. Diese Möglichkeit ist daher für große Datenvolumen gegenüber SMS/GSM vorzuziehen.

3.5.3 Alternativwege

Sowohl über SMS als auch über UMTS wäre eine Datenverbindung möglich, ohne vom Backbone des Internets abhängig zu sein. Dazu müsste ein Gateway installiert werden, welches bei einem Ausfall der Verbindung zum Internet, wie es in Abbildung 3.13 zu erkennen ist, zwischengeschaltet wird.

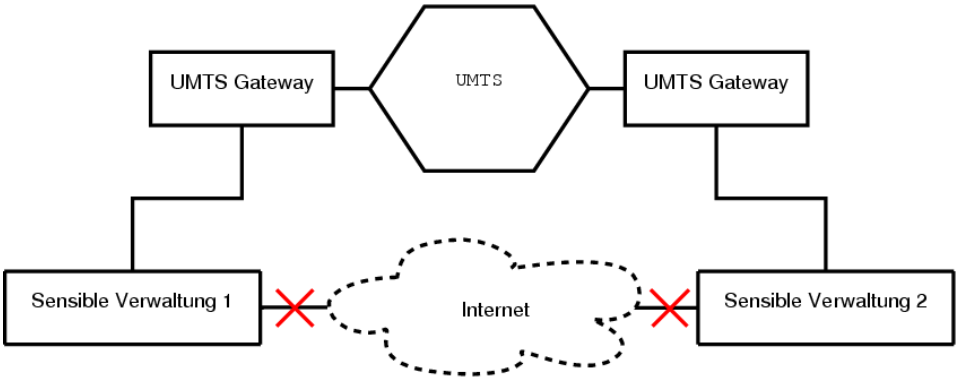

Abbildung 3.13: Alternativweg über UMTS

Auf diesem Wege könnten sensible Verwaltungsstellen, beispielsweise über UMTS, weiterhin miteinander kommunizieren, ohne den Weg über Internetrouter beschreiten zu müssen. So wäre ein Datenaustausch dieser Verwaltungen auch zum Zeitpunkt eines größeren Ausfalls des Internets möglich. Diese würden dann über ein UMTS-Gateway kommunizieren, welches die Schnittstelle zwischen UMTS und dem internen Netzwerk bildet.

Diese Variante ist jedoch für große Webseitenbetreiber keine wirkliche Alternative. Dies würde bedeuten, dass jeder einzelne Internetprovider ein solches UMTS Gateway einrichtet. Somit würde seinen Kunden ermöglicht, eine direkte Verbindung zu einer beliebigen, jedoch festen Webseite, die ebenfalls über ein UMTS-Gateway verfügen muss, aufzubauen. Diese wird dann bei Bedarf mit dem UMTS-Gateway des Providers verbunden und kann dann vom Kunden des Providers abgerufen werden.

Es ist klar zu erkennen, dass das Erreichen von Webseiten vieler verschiedener Anbieter, von jedem Endbenutzer unmöglich zu realisieren ist, da jede Verbindung zu einer Webseite auch das Herstellen einer neuen UMTS-Verbindung nach sich ziehen würde. Bei Millionen von Webseiten weltweit und oftmals tausenden Kunden pro Provider wäre der „Umweg“ über UMTS für einzelne Webseite praktisch nicht erreichbar.

Aus der Theorie ist erkennbar, dass es ein breites Spektrum an Gefahren gibt, die Netzwerke, also auch Kritische Infrastrukturen, bedrohen. Diese Bedrohungen möglichst gering zu halten, sowie die Auswirkungen eines Angriffs zu vermindern oder gar zu eliminie-

ren, ist Aufgabe der zukünftigen Forschung. Es müssen in Zukunft immer wieder Wege gefunden werden, die den Angreifern das Arbeiten zu erschweren oder es ihnen ganz und gar unmöglich zu machen, Infrastrukturen weltweit zu gefährden. Es sind schon einige Mechanismen bekannt und in der Anwendung, jedoch lassen sich findige „Bösewichte“ immer wieder neue Strategien einfallen, die dann zu neuen Bedrohungen werden. Im nun folgenen praktischen Teil wird ein möglicher Angriff eines Netzwerks und eine geeignete Gegenmaßnahme in einem Simulator implementiert, um die theoretisch erläuterten Probleme zu illustrieren und die daraus gewonnenen Daten zu bewerten.

Kapitel 4

Praktische Implementierung

Im vorhergehenden Kapitel wurden die theoretischen Grundlagen beleuchtet, die für den Angriff eines Netzwerkes von Bedeutung sind. Im Folgenden wird exemplarisch ein Angriffsszenario ausgewählt. Dieses wird in einem Netzwerksimulator visualisiert und untersucht. Dieser Angriff wird dabei mit einer geeigneten Maßnahme bekämpft.

4.1 Auswahl eines Szenarios

Der Grundgedanke bei der Implementierung ist die Sicherstellung eines Nachrichtenaustauschs während eines Netzwerkangriffs. Hierbei lässt sich eine Kritische Infrastruktur simulieren, deren Kommunikationssystem unter allen Umständen noch funktionieren sollte, wenn ein Angriff gestartet wird. Als Beispiel lässt sich hier ein Atomkraftwerk anführen, dessen Sensoren und Kontrollstationen über ein Netzwerk kommunizieren. Deren Kommunikation ist für die Aufrechterhaltung der Sicherheit eines Kraftwerks von immenser Bedeutung.

Als weiteres Beispiel sei hier die eMail-Kommunikation zwischen Banken, die beispielsweise auf diesem Weg Informationen über das aktuelle Börsengeschehen austauschen, angeführt. Ein Ausfall oder gar Manipulation dieser Kommunikation könnte möglicherweise zu finanziellen Schäden bei Anlegern aber auch bei der bankinternen Bilanz führen.

Die verschiedenen Kommunikationsvarianten können nun auf den vier verschiedenen Ebenen der Netzwerkstruktur angegriffen werden.

1. Physikalische Ebene, z.B. durch Lichtwellen
2. Netzwerkebene, z.B. IP
3. Transportebene, z.B. TCP
4. Applikationsebene, z.B. HTTP

Jede dieser Ebenen ist potentiell angreifbar. Im folgenden werden die einzelnen Ebenen nach möglichen Gefahrenpunkten hin untersucht, und Varianten erläutert, die den Nachrichtenaustausch trotz eines Angriffs dieser Ebene noch gewährleisten. Die Gewährleistung dieses Nachrichtenaustauschs wird im weiteren Verlauf so definiert, dass es bedeutet, dass die Kommunikation, wie sie vor dem Angriff stattgefunden hat, auch während bzw. nach dem Angriff weiterhin so funktioniert, als hätte kein Angriff stattgefunden.

Jede Ebene ist von der unter ihr liegenden Ebene abhängig. Das heißt, je weiter unten der Angriff im Ebenenmodell stattfindet, desto größer ist die Beeinträchtigung des gesamten Netzwerks.

Physikalische Ebene Ein Angriff auf unterster Ebene ist die wirkungsvollste, wenn auch nicht praktikabelste Variante. Ein solcher Angriff wäre beispielsweise das einfache Kappen eines Kabels oder auch die gewaltsame, physische Zerstörung eines Routers.

Gegenmaßnahmen für diese Art von Angriff sind ebenfalls physisch. So sind sensible Rechnersysteme in speziell gesicherten Räumen untergebracht, deren Zutritt nur autorisierten Personen möglich ist. Kabelstränge können zum einen verzweigt werden um einem *Single-Point-Of-Failure* zu umgehen, zum anderen aber auch in massive Kabelrohre eingebettet werden, um einen physischen Zugang zu erschweren.

Netzwerkebene Auf dieser Ebene befinden sich die Router. Dortige Angriffe können entweder direkt auf die Router zielen oder auf Routingprotokolle, mit deren Algorithmen das korrekte Weiterversenden von eingehenden Paketen berechnet wird.

Eine Maßnahme zur Verhinderung von Angriffen bzw. zur Gewährleistung des Routings liegt in der Verschlüsselung von Paketen, im speziellen in der Verschlüsselung von Routinginformationen. Durch eine solche Maßnahme wird es dem Angreifer erschwert, die enthaltenen Informationen auszulesen oder zu manipulieren.

Transportebene Ein Angriff auf der Transportebene ist beispielsweise SYN-Flood. Hierbei wird der Verbindungsstatus von TCP ausgenutzt um einen Server zu blockieren.

Eine Gegenmaßnahme wäre hierbei ein alternativer Transport über UDP, bei dem die Applikationsebene die zusätzlichen Funktionen von TCP nachbildet. Eine andere Möglichkeit wäre die Beschränkung der Anzahl von frei zu vergebenen TCP-Ports. So könnte eine Liste erstellt werden, die nur autorisierten Nutzern einen TCP-Port zur Verfügung stellt.

Applikationsebene Bei Angriffen auf Applikationsebene geht es im einzelnen um die Serverdienste. So werden immer wieder Sicherheitslücken in Serverapplikationen entdeckt, die dem Angreifer ermöglichen sensible Konfigurationen des Servers zu ändern oder auch diesen zum Absturz zu bringen.

Die bedeutendste Gegenmaßnahme ist die regelmäßige Installation von Sicherheitsupdates der Serverssoftware. Diese Updates schließen die vorhandenen, bekannten Sicherheitslücken. Im Weiteren könnte überlegt werden, die gesamte Serverapplikation redundant auszulegen, und eine zweite Serverapplikation mit Software eines anderen Entwicklerteams zu nehmen. In seltenen Fällen besitzen zwei verschiedenen Serverapplikationen die gleichen Sicherheitslücken. So könnte beim gezielten Angriff auf die Lücke des einen Servers der andere Server alternativ die Aufgaben des angegriffenen Servers übernehmen.

4.2 Der Simulator

Die gewählte Implementierung eines Angriffs auf ein Netzwerk wurde in einem Netzwerksimulator realisiert. Dieser Simulator ist eine Entwicklung des DAI-Labors[1] der TU-Berlin. Der Simulator dient der Entwicklung und des Testens von neuer Sicherheitssoftware. So können beispielsweise verschiedene Intrusion-Detection-Systeme (siehe Abschnitt 3.4.2.1) auf ihre Wirksamkeit überprüft werden.

Auf Basis von Eclipse[2] als Entwicklungsumgebung, der Programmiersprache Java und JIAC als ein auf Agententechnologien beruhendes Serviceware-Framework wurde der Simulator entwickelt. Eine grafische Benutzeroberfläche ermöglicht den Aufbau eines simulierten Netzwerkes ohne tiefer gehende Kenntnisse der zugrunde liegenden Architektur der Software. Beispielsweise können Web-Clients, Web-Server und Router platziert und über simulierte Netzwerklinks miteinander verbunden werden. Weitere Geräte, die bisher in den Simulator eingebunden worden sind: Mail-Server und Proxy-Server.

Bei der Realisierung der Netzwerkverbindungen wurde bewusst auf die Java-Netzwerksockets verzichtet. Alternativ wurden eigene Sockets programmiert, die auf der TCP/IP-Architektur aufbauen. So gibt es einen IP-Layer, der für das Erstellen von IP-Paketen sowie deren Fragmentierung zuständig ist. Es gibt darüber hinaus einen Network-Layer, der das Routing steuert und einen Transport-Layer, indem die Funktionalität von TCP und UDP abgebildet wird. Jedes Gerät im Netzwerk enthält demnach ein System aus mehreren Ebenen, die jede für sich spezielle Aufgaben erfüllen. Außerdem existiert auf den Endgeräten, also Servern und Clients, noch eine Applikationsebene, welche die Server- bzw. Client-Applikationen enthält.

[1] http://www.dai-labor.de
[2] http://www.eclipse.org

4.2.1 Relevante Geräte

Die für die Implementierung des Angriffsszenarios relevanten Geräte sind Web-Clients, Web-Server und Router.

Web-Client Der Web-Client ist in der Lage Webseiten von allen Web-Servern, die im gesamten Netzwerk vorhanden sind, anzufordern. Dazu wird auf der Applikationsebene des Clients ein HTTP-Request generiert und dort an die unteren Netzwerkebenen weitergegeben. Kommt auf dieses HTTP-Request ein HTTP-Response, also eine Antwort vom Web-Server, zurück, so wird diese im Browserfenster angezeigt.

Web-Server Der Web-Server wartet auf HTTP-Requests von Web-Clients. Erhält er ein solches Request, so wird ein Server-Thread gestartet, der dann wiederum die Antwort an den anfragenden Client zurücksendet. Dies geschieht konform zur Spezifikation des HTTP-Protokolls.

Router Die Router sind für die Verteilung der IP-Pakete im Netzwerk zuständig. Für das Erstellen seiner Routinginformationen nutzt der Router standardmäßig OSPF (*Open Shortest Path First*), ein Link-State Routing Protokoll. Wahlweise lässt sich hier auch ISIS (*Intermediate System to Intermediate System Protocol*) als Routingprotokoll einstellen.

Der einfache Aufbau eines Netzwerks, bestehend aus mehreren Web-Servern, Web-Cients und Routern, könnte dann so aussehen, wie z.B. in Abbildung 4.1.

Dies ist die Ausgangsimplemetierung des Netzwerksimulators NeSSi. Sie wird nun anhand eines ausgewählten, beispielhaften Angriffsszenarios erweitert. Dieser Angriff soll folgend mit einer Gegenmaßnahme abgeschwächt werden, so dass ein Nachrichtenaustausch weiterhin möglich ist. Diese Maßnahme wird ebenfalls in den Simulator integriert und dort getestet.

4.3 Implementierung

Wie in Abschnitt 3.2.2 schon erwähnt wurde, werden DDoS-Angriffe in der Regel über einen IRC gestartet. Dazu sind eine große Anzahl, beispielsweise über Trojaner infizierter Rechner über ein IRC-Netzwerk miteinander verbunden und können so kommunizieren. Der Angreifer hat dafür gesorgt, dass er nun viele Rechner weltweit ansteuern und ihnen Befehle erteilen kann. Sie wurden zu Bots. Es kann Befehle absetzen, die z.B. einen SYN-Flood auf ein bestimmtes Ziel veranlassen. Alle verbundenen Bots erhalten diesen Befehl und werden das, vom Angreifer gewünschten Ziel, mit SYN-Paketen „bombardieren“. Dieses Vorsehen führt dazu, dass viele TCP-Verbindungen geöffnet werden. Die

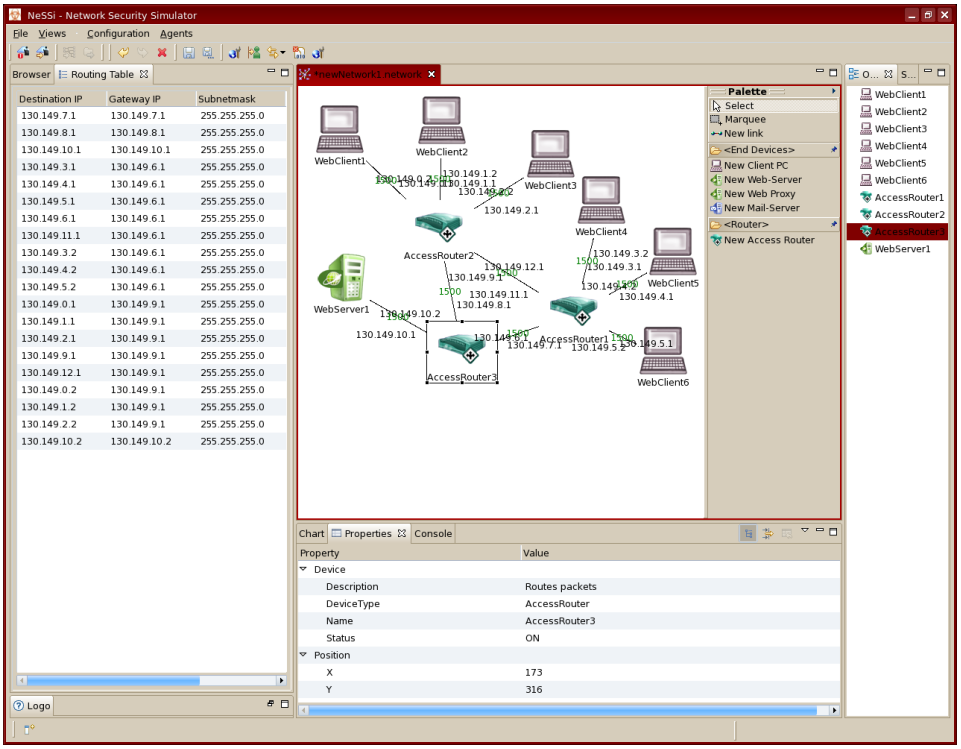

Abbildung 4.1: Beispielhafter Aufbau eines kleinen Netzwerks in NeSSi

Verbindungen sind jedoch nicht komplett geöffnet, da das Antwortpaket des Angreifenden ausbleibt (3-Wege-Handshake: siehe Abschnitt 3.2.1.1). Da das Betriebssystem nur begrenzt halboffene Verbindungen bereit stellt, ist es ab einem gewissen Zeitpunkt nicht mehr möglich, neue TCP-Verbindungen zu öffnen. Dies schließt auch die Versuche von Benutzern ein, die nicht Teil des Angriffs sind.

Hier muss die Gegenmaßnahme ansetzen. Während eines solchen DDoS-Angriffs, eines SYN-Floods, sollen trotzdem noch neue TCP-Verbindungen möglich sein. Hierzu muss für legitime neue Verbindungen eine Reserve zurückgehalten werden. Aus dieser Reserve werden nur dann neue Verbindungen entnommen, wenn die Anfrage von einer autorisierten Stelle kommt. Die IP-Adressen autorisierter Clients werden in einer *Whitelist* gespeichert, die sowohl statische als auch dynamische Elemente enthält. Ein bestimmter Anteil der maximal möglichen Anzahl an halboffenen Verbindungen wird für diese Whitelist reserviert.

4.3.1 IRC-Szenario

Für die Realisierung eines Angriffs mittels eines IRC-Netzwerks, sind weitere Geräte im Simulator zu entwickeln. Zum einen wird ein Angreifer benötigt, der die einzelnen

Bots mit Hilfe des IRC-Protokolls steuert und ihnen den Befehl zum SYN-Flood eines ausgewählten Ziels erteilt. Dieses Ziel könnte beispielsweise ein Web-Server sein.

Zum anderen wird für die Kommunikation der Bots und des Angreifers ein IRC-Server benötigt. Dieser IRC-Server ist als vermittelnde Einheit zwischen den Bots und dem Angreifer zu sehen. Alle Bots sind auf diesem IRC-Server eingewählt, und sind Teilnehmer desselben IRC-Kanals auf diesem Server.

Neben diesen neuen Endgeräten ist auch noch die Modifizierung der Web-Clients nötig. Damit diese Clients nicht nur in der Lage sind Webseiten von diversen Web-Servern abzurufen sondern auch SYN-Floods zu tätigen, ist dafür zu sorgen, dass sie als Bots in einem Bot-Netzwerk teilnehmen können. Diese Bots können sich auf einem beliebigen IRC-Server einwählen und dort in einem definierten IRC-Kanal auf Anweisungen warten.

4.3.1.1 IRC-Server

Zur Integration des IRC-Servers wurde eine bereits bestehende GPL-lizenzierte Implementierung namens *Sonata IRC Network*[3] genutzt. Es wurde ein neues Gerät erstellt, auf dem im Basiszustand ein Thread läuft, der diesen IRC-Dämon enthält. Als IRC-Dämon wird im weiteren Verlauf die verwendete Software des IRC-Servers bezeichnet. Die Implementierung dieses IRC-Dämons musste an die Socketimplementierung von NeSSi angepasst werden.

In der Version mit Java-Sockets wird im IRC-Dämon der Server-Socket erstellt. Dieser wartet dann auch eingehende Verbindungen. Gehen Anfragen auf neue TCP-Verbindungen ein, so wird ein Client-Socket erstellt, der dann wiederum einem neu erstellten Client-Thread übergeben wird. Dieser neu erstellte Thread kommuniziert über den übergeben Socket mit dem IRC-Client.

Mit der Socket-Implementierung von NeSSi muss dies im geringen Umfang ummodelliert werden. Hier wird bereits in der IRC-Server-Applikation, die den IRC-Dämon als Thread enthält der Server-Socket kreiert. Dieser wird dem IRC-Dämon übergeben. Eingehende Verbindungsanfragen auf dem erzeugten Server-Socket werden in der Transportebene behandelt. Dort wird ein Client-Socket erstellt, der an die Applikationsebene des IRC-Servers weitergegeben wird. Dort wird der erhaltenen Client-Socket wiederum an den IRC-Dämon übergeben, der dann intern einen neuen Client-Thread mit diesem erhaltenen Socket erstellt.

[3] http://sourceforge.net/projects/sonata/

4.3.1.2 Bots

Der Web-Client in der Ausgangsversion enthält Applikationen zum Browsen und zur Versendung von UDP-Paketen. Um nun die Kommunikation mit einem IRC-Server möglich zu machen muss auf dem Web-Client eine weitere Applikation laufen, die die Verbindung zum IRC-Server steuert. Diese Applikation heißt Bot-Applikation. Auch für diese Applikation wurde eine bereits existierende Implementierung eines IRC-Clients mit dem Namen PircBot[4] benutzt. Die Bot-Applikation des Web-Clients besitzt einen Thread, der diesen PircBot enthält.

Auch die Implementierung des PircBot musste für die eigene Socket-Implementierung von NeSSi umgestaltet werden. Sobald die Verbindung zu einem IRC-Server aufgebaut werden soll, wird auf der Transportebene des Web-Clients ein Client-Socket erzeugt. Dieser Socket kann von der Bot-Applikation benutzt werden. Folgend wird dieser erzeugte Socket an den PircBot weitergegeben. Innerhalb des PircBot ist die Erstellung eines Sockets demnach nicht nötig. Der erhaltene Socket kann somit zur Einwahl auf dem IRC-Server und zur weiteren Kommunikation mit diesem genutzt werden.

Um von der grafischen Benutzerschnittstelle in der Bot-Applikation eines Web-Clients die Einwahl auf einem vorhandenen IRC-Server zu erreichen, wird ein Event generiert, welches Daten zur IP-Adresse des IRC-Servers und dem IRC-Kanal enthält. Diese Daten wurden zuvor über einen Dialog ermittelt. Der generierte Event löst in der Bot-Applikation des Web-Clients die Ausführung einer bestimmten Methode aus. Diese Methode erzeugt dann den Thread, der mittels des PircBot und der übermittelten Daten eine TCP-Verbindung mit dem IRC-Server herstellt, und auf dem IRC-Server dem gewünschten IRC-Kanal beitritt.

Nach der erfolgreichen Einwahl ist der Web-Client nun ein Bot, der auf eingehende Textnachrichten wartet. Speziell codierte Textnachrichten, die vom Angreifer in dem IRC-Kanal gesendet werden, lösen in der Bot-Applikation bestimmte Reaktionen aus.

4.3.1.3 Angreifer

Der Angreifer enthält, genau wie der Web-Client, eine Bot-Applikation. Auch hier wird, ausgehend von der grafischen Oberfläche, ein Event generiert, welches die Bot-Applikation dazu veranlasst, einen Thread mit enthaltenem PircBot zu starten. Hier wird über einen Dialog der IRC-Server und IRC-Kanal gewählt. Zusätzlich wird ein Befehl erwartet, der allen Bots, die auf gewählten IRC-Kanal des gegebenen IRC-Servers auf Nachrichten warten, gesendet wird. Für die Ausführung eines SYN-Floods wird folgende Syntax als Textnachricht versendet:

[4] http://www.jibble.org/pircbot.php

```
synflood <ip1> <ip2> <count>
```

`ip1` ist hierbei die gespoofte Absenderadresse des SYN-Pakets, `ip2` die Zieladresse. `count` bezeichnet die Anzahl der zu sendenden SYN-Pakete pro Bot. Wird nun `synflood 130.149.5.2 130.149.11.2 1000` an Kanal X des IRC-Servers Y gesendet, so werden alle Bots, die auf Kanal X des IRC-Servers Y horchen, einen SYN-Flood auf den Rechner mit der IP `130.149.11.2` starten und sich als `130.149.5.2` ausgeben. Die Flutung besteht aus 1000 SYN-Paketen.

Ein solches SYN-Flood könnte nun auf einen Web-Server zielen. Der Aufbau eines einfachen Netzwerks ist in Abbildung 4.2 zu sehen. Teilnehmer des Netzwerks sind IRC-Server, Angreifer (Attacker), Bots (Web-Clients) und ein anzugreifender Web-Server.

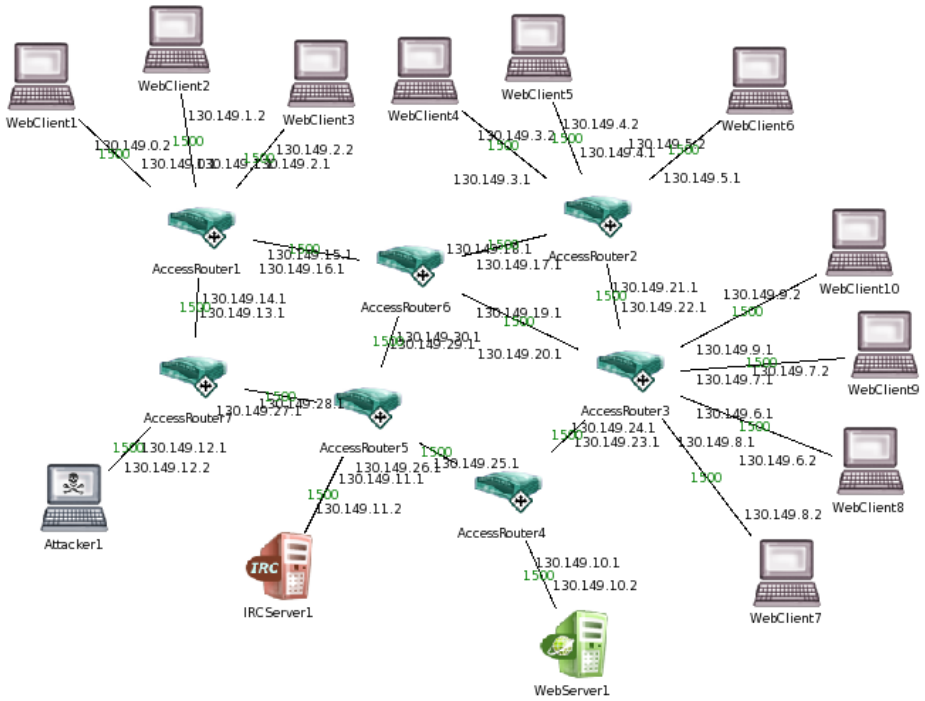

Abbildung 4.2: Aufbau eines Gesamtsystems im Angriffsszenario

4.3.2 Halboffene Verbindungen in TCP

Da in der bestehenden Implementierung von NeSSi die Limitierung von halboffenen TCP-Verbindungen noch nicht eingebaut ist, musste diese zunächst verwirklicht werden. Zu diesem Zweck wurde eine Klasse *HalfOpenedConnections* entwickelt, welche diese

halboffenen Verbindungen verwaltet. Sie enthält eine Liste von (Socket,Zeit)-Paaren, die mit Hilfe der enthaltenen Methoden aktualisiert werden. Jedes Element der Liste besteht demnach aus einem Socket und einem Zeitstempel, der für die spätere Aktualisierung der Liste notwendig ist.

Bei jeder eingehenden Anfrage einer TCP-Verbindung, also dem Erhalt eines SYN-Pakets wird der erzeugte Client-Socket in die Liste der halboffenen Verbindungen eingefügt. Beim Einfügen wird als Zeitstempel die aktuelle Zeit benutzt.

Beim Erhalt eines ACK-Pakets, also der Bestätigung der Verbindung ändert sich der Status der TCP-Verbindung von *halboffen* nach *offen*. Somit kann der Socket, dem dieses TCP-Verbindung zugeordnet wird aus der Liste der halboffenen Verbindungen entfernt werden.

Neben der Entfernung nach erfolgreichem Verbindungsaufbau gibt es noch die Möglichkeit eines *timeout*. Sollte nach einer definierten Zeit kein ACK-Paket zur Bestätigung der TCP-Verbindung eintreffen, so wird der Socket, dessen „Zeit abgelaufen ist“, zur Schonung von Ressourcen aus der Liste genommen. Die Überprüfung solcher *timeouts* geschieht direkt nach dem Eintreffen eines neuen SYN-Pakets. Dabei wird für alle Elemente in der Liste überprüft, ob die aktuelle Zeit vor oder nach der Zeit liegt, die sich ergibt, wenn Zeitstempel und timeout-Zeit addiert werden. Liegt die errechnete Zeit nach der aktuellen Zeit, so liegt ein timeout vor und das zugehörige Element wird aus der Liste gelöscht.

4.3.3 Whitelist-Prinzip

Mit der Implementierung der begrenzten Liste von halboffenen Verbindungen ist nun für die Angreifer die Möglichkeit gegeben, diese Begrenzung zu missbrauchen. Da nun nur begrenzt halboffene Verbindungen möglich sind, erreicht der Angreifer mit einem SYN-Flood den Effekt, dass legitime Anfragen bei voller Liste von halboffenen Verbindungen ignoriert werden.

Hier setzt die Whitelist an. Beispielsweise die Hälfte der begrenzten Anzahl an halboffenen Verbindungen wird nun nur noch für Anfragen benutzt, deren Quell-IP-Adresse in der Whitelist enthalten sind. Diese Whitelist enthält zwei verschiedene Listen. Eine statische Liste, welche IP-Adressen enthält, die für unbegrenzte Zeit priorisiert werden sollen. Zum zweiten gibt es eine dynamische Liste, deren Einträge laufend geändert werden. So wird eine IP-Adresse in diese dynamische Liste eingefügt, wenn sie die TCP-Verbindung regulär, also mit einem FIN-Paket, beendet. So wird erreicht, dass Angreifer, die nur SYN-Pakete senden, in diese Liste aufgenommen werden.

Wie auch bei der Liste der halboffenen Verbindungen bekommt jede IP-Adresse, die in die dynamische Liste eingefügt wird, einen Zeitstempel zugeordnet. Jeder Eintrag darf

nur für eine fest definierte Zeit in der dynamischen Liste geführt werden. Sie ist nach dem gleichen Prinzip timeout-gesteuert wie die Liste der halboffenen Verbindungen. Somit ist gewährleistet, dass die dynamische Liste nicht zu groß wird und irgendwann so viele IP-Adressen enthält, dass die gespooften Absender-IP-Adressen der SYN-Floods in zu großer Anzahl in der Liste vorhanden sind, und die Whitelist ihre Wirkung verliert.

Um diese Implementierung zu testen, wurden verschiedene Szenarien erstellt.

4.4 Test

Um die Funktionalität und Wirksamkeit des Whitelist-Ansatzes zu testen, wurde ein Netzwerkaufbau (Abbildung 4.3) konstruiert. In diesem Netzwerk existieren 30 Web-

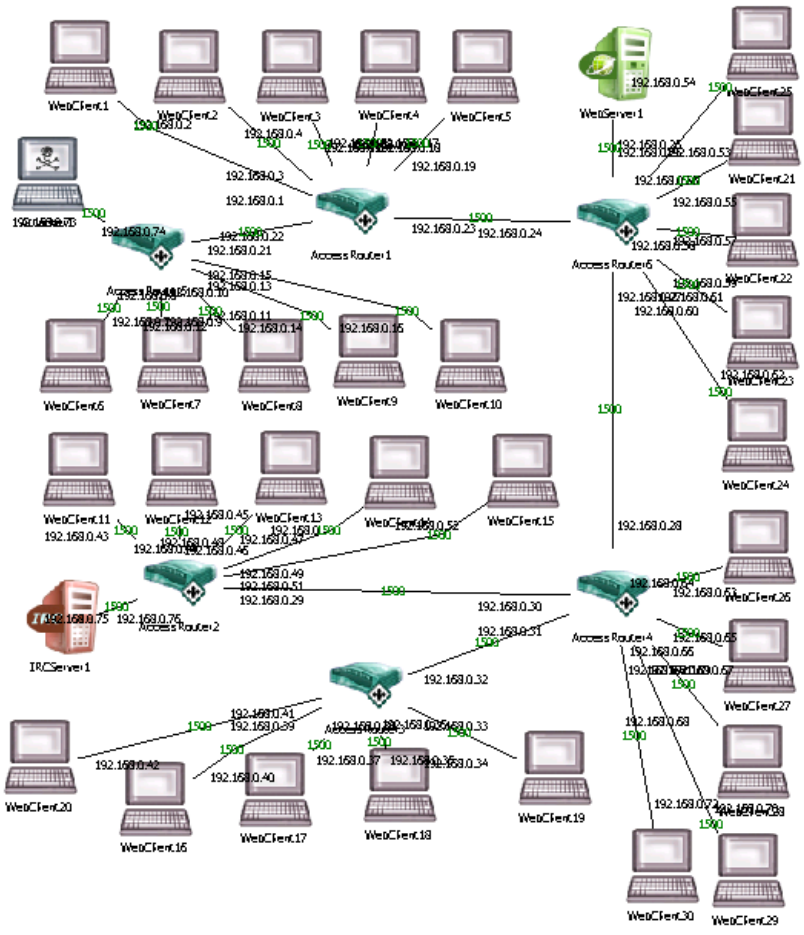

Abbildung 4.3: Testaufbau

Clients, ein Web-Server, ein IRC-Server und ein Angreifer. In den folgenden drei Szenarien wurden Daten gesammelt.

4.4.1 Szenario 1

Im ersten Szenario hat jeder der 20 Web-Clients auf der linken Seite alle 0,5 Sekunden einen Browse-Versuch unternommen. Jeder der Web-Clients kann vier Browse-Vorgänge gleichzeitig durchführen. Die Liste der halboffenen TCP-Verbindungen auf dem Web-Server hat eine Länge von 100.

4.4.2 Szenario 2

Im zweiten Szenario tätigte wiederum jeder der 20 Web-Clients Browse-Versuche. Hierbei wurde nun ein DDoS Angriff auf den Web-Server unternommen. Die 10 Web-Clients, die sich rechts befinden, registrierten sich dazu beim IRC-Server. Der Angreifer registrierte sich ebenfalls beim IRC-Server und gab den Befehl des SYN-Flood auf den Web-Server. Alle 0,2 Sekunden wurde von jedem Web-Client ein SYN-Paket an den Web-Server gesendet.

4.4.3 Szenario 3

Auch im dritten Szenario tätigte jeder der 20 Web-Clients Browse-Versuche. Auch hier wurde ein DDoS Angriff auf den Web-Server unternommen. Weiterhin wurde auf dem Web-Server das Whitelist-Prinzip aktiviert. Jede erfolgreicher Verbindungsaufbau führte dazu, dass der, dieser Verbindung zugeordnete Client, in die Whitelist aufgenommen wurde.

4.4.4 Auswertung

In jedem Szenario wurden Daten in einer Datei gespeichert. Sowohl die versuchten, als auch die erfolgreich aufgebauten Verbindungen wurden registriert. Die Daten wurden in ein Diagramm, wie in Abbildung 4.4, exportiert. Der Web-Server hat eine Beschränkung in der Anzahl der halboffenen Verbindungen. Es sind im Zustand ohne Whitelist-Prinzip 100 halboffene Verbindungen möglich. Jede weitere Verbindungsanfrage wird verworfen.

Es ist zu erkennen, dass die Kurve des ersten Szenarios („Ohne DDoS“) kontinuierlich und annähernd linear steigt. Dies war aufgrund des Testaufbaus auch so zu erwarten.

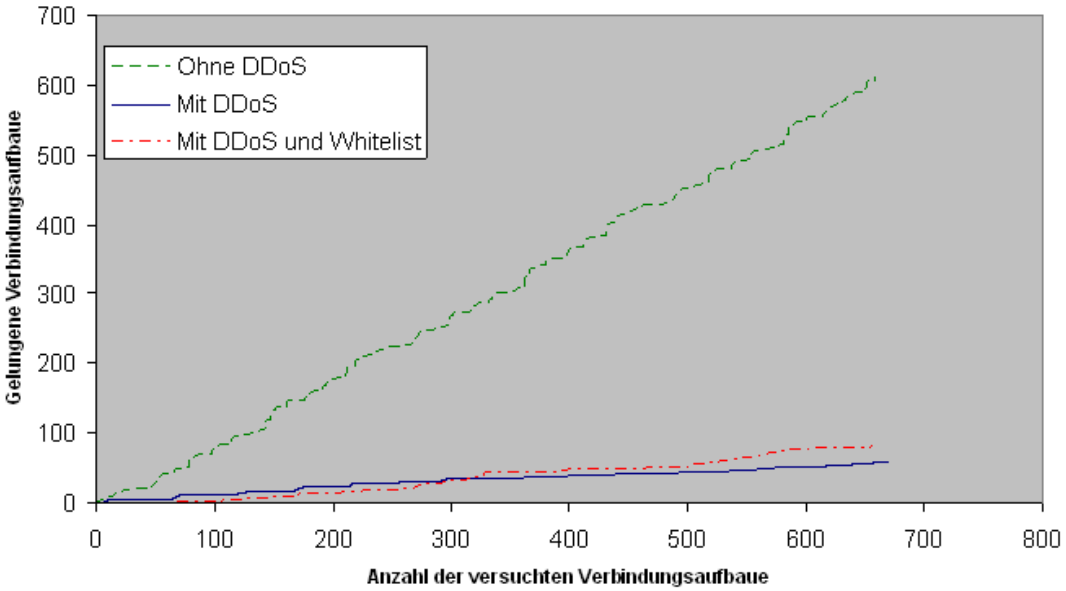

Abbildung 4.4: Auswertungsdiagramm

Da alle 20 Web-Clients insgesamt maximal 80 Browse-Anfragen stellen können, wurde die maximale Anzahl an halboffenen Verbindungen des Web-Servers nicht erreicht. Daher wird jede Anfrage akzeptiert. Es ist anzumerken, dass die Kurve aufgrund eines geringeren Datenvolumens künstlich verlängert wurde. Dies dient der Vergleichbarkeit der Kurven. Es ist möglich, da die Kurve linear verläuft.

Beim zweiten Szenario („Mit DDoS“) werden wesentlich weniger Verbindungen zugelassen. Dies resultiert aus dem SYN-Flood. Dieser bewirkt eine ständige Überfüllung der Liste an halboffenen Verbindungen im Web-Server. Nach einem Timeout von 30 Sekunden wird jede Verbindungsanfrage verworfen. Es besteht daher immer wieder die Möglichkeit, dass nicht jede legitime Verbindungsanfrage verworfen wird. Legitim sind hierbei die Browse-Anfragen der 20 Web-Clients. Die Anzahl der gelungenen Verbindungsaufbaue fällt mit 10% jedoch sehr gering aus.

Die Kurve des dritten Szenarios („Mit DDoS und Whitelist“) verläuft ähnlich wie die des zweiten Szenarios. Es ist jedoch zu sehen, dass ab einer Anzahl von etwa 300 Anfragen die Kurve eine höhere Steigung aufweist. Dies ist dadurch zu erklären, dass die aktivierte Whitelist den Aufbau von legitimen neuen Verbindungen erleichtert. Der Web-Server reserviert 50 Plätze auf der Liste der halboffenen Verbindungen für Mitglieder der Whitelist. Diese Plätze sind vom SYN-Flood nicht betroffen. Mit fortlaufender Zeit werden immer mehr legitime Web-Clients Mitglieder der Whitelist. Somit steigt die Wahrscheinlichkeit eines erfolgreichen Verbindungsaufbaus.

Insgesamt ist zu erkennen, dass der Whitelist-Ansatz eine Lösung anbietet. Die Wirkung ist jedoch gering. Weitere Bewertungen und Ausblicke werden daher in dem folgenden Kapitel erörtert.

Kapitel 5

Fazit

Nachrichtenaustausch unter Angriff. Was Nachrichtenaustausch in Netzwerken bedeutet, und wie wichtig er für die Gesellschaft ist, wurde in dieser Arbeit herausgestellt. Der Austausch von Nachrichten ist immer häufiger Bedrohungen ausgesetzt. Diese Bedrohungen zu erkennen und zu bekämpfen ist somit sehr wichtig geworden. In dieser Arbeit wurde eine Angriffsvariante in einem Simulator implementiert. Diesem Angriff wurde mit einer Gegenmaßnahme entgegengewirkt, welche hier als Whitelist-Prinzip bezeichnet wurde.

Aus dem Diagramm (Abbildung 4.4) ist die Wirksamkeit des Whitelist-Ansatzes zu erkennen. Mit steigender Versuchsdauer erhöhte sich die Wahrscheinlichkeit eines erfolgreichen Verbindungsaufbaus, deren Wirkung jedoch nur sehr gering ist. Es ist zu erwarten, dass in einem länger andauernden Szenario, die Kurve „Mit DDoS und Whitelist“ noch weiter an Steigung zunimmt. Dies liegt in der Aufnahme von weiteren Web-Clients in die Whitelist begründet.

Während eines stattfindenden SYN-Flood ist die Aufnahme in die Whitelist nur schwer möglich. Ein Timeout verkleinert die Liste der halboffenen Verbindungen um jeweils ein Element. Nur in diesem Fall besteht Möglichkeit, dass eine Anfrage nicht verworfen wird. Da neben den Anfragen der Web-Clients aber weiterhin noch SYN-Pakete von Bots gesendet werden, ist die Wahrscheinlichkeit sehr gering, dass der Anfrage des Web-Clients genau dieser frei gewordene Platz zugeteilt wird. Nur wenn das SYN-Paket des legitimen Web-Clients nicht verworfen wird, besteht die Möglichkeit, dass die Verbindung zustande kommt. Einzig in diesem Fall wird der Web-Client in die Whitelist aufgenommen.

In diesem Punkt zeigt das Whitelist-Prinzip einen Schwachpunkt. Der Angreifer könnte beispielsweise die Methodik der Aufnahme in die dynamische Whitelist erraten oder durch Probieren heraus finden.

Bei dem hier implementierten Ansatz erfolgt die Aufnahme nach dem Senden eines ACK-Pakets. Die Bots des Angreifers könnten ein solches Paket aber auch senden. Damit würden sie in die Whitelist aufgenommen. Dies hätte die Unwirksamkeit der Whitelist zur

Folge. So könnten andere Algorithmen die Aufnahme in die Whitelist steuern. Beispielsweise könnte nach dem FIN-Paket, welches zur Beendingung einer TCP-Verbindung gesendet wird, anstelle des ACK-Pakets, die Aufnahme in die Whitelist erfolgen. Um das Raten der Aufnahmemethode weiter zu erschweren, könnten verschiedene Methoden abwechselnd genutzt werden.

Diese Whitelist-Methode ist nur eine von vielen Methoden zur Abwehr von Angriffen bzw. zur Begrenzung von Schäden durch Angriffe. Weitere Methoden zur Abwehr von Angriffen auf die verschiedenen Ebenen der Netzwerkarchitektur müssen entwickelt werden. Nur so kann, bei wachsender Nutzung von Netzwerken in wirtschaftlichen, politischen und gesellschaftlichen Bereichen, der Austausch von Nachrichten weiterhin sichergestellt werden. Hauptmerkmale der Sicherstellung sollten Verfügbarkeit, Integrität und Vertraulichkeit sein.

Verfügbarkeit Nachrichtenaustausch sollte zu jeder Zeit funktionieren.

Integrität Die versendeten Daten sollten unverändert beim Empfänger ankommen.

Vertraulichkeit Das Mitlesen von Informationen sollte nicht möglich sein.

Mit Hilfe von NeSSi könnten sich weitere Strategien zur Abwehr von Angriffen entwickeln lassen. Das Einbinden des IRC-Protokolls als Kommunikationsmedium für Angreifer und Bots hat dem Simulator zusätzliche Flexibilität verliehen. Auf dieser Grundlage bestände die Möglichkeit, weitere Gegenmaßnahmen zur DDoS-Abwehr zu entwickeln. Zu diesem Zweck können weitere Angriffsszenarien implementiert werden. Die Maßnahmen zur Abwehr dieser Angriffe könnten wiederum im Simulator getestet und bewertet werden. So eröffnet die hier implementierte Kommunikation über IRC Perspektiven zukünftiger Forschungsanstrengungen.

Literaturverzeichnis

[1] 3GPP. *High Speed Circuit Switched Data (HSCSD)*. http://www.3gpp.org/ftp/Specs/html-info/0234.htm. 25.06.1999.

[2] 3GPP. *Technical realization of the Short Message Service (SMS)*. http://www.3gpp.org/ftp/Specs/html-info/0340.htm. 11.01.2002.

[3] 3GPP. *UMTS Phase 1*. http://www.3gpp.org/ftp/Specs/html-info/22100.htm. 05.10.2001.

[4] D. Anderson. *Mayday: Distributed Filtering for Internet Services*. 3rd Usenix USITS, 2003.

[5] M.M. Breuning, Kriegel H.-P., R.T. Ng, and J. Sander. Lof: Identifying density-based local outliers. *Proceedings of the ACM SIGMOD Conference*, 2000.

[6] BSI. *Virenwarnung: W32.Novarg.A@mm*. http://www.bsi.de/av/vb/novarg.htm. 26.01.2004.

[7] Statistisches Bundesamt. *Informationstechnologie in Unternehmen und Haushalten 2005*. http://destatis.de/download/d/veroe/Pressebroschuere_IKT2005.pdf. Stand: 19.07.2006.

[8] V. Cerf and R. Kahn. A protokoll for packet network interconnection. *IEEE Transactions on Communications*, 1974.

[9] Cisco. *Neue Infrastruktur für Service Provider, die 'any over IP' ermöglicht - Pressemitteilung 01.05.2001*. http://www.cisco.com/global/AT/presse/archiv/pressemitteilungen/ar_home_s124.shtml.

[10] CNNMoney. *More sites hacked in wake of Yahoo!* http://money.cnn.com/2000/02/08/technology/yahoo/. 08.02.2000.

[11] ICPC International Cable Protection Committee. *Cable Database*. http://www.iscpc.org/cabledb/cabledb.htm. Stand: 02.08.2006.

[12] Symantec Corp. *Bericht zu Bedrohungen aus dem Internet, Band VIII - 2005*. http://enterprisesecurity.symantec.de/Content/displaypdf.cfm?PDFID=2221&EID=0. Stand: 28.06.2006.

[13] Symantec Corp. *W32.Mydoom.A@mm - Warnung*. http://www.symantec.com/region/de/techsupp/avcenter/venc/data/de-w32.mydoom.a@mm.html. 27.01.2004.

[14] Dai Davies. *Report on present status of international connectivity in Europe and to other continents.* http://www.serenate.org/publications/d6-serenate.pdf. 11.02.2003.

[15] DE-CIX. http://www.de-cix.net/info/connected.html. Stand: 02.08.2006.

[16] EUQUINIX. https://ecc.equinix.com/.

[17] Amsterdam Internet Exchange. http://www.ams-ix.net. Stand: 24.11.2006.

[18] P. Ferguson and D. Senie. *Network Ingress Filtering: Defeating Dinial of Service Attacks that Employ IP Source Address Spoofing.* 2000.

[19] Security Focus. *Cisco SNMP configuration attack with a GRE tunnel.* http://www.securityfocus.com/infocus/1847. 16.09.2005.

[20] Security Focus. *Exploiting Cisco Routers: Part 1.* http://www.securityfocus.com/infocus/1734. 29.09.2003.

[21] Bundesministerium für Sicherheit in der Informationstechnik (BSI). *KRITIS - Einführung.* http://www.bsi.bund.de/fachthem/kritis/KRITIS_Einfuehrung.pdf. Stand: 19.04.2006.

[22] R. Heady, G. Luger, A. Maccabe, and M. Servilla. *The architecture of a network level intrusion detection.* Computer Science Department, 1990. University of New Mexico.

[23] heise.de News. *Internet steckte DE-CIX-Ausfall gut weg.* http://www.heise.de/newsticker/meldung/36192. 16.04.2003.

[24] IETF. *Domain Names - Concepts and Facilities.* http://www.ietf.org/rfc/rfc1034.txt.

[25] IETF. *Domain Names - Implementation and Specification.* http://www.ietf.org/rfc/rfc1035.txt.

[26] IETF. *RFC 2616 - Hypertext Transport Protocol v1.1.* http://www.ietf.org/rfc/rfc2616.txt.

[27] IETF. *RFC 2827 - Network Ingress Filtering.* http://www.ietf.org/rfc/rfc2827.txt.

[28] IETF. *RFC 4251 - Secure Shell Protocol.* http://www.ietf.org/rfc/rfc4251.txt.

[29] IETF. *RFC 792 - Internet Control Management Protocol.* http://www.ietf.org/rfc/rfc792.txt.

[30] IETF. *RFC 959 - File Transport Protocol.* http://www.ietf.org/rfc/rfc0959.txt.

[31] Infratest. *8. Faktenbericht - Juni 2005 - Abbildung 186, Seite 190.* http://www.tns-infratest.com/06_BI/bmwa/Faktenbericht_8/06480_index_bmwa.asp. Stand: 19.07.2006.

[32] Infratest. *8. Faktenbericht - Juni 2005 - Abbildung 215, Seite 211.* http://www.tns-infratest.com/06_BI/bmwa/Faktenbericht_8/06480_index_bmwa.asp. Stand: 19.07.2006.

[33] Infratest. *8. Faktenbericht - Juni 2005 - Abbildung 216, Seite 212.* http://www.tns-infratest.com/06_BI/bmwa/Faktenbericht_8/06480_index_bmwa.asp. Stand: 19.07.2006.

[34] Infratest. *8. Faktenbericht - Juni 2005 - Abbildung 217, Seite 215.* http://www.tns-infratest.com/06_BI/bmwa/Faktenbericht_8/06480_index_bmwa.asp. Stand: 19.07.2006.

[35] J. Ioannidis and S. Bellovin. *Implementing Pushback: Router-Based Defense Against DoS Attacks.* NDSS, 2002.

[36] Angelos D. Keromytis, Vishal Misra, and Dan Rubenstein. *SOS: An Architecture for Mitigating DDoS Attacks.* IEEE Jounal on Selected Areas in Communications, Special Issue on Service Overlay Networks, 2003.

[37] Lilia Lajmi. *Paketsubstitution in Audiosignalen bei paketorientierter Audioübertragung.* http://ftsu01.nue.tu-berlin.de/elvera/files/0937Lajmi2003.pdf, 2003.

[38] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. *A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection.* Computer Science Department, 2003. University of Minnesota.

[39] Felix Lindner. *Route 666 - c't 05/04, S. 210.* http://www.heise.de/security/artikel/44824/0. Stand: 28.07.2006.

[40] LINX. http://www.linx.net. Stand: 02.08.2006.

[41] The Local. *Hackers crash Swedish government web site.* http://www.thelocal.se/article.php?ID=3983&date=20060604. 04.06.2006.

[42] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxxon, and S. Shenker. *Controlling High Bandwidth Aggregates in the Network.* Computer Communications Review. 2002.

[43] Internet Security News. *Dutch government sites attacked.* http://www.landfield.com/isn/mail-archive/2004/Oct/0003.html. 06.10.2004.

[44] news.com. *How a basic attack crippled Yahoo.* http://news.com.com/2100-1023-236621.html. 07.02.2000.

[45] news.com. *Leading Web sites under attack.* http://news.com.com/2100-1017-236683.html. 09.02.2000.

[46] NYIIX. http://www.nyixx.net. Stand: 02.08.2006.

[47] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. *Practical Network Support for IP Traceback*. ACM SIGCOMM, 2000.

[48] D. Song and A. Perrig. *Advance and Authenticated Marking Schemes for IP Traceback*. IEEE Infocom, 2001.

[49] SOPHOS. *Top Ten der im Januar 2004 an Sophos gemeldeten Viren und HoaxesTop Ten der im Januar 2004 an Sophos gemeldeten Viren und Hoaxes*. http://www.sophos.de/pressoffice/news/articles/2004/01/pr_20040130topten.html. 30.01.2004.

[50] Joe Stewart. *DNS Cache Poisoning - The Next Generation*. http://www.lurhq.com/dnscache.pdf. Stand: 25.07.2006.

[51] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall PTR, 1996.

[52] Randal Vaughn and Gadi Evron. *DNS Amplification Attacks*. http://www.isotf.org/news/DNS-Amplification-Attacks.pdf. 17.03.2006.

[53] Wikipedia. *MyDoom*. http://de.wikipedia.org/wiki/Mydoom. 09.08.2006.

[54] Xiaowei Yang, David Wetherall, and Thomas Anderson. *A DoS-limiting Network Architecture*. ACM SIGCOMM, 2005.

[55] zdnet.com News. *Hack attack knocks out FBI site*. http://news.zdnet.com/2100-9595_22-514742.html. 26.05.1999.

Erklärung der Urheberschaft

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum Unterschrift