INSTITUT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Diplomarbeit

Sicherheitskonzept für den BMW-Internet-Zugang

Rainer Hauck

Aufgabensteller:	Prof. Dr. Heinz-Gerd Hegering
Betreuer:	Herr Trapa, BMW AG
	Thomas Paintmayer
	Dr. Bernhard Neumair
Abgabedatum:	15. August 1995

Ich versichere, daß ich diese Diplomarbeit selbständig verfaßt und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, 14. August 1995

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Thememstellung

Im Zuge der zunehmenden kommerziellen Nutzung des Internet plant die BMW AG, den bereits vorhandenen Internet-Zugang erheblich auszuweiten. Es sollen ausgewählte Dienste (z. B. *e-mail, netnews*) unternehmensweit zur Verfügung stehen. Dies birgt aber große Risiken, da durch den Anschluß an das Internet umgekehrt auch der Zugang aus dem Internet in das Unternehmensnetz möglich wird. Damit hieraus keine Angriffe resultieren können, soll ein sogenannter Firewall eingesetzt werden, d. h. ein Programm, das den Anschluß dahingehend überwacht, daß nur die Benutzer die Internet-Dienste nutzen können, die durch die Sicherheitspolitik des Unternehmens dazu autorisiert sind.

Da inzwischen eine relativ große Auswahl an Firewalls existiert, ist es erforderlich, die verschiedenen Lösungen einander gegenüberzustellen, um so die für die BMW AG geeigneteste Lösung auswählen zu können.

1.2 Ziele der Arbeit

Die wichtigsten Ziele dieser Diplomarbeit sind im einzelnen die folgenden:

- Untersuchung der wichtigsten Internet-Dienste und Darstellung ihrer Schwachstellen
- Erstellung eines Kriterienkataloges, der den Vergleich unterschiedlicher Firewalls ermöglicht
- Auswahl des geeignetesten Firewalls für den Einsatz zur Sicherung des Internet-Zugangs der BMW AG

1.3 Aufbau der Arbeit

Die hier vorliegende Arbeit ist folgendermaßen aufgebaut:

- Kapitel 2: Erstellen einer Sicherheitspolitik

  In diesem Kapitel wird zuerst eine Untersuchung des TCP/IP-Protokollstacks durchgeführt. Kapitel 2.1 erläutert zunächst die wichtigsten Protokolle der Schichten 3 und 4, auf denen alle anderen Protokolle aufsetzen. Die eigentlichen Anwendungsprotokolle werden dann in Kapitel 2.2 untersucht und beschrieben. Zum Abschluß des Kapitels 2 wird schließlich eine Sicherheitspolitik aufgestellt, wie sie aufgrund der bereits durchgeführten Untersuchungen aussehen könnte.

- Kapitel 3: Erstellung eines Kriterienkataloges

  Hier wird ein Kriterienkatalog aufgestellt, der den Vergleich unterschiedlicher Firewalls ermöglichen soll. In Kapitel 3.1 werden Kriterien aus den Definitionen der verschiedenen Internet-Dienste hergeleitet, während in Kapitel 3.2 die Herleitung der Kriterien aus den verschiedenen Angriffsmöglichkeiten erfolgt, die sich einem Angreifer bieten können. Die Erstellung des Kriterienkataloges wird in Kapitel 3.3 abgeschlossen, das Kriterien enthält, die keinem der oben erwähnten Kapitel zugeordnet werden konnten.

- Kapitel 4: Vorstellung allgemeiner Firewall-Konzepte

  Es existieren drei große Konzepte, wie ein Firewall realisiert werden kann. Diese sind in diesem Kapitel sowohl beschrieben als auch nach den Kriterien, die in Kapitel 3 aufgestellt wurden, bewertet. Kapitel 4.2 behandelt das Konzept des Paket-Filters, Kapitel 4.3 das TCP-Relay und Kapitel 4.4 befaßt sich mit dem Proxy-Konzept.

- Kapitel 5: Vorstellung existierender Firewall-Lösungen

  In diesem Kapitel werden nun existierende Firewall-Lösungen vorgestellt und ebenfalls mit Hilfe des erstellten Kriterienkataloges bewertet. Im einzelnen handelt es sich um den IBM-Firewall NetSP, der in Kapitel 5.1 behandelt wird, den Sun-Firewall FW-I in Kapitel 5.2 sowie den DEC-Firewall SEAL, um den es in Kapitel 5.4 geht. Als letzter Firewall wird dann noch in Kapitel 5.3 der Firewall-Toolkit der Firma TIS vorgestellt.

- Kapitel 6: Auswahl einer geeigneten Lösung für die BMW AG

  Nachdem die Untersuchung der unterschiedlichen Firewalls abgeschlossen ist, wird eine Lösung ausgewählt, die für den Einsatz bei der BMW AG am geeignetesten erscheint. Hierbei wird die Auswahl der Lösung beschrieben sowie Hinweise gegeben (Kapitel 6.2), wie dieser Firewall am günstigsten konfiguriert werden sollte.

- Anhang A: Authentifikationmechanismen

  Dieses Kapitel stellt die derzeit gebräuchlichsten Methoden zur Authentifizierung eines Benutzers vor. Drei Produkte, die von den meisten derzeit erhältlichen Firewalls unterstützt werden, werden näher betrachtet.

- Anhang B: Vorgehen bei der Auswahl eines Firewalls

  Dieses Kapitel dient dazu, zu beschreiben, was bei der Auswahl eines Firewalls zu beachten ist. In Anhang B.1 wird beschrieben, auf welche Kriterien besonderer Wert gelegt werden sollte, wenn man sich zwischen verschiedenen Lösungen entscheiden möchte, während Anhang B.2 allgemeine Hinweise gibt, was bei der Konfiguration eines beliebigen Firewalls zu beachten ist.

Kapitel 2

Erstellen einer Sicherheitspolitik

Die Anforderungen an die bereitzustellenden Dienste sowie an die Sicherheit der gespeicherten Daten unterscheiden sich von Organisation zu Organisation. Beispielsweise wird eine militärische Organisation deutlich stärkeres Interesse an der Geheimhaltung ihrer Daten haben als eine Universität. Aus diesem Grund ist der erste Schritt bei der Errichtung eines Internet-Zugangs die Erstellung einer Sicherheitspolitik, die festlegt, welche Dienste angeboten werden sollen bzw. welche Dienste ein zu großes Risiko darstellen würden. Die allgemeine Vorgehensweise hierbei ist die folgende:

- Bedarfsanalyse

  Im Rahmen einer Bedarfsanalyse muß festgestellt werden, welche Anforderungen an das Dienstangebot von Seiten der Benutzer vorliegen. Insbesondere muß geklärt werden, welche Dienste benötigt werden sowie auf welche Dienste verzichtet werden kann.

- Risikoanalyse

  Die Dienste, die bei der Bedarfsanalyse als notwendig eingestuft wurden, müssen nun daraufhin untersucht werden, ob sie Möglichkeiten bereitstellen, die es einem Angreifer ermöglichen, nicht-autorisierten Zugang zu Unternehmensdaten zu erlangen, diese zu verändern oder den reibungslosen Netzverkehr zu behindern.

- Aufstellen der Sicherheitspolitik

  Nachdem die Anforderungen sowie die daraus resultierenden Risiken bekannt sind, muß ein Kompromiß gefunden werden, der die Benutzer nicht zu sehr einschränkt, die Sicherheit des Netzes aber ebenso berücksichtigt. Oberster Grundsatz hierbei sollte immer sein: "Alles was nicht ausdrücklich erlaubt ist, ist verboten."

2.1 Die Protokolle der Netzwerk- und Transportschicht

Alle Standard-Dienste, die im Internet eingesetzt werden, verwenden auf Schicht 3 des OSI-Referenzmodells (Netzwerkebene) das *Internet Protocol* (IP) und auf Schicht 4 (Transportebene) entweder das *Transmission Control Protocol* (TCP) oder das *User Datagram Protocol*

(UDP). Dies erklärt auch den weitverbreiteten Namen für die Gesamtheit aller im Internet verwendeten Dienste, TCP/IP. Da viele Sicherheitsprobleme in den Netzwerk– und Transportprotokollen begründet liegen, werden diese im folgenden kurz vorgestellt [Lynch93], bevor die darauf aufsetzenden Dienste beschrieben werden.

2.1.1 Das Internet Protocol (IP)

Kern des TCP/IP-Stacks ist das auf OSI-Schicht 3 anzusiedelnde *Internet Protocol* (IP) [RFC791], auf dem praktisch alle anderen Protokolle aufsetzen. Es stellt einen ungesicherten, verbindungslosen Datagrammdienst zur Verfügung, mit dessen Hilfe Daten von einem Quellrechner zu einem Zielrechner transportiert werden können. Unter einem Datagramm versteht man eine bestimmte Anzahl von Bits, die in zwei Teile, den Header sowie die Nutzdaten, unterteilt werden können. Während die Nutzdaten die tatsächlich zu übertragenden Daten sind, enthält der Header Daten, die benötigt werden, damit das Paket sein Ziel erreichen kann, wie z.B. die Adresse des Zielrechners sowie des Absenders.

Der Transport des Paketes erfolgt entweder direkt von Host zu Host oder über einen oder mehrere Router. Die Router entscheiden aufgrund der Zieladresse des Paketes sowie sogenannter Routing-Tabellen, ob das Paket direkt an den Zielrechner ausgeliefert werden kann oder an einen weiteren Router weitergeleitet werden muß. Dies geschieht solange, bis das Paket schließlich den Zielhost erreicht hat.

Die Header-Felder

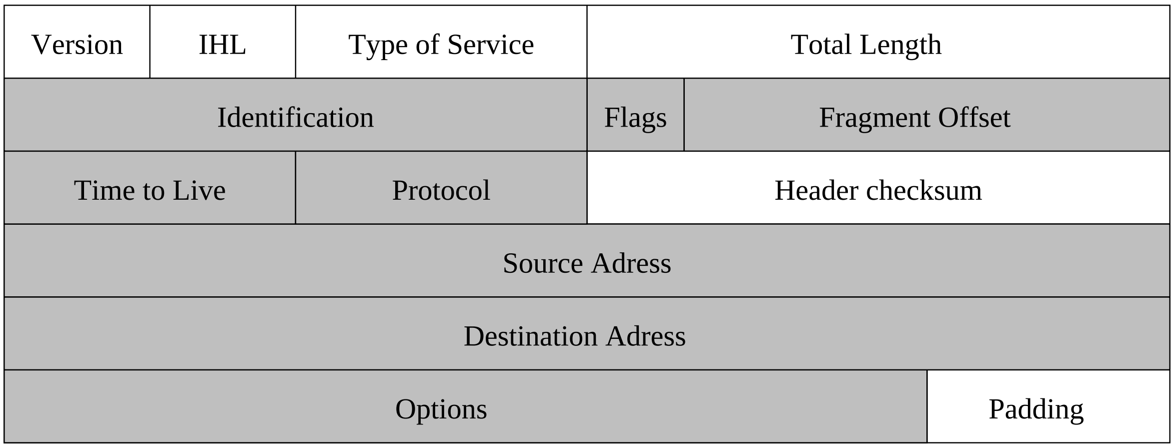

Abbildung 2.1: IP-Header

Abbildung 2.1.1 stellt den IP-Header dar, wobei die aus Sicherheitssicht relevanten Felder grau unterlegt sind. Diese sollen nun näher beschrieben werden.

- Source Adress / Destination Adress

  Es handelt sich um 32-Bit-Adressen, die den Sender bzw. den Empfänger weltweit eindeutig identifizieren.

- Identification / Flags / Fragment Offset

  Diese drei Felder dienen zur Fragmentierung von Paketen, wenn die Übertragung zum nächsten Router bzw. zum Zielrechner ein Paket der ursprünglichen Größe nicht zulassen würde. Die Zusammensetzung der einzelnenen Fragmente erfolgt erst im Zielrechner. Das *Flags*-Feld zeigt an, ob eine derartige Zerlegung für das jeweilige Paket zulässig ist, bzw. wenn bereits eine Fragmentierung stattgefunden hat, ob das entsprechende Teilpaket das letzte dieser Zerlegung ist. Das *Identification*-Feld enthält eine Nummer, die für die einzelnen Fragmente eines Paketes gleich ist, um ein späteres Zusammensetzen des Paketes zu ermöglichen, während das *Fragment Offset*-Feld die Position eines Teilpaketes innerhalb des ursprünglichen Paketes anzeigt. Dies ist erforderlich, da IP nicht garantiert, daß die Pakete in der Reihenfolge des Abschickens auch empfangen werden.

- Protocol

  Das *Protocol*-Feld gibt an, welches Protokoll der Transportebene (z.B. TCP / UDP) in diesem Paket transportiert wird.

- Time to Live (TTL)

  Es handelt sich um einen 8-Bit Zähler, der angibt wieviele Router ein Paket auf seinem Weg von der Quelle zum Ziel durchlaufen darf. Er wird verwendet, um bei Fehlern in den Routingtabellen ein endloses Zirkulieren eines Paketes auf dem Netz zu verhindern. Jeder Router, der das Paket weiterleitet, verringert den Wert dieses Feldes um eins und überprüft, ob der Wert null ereicht wurde. Sollte dies der Fall sein, so wird das Paket verworfen.

- Options

  Die aus Sicherheitssicht relevanten Optionen sind *strict source routing* bzw. *loose source routing*. Mit Hilfe dieser Optionen ist es dem Sender möglich, den Weg vorherzubestimmen, den das Paket nehmen muß, um den Empfänger zu erreichen. Der Empfänger muß seine Antwortpakete auf genau dem umgekehrten Weg abschicken, auf dem sie angekommen sind. Dies kann von Angreifern folgendermaßen ausgenutzt werden: Der Angreifer gibt eine falsche Absenderadresse an, um die Identität eines anderen Rechners anzunehmen. Dies würde aber bedeuten, daß Antworten an den Rechner geschickt werden, dessen Identität der Angreifer übernehmen möchte. Wird jedem Paket nun ein Weg vorgeschrieben, auf dem es transportiert werden muß, so kann es auch zum Rechner des Angreifers dirigiert werden, wo es vom Netz genommen wird. Beim *strict source routing* muß ein Paket den vorbestimmten Weg exakt einhalten, während beim *loose source routing* die Möglichkeit besteht, außer den angegebenen Routern noch weitere zu benutzen. Die sog. *basic security option*, mit derem Hilfe man einem Paket einen *security level* zwischen *unclassified* und *top secret* zuweisen kann, hat derzeit keinerlei praktische Bedeutung.

2.1.2 Das Transmission Control Protocol (TCP)

TCP [RFC793] setzt auf dem IP-Protokoll auf und stellt einen gesicherten, verbindungsorientierten *end-to-end* -Dienst zur Verfügung, der ein Multiplexing mehrerer Verbindungen zwischen zwei Hosts erlaubt.

Die Header-Felder

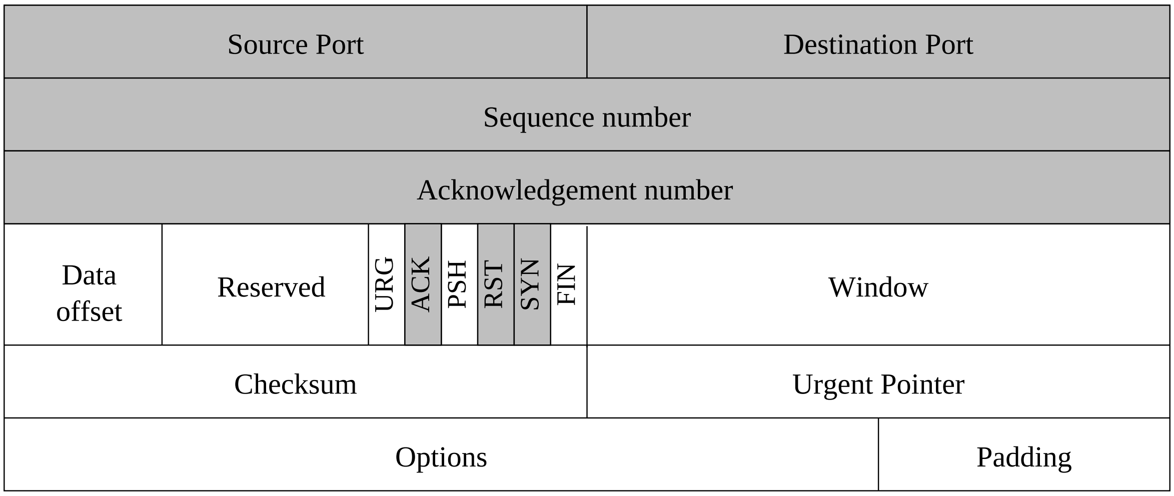

Abbildung 2.2: Der TCP-Header

TCP-Pakete enthalten, ähnlich wie bei IP, einen Header sowie Nutzdaten. Die wichtigsten Felder des im Abbildung 2.2 dargestellten Headers sind die folgenden:

- Source Port / Destination Port

  Die Port-Nummer ist eine 16-Bit Adresse, mit deren Hilfe das Betriebssystem entscheiden kann, für welchen Prozeß ein eintreffendes Paket bestimmt ist. Aufgrund der Informationen *source adress*, *destination adress*, *source port* sowie *destination port* ist eine TCP-Verbindung zwischen zwei Prozessen auf zwei evtl. verschiedenen Rechnern eindeutig bestimmt. Die Ports im Bereich von 0 bis 1023 werden als privilegierte Ports bezeichnet, da sie auf UNIX-Maschinen üblicherweise nur von Prozessen benutzt werden können, die über *root*-Privilegien verfügen. Die meisten *server*, d.h. Prozesse, die über das Netz Dienste an die Benutzer anbieten, 'hören' auf einem privilegierten Port, ob ein Verbindungswunsch eintrifft. Die Standard-*server*, die im folgenden noch einzeln beschrieben werden, verwenden meist einem Port, der sich im gesamten Internet als der übliche Port für diesen Server durchgesetzt hat, man spricht von sogenannten *well-known-ports*.

- Sequence Number

  Jedes Paket enthält eine 32-Bit-*sequence number*, die bei Verbindungsaufbau zufällig gewählt wird und dann bei jedem übertragenen Paket um eins erhöht wird. Somit kann sichergestellt werden, daß die Reihenfolge des Abschickens wieder eingenommen werden kann, da Übertragung mittels IP dies ja nicht gewährleistet.

- Acknowledgement Number

  Diese Nummer gibt die *sequence number* des als nächstes erwarteten Paketes an. Hierdurch werden alle Pakete mit niedrigerer *sequence number* als korrekt empfangen bestätigt. Dies ermöglicht es zu erkennen, ob einzelne Pakete auf dem Transport verlorengegangen sind.

- Acknowledgement Field Is Valid (ACK-Flag)

  Dieses Flag zeigt an, ob die Daten im *acknowledgement number*-Feld gültig sind oder nicht. Dies ist immer der Fall, außer es handelt sich um das allererste Paket einer Verbindung. Somit ist es möglich zu entscheiden, ob ein Paket Teil einer bestehenden Verbindung ist oder zur Initiierung einer neuen Verbindung dient. Dies kann von Bedeutung sein, wenn man nur internen Benutzern den Aufbau einer Verbindung gestatten möchte.

- Synchronize Sequence Numbers (SYN-Flag)

  Das SYN-Flag wird beim Verbindungsaufbau gesetzt, um dem Partner anzuzeigen, daß die übertragene *sequence number* als neue gültige Nummer zu betrachten ist. Der Empfänger schickt nun seinerseits ein Paket mit dem SYN-Flag sowie seiner *sequence number*, worauf wiederum eine Bestätigung erfolgt. Dieser sogenannte *three-way handshake* ist in Tabelle 2.1 [Lynch93] genauer dargestellt.

TCP A State		TCP B State
closed		listen
SYN-sent	→ <SEQ=100> <CTL=SYN>	→ SYN-received
established	← <SEQ=300> <ACK=101> <CTL=SYN,ACK>	← SYN-received
established	→ <SEQ=101> <ACK=301> <CTL=ACK>	→ established

Tabelle 2.1: Verbindungsaufbau bei TCP (three-way handshake)

2.1.3 Das User Datagram Protocol (UDP)

UDP [RFC768] ist ein sehr einfaches Transportprotokoll, das ebenfalls auf IP aufsetzt und deutlich weniger Overhead erzeugt als TCP. Es wird eingesetzt, wenn häufig kleine Datenmengen übertragen werden sollen und der Aufwand des Verbindungsaufbaus in keinem Verhältnis zu der übertragenen Datenmenge stehen würde. Es bietet einen ungesicherten, verbindungslosen Datagramm-Dienst, der ebenfalls das Multiplexing mehrerer Verbindungen zwischen zwei Hosts ermöglicht. Da weder eine *sequence number* noch ein Mechanismus zur Bestätigung empfangener Pakete vorgesehen ist, ist es möglich, daß Pakete nicht in der Reihenfolge des Abschickens eintreffen bzw. verlorene Pakete nicht bemerkt werden. Obwohl UDP ein verbindungsloses Protokoll ist, wird im Verlauf dieser Arbeit von UDP-Verbindungen gesprochen. Hiermit ist natürlich jeweils eine logische Verbindung gemeint, d. h. jeweils zusammengehörige Anfragen und Antworten.

Die Header-Felder

In Abbildung 2.3 erkennt man, daß der UDP-Header nur sehr wenige Daten enthält. Von diesen sind nur der *source port* sowie der *destinaton port* für die Sicherheitsbetrachtung relevant.

Source Port	Destination Port
UDP length	UDP checksum

Abbildung 2.3: Der UDP-Header

- Source Port / Destination Port

  Genau wie bei TCP dienen die Port-Nummern auch bei UDP dazu, den einzelnen Prozessen auf einem Rechner bestimmte Verbindungen zuzuordnen. Es werden ebenfalls 16-Bit Zahlen verwendet, wobei UDP- und TCP-Ports trotz gleicher Nummer unterschiedlichen Prozessen zugeordnet werden können.

2.1.4 Das Internet Control Message Protocol (ICMP)

ICMP ist eine Erweiterung des IP-Protokolls und wird verwendet, um TCP- sowie UDP-Verbindungen zu beinflussen. Es kann benutzt werden, um einem Sender einen besseren Weg zu seinem Ziel anzuzeigen, um bestimmte Fehlersituationen mitzuteilen oder auch um Verbindungen aufgrund von Netzproblemen zu beenden. Diese Nachrichten können natürlich auch von Angreifern für ihre Zwecke mißbraucht werden. Eine kurze Beschreibung einiger wichtiger ICMP-Nachrichten wird im folgenden gegeben:

- Redirect

  Eine *redirect*-Nachricht weist den Sender an, einen anderen Router zu benutzen, um einen bestimmten Zielrechner zu erreichen.

- Destination Unreachable

  Nachrichten diesen Typs werden verwendet um mitzuteilen, daß das Ziel, ein bestimmter Rechner oder auch ein ganzes Netz, im Moment nicht erreichbar ist. Somit kann eine Verbindung aufgrund von Netzwerkproblemen abgebrochen werden, ohne daß auf einen *timeout* der Verbindung gewartet werden muß.

- Time Exceeded

  Sollte die *time to live* eines Paketes abgelaufen sein, bevor es sein Ziel erreicht hat, so wird das Paket verworfen und dem Sender mit Hilfe der *time exceeded*-Nachricht mitgeteilt, daß das Ziel nicht erreicht werden konnte. Dies kann der Fall sein, wenn ein Paket aufgrund eines Fehlers in den Routingtabellen in einer Schleife zwischen zwei oder mehreren Routern pendelt.

- Adress Mask Request and Reply

  Hierbei handelt es sich um ein Paar von Nachrichten, wobei in der ersten der beiden nach der *subnet mask* eines Netzes gefragt wird, d. h. nach einer Bitmaske, die angibt, welcher Teil der 32-Bit-Adresse als Adresse des Subnetzes angesehen werden soll. Die zweite Nachricht enthält die Antwort auf diese Frage. Dies kann z.B. während des Bootvorganges eines Rechners notwendig sein, um ihn über die verwendete *subnet mask* in den angeschlossenen Netzen zu informieren.

2.2 Analyse der Standard-Dienste

Im Internet haben sich eine Reihe von Diensten als Standards durchgesetzt [Ches94] [Lynch93]. Sie alle benutzen auf Netzwerkebene das IP-Protokoll sowie auf Transportebene TCP bzw. UDP. In diesem Kapitel werden die wichtigsten Internet-Dienste vorgestellt, die Notwendigkeit des Einsatzes dieses Dienstes den damit verbundenen Risiken gegenübergestellt sowie mögliche Politiken, wie der Dienst behandelt werden kann, dargestellt. Sollte es sinnvoll sein, einen Dienst in einer eingeschränkten Form anzubieten, so werden hier ebenfalls verschiedene Möglichkeiten vorgestellt. Hierbei wird allerdings nur auf Risiken eingegangen, die in der Spezifikation des Protokolls begründet liegen. Andere Gefahren wie z.B. fehlerhafte Server, werden erst später betrachtet.

2.2.1 Telnet

Das Telnet-Protokoll [RFC854] erlaubt Terminal-Zugriff auf einen entfernten Rechner, als ob das Terminal an diesen Rechner direkt angeschlossen wäre. Hierzu wird ein *network virtual terminal* (NVT), also ein virtuelles Standard-Terminal, definiert, auf dessen Eigenschaften die Charakteristika des reellen Terminals abgebildet werden müssen. Der *telnet-client* auf der Maschine des Benutzers sowie der *telnet-server* auf dem entfernten Rechner führen diese Umsetzung durch. Der Standard-Port, auf dem ein *telnet-server* auf eintreffende Verbindungen wartet, ist Port 23. Möchte nun ein Benutzer auf einen entfernten Rechner zugreifen, so initiiert er mit Hilfe der entsprechenden *client*-Software eine TCP-Verbindung zum *telnet*-Port des gewünschten Rechners. Der *server* verlangt zur Authentifizierung des Benutzers einen Account-Namen sowie ein Paßwort, deren Gültigkeit er über das übliche *login*-Programm des UNIX-Betriebssystems überprüft.

Bedarf

telnet ist ein wichtiger Dienst, der sowohl von außen nach innen als auch von innen nach außen benötigt wird. Ein Benutzer an einem internen Rechner hat über das *telnet*-Protokoll die Möglichkeit, Ressourcen externer Rechner zu nutzen und dort z.B. Programme auszuführen. Umgekehrt können sich Mitarbeiter, die sich aufgrund einer Dienstreise nicht in der Firma aufhalten können, von einem externen Rechner in einen internen einloggen und dort beispielsweise ihre Mail lesen. Auch kann es erforderlich sein, Herstellern von Software den Zugang über Telnet zu ermöglichen, um Wartungsarbeiten schnell und kostengünstig durchführen zu können.

Gefahren

Leider hat das *telnet*-Protokoll einige Schwachstellen, die eine sichere Nutzung deutlich erschweren. An erster Stelle steht hier die Verwendung des üblichen UNIX-Paßwort-Konzeptes, das sich als zu schwach erweist, um wirkungsvollen Schutz gegen einen Angreifer bieten zu können. Da die Benutzer ihre Paßworte relativ frei wählen können, ist es ziemlich wahrscheinlich, daß einzelne Mitarbeiter Paßworte wählen, die für einen Angreifer leicht zu erraten sind. Eine weitere Gefahr ergibt sich durch die Möglichkeit, den Netzverkehr abzuhören und sämtliche Pakete mitzuprotokollieren. Da *telnet* das eingegebene Paßwort im Klartext über das Netz überträgt, kann ein Angreifer, der in der Lage ist, das Netz abzuhören, bei Beginn einer Telnet-Sitzung das Paßwort erhalten und später für seine eigenen Zwecke mißbrauchen. Ist ein Angreifer erst einmal auf einen firmeneigenen Rechner eingeloggt, so hat er eine Vielzahl von Möglichkeiten, Daten zu manipulieren bzw. seinen Angriff auf weitere Rechner auszudehnen.

mögliche Sicherheitspolitik

Eine Möglichkeit, die durch *telnet* entstehenden Gefahren gering zu halten, ist es, den Zugang nur von innen nach außen zu gestatten. Somit ergibt sich für einen Angreifer selbst bei Kenntnis eines Paßwortes nicht die Möglichkeit, dies für seine Zwecke einzusetzen. Möchte man aber auch Terminalzugriff von außen nach innen, so gilt es zu beachten, daß nie ein wiederverwendbares Paßwort über das ungesicherte äußere Netz übertragen wird. Aus diesem Grund stehen Methoden zur Verfügung, mit Hilfe sogenannter *one-time-passwords* die Identität des Benutzers zu überprüfen, ohne daß ein Angreifer ein derartiges Paßwort zu einem späteren Zeitpunkt nochmals einsetzen kann (Siehe Anhang A). Weiterhin ist es ratsam, im Rahmen eines umfangreichen Loggings insbesondere alle gescheiterten Versuche, eine *telnet*-Verbindung aufzubauen, mitzuprotokollieren. So können versuchte Angriffe meist bereits frühzeitig entdeckt werden und die notwendigen Gegenmaßnahmen getroffen werden.

2.2.2 Rlogin

Mit Hilfe des *rlogin*-Protokolls [RFC1258] kann man sich, genau wie bei *telnet* beschrieben, in einen anderen Rechner einloggen. Der wesentliche Unterschied ist aber, daß bei Verwendung von *rlogin* kein Paßwort angegeben werden muß, wenn die folgenden drei Kriterien erfüllt sind:

- Die Verbindung muß von einem privilegierten TCP-Port stammen.
- Sowohl der Benutzer als auch der Rechner, von dem der Verbindungswunsch ausging, müssen in einer Liste von zugelassenen Kommunikationspartnern eingetragen sein (normalerweise `/etc/hosts.equiv` oder `$HOME/.rhosts`).
- Der Name des Rechners muß mit seiner IP-Adresse übereinstimmen.

Trifft eine dieser Bedingungen nicht zu, so verhält sich *rlogin* analog zu *telnet*, d. h. es muß ein Paßwort angegeben werden, bevor die Verbindung zustande kommt.

Bedarf

Rlogin wird von den Benutzern üblicherweise gegenüber *telnet* bevorzugt, da bei Verbindungen zu häufig kontaktierten Rechnern die Angabe eines Paßwortes entfallen kann. Handelt es sich um eine Verbindung über ein unsicheres Netz, so kann dies auch einen Sicherheitsgewinn darstellen, da keine Gefahr des Abhörens von Paßworten besteht.

Gefahren

Demgegenüber stehen aber auch große Gefahren. Da die Zugangskontrolle von den Benutzern selber verwaltet werden kann, kann man sich nicht darauf verlassen, daß dies mit ausreichendem Blick auf die Sicherheit des Netzes geschieht. Wählt ein Benutzer z. B. die Rechte für sein `.rhosts`-File so, daß andere Benutzer Schreibrecht haben, so kann sich ein Angreifer einen Eintrag in diesem File erzeugen, der ihm ein Einloggen als dieser Benutzer ermöglicht. Dies wird auch oft ausgenutzt, um nach einem geglückten Angriff eine unauffällige Möglichkeit zu schaffen, zu einem späteren Zeitpunkt wieder Zugang zum System zu erlangen.

Ein weiteres Problem ist es, daß sich dieses Vertrauen der Rechner untereinander häufig über das gesamte Unternehmen erstreckt. Das bedeutet, daß ein Angreifer, der Zugang zu einem Rechner erlangt hat, gleichzeitig Zugang zu allen Rechnern erlangt, die diesem vertrauen. Da auch diese Rechner wiederum anderen vertrauen, kann sich ein Angriff so über das ganze Netz ausdehnen.

mögliche Sicherheitspolitik

Es ist ratsam, *rlogin* nicht durch einen Firewall zu erlauben, da man sich keinesfalls darauf verlassen kann, daß die Benutzer auf die Sicherheit bei der Erstellung ihrer `.rhosts`-Files achten. Findet allerdings am Firewall bereits eine Authentifizierung des Benutzers statt, so kann die weitere Verbindung mittels *rlogin* erfolgen, um einerseits eine zweifache Authentifizierung des Benutzers zu vermeiden, andererseits aber auch die Übertragung von Paßworten im Klartext über das ungesicherte Netz zu verhindern.

2.2.3 Electronic Mail

Einer der wichtigsten Internet-Dienste ist wohl die Möglichkeit, sogenannte *e-Mails* an andere User zu verschicken. Das im Internet zum Versenden von Mail eingesetzte Protokoll ist das auf TCP aufsetzende *simple mail transfer protocol* (SMTP) [RFC821].

Ein Benutzer, der eine Mail verschicken möchte, bereitet diese üblicherweise mit einem speziellen Programm vor, das die Nachricht daraufhin im Filesystem an einer vorher festgelegten Stelle ablegt. Das SMTP-Modul findet die dort abgelegte Mail und baut eine Verbindung zu dem SMTP-Server des adressierten Rechners auf. Der *well-known Port*, auf dem normalerweise Mailübertragung stattfindet, ist Port 25. Dem empfangenden Modul werden einige Daten wie z.B. der Absender oder der/die Empfänger mitgeteilt und daraufhin der Inhalt der Nachricht übertragen. Nach Abschluß der Übertragung wird die empfangene Nachricht in einem speziellen Verzeichnis, der Mailbox des Empfängers abgelegt, wo er sie wiederum mit Hilfe eines Programmes lesen kann. Kann das sendende Modul den Zielrechner nicht direkt erreichen, so gibt es weiterhin die Möglichkeit, die Nachricht an einen anderen Host weiterzuleiten, der dann die weitere Übertragung der Nachricht übernimmt.

Bedarf

Electronic Mail ist einer der meistgenutzten Internet-Dienste. Fast jeder Benutzer verschickt und empfängt regelmäßig Nachrichten, wobei auch häufig Kontakte zu externen Benutzern bestehen. Somit ist es praktisch unumgänglich, *e-mail* uneingeschränkt zu gestatten.

Gefahren

Im Zusammenhang mit SMTP zeigen sich mehrere Gefahren:

So besteht die Möglichkeit, daß ein Mitarbeiter mit Hilfe einer Mail wichtige Daten nach außen schleust. Da den Mitarbeitern aber auch andere Möglichkeiten zur Verfügung stehen, interne Daten nach außen zu schaffen, die nicht mit Hilfe eines Firewalls unterbunden werden können (z.B. Disketten), erscheint es aber nicht sinnvoll, das Verschicken von *e-mail* auf Nachrichten von innen nach außen zu begrenzen, da dies auch zu einer großen Unzufriedenheit bei den Benutzern führen dürfte.

Außerdem ist es nicht möglich, den Absender einer Mail mit Sicherheit zu bestimmen, was es einem Angreifer ermöglicht, Mail unter falschem Namen zu versenden. Insbesondere kann ein Angreifer unter dem Namen *root* Benutzer auffordern, ihr Paßwort zu ändern oder sonstige Tätigkeiten auszuführen. Ein unbedarfter Benutzer wird dieser Aufforderung vielleicht Folge leisten und dem Angreifer somit Möglichkeiten eröffnen, weiterführende Angriffe zu starten.

Ein weiteres Sicherheitsproblem stellt die Möglichkeit dar, Mail-Aliases auf den tatsächlichen Login-Namen abzubilden. Mit Hilfe des 'VRFY'-Kommandos, wird zu einem gegebenen Alias der Login-Name ausgegeben. Dies ermöglicht es dem Angreifer einerseits, zu testen, welche Logins auf dem Rechner existieren, um daraufhin zu versuchen, ein Paßwort eines dieser Accounts zu erraten. Andererseits besteht die Möglichkeit, Aliases wie *postmaster* bzw. *root* auf den tatsächlichen Login-Namen abzubilden, um somit Informationen zu erhalten, welche Accounts ein besonders lohnendes Ziel für einen Angriff darstellen.

Sendmail, die verbreitetste Implementierung des SMTP-Protokolls, enthält in sich eine große Anzahl von Sicherheitslücken, auf die allerdings erst an späterer Stelle eingegangen wird (siehe Kap. 5.1.2).

mögliche Sicherheitspolitik

Wie bereits erwähnt, ist eine Einschränkung der Möglichkeit, Mail zu empfangen bzw. zu versenden, nicht empfehlenswert. Man muß sich also darauf beschränken, sämtliche Mails zu protokollieren, falls nicht aus Datenschutzgründen davon abgesehen werden muß. Weiterhin sollten die Benutzer darauf aufmerksam gemacht werden, daß sie sich nicht darauf verlassen können, daß der auf einer Mail angegebene Absender auch mit dem tatsächlichen Absender übereinstimmt und daß die Möglichkeit besteht, die Inhalte einer Mail abzuhören.

2.2.4 File Transfer

Ein ebenfalls häufig genutzter Internet-Dienst ist *File Transfer*, d. h. die Übertragung von Dateien von einem Rechner zu einem anderen. Das im Internet verbreitetste Protokoll zur

Übertragung von Dateien ist das *File Transfer Protocol* (FTP) [RFC959]. Es verwendet auf Transportebene das TCP-Protokoll, wobei der FTP-*server* üblicherweise auf Port 21 angesiedelt ist.

Möchte ein Benutzer ein File von einem Rechner A auf seinen eigenen Rechner, den Rechner B übertragen, so geht er folgendermaßen vor. Mit Hilfe seines FTP-*client* eröffnet er einen sogenannten *control channel* zwischen den beiden Rechnern. Zur Authentifizierung seiner Identität wird er aufgefordert, einen Account-Namen sowie ein gültiges Paßwort anzugeben. Dann stehen ihm Kommandos zur Verfügung, mit denen er sich in der Verzeichnisstruktur von Rechner A bewegen kann, wie z.B. die bekannten Kommandos cd und ls. Möchte er nun ein File übertragen, so kann er den Get-Befehl benutzen, um Daten von Rechner A auf Rechner B zu übertragen, sowie den Put-Befehl zur Übertragung in umgekehrter Richtung.

Die tatsächlichen Daten werden nicht auf dem *control channel* übertagen, sondern es wird ein zweiter Kanal, der sogenannte *data channel* eröffnet. Der *server* verwendet Port 20 für diesen Datenkanal, während der *client* normalerweise denselben Port verwenden sollte, den er auch für den Kontrollkanal benutzt. Da aber praktisch alle FTP-Implementierungen für jede Datenübertragung den Kanal neu öffnen und bei Ende der Übertragung wieder schließen und TCP außerdem die Eigenschaft hat, eine Neuverbindung derselben Ports erst nach Ablauf eines bestimmten Zeitintervalls zu gestatten, muß für jede Datenübertragung ein neuer Port gewählt werden. Der *client* wählt also einen neuen Port aus und teilt diesen dem *server* über das Port-Kommando mit, worauf der *server* einen Datenkanal zu ebendiesem Port des *client*–Rechner eröffnet.

Das Protokoll beinhaltet eine Möglichkeit, von einem dritten Rechner aus die Übertragung von Daten zwischen zwei Rechnern anzustoßen. Ein Benutzer an einem Rechner C möchte ein File von einem Rechner A auf einen Rechner B übertragen. Er eröffnet *control channels* zu beiden Rechnern. Daraufhin schickt er an einen der beiden Rechner, z.B. Rechner B ein sogenanntes PASV-Kommando, was diesen veranlaßt, dem *client* einen Port mitzuteilen, auf dem er auf eintreffende Verbindungen wartet, anstatt selber eine Verbindung aufzubauen, wenn ein Transferkommando eintrifft. Daraufhin wird an Rechner A ein PORT-Kommando abgesetzt, das diesen anweist, einen Datenkanal zu ebendiesem Port des Rechners B zu erzeugen. Dann kann man den beiden Rechnern die jeweils korrespondierenden Kommandos schicken, `get` *filename* an Rechner A, sowie `put` *filename* an Rechner B. Daraufhin wird Rechner A einen Datenkanal zu Rechner B erzeugen und die Datenübertragung kann stattfinden. Wie dies ausgenutzt werden kann, um die Sicherheit des FTP-Protokolls zu erhöhen, wird in einem späteren Kapitel (Kap. 4.2.2) dargestellt.

Anonymous FTP Zusätzlich zum Filetransfer von bzw. zu Rechnern, auf denen der Benutzer ein Account hat, wird von vielen Netzteilnehmern ein *anonymous ftp server* angeboten, d.h. ein *server*, von dem man sich Daten übertragen kann, ohne dort ein Account haben zu müssen. Als Account-Name verwendet man *anonymous* und als Paßwort üblicherweise seine eigene E-Mail-Adresse, deren Richtigkeit vom *server* aber nicht überprüft werden kann.

Bedarf

File transfer ist ein sehr häufig genutzer Dienst, für den auch bei der Firma BMW Bedarf besteht. Allerdings ist es ausreichend, die Möglichkeit *file transfer* auszuführen auf interne

Benutzer zu beschränken. Sollte ein Mitarbeiter von außen FTP nutzen müssen, so besteht die Möglichkeit, sich über *telnet* in einen internen Rechner einzuloggen und die Datenübertragung von dort aus zu starten. Bedarf für einen *anonymous ftp server* besteht nicht.

Gefahren

Die Hauptgefahr in Verbindung mit FTP, ist die Übertragung firmeninterner Daten nach außen. Wenn ein Angreifer ein Paßwort eines Benutzers ermittelt hat, so kann er eine FTP-Verbindung zu dem entsprechenden Rechner aufbauen, sämtliche Files, auf die dieser Benutzer Zugriff hat, auf seinen Rechner kopieren und dort in Ruhe auswerten. Dies schließt meist auch die Paßwort-Datei der Maschine mit ein, deren Besitz es dem Angreifer ermöglicht, weitere Paßwörter zu knacken und den Angriff auf weitere Benutzer auszudehnen. Selbst *anonymous ftp server* stellen zum Teil Paßwort-Files zur Verfügung.

Weiterhin kann der Angreifer auch Daten des Benutzers ändern, beispielsweise ein `.rhosts`-File anlegen, welches ihm ein späteres Einloggen in den Rechner erlaubt, auch wenn der Benutzer zwischenzeitlich sein Paßwort ändern sollte.

FTP kann auch benutzt werden, um Paßwörter der einzelnen Benutzer erst einmal zu ermitteln. Ähnlich wie bei Telnet, wird das Paßwort beim Einloggvorgang im Klartext übertragen, womit es jedem Angreifer, der Zugang zu diesem Netz hat, möglich ist, dieses Paßwort mitzuprotokollieren. Dies verbietet FTP-Zugang über normale Paßwort-Authentifizierzung aus einem nicht gesicherten Netz völlig.

mögliche Sicherheitspolitik

Da kein Bedarf an einem FTP-Zugang von außen besteht, bleibt nur zu überlegen, ob man eine weitere Einschränkung vornehmen soll, die es internen Benutzern verbietet, Daten nach außen zu transportieren. Da dies einem internen Benutzer aber auch auf viele andere Arten moglich ist (z. B. über *E-Mail*), kann davon abgesehen werden. Man könnte somit aber verhindern, daß ein Angreifer, der ein Account auf einem internen Rechner erfolgreich angegriffen hat, Daten von dort nach außen schaffen kann.

2.2.5 News

Von ebenfalls großer Bedeutung ist der Austausch von *news* über das Internet. Der Unterschied zur normalen *e-mail* besteht darin, daß eine Nachricht nicht an einen bestimmten Empfänger geschrieben wird, sondern weltweit lesbar gemacht wird. Somit ergibt sich die Möglichkeit, Fragen an die gesamte Internet-Gemeinde zu richten, wobei natürlich die Chance auf eine hilfreiche Antwort aufgrund der großen Anzahl von Teilnehmern sehr gut ist. Der Einsatz von *news* beschränkt sich nicht nur auf technische Fragestellungen, sondern dehnt sich in praktisch alle Bereiche des täglichen Lebens aus. Um bei der großen Anzahl von *news*-Artikeln die Übersicht behalten zu können, existierten verschiedene *newsgroups*, die sich mit unterschiedlichen Themen beschäftigen.

Der Austausch von *news* zwischen zwei Rechnern erfolgt über das *network news transport protocol* (NNTP)[RFC977]. Hierbei handelt es sich um ein auf TCP basierendes Protokoll.

Ein *newsserver* baut eine Verbindung zu seinem Nachbar–*server* auf und überträgt sämtliche *news*, die dort seit der letzten Verbindung neu eingetroffen sind. Umgekehrt gibt er seine neu eingetroffenen *news* ab. So verbreitet sich ein neu geschriebener Artikel in kürzester Zeit im gesamten Internet. Der normalerweise für *NNTP* verwendete Port ist 119.

Bedarf

Es besteht sowohl Bedarf daran, Artikel innerhalb des Unternehmens zu verbreiten, als auch daran, Fragen an alle Internet-Teilnehmer richten zu können. Der Austausch von Artikeln innerhalb des Unternehmens dient hauptsächlich der Verbreitung von Informationen an alle Mitarbeiter, während der Austauch mit allen Internet-Teilnehmern der schnellen Beantwortung von Fragen dient.

Gefahren

Da der interne *newsserver* die externen Artikel ausschließlich von einem einzigen, vorherbestimmten *server* erhält, gestaltet sich die Sicherung von *NNTP* relativ unproblematisch. Man kann jedem anderen Rechner den Zugang verbieten und somit einem Angreifer kaum Ansatzmöglichkeiten für einen Angriff bieten. Allerdings besteht die Gefahr, daß interne Informationen an die Öffentlichkeit gelangen, wenn sie über *news* zugänglich gemacht werden. Außerdem kann der Rechner, der die Behandlung der *news* übernimmt, aufgrund der großen Anzahl von eintreffenden Artikeln überlastet werden und somit auch nicht mehr für seine anderen Aufgaben zur Verfügung stehen.

mögliche Sicherheitspolitik

Es erscheint sinnvoll, *newsgroups* zu definieren, deren Artikel nicht an externe *server* abgegeben werden. Somit können Informationen, die ausschließlich für Mitarbeiter des eigenen Unternehmens gedacht sind, geheimgehalten werden. Gegen die mögliche Überlastung des *servers* kann man nichts unternehmen. Deshalb sollte als *newsserver* ein Rechner eingesetzt werden, der keine weiteren Aufgaben zu erfüllen hat. Eine Überlastung dieses Rechners bedeutet dann nur eine Verzögerung in der Verbreitung der Artikel bzw. möglicherweise Wartezeiten beim Lesen von Artikeln.

2.2.6 Domain Name Service (DNS)

Jeder Host im Internet besitzt eine 32-Bit-Adresse, die ihn weltweit eindeutig identifiziert. Um dem menschlichen Benutzer die Arbeit zu erleichtern, hat man ein System von Namen eingeführt, die auf die entsprechende Internet-Adresse abgebildet werden können. Diese Abbildung wird von sogenannten *domain name servern* [RFC1034] [RFC1035] durchgeführt, d. h. Rechnern, die Tabellen enthalten, die es ihnen erlauben, zu einem bestimmten Rechnernamen die korrespondierende Adresse auszugeben. Außer der IP-Adresse liefern diese *server* auch noch weitere Informationen über den gewünschten Host, wie z. B. einen *mail–server*, der für diesen Host Mail empfangen kann. Die Anfragen an die *name server* laufen üblicherweise über UDP, obwohl es ebenfalls möglich wäre, TCP zu verwenden.

Der verwendete Namensraum ist baumartig organisiert, was es ermöglicht, Teilbäume und somit ganze Teilnetze an untergeordnete Server zu delegieren. Somit entsteht eine verteilte Datenbank, in der der jeweilige Netzbetreiber für die Einrichtung eines *name servers* für seine Hosts verantwortlich ist. Um umgekehrt eine Abbildung von IP-Adressen auf Namen durchführen zu können, existiert ein zweiter Baum.

Damit ein Ausfall eines Servers nicht zum Ausfall des gesamten betreuten Netzes führt, ist eine Möglichkeit vorgesehen, einen oder mehrere *secondary server* einzurichten, die regelmäßig die Daten ihres jeweiligen *primary server* über einen sogenannten *zone transfer* erhalten. Für diese Transfers wird TCP als Transportprotokoll eingesetzt.

Bedarf

Es besteht die unbedingte Notwendigkeit, einen *name server* für das Firmennetz einzurichten, da die Verwendung von IP-Adressen statt Rechnernamen für menschliche Benutzer nicht zumutbar ist. Soll ein Zugang von außen möglich sein, so muß sichergestelt sein, daß auch nach außen hin ausreichende Informationen verbreitet werden, so daß der entsprechende interne Rechner erreicht werden kann.

Gefahren

Eine gewisse Gefahr ergibt sich dadurch, daß ein möglicher Angreifer über Anfragen an den *name server* Kenntnis über die Adressen der internen Rechner erlangen kann. Dies ist ein erster Schritt zur Durchführung eines Angriffes, da ein Rechner, dessen Adresse nicht bekannt ist, auch nicht angegriffen werden kann. Insbesondere mit Hilfe eines *zone transfers* kann ein Angreifer eine große Anzahl potentieller Angriffsziele erhalten.

mögliche Sicherheitspolitik

Obwohl Angreifer auch über anderen Möglichkeiten verfügen, Adressen von möglichen Angriffzielen zu ermitteln, sollte ihnen der Zugriff zum *name server* weitestgehend untersagt werden. Insbesondere ein *zone transfer* darf nur dem *secondary server* möglich sein. Um den internen Benutzern alle Adressen zugänglich zu machen, den externen diese Informationen aber vorzuenthalten, besteht die Möglichkeit, einen zweigeteilten *name server* einzurichten. Dieser bietet vollen *name service* für alle internen Rechner, während auf Anfragen von außen nur die Adresse des Firewalls bekanntgegeben wird, der dann die weitere Behandlung der eintreffenden Pakete übernehmen muß. Dies stellt aber, wie bereits erwähnt, keine Garantie dafür dar, daß ein Angreifer keine Informationen über die Adressen der internen Rechner erhält.

2.2.7 Finger

Das *finger*-Protokoll [RFC742] wird eingesetzt, um Informationen über einen bestimmten Benutzer zu erhalten oder über sämtliche Benutzer, die auf einem Rechner eingeloggt sind. Es handelt sich um ein sehr einfaches Protokoll auf Basis von TCP, bei dem der *client* eine einzeilige Anfrage an den Server stellt, dieser eine Antwort generiert, dem *client* übermittelt und die Verbindung wieder beendet.

Eine Anfrage in der Form `finger @some.host` ergibt als Antwort die *login*-Namen aller Benutzer, die momentan am Rechner `some.host` eingeloggt sind, ihren richtigen Namen, sowie weitere Informationen, wie den Zeitpunkt des Einloggens und die Zeit, seit der dieser Benutzer untätig war.

Möchte man nun über einen Benutzer weitere Informationen erhalten, so kann man mit Hilfe des Kommandos `finger user@some.host` genauere Details über den bestimmten Benutzer erhalten, egal ob er momentan eingeloggt ist oder nicht. Dies umfaßt den Zeitpunkt, zu dem der Benutzer zuletzt eingeloggt war, sein *home directory* und einige weitere Informationen.

Bedarf

Mit Hilfe von *finger* läßt sich z. B. feststellen, ob ein bestimmter Benutzer anwesend ist oder wann dieser Benutzer zuletzt seine *mail* gelesen hat. Auch kann die *mail*-Adresse eines Benutzers ermittelt werden, um ihm eine Nachricht zustellen zu können. Es handelt sich also um einen eher unwichtigen Dienst.

Gefahren

Die Gefahr, die der *finger*-Dienst mit sich bringt, wenn er für externe Benutzer gestattet wird, ist die, daß jeder Angreifer Informationen über mögliche Angriffspunkte erhalten kann. so kann beispielsweise erkannt werden, wenn bestimmte Benutzer seit langer Zeit nicht eingeloggt waren, weshalb deren *account* für einen Angriff besonders in Frage kommt. Selbst ein geglückter Angriff könnte hier nämlich über längere Zeit unbemerkt bleiben.

mögliche Sicherheitspoltik

Die meisten Unternehmen verbieten *finger* von außen vollständig. Als Alternative wäre es zu überlegen, auf Anfragen von außen eine allgemeine Information auszugeben, die z. B. die Adressierung von *mail* an die Mitarbeiter dieses Unternehmens erläutert.

2.2.8 X11

X11 ist das derzeit im Internet vorherrschende *Window System*. Auf dem Rechner des Benutzers läuft ein sogenannter *X-server*, der die Kontrolle über die Ein- und Ausgabegeräte, also z. B. Bildschirm, Tastatur und Maus, ausübt. Über das *X11*-Protokoll können nun Anwendungen, sogenannte *X-clients* mit dem *server* kommunizieren und diesem mitteilen, was am Bildschirm ausgegeben werden soll. Im Gegenzug erhalten sie vom *server* z. B. Informationen über Tastatureingaben und Mausbewegungen. Dies ermöglicht es, herstellerspezifische Implementierungen auf den *server* zu beschränken, sodaß auch in einer heterogenen Systemumgebung von jedem Terminal aus die gleichen *clients* verwendet werden können. *X-server* laufen normalerweise auf einem Port im Bereich von 6000 bis 6100, meist wird Port 6000 verwendet.

Damit ein *client* eine Verbindung zu einem *X-server* aufbauen kann, muß er hierzu autorisiert werden. In der ursprünglichen Definition des *X11*-Protokolls gab es nur die Autorisierung aufgrund der IP-Adresse des *client*-Rechners. Ein Benutzer, der auf einem entfernten Rechner eine Anwendung starten will, deren Ausgabe auf seinem Bildschirm erfolgen soll, kann mit Hilfe des `xhost`-Kommandos dem entfernen Rechner Zugang zu seinem *X-server* erlauben. Hierzu setzt er ein Kommando der Form `XHOST+ System_Name` ab, wobei `System_Name` der Name des entfernten Rechners ist. Diesem Rechner ist somit bis zum Ausloggen des Benutzers der Zugang zum *X-server* erlaubt. Entfällt die Angabe eines Systemnamens, wird die Zugangskontrolle deaktiviert. Mit dem Kommando `XHOST- System_Name` kann der Zugang wieder verwehrt werden.

Mit *Release 4* des *X11*-Protokolls besteht nun die Möglichkeit, die Autorisierung auf ein Benutzer/Rechner-Paar auszuweiten [Mui92] [Shel92]. Einem Benutzer wird bei Beginn seiner Sitzung ein geheimes Paßwort, ein sogenanntes *magic cookie*, in einem File namens `.Xauthority` mitgeteilt. Dieses Paßwort muß nun verwendet werden, um eine X-Verbindung von einem entfernten Rechner aus zu initiieren. Der Benutzer überträgt hierzu das *magic cookie* mit Hilfe des `Xauth`-Kommandos an den entfernten Rechner, wo es in das dortige `.Xauthority`-File aufgenommen werden muß. Damit hat der Benutzer die Kontrolle, wer die Erlaubnis hat, auf seinen *X-server* zuzugreifen.

Bedarf

Von vielen Benutzern besteht der Wunsch, *X11* durch den Firewall zu nutzen. Zum einen möchten sie Anwendungen auf externen Rechnern laufen lassen, deren Ausgaben an ihrem internen Terminal angezeigt werden, zum anderen kann es auch sein, daß Mitarbeiter, die sich vorübergehend nicht in der Firma aufhalten, Anwendungen auf internen Rechnern von externen Terminals aus benutzen. Da für die meisten Dienste auch *clients* existieren, die ohne das *X11*-Protokoll auskommen, ist dies meist aber nicht unbedingt notwendig.

Gefahren

Ein Angreifer, der Zugang zum *X-server* eines Rechners hat, hat z. B. die Möglichkeit, Bildschirminhalte zu kopieren, Tastendrucke zu protokollieren oder Mausbewegungen zu erzeugen [Bisch94]. Er könnte also in den Besitz von geheimen Informationen gelangen, wenn er eine Kopie des Bildschirms erstellt, während der Benutzer gerade ein Dokument mit den entsprechenden Daten betrachtet. Auch kann er Paßworte erlangen, indem er sämtliche Tastatureingaben mitprotokolliert, die auf dem entsprechenden Rechner gemacht werden. Eröffnet der dort eingeloggte Benutzer dann z. B. eine *telnet*-Verbindung, so kann der Angreifer das verwendete Paßwort mitlesen.

Wie oben bereits erwähnt, sind vor allem die älteren Versionen des *X11*-Protokolls nicht mit ausreichenden Zugangskontrollen versehen, um unberechtigten Benutzern den Zugang zum *server* zu verwehren. Auch die neue Version, die mit Hilfe des *magic cookies* die Kontrolle auch auf einzelne Benutzer ausdehnt, kann nicht als ausreichend sicher angesehen werden, da viele Benutzer mehr Wert auf Bequemlichkeit anstatt auf Sicherheit legen. Insbesondere wenn ein Benutzer verschiedene Anwendungen auf verschiedenen Rechnern laufen läßt, besteht die Gefahr, daß er auf die umständliche Übertragung des *magic cookies* verzichtet und stattdessen die Kontrolle vollständig deaktiviert. Viele Benutzer gestatten somit jedem Angreifer, eine Verbindung zum *X-server* aufzubauen, um sich die Mühe zu ersparen, jeden *X-client* einzeln zu autorisieren und gefährden somit unwissentlich die Sicherheit ihres Rechners und des gesamten Netzes.

mögliche Sicherheitspolitik

Aufgrund der oben beschriebenen Gefahren sollte *X11* nur innerhalb des internen Netzes erlaubt sein. Andernfalls muß davon ausgegangen werden, daß unbeabsichtigte Fehler von internen Benutzern zu Angriffen ausgenutzt werden können. Da es aber manchmal unvermeidlich ist, *X11* durch den Firewall zu gestatten, gibt es inzwischen Möglichkeiten, einen gewissen Schutz auszuüben. Dabei wird, bevor eine Verbindung gestattet wird, eine explizite Bestätigung des Benutzers verlangt, daß er diese Verbindung gestattet. Somit kann zumindest kein Angreifer einen Kontakt zum *server* herstellen, ohne daß der Benutzer dies bemerkt.

2.2.9 World Wide Web (WWW)

Das World Wide Web ist ein Informationsdienst, der über das *hypertext transfer protocol* (HTTP) Informationen eines *WWW-servers* einem *client* zur Verfügung stellt. Es handelt sich um ein einfaches, zustandsloses Protokoll, das TCP als Transportprotokoll verwendet. Eine typische HTTP-Anfrage läuft nach folgendem Schema ab:

- Connection

  Der *client* baut eine TCP-Verbindung zum *server* auf, üblicherweise über den *well-known port* 80.

- Request

  Der Client schickt eine Anfrage an den *server*, welche Informationen er erhalten möchte.

- Response

  Der *server* liefert die gewünschten Informationen oder eine Fehlermeldung.

- Close

  Nach Ende der Datenübertragung schließt der *server* die Verbindung wieder.

Die *request* entält einen sogenannten *uniform resource locator* (URL), der eine Ressource im Internet weltweit eindeutig identifiziert. Das Konzept des URLs erlaubt die Angabe eines

Protokolls, über das die Kommunikation ablaufen soll. Dies ermöglicht es einem *WWW-client*, Daten, die nicht auf *WWW-servern* sondern z. B. auf *FTP*-, *Gopher*-, *WAIS*- oder *NNTP-servern* existieren, anzufordern. Hierzu muß der *client* natürlich all diese Protokolle beherrschen, was andererseits aber die Notwendigkeit, für jedes dieser Protokolle spezielle *clients* zu verwenden, verringert. Über das sogenannte *common gateway interface* (CGI) können Dokumente im Moment der Anfrage dynamisch erzeugt werden. Hierzu wird bei entsprechender Anfrage ein Programm ausführt, welches das gewünschte Dokument generiert. Dies kann natürlich auch verwendet werden, um Programme zu starten, die keine Informationen liefern sondern sonstige Aktionen ausführen.

Bedarf

Die Möglichkeit, Informationen über *WWW* zu erhalten, wird heute in großem Umfang genutzt. Es ist also ratsam, den internen Benutzern die Möglichkeit zu geben, diesen Dienst zu nutzen. Auch zur Präsentation des eigenen Unternehmens über das Internet, wird *WWW* sehr stark eingesetzt. Viele Unternehmen wollen also auch externen Benutzern den Zugang über *WWW* gestatten.

Gefahren

Es ist natürlich möglich, daß ein Angreifer über *WWW* in den Besitz von Dokumenten gelangt, die geheim gehalten werden sollten. Die weitaus größere Gefahr ergibt sich aber nicht durch das einfache Lesen von Dokumenten sondern durch das Ausführen von Programmen über das *CGI*. Hat der Angreifer die Möglichkeit, z. B. über *FTP* ein ausführbares Programm auf einem Rechner abzulegen, so kann er dieses eventuell über das *WWW*-Protokoll zur Ausführung bringen.

mögliche Sicherheitspolitik

Es ist zu empfehlen, internen Benutzern ungehinderte Nutzung des *WWW*-Protokolls zu gestatten, während externe Benutzer nur Zugriff auf einen speziellen *WWW-server* erlangen können. Dieser *server* muß dann natürlich sehr sorgfältig programmiert und konfiguriert sein, damit ein Angreifer keine Ansatzpunkte für eventuelle Angriffe erhält. Insbesondere darf der *WWW-server* keinesfalls in einer Umgebung laufen, in der Dateien über *anonymous FTP* abgelegt werden können.

2.3 Aufstellen der Sicherheitspolitik

Nachdem bekannt ist, welche Risiken die verschiedenen Internet-Dienste bergen und an welchen Diensten Bedarf besteht, kann eine Sicherheitspolitik aufgestellt werden. Tabelle 2.2 stellt beipielhaft dar, welche Politik von einem Unternehmen verfolgt werden könnte.

Dienst	innen → außen	außen → innen
Telnet	uneingeschränkt	Nur nach Authentifikation über one-time passwords
Rlogin	verboten	verboten
E-Mail	uneingeschränkt	uneingeschränkt
FTP	uneingeschränkt	Nur nach Authentifikation über one-time passwords
News	uneingeschränkt	nur bestimmte Newsgruppen sichtbar
Finger	verboten	verboten
X11	verboten	verboten
WWW	uneingeschränkt	Zugriff nur auf einen bestimmten Server

Tabelle 2.2: mögliche Sicherheitspolitik

Kapitel 3

Erstellung eines Kriterienkataloges

Auf dem Firewall-Markt herrscht große Unsicherheit, welches Produkt für die speziellen Bedürfnisse eines Unternehmens am geeignetsten ist. Da jedes Unternehmen eine andere Sicherheitspolitik verfolgt, ist es nicht möglich, allgemeingültige Ratschläge zu erteilen. Vielmehr müssen Bewertungskriterien erstellt werden, die einen objektiven Vergleich von Firewall-Systemen ermöglichen und so zur Auswahl einer geeigneten Lösung bei einer bestimmten Sicherheitspolitik führen können.

Da die Kriterien möglichst allgemeingültig sein sollen und auch noch für zukünftige Entwicklungen einsetzbar sein sollen, ist der sogenannte *bottom-up*-Ansatz nicht geeignet, da bei diesem Verfahren die Kriterien aus einer Untersuchung der Möglichkeiten existierender Implementierungen hervorgehen. Eine bessere Vorgehensweise stellt der *top-down*-Ansatz dar, bei dem die Kriterien aufgrund der Anforderungen an ein Firewall-System aufgestellt werden, gleichgültig ob es derzeit möglich ist, derartige Anforderungen zu erfüllen oder nicht. Der hier vorgestellte Kriterienkatalog wurde nach dem *top-down*-Verfahren erstellt, wobei sich die Kriterien aufgrund ihrer Herleitung folgendermaßen einteilen lassen:

- Herleitung der Kriterien aus den Dienstspezifikationen

  Von entscheidender Bedeutung für die Bewertung eines Firewalls ist es, festzustellen welche Dienste wie gesichert werden können bzw. welche Einschränkungen gemacht werden müssen, um die Dienste sicher anbieten zu können. Um hierbei nicht jeden Dienst einzeln untersuchen zu müssen, wird eine Klassifizierung der Internet-Dienste vorgenommen, die allgemeine Aussagen über Klassen von Diensten erlaubt.

- Herleitung der Kriterien aus den möglichen Bedrohungen

  Hier wurde versucht, die Bedrohungen, die ein Computer-Netz betreffen können, zu klassifizieren, um untersuchen zu können, welcher Firewall welche Art von Angriffen verhindern kann, bzw. welche Einschränkungen für den Netzverkehr entstehen, wenn eine derartiger Angriff unmöglich gemacht werden soll.

- weitere Kriterien

  Zusätzlich stehen noch weitere Kriterien zur Verfügung, deren Herleitung sich nicht einer der obengenannten Methoden zuordnen läßt, die aber wichtige Randbedingungen für die Auswahl eines Firewall-Systems darstellen können.

Die so entstehenden Kriterien werden in späteren Kapiteln auf die allgemeinen Firewall-Konzepte sowie auf eine Auswahl existierender Produkte angewandt. Um den Rahmen der Arbeit nicht zu sprengen, wurde auf eine Anwendung der Kriterien aus den möglichen Bedrohungen auf sämtliche untersuchten Firewalls verzichtet. Hier werden nur einige Besonderheiten erwähnt, in denen sich die entsprechenden Firewalls von den allgemeinen Konzepten unterscheiden.

3.1 Herleitung von Kriterien aus den Dienstdefinitionen

Im Hinblick auf die Bewertung von Firewalls, erscheint eine Unterscheidung der Internet-Dienste in zweierlei Hinsicht sinnvoll, eine Unterscheidung nach dem verwendeten Transportprotokoll sowie eine Unterscheidung nach der Art der Verbindung, *client/server* oder *peer-to-peer*.

- Unterscheidung nach dem Transportprotokoll

  Sämtliche Internet-Dienste setzen entweder auf dem *Transmission Control Protocol* (TCP) oder dem *User Datagram Protocol* (UDP) auf.

  - TCP

    TCP ist ein gesichertes, verbindungsorientiertes Transportprotokoll (siehe Kap. 2.1.2), d. h. es garantiert, daß abgeschickte Pakete in der Reihenfolge ihres Abschickens verläßlich ihr Ziel erreichen. Jedes TCP-Paket enthält Informationen darüber, ob es Teil einer bereits bestehenden Verbindung ist oder nicht. Dies kann ausgenutzt werden, um den Aufbau einer TCP-Verbindung nur in einer Richtung zu gestatten, gleichzeitig aber die Fortsetzung existierender Verbindungen in beiden Richtungen zu erlauben. Dienste, denen TCP zugrundeliegt sind beispielsweise *Telnet* oder *Finger*.

  - UDP

    Im Gegensatz zu TCP handelt es sich bei UDP um ein ungesichertes, verbindungsloses Protokoll (siehe Kap. 2.1.3), das es deshalb auch nicht gestattet, ein eintreffendes Paket dahingehend zu untersuchen, ob es Teil einer bestehenden (logischen) Verbindung ist oder nicht.

- Unterscheidung nach der Art der Verbindung

  Wie bereits erwähnt, lassen sich zwei unterschiedliche Arten von Verbindungen angeben, die *client/server*-Verbindung und die *peer-to-peer*-Verbindung.

  - Client / Server

    Client/Server-Verbindungen sind der weitaus häufiger anzutreffende Verbindungstyp. Ein sogenannter *server* stellt einen Dienst zur Verfügung, der von den sogenannten *clients* in Anspruch genommen werden kann. Der *server* verwendet meist einen für diesen Dienst bekannten *well-known* Port, der, von wenigen Ausnahmen abgesehen, im privilegierten Bereich liegt, während der *client* bei jedem Verbindungsaufbau einen beliebigen Port aus dem nicht-privilegierten Bereich zugewiesen bekommt. Typische Beispiele für *client/server*-Dienste sind *Telnet* und *WWW*.

  - Peer-to-Peer

    Im Gegensatz hierzu stehen die *peer-to-peer*-Verbindungen, bei denen sich die zwei Kommunikationspartner gleichberechtigt gegenüberstehen und gegenseitig mit Informationen versorgen. Bei diesen Diensten haben beide Partner im vorhinein bekannte Ports, was eine Sicherung dieser Dienste erheblich vereinfacht.

- Sonderfälle

  Mit Hilfe der hier aufgezeigten Klassifizierung läßt sich die Beschreibung der Sicherung der meisten Internet-Dienste vornehmen. Allerdings gibt es auch einige Sonderfälle, die einer besonderen Behandlung bedürfen:

  - FTP

    FTP (siehe Kap. 2.2.4) ist ein TCP-basierendes *Client/Server*-Protokoll, das allerdings die Besonderheit enthält, daß auf eine Anfrage eines *clients* hin von Seiten des *servers* aus eine neue Verbindung aufgebaut wird, auf der die Daten übertragen werden. Das bedeutet, daß eine von einem internen Benutzer initiierte Verbindung einen Verbindungsaufbau von außen nach sich zieht, was die Sicherung dieses Dienstes erheblich erschwert.

  - SMTP

    Da *electronic mail* (vgl. Kap. 2.2.3) sowohl von außen als auch von innen ungehindert passieren können muß, muß dieser Dienst natürlich gesondert behandelt werden. Einerseits sollte es möglich sein, für jede eintreffende *mail* einen Log-Eintrag zu erzeugen, andererseits sollten die Nachrichten daraufhin untersucht werden können, ob sie irgendwelche verdächtigen Dinge beinhalten. Ein beliebter Angriff ist es z. B. gewisse Fehler des *sendmail*-Programmes auszunutzen, sodaß dieses Programm die im Inhalt oder Absender der Nachricht angegebenen Kommandos ausführt. Befinden sich also ausführbare Kommandos in einer *mail*, so ist immer Vorsicht geboten, da es sich um einen Angriffsversuch handeln könnte.

  - X11

    Wenn ein Benutzer seinen *x-server* für einen anderen Host zugänglich macht, damit Anwendungen, die auf diesem Rechner laufen, darauf zugreifen können, kann jeder Benutzer dieses Rechners auf den *x-server* zugreifen. Deshalb kann *X11* durch einen Firewall nur mit sehr ausgefeilten Sicherheitsmechanismen gestattet werden (siehe auch Kap. 2.2.8).

  - RPC-basierende Protokolle

    Dienste, die den *Remote Procedure Call* (RPC) [RFC1057] verwenden, müssen ebenfalls als Sonderfälle betrachtet werden, da hierbei die *server* keine festen Portnummern besitzen, sondern vom Betriebssystem dynamisch Portnummern zugewiesen bekommen. Diese Portnummern werden von einem Programm, dem sogenannten *portmapper* verwaltet und können dort erfragt werden. Es ist also schwierig, ein eintreffendes Paket einem bestimmten Dienst zuzuordnen, da die Portnummer keinen Aufschluß darüber gibt, an welchen *server* das Paket adressiert ist. Weiterhin ist problematisch, daß die überwiegende Mehrheit der Dienste auf UDP aufsetzt, wodurch die Behandlung dieser Dienste weiter erschwert wird.

Die Kriterien, die sich aus der Definition der unterschiedlichen Internet-Dienste herleiten lassen, sind also die folgenden:

- Behandlung von TCP-basierenden Client/Server-Diensten
- Behandlung von UDP-basierenden Client/Server-Diensten
- Behandlung von TCP-basierenden Peer-to-Peer-Diensten
- Behandlung von UDP-basierenden Peer-to-Peer-Diensten
- Behandlung von Sonderfällen
  - Behandlung von FTP
  - Behandlung von SMTP
  - Behandlung von X11
  - Behandlung von RPC-basierenden Diensten

3.2 Herleitung von Kriterien aus den möglichen Angriffen

Die Angriffe, denen das zu schützende Netz ausgesetzt ist, lassen sich wie in Abbildung 3.1 dargestellt klassifizieren. Dies ermöglicht es, allgemeine Kriterien herzuleiten, die die Eignung eines Firewalls zur Abwehr von bestimmten Klassen von Angriffen beschreiben. Der folgende Abschnitt beschreibt diese Klassifizierung näher.

3.2.1 Klassifizierung der möglichen Angriffe

Unterscheidung nach der Art des Angriffes

Einem Angreifer stehen zwei grundsätzliche Arten von Angriffen zur Verfügung, die aktiven und die passiven Angriffe:

- passiver Angriff

  Unter einem passiven Angriff versteht man einen Angriff, bei dem der ursprüngliche Zustand des Systems unverändert erhalten bleibt [X.800].

- aktiver Angriff

  Im Gegensatz zum passiven Angriff wird beim aktiven Angriff der Systemzustand geändert [X.800].

  Hierbei ist es unerheblich, ob die Änderung des Zustandes den Grund für den Angriff darstellte oder nur in Kauf genommen wurde, um das eigentliche Ziel des Angriffs zu erreichen.

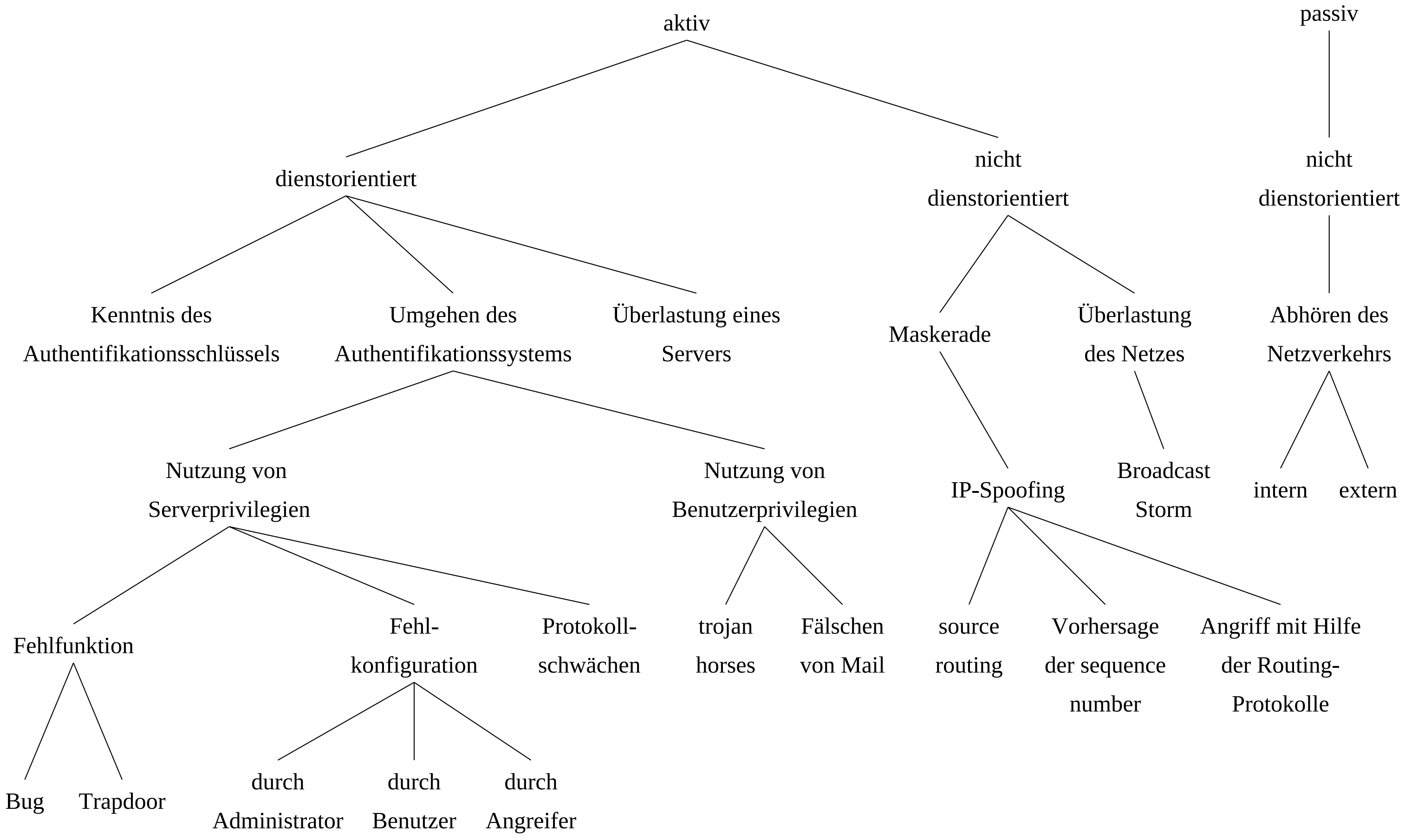

Abbildung 3.1: Klassifizierung der möglichen Angriffe

Unterscheidung in dienstorientierte und nicht-dienstorientierte Angriffe

Je nachdem, ob sich ein Angriff gegen einen bestimmten Dienst richtet oder nicht, kann man dienstorientierte und nicht-dienstorientierte Angriffe unterscheiden.

- dienstorientierte Angriffe

  Jeder Angriff, der einen bestimmten Dienst bzw. einen bestimmten *server* zum Ziel hat, wird im Verlauf dieser Arbeit als dienstorientierter Angriff bezeichnet.

- nicht-dienstorientierte Angriffe

  Nicht-dienstorientierte Angriffe sind Angriffe, die z. B. auf Transport- oder Netzwerkebene angesiedelt sind und somit eingesetzt werden können, um unterschiedlichste Dienste zu attackieren.

Während die passiven Angriffe ausschließlich nicht-dienstorientiert sind, können bei den aktiven Angriffen beide Arten, also dienstorientierte und nicht-dienstorientierte Angriffe auftreten.

Unterscheidung nach der Angriffsmethode

Eine weitere Unterteilung der Angriffe läßt sich vornehmen, wenn man die Angriffsmethoden unterscheidet, die vom Angreifer verwendet werden. Zunächst werden die unterschiedlichen Angriffsmethoden dienstorientierter Angriffe beschrieben:

- Kenntnis des Authentifikationsschlüssels

  Gelingt es dem Angreifer in den Besitz des Authentifikationsschlüssels zu gelangen, so stehen ihm sämtliche Möglichkeiten offen, die auch einem autorisierten Benutzer des Systems zur Verfügung stehen würden.

  Diese Schlüssel, meist handelt es sich um einfache Paßworte, kann der Angreifer mit verschiedensten Methoden erlangen. Ist er erst einmal im Besitz eines gültigen Paßwortes, so kann kein Firewall-System mehr unterscheiden, ob es sich um einen rechtmäßigen Benutzer oder um einen Angreifer handelt. Eine wirksame Verteidigung gegen diese Angriffsmethode ist also nicht vorhanden. Stattdessen muß der Angreifer daran gehindert werden, den Authentifikationsschlüssel zu erlangen bzw. die Wiederverwendbarkeit eines bereits verwendeten Schlüssels muß ausgeschlossen sein.

- Umgehen des Authentifikationssystems

  Bei dieser weitverbreiteten Angriffsmethode versucht der Angreifer, ohne Kenntnis des entsprechenden Schlüssels, Resourcen zu nutzen bzw. Daten zu manipulieren. Hierzu stehen ihm zwei Möglichkeiten zur Verfügung:

  – Nutzung von Server-Privilegien

    Server sind Programme, die dazu dienen, autorisierten Netzteilnehmern gewisse Dienste zu erfüllen, z. B. Daten zur Verfügung zu stellen. Gelingt es nun einem

Angreifer, das Authentifikationssystem des *servers* zu umgehen, so kann er die Dienste des *servers* in Anspruch nehmen, ohne über den erforderlichen Schlüssel zu verfügen.

  – Nutzung von Benutzer-Privilegien

    Unter Umständen kann ein Angreifer einen autorisierten Benutzer dazu bringen, bestimmte Kommandos auszuführen, die entweder den Angriff selber darstellen oder aber weiterführende Angriffe gestatten.

- Überlastung eines Servers

  Wenn ein Angreifer einen *server* mit einer großen Anzahl von Anfragen belastet, so kann es passieren, daß der Server nicht mehr in der Lage ist, sämtliche Anfragen korrekt zu bearbeiten. Der überlastete *server* wird einen Großteil seiner Ressourcen einsetzen, um die Anfragen des Angreifers zu beantworten, was im Extremfall bis zum Ausfall des *servers* aufgrund eines Fehlers oder eines Speicherengpasses wegen der Zwischenspeicherung der eintreffenden Pakete führen kann.

Ebenso wie bei den dienstorientierten Angriffen lassen sich auch die nicht-dienstorientierten Angriffe nach der Angriffsmethode klassifizieren:

- Maskerade

  Maskerade ist der Angriff, bei dem eine bestimmte Einheit, z. B. ein bestimmter Rechner, vorgibt, eine andere Einheit zu sein [X.800]

  Diese Angriffsmethode wird vorwiegend eingesetzt, um *server* zu täuschen, die die IP-Adresse des Absenders als einzige Authentifikation benutzen, sie wird auch als *IP-Spoofing* bezeichnet.

- Überlastung des Netzes

  Ähnlich wie bei der Überlastung eines bestimmten Servers kann man auch das gesamte Netz überlasten, um es für den normalen Datenverkehr unbrauchbar zu machen.

  Beispiel hierfür ist das Auslösen eines sogenannten *broadcast storm*, d. h. das Versenden einer Nachricht an alle Rechner des Netzes, die diese auffordert, ebenfalls eine derartige Nachricht zu verschicken. Somit kann in kurzer Zeit der gesamte Verkehr im Netz zum Erliegen gebracht werden.

- Abhören des Netzverkehrs

  Das Abhören des Netzverkehrs ist die einzige Methode, die für passive Angriffe zur Verfügung steht. Hierbei kann ein Angreifer mit Hilfe geeigneter Software den gesamten Netzverkehr überwachen und somit in den Besitz von Daten gelangen, für die er keine Autorisation besitzt.

  Dieses Abhören des Netzverkehrs, häufig auch als *wire-tapping* bezeichnet, kann man wieder unterscheiden in das Abhören eines externen Netzes sowie das Abhören des internen Netzes.

- Abhören eines externen Netzes

  Das Abhören eines externen Netzes ist jedem möglich, der Zugang zu diesem Netz hat. Somit kann man also nie sicher sein, wenn eine Nachricht über ein öffentliches Netz übertragen wird, daß der Inhalt der Nachricht geheim bleibt. Als Abwehrmöglichkeit bleibt die Verschlüsselung sämtlicher Pakete, die über ein unsicheres Netz transportiert werden und Informationen enthalten, die geheimzuhalten sind.

- Abhören des internen Netzes

  Wenn es einem Angreifer gelungen ist, auf einem internen Rechner *root*-Berechtigung zu erlangen, so kann er auf diesem Rechner Software installieren, die den Verkehr auf dem internen Netz überwacht. Dies ermöglicht es ihm insbesondere, eine große Anzahl von Paßworten mitzuprotokollieren bzw. geheime Informationen zu erfahren. Die Abwehr hiergegen muß natürlich darin bestehen, dem Angreifer den Zugang zum Rechner zu verwehren. Ist er erst einmal in den Rechner eingeloggt, so kann das Abhören kaum mehr verhindert werden.

Unterscheidung nach der Ursache der Angriffsmöglichkeit

Die einzelnen Angriffsmethoden lassen sich wiederum weiter aufgliedern und nach den Ursachen, die den Angriffsmöglichkeiten zugrundeliegen, klassifizieren:

- Nutzung von Server-Privilegien

  Die vorher bereits beschriebene Angriffsmethode der Nutzung von *server*-Privilegien zur Umgehung des Authentifikationssystems läßt sich auf verschiedene Ursachen zurückführen:

  - Fehlfunktion eines Servers

    Ein *server* kann eventuell bei bestimmten Anfragen bzw. Eingaben Funktionen ausführen, die vom Betreiber nicht beabsichtigt sind. Kennt ein Angreifer diese Fehler, so kann er versuchen, sie zu seinen Zwecken zu mißbrauchen. Handelt es sich um einen echten Fehler im Programmcode, also einen Fehler der bei der Erstellung des Programms unbeabsichtigt entstand, so spricht man von einem sogenannten *bug*. Ist die Fehlfunktion vom Entwickler des *servers* gewollt, um später Zugang zum System erlangen zu können, so spricht man von einem *trapdoor*.

  - Fehlkonfiguration eines Servers

    Da die Anforderungen an einen *server* in jeder Umgebung unterschiedlich ausfallen können, muß beim Start des *servers* eine Konfiguration vorgenommen werden. Hier wird beispielsweise festgelegt, welche Benutzer die Möglichkeit haben sollen, die Dienste dieses *servers* in Anspruch zu nehmen. Werden bei der Konfiguration Fehler gemacht, so kann es sein, daß ein Angreifer Zugang zum System erlangt, ohne über die entsprechende Autorisation zu verfügen.

    Die Fehlkonfigurationen können einerseits durch den Systemadministrator verschuldet sein, der beim Start des *servers* eine falsche Konfiguration eingibt, können andererseits aber auch durch unbedarfte Benutzer entstehen. Manche *server* erlauben den Benutzern eine eigenmächtige Erweiterung der Konfiguration, wobei den im Umgang mit sicherheitskritischen Problemen nicht vertrauten Benutzern teilweise beträchtliche Fehler unterlaufen. Der dritte Punkt, wie eine Fehlkonfiguration zustande kommen kann, ist eine Umkonfiguration durch einen Angreifer. Da der Angreifer hierzu aber bereits Zugang zum System haben muß, ist es vorrangig, diesen ersten Angriff zu unterbinden, anstatt zu versuchen, die Konfiguration des *servers* zu schützen.

  - Schwächen in der Protokolldefinition

    Manche Protokolle sind zu einem Zeitpunkt definiert worden, als das Sicherheitsproblem noch nicht seine heutige Bedeutung genoß. Deswegen können diese Protokolle Möglichkeiten bereitstellen, die es einem Angreifer erlauben, Zugang zum System zu erlangen. Insbesondere die Authentifikationsmechanismen vieler Dienste sind viel zu schwach, um einen ausreichenden Schutz vor Angreifern zu bieten.

- Nutzung von Benutzer-Privilegien

  Einen Benutzer kann man auf vielerlei Arten dazu bringen, Kommandos auszuführen, die dem Angreifer Möglichkeiten bereitstellen, einen Angriff auszuführen. Zwei typische Beipiele sind im folgenden erwähnt:

  - trojan horses

    Unter *trojan horses* verteht man Programme, die den Anschein erwecken, eine bestimmte Funktion zu erfüllen, gleichzeitig aber irgendwelche anderen Funktionen ausführen [Garf92].

    Führt nun ein Benutzer ein derartiges Programm aus, so kann das Programm gleichzeitig beliebige Aktionen durchführen, wobei ihm die Privilegien dieses Benutzers zur Verfügung stehen. Ein beliebtes Beispiel hierfür ist die Ersetzung des `ls`-Kommandos durch ein Programm, das zwar ebenfalls das gewünschte Verzeichnis anzeigt, zusätzlich aber noch beliebige andere Funktionen erfüllen kann. Die Verteidigung gehen einen derartigen Angriff muß es wiederum sein, die Installation des *trojan horse* zu verhindern, da zu einem späteren Zeitpunkt kaum mehr eine Verteidigung möglich ist.

  - Fälschen von Mail

    Eine andere Methode, Kommandos unter dem Namen eines rechtmäßigen Benutzers auszuführen ist es, dem Benutzer *mail* unter falschem Namen zu senden, die ihn auffordert, bestimmte Tätigkeiten auszuführen. Beispielsweise ist es denkbar, einem Benutzer unter dem Namen des Systemadministrators eine Nachricht zu schicken, die ihn auffordert, ein bestimmtes vorgegebenes Paßwort statt seinem alten zu verwenden. Ein unbedarfter Benutzer wird einer solchen Nachricht vielleicht Glauben schenken und die gewünschte Aktion ausführen.

- IP-Spoofing

  Es gibt verschieden Möglichkeiten, einen Kommunikationspartner mit einer falschen Absenderadresse zu täuschen. Das große Problem für den Angreifer ist es hierbei, daß die Antworten des *servers* natürlich nicht beim Absender landen sondern an die angegebene (falsche) Adresse gesandt werden. Die wichtigsten Möglichkeiten des *IP-Spoofing* sind die folgenden:

  - Source Routing

    Beim *source routing* [Ches94] bestimmt der Absender den genauen Weg, den das Paket auf dem Weg zum Ziel zurückzulegen hat. Der Empfänger muß seine Antworten auf genau dem umgekehrten Weg versenden. Somit kann man den Empfänger dazu bringen, die Antwortpakete auf einem Umweg über den Rechner des Angreifers zu senden, wo sie vom Netz genommen werden. Auf diese Weise kann ein

Angreifer die Identität eines anderen Rechners übernehmen, ohne das der angegriffene Rechner dies bemerkt (siehe auch Kap. 2.1.1).

– Attacken ohne Rückantwort

  Manche Angriffe erfordern es nicht, daß der Angreifer eine Rückantwort erhält. Hierbei ist es wieder von großer Bedeutung, ob als Transportprotokoll TCP oder UDP verwendert wird.

  * UDP-basierende Dienste

    Bei UDP-basierenden Diensten ist es egal, wenn der Angreifer keine Antworten empfängt, da der angegriffene Rechner keine Informationen darüber erhält, ob die Pakete angekommen sind oder nicht.

  * TCP-basierende Dienste

    Da TCP einen *three-way handshake* beim Verbindungsaufbau durchführt (siehe Kap. 2.1.2), ist es normalerweise erforderlich, daß die Antwortpakete beim richtigen Absender eintreffen. Dieser muß nämlich die vom seinem Kommunikationspartner übermittelte *sequence number* bestätigen. Gelingt es nun aber, diese Nummer vorauszusagen, so kann eine Verbindung aufgebaut werden, ohne daß der Angreifer je ein Antwortpaket erhalten muß [Bello89].

– Attacken mit Hilfe der Routing-Protokolle

  Die dritte Möglichkeit ist nun mit Hilfe der Routing-Protokolle [Bello89]. Verbreitet ein Rechner die Nachricht, daß er eine besonders günstige Route zu einem bestimmten Rechner anbieten kann, so werden die anderen Router und Gateways Pakete für diesen Rechner auf diesem Weg versenden. Somit kann, ähnlich wie beim *source routing*, die Identität eines anderen Rechners angenommen werden.

3.2.2 Die Ziele der unterschiedlichen Angriffe

Ein Angreifer kann mit einem Angriff unterschiedliche Ziele verfolgen. Die vier wichtigsten Ziele sollen hier kurz dargestellt werden:

- nicht-autorisierte Kenntnis von Daten

  Ziel eines Angriffs kann es sein, in den Besitz von Informationen zu gelangen, die autorisierten Benutzern vorbehalten sind. Hierbei kann es sich um geheime Unternehmensdaten handeln, deren Kenntnis dem Angreifer materielle Vorteile bringen kann, es kann sich aber auch um Informationen handeln, die er für die Durchführung weiterer Angriffe benötigt. Insbesondere die Kenntnis von Paßworten ist ein wichtiges Ziel jedes Angreifers.

- nicht-autorisierte Manipulation von Daten

  Ein weiteres Ziel eines Angriffs kann es sein, Daten zu manipulieren. Beispiel hierfür ist die Manipulation von Kontoständen in einer Bank. Auch hier ist es wieder möglich, daß die Veränderung der Daten nicht das eigentliche Angriffsziel darstellt, sondern nur für weiterführende Angriffe dient. Beispielsweise kann ein Angreifer eine Konfigurationsdatei dahingehend manipulieren, daß ihm der Zugang zu bestimmten *servern* gewährt wird.

	Kenntnis von Daten	Manipulation von Daten	Nutzung von Ressourcen	Denial of Service
Kenntnis der Authentifikationsschlüssels	+	+	+	+
Umgehen des Authentifikationssystems	+	+	+	+
Maskerade	+	+	+	+
Überlastung eines Servers	–	–	–	+
Überlastung des Netzes	–	–	–	+
Abhören des Netzverkehrs	+	–	–	–

Tabelle 3.1: Zusammenhang zwischen Angriffsmethoden und -zielen

- nicht-autorisierte Nutzung von Ressourcen

  Ein Angreifer kann ebensogut versuchen, bestimmte Ressourcen zu nutzen, die in einem Unternehmen zur Verfügung stehen. Beispielsweise besonders leistungsfähige Rechner können ein Ziel eines derartigen Angriffs darstellen.

- Denial-of-Service

  Denial-of-Service-Angriffe dienen dazu, die Benutzung einer bestimmten Ressource im Netz zu verhindern. Da heutzutage der gesamte Arbeitsablauf in sehr starkem Maße von Computern und deren Vernetzung abhängt, kann man mit einem derartigen Angriff die Produktivität eines Unternehmens empfindlich stören.

In der Tabelle 3.1 ist der Zusammenhang der unterschiedlichen Ziele von Angriffen mit den verschiedenen Angriffsmethoden dargestellt.

3.3 weitere Kriterien

Neben den bereits erwähnten Kriterien, die sich aus den Angriffen bzw. aus den Dienstspezifikationen ergeben, existieren noch weitere, die zum Teil erheblichen Einfluß auf die Auswahl einer bestimmten Firewall-Lösung haben können. Einige dieser Kriterien sind im folgenden aufgezählt.

1. **Audit**

   - Logging
     Um einen versuchten oder geglückten Angriff erkennen zu können und entsprechende Gegenmaßnahmen einleiten zu können, ist es erforderlich, daß der Firewall

ein umfangreiches Logging zur Verfügung stellt. Unter Logging versteht man das Mitprotokollieren sämtlicher Ereignisse, die für den Administrator von Bedeutung sein könnten. Hierbei kann es sich z. B. um folgende Ereignisse handeln:

- Transport eines Paketes
- Verwerfen eines Paketes
- Verbindungsaufbau
- Verbindungsabbau
- Authentifikationsfehler
- Konfigurationsänderungen

Von Bedeutung ist natürlich auch die Art der Darstellung dieser Informationen. Aufgrund der großen Anzahl von Einträgen, ist es einem menschlichen Betrachter unmöglich, diese zu überwachen. Es wäre also wünschenswert, wenn z. B. alle Pakete einer Verbindung zu einem einzigen Eintrag zusammengefaßt werden könnten. Auch ein Eintrag in unterschiedliche Dateien, je nach Art des Ereignisses, ist sinnvoll.

Möchte man die Kosten des Internet-Anschlusses auf die Benutzer umlegen, die den Anschluß tatsächlich in Anspruch genommen haben, so ist es weiterhin erforderlich, daß die Log-Einträge die Benutzernamen sowie das übertragene Datenvolumen enthalten. Mit den entsprechenden Tools ist dann ein problemloses Accounting denkbar.

- Alerting

  Unter Alerting versteht man die Fähigkeiten eines Firewalls, im Falle eines versuchten Angriffes eine verantwortliche Person zu verständigen. Unterschiede hinsichtlich der Alerting-Fähigkeiten ergeben sich insbesondere im Hinblick auf die Auswahl der Ereignisse, bei denen eine Benachrichtigung erfolgen soll sowie in der Art der Benachrichtigung. Hier ist es natürlich erstrebenswert, daß der Firewall-Betreiber selbst definieren kann, bei welchem Ereignis eine Benachrichtigung erfolgen soll. Mögliche Methoden der Benachrichtigung sind z. B.

  - E-Mail
    Für Ereignisse, die keine sofortige Reaktion erfordern, kann es ausreichen, einen Verantwortlichen über E-Mail zu benachrichtigen.
  - Öffnen eines Fenster
    Ebenso kann man auf dem Monitor des entsprechenden Mitarbeiters ein Fenster öffnen, das die notwendigen Informationen enthält.
  - akustische Signale
    Der Firewall kann ein akustisches Signal abgeben, das einen Verantwortlichen zur Untersuchung der Ursache des Signals auffordert.
  - Versenden eines Fax
    Tritt ein Ereignis außerhalb der normalen Arbeitszeit auf, so ist nicht unbedingt davon auszugehen, daß der verantwortliche Mitarbeiter bei Verwendung der bisher erwähnten Methoden so rechtzeitig davon Kenntnis erlangt, daß er Gegenmaßnahmen einleiten kann. Hier bietet sich die Möglichkeit, ihn mittels eines Faxes zu informieren.

   - Verwendung eines Pagers
     Möchte man sichergehen, daß der Mitarbeiter schnellstmöglich über das Ereignis informiert wird, so bietet sich die Möglichkeit, ihn mit Hilfe eines Pagers zu verständigen.
   - benutzerdefinierte Kommandos
     Ideal ist natürlich auch die Möglichkeit, selber ein Kommando oder ein Programm angeben zu können, daß im Falle eines bestimmten Ereignisses zur Ausführung kommt.

2. sicherheitrelevante Kriterien

   - Authentifikationsmöglichkeiten
     Möchte man den Zugang von außen in das interne Netz zulassen, so stehen verschiedene Maßnahmen zur Authentifikation des Benutzers zur Verfügung. Diese bieten ein unterschiedliches Maß an Sicherheit und Benutzerfreundlichkeit. Die derzeit verbreitetsten Authentifikationsmechanismen werden im folgenden genannt:
     - Standard UNIX-Paßworte
       Die Verwendung von Standard UNIX-Paßworten zur Sicherung des Zugangs von außen stellt ein großes Sicherheitsrisiko dar. Da die Paßworte im Klartext über das Netz übertragen werden, kann ein Angreifer, der das Netz abhört, Paßworte mithören und sich später mit Hilfe dieser Paßworte in den entsprechenden Rechner einloggen.
     - one-time passwords
       One-time passwords können nur ein einziges Mal verwendet werden. Gelingt es einem Angreifer also, ein derartiges Paßwort abzuhören, so kann er es nicht verwenden, um sich zu einem späteren Zeitpunkt Zugang zu dem entsprechenden System zu verschaffen. Die verschiedenen Möglichkeiten der Realisierung von *one-time passwords* werden in Anhang A beschrieben. Die wichtigsten Produkte, die derzeit auf dem Markt erhältlich sind, sind die folgenden:
       * Security Dynamics SecureID Cards
         Bei diesem System besitzt der Benutzer ein Gerät, daß abhängig von der Zeit und einer Geheimzahl Paßworte erzeugt.
       * Digital Pathways Secure NetKey (SNK)
         Es handelt sich um ein *challenge/response*-Verfahren, bei dem der Benutzer vom *server* eine Zeichenkette erhält, aus der er mit Hilfe eines Gerätes und einer Geheimzahl eine Antwort errechnen muß.
       * Bellcore S/Key
         Dieses Verfahren berechnet eine gewisse Anzahl von Paßworten im voraus, die dann ausgedruckt und der Reihe nach eingesetzt werden können.
   - **Sicherheit der Firewall-Komponenten**
     Von großer Bedeutung ist auch die Sicherheit der Firewall-Komponenten selbst. Hat ein Angreifer die Möglichkeit, die Konfiguration des Firewalls zu ändern, so ist das gesamte Unternehmensnetz ungeschützt. Kriterien, die die Sicherheit der Firewall-Komponenten bestimmen, sind z. B.:

- Fehlerfreiheit der Firewall-Software
  Enthält die Firewall-Software selber Fehler, die zu Angriffen führen können, so ist der entsprechende Firewall natürlich praktisch nutzlos. Insbesondere besteht diese Gefahr, wenn die Hersteller auf frei verfügbare Produkte zurückgreifen, deren *source-code* allen Angreifern zur Verfügung steht, so daß eventuell vorhandene Fehler von diesen auch relativ schnell gefunden werden können.
  Von Vorteil ist es hier, wenn die vom Firewall eingesetzten Programme in einer Umgebung laufen, in der selbst ein Fehler des Programms keine größeren Schwierigkeiten bereiten kann. Insbesondere sollte vor Start des Programms ein `chroot`-Kommando ausgeführt werden, mit dessen Hilfe dem Programm ein neues *root*-Verzeichnis zugewiesen werden kann. Enthält dieses Verzeichnis dann keine wichtigen Systemdateien und auch keine ausführbaren Files, so kann auch ein eventueller Fehler nur sehr schwer ausgenutzt werden. Weiterhin ist wichtig, daß die Firewall-Software möglichst nicht unter *root*-Privilegien läuft, damit ein eventueller Fehler nicht allzu großen Schaden anrichten kann.
- Existenz von User-Accounts
  Auf dem Firewall sollten keinerlei User-Accounts existieren, da man normalerweise nicht davon ausgehen kann, daß ein Benutzer ein hinreichend sicheres Paßwort wählt. Existieren User-Accounts auf dem Firewall, so dürfen sie keinesfalls nur mit Standard UNIX-Paßworten gesichert sein. Ein weiteres Problem ist, daß bei einer größeren Anzahl eingeloggter Personen die Entdeckung eines geglückten Angriffs natürlich deutlich erschwert wird.
- periodische Prüfsummenbildung
  Die Firewall-Software kann in regelmäßigen Abständen mit Hilfe einer Prüfsumme überprüfen, ob Änderungen an der Konfiguration vorgenommen wurden. Sollte dies der Fall sein, so ist umgehend jede bestehende Verbindung abzubrechen und auch keine weitere Verbindung mehr zuzulassen.
- administrativer Zugang über das Netz
  Verbietet man den administrativen Zugang zum Firewall über das Netz, so steigt die Sicherheit natürlich erheblich, da ein Angreifer, der die Konfiguration des Firewalls verändern möchte, Zugang zur Firewall-Konsole haben müßte. Andererseits ist ein derartiger Zugang aus Gründen der Benutzerfreundlichkeit anzustreben.

• hierarchische Sicherheitsabstufungen
  Es kann von Vorteil sein, wenn der Firewall nicht nur das interne Netz gegen das externe absichern kann, sondern wenn er weiterhin noch bestimmte Subnetze oder sogar einzelne *server* besonders schützen kann. Das kann bedeuten, daß bestimmte Subnetze auch vor Zugriff aus dem internen Netz geschützt werden sollen oder das bestimmte Subnetze überhaupt keinen Zugang zum externen Netz haben sollen, da die dort gespeicherten Daten um jeden Preis geheim gehalten werden müssen.

• Domain Name Service
  Gelingt es einem Netzbetreiber, die Adressen, die er innerhalb seines Netzes verwendet, geheimzuhalten, so bietet er einem Angreifer keinen Ansatzpunkt, wie er seinen Angriff beginnen könnte. Deshalb ersetzen einige Firewalls die Absenderadresse jedes Pakets durch die Adresse des Firewalls und tragen erst bei Eintreffen des Antwortpaketes die entsprechende Zieladresse wieder ein. Somit erfährt weder

der Kommunikationspartner noch ein eventueller Angreifer, der das externe Netz abhört, eine andere Adresse als die des Firewalls selber. Solange nur die Adresse des Firewalls bekannt sein muß, kann man auch einen zweigeteilten *name server* verwenden, der nach außen hin nur die Adresse des Firewalls sowie die Adresse eines Rechners zur Weiterleitung von Mail bekannt macht, während nach innen alle Rechner des internen und externen Netzes bekannt sind. Ob dies aber einen echten Sicherheitsgewinn darstellt, ist fraglich, da einem Angreifer noch verschiedene andere Möglichkeiten zur Verfügung stehen, Adressen möglicher Angriffsziele herauszufinden. Da dies aber auf jeden Fall durch die Geheimhaltung deutlich erschwert wird, sollte nach Möglichkeit nicht darauf verzichtet werden.

Ein weiterer Grund für die Verwendung eines zweigeteilten *name servers* kann sein, daß das interne Netz keine offiziell registrierte Adressen verwendet. Dies ist zwar nicht empfehlenswert, läßt sich aber bei der derzeitigen Knappheit an Adressen nicht unbedingt vermeiden. Besitzt man einen zweigeteilten *server*, so kann man beliebige Adressen im internen Netz verwenden, da diese nie der Außenwelt bekanntgegeben werden.

3. Administration

- Installation und Konfiguration

  Ein wichtiges Kriterium ist die Installation und Konfiguration des Firewalls. Gestaltet sie sich schwierig, so ist entweder Unterstützung des Herstellers notwendig oder es besteht die Möglichkeit, daß bereits hier Fehler passieren, die die spätere einwandfreie Funktion des Firewalls beeinträchtigen können.

  Zu untersuchen ist hier insbesondere, ob die Installation und Konfiguration mit Hilfe einer umständlichen und schwer zu erlernenden Kommandosprache erfolgt oder ob eine graphische Benutzeroberfläche existiert, die ebenso übersichtlich wie leicht zu erlernen ist.

  Ebenso ist es wichtig, das der Firewall offensichtliche Konfigurationsfehler selbstständig erkennt und den Betrieb erst aufnimmt, wenn eine sinnvolle und konsistente Konfiguration vorliegt.

- Support

  Der Support durch den Hersteller ist natürlich speziell bei schwer zu bedienenden Systemen von großer Bedeutung. Ausreichender Support ist sehr wichtig, um die reibungslose Funktion des Firewalls zu gewährleisten. Hierzu zählt ebenfalls die rechtzeitige Versorgung mit neuen Versionen des Firewalls, falls eine Angriffsmöglichkeit bekannt geworden ist, die mit Hilfe dieses Firewalls nur unzureichend abgesichert ist.

- Dokumentation

  Um die richtige Konfiguration des Firewalls zu ermöglichen, ist eine umfangreiche Dokumentation unerläßlich. Diese muß neben den Konfigurationshinweisen auch ausführliche Methoden zum Testen der erstellten Konfiguration enthalten, da die wenigsten Firewalls so übersichtlich konfigurierbar sind, daß man sich auf fehlerfreie Funktion ohne Tests verlassen kann.

- SNMP-Unterstützung

  Ein weiterer interessanter Aspekt ist die SNMP-Unterstützung[RFC1157]. Inzwischen werden bereits Firewalls angeboten, die über einen SNMP-Agenten verfügen. Dies ermöglicht es, Statusinformationen über den Firewall mit Hilfe eines Managementsystems anzufordern. Hier besteht natürlich die Gefahr, daß diese Informationen einem potentiellen Angreifer in die Hände fallen könnten, der sie wiederum als wichtige Informationen für die Durchführung eines Angriffs einsetzen könnte.

  Das Konfigurieren des Firewalls über SNMP ist derzeit auf keinen Fall anzuraten, da das SNMP-Protokoll zu viele Sicherheitsmängel beinhaltet und somit möglicherweise ein Angreifer in die Lage versetzt wird, die Konfiguration des Firewalls über SNMP zu verändern.

- Behandlung proprietärer bzw. neuer Dienste

  Möchte man proprietäre Dienste über den Firewall einsetzen, so muß er Methoden bereitstellen, auch diese Dienste sicher zu behandeln. Dies läuft normalerweise über ein dienstunabhängiges Firewall-Konzept wie z. B. einen Paket-Filter oder ein TCP-Relay, die um die entsprechenden Filterregeln erweitert werden müssen.

  Ebenso verhält es sich bei der Entwicklung neuer Standard-Dienste. Das Beispiel *WWW* zeigt, mit welcher Geschwindigkeit sich ein neuer Dienst im Internet etablieren kann, so daß nicht auszuschließen ist, daß bereits in einem oder zwei Jahren neuartige Dienste existieren, die ebenfalls geschützt werden müssen. Verfügt der Firewall dann über kein erweiterbares Konzept, so muß auf diese neuen Dienste verzichtet oder eine neue Version des Firewalls erworben werden.

4. Benutzerfreundlichkeit

- Transparenz

  Ein ganz wesentlicher Punkt ist die Transparenz des Firewalls. Man kann drei unterschiedliche Arten von Transparenz unterscheiden:

  - Transparenz für den Benutzer
    Ein anzustrebendes Ziel ist es, den Firewall für den Benutzer transparent zu gestalten. Wird der Verbindungsaufbau durch mehrstufige Sicherheitsabfragen verkompliziert, so müssen die Benutzer ihre Arbeitsgewohnheiten ändern. Kommt es zu Unzufriedenheit bei den Benutzern aufgrund der umständlichen Handhabung des Firewall-Systems, so kann es passieren, daß sie versuchen, den Firewall zu umgehen und somit die Sicherheit des ganzen Netzes gefährden.
  - Transparenz für die Anwendungen
    Ebenso bedeutsam ist es, ob der Firewall für die Anwendungen transparent ist. Ist dies nicht der Fall, so müssen sämtliche *clients* abgeändert werden, sodaß sie mit dem Firewall zusammenarbeiten. Abgesehen von der Tatsache, daß dies bei großen Netzen einen riesigen Aufwand darstellen kann, ist es nur für *client*-Programme möglich, für die der *source-code* vorliegt.
  - völlige Transparenz
    Optimal wäre natürlich die völlige Transparenz des Firewalls sowohl gegenüber den Anwendungen als auch gegenüber den Benutzern.

- Performance

  Die Performance des Firewalls ist selbstverständlich auch von großer Bedeutung. Wenn aufgrund der vielen Überprüfungen, denen die einzelnen Pakete unterzogen werden, die Übertragungsleistung zu gering wird, so kann es sein, daß unzufriedene Benutzer dazu übergehen, den Firewall zu umgehen. Dies ist zm Beispiel über einen eigenen Modem-Anschluß denkbar und setzt die Sicherheit des gesamten Unternehmensnetzes aufs Spiel.

5. **Kosten**

- Kaufpreis

  Die Preise der einzelnen Firewalls unterscheiden sich ganz erheblich. Natürlich ist beim Erwerb eines Firewalls dessen Preis ein zu bedenkender Faktor, wobei aber aufgrund eines günstigeren Preises keine Abstriche in Bezug auf die Sicherheit gemacht werden sollten.

- Hardware-Voraussetzungen

  Ähnlich wie beim Preis verhält es sich bei den Hardware-Voraussetzungen. Es gibt Lösungen, die einen PC erforden, es gibt aber auch Lösungen, für die drei Workstations notwendig sind. Auch dies stellt einen großen Kostenfaktor dar. Auch der Einsatz eines zusätzlichen Routers kann die Kosten deutlich erhöhen. Stehen in einem Unternehmen bereits Rechner zur Verfügung, die für den Firewall verwendet werden können, so ist natürlich von Bedeutung, auf welchen Rechnertypen der Firewall lauffähig ist. Man sollte aber keinesfalls einen weniger geeigneten Firewall einsetzen, nur weil bereits die entsprechenden Rechner zur Verfügung stehen.

Kapitel 4

Vorstellung allgemeiner Firewall-Konzepte

Nachdem in Kapitel 2.2 die wichtigsten Internet-Dienste vorgestellt wurden, wobei auch darauf eingegangen wurde, in welcher zum Teil eingeschränkten Form diese angeboten werden sollten, soll nun aufgezeigt werden, welche Möglichkeiten zur Verfügung stehen, diese Dienste zu sichern. Hierzu wird zuerst einmal untersucht, welche Informationen den verschiedenen in Frage kommenden Geräten zur Verfügung stehen, bevor dann die Vor- und Nachteile der einzelnen Lösungen betrachtet werden.

4.1 verfügbare Informationen

Um entscheiden zu können, ob ein eintreffendes Paket mit der vom Unternehmen vorgegebenen Sicherheitspolitik vereinbar ist, müssen Informationen über dieses Paket zur Verfügung stehen, die mit vorgegebenen Anforderungen verglichen werden können. Hierzu bietet sich der Paket-Header an, der wichtige Informationen über das jeweilige Paket enthält. Je nachdem, welches Gerät als Firewall eingesetzt wird, kann nun der Paket-Header bis zu einer bestimmten Schicht untersucht werden. In Tabelle 4.1 wird aufgezeigt, welchem Gerät welche Informationen zur Verfügung stehen.

Gerät	OSI-Schicht	verfügbare Informationen
Host	7	Kommandos, Inhalte
	4	Port-Nummern, TCP-Flags
Router	3	IP-Adressen, Transportprotokoll
Brücke	2	MAC-Adressen
	1	Interface

Tabelle 4.1: Informationen, die den unterschiedlichen Geräten zur Verfügung stehen

Die Brücke

Die Brücke ist ein Koppelelement, das auf Schicht 2 des OSI-Referenzmodells angesiedelt ist. Daher kann sie nur Informationen der Schichten eins und zwei verwenden, um eine Entscheidung zu treffen, ob das Paket weitergeleitet werden soll oder nicht. Die hier verwendbaren Informationen sind:

- Interface

  Man kann erkennen, an welchem Interface ein Paket eintrifft und somit bestimmen, ob es aus dem internen oder aus dem externen Netz stammt.

- MAC-Adresse

  Jedes Paket entält im Header die MAC-Adresse des Absenders sowie des Ziels. Somit ergibt sich die Möglichkeit, Pakete nur von bestimmten Adressen an bestimmte Adressen zuzulassen. Da aber die MAC-Adresse nicht unbedingt den tatsächlichen Absender bzw. den tatsächlichen Empfänger angibt, sondern eventuell nur den jeweils vorhergehenden bzw. folgenden Router, kann eine Transportentscheidung aufgrund der MAC-Adresse nur in wenigen Fällen sinnvoll eingesetzt werden.

Es zeigt sich also, daß die Informationen, die mit Hilfe einer Brücke gewonnen werden können, nicht ausreichen, um die Sicherheitspolitik eines Unternehmens ausreichend zu realisieren. Es ist also erforderlich, Geräte einzusetzen, die mehr Daten zur Verfügung stellen.

Der Router

Ein Router wird zur Koppelung auf Schicht 3 eingesetzt. Ihm stehen also zusätzlich zu den Daten, die auch die Brücke erkennt, weitere Informationen zur Verfügung. Aus Sicherheitssicht erscheinen die folgenden am bedeutendsten (vgl. Kap. 2.1.1):

- IP-Adresse

  Die weltweit eindeutige Adresse des Absenders bzw. Empfängers des Pakets. Hier bietet sich die Möglichkeit, nur bestimmten externen Rechnern eine Verbindung zum internen Netzwerk zu gestatten bzw. umgekehrt, sowie weiterhin für gewisse interne Rechner jeden Zugang von und nach außen zu verweigern, um diese somit vor Angriffen zu schützen.

- Protokoll

  Auf Schicht 3 läßt sich weiterhin das auf Schicht 4 eingesetzte Transportprotokoll erkennen. Da, wie später noch genauer erklärt wird, UDP gewisse Probleme bei der Sicherung bereitet, kann z.B. der UDP-Verkehr aufgrund dieser Information vollständig unterbunden werden.

- Port-Nummer

  Auch wenn die Funktionalität des Routers auf Schicht 3 angesiedelt ist, kann er trotzdem den Header der Pakete bis zur Schicht 4 inspizieren und somit auch die Port-Nummer, sowohl des Senders als auch des Empfängers, ermitteln (siehe Kap. 2.1.2). Das bedeutet, er kann feststellen, an welchen Prozeß ein Paket adressiert ist und damit die Sicherheitspolitik des Unternehmens recht effizient implementieren.

- Flags

  Ebenfalls der Schicht 4 zuzurechnen sind die Flags des TCP-Headers, mit deren Hilfe sich ermitteln läßt, ob ein von außen eintreffenden Paket Teil einer existierenden TCP-Verbindung ist oder dem Neuaufbau einer solchen Verbindung dient. So kann man Verbindungen zwischen zwei Rechnern erlauben, wenn sie von innen initiiert wurden und verbieten, wenn der Verbindungswunsch von außen stammte.

Man kann erkennen, daß mit Hilfe eines Routers deutlich mehr Informationen ermittelt werden können als mit einer Brücke. Router lassen sich einsetzen, um einfache aber sinnvolle Firewalls zu implementieren.

Der Host

Die Möglichkeiten, die ein Host bietet, gehen wiederum deutlich über die Möglichkeiten eines Routers hinaus. Die wichtigsten Informationen, die mit einem Host im Gegensatz zum Router untersucht werden können, sind die folgenden:

- Kommando

  Da ein Host in der Lage ist, die Protokolle bis zur Schicht 7 abzuwickeln, ist es möglich, zu erkennen, welches Kommando vom Benutzer gesendet wurde. Bei Protokollen, bei denen nur bestimmte Kommandos ein Sicherheitsproblem darstellen würden, ist es somit möglich, genau diese Kommandos zu unterbinden und den Dienst somit trotzdem, wenn auch in einer eingeschränkten Form, anzubieten.

- Inhalte

  Ebenso wie die übermittelten Kommandos können auch die zu übertragenden Paketinhalte einer näheren Untersuchung unterzogen werden. Z. B. ist es möglich, E-Mails daraufhin zu untersuchen, ob sie ausführbare Kommados enthalten, ein Trick der bei vielen Einbruchversuchen verwendet wird (siehe Kap. 3.1). Derartige Mails können somit verworfen werden, ohne den *mail*-Dienst für die Benutzer einzuschränken.

4.2 Der Paket-Filter

Unter einem Paket-Filter [Ches94] versteht man einen Firewall, der die Transportentscheidung über ein eintreffendes Paket aufgrund der Header-Informationen bis zur Schicht 4 des OSI-Modells trifft, also aufgrund von (siehe Kapitel 2.1):

- Adresse des Absenders
- Port des Absenders

- Adresse des Empfängers
- Port des Empfängers
- Transportprotokoll
- Flags
- Interface

Aus diesem Grund kommen hierfür Router in Frage, wobei es natürlich auch möglich ist, einen Host als Paket-Filter zu verwenden. Ein Paket-Filter verfügt üblicherweise über zwei oder mehr Interfaces, was eine Konfiguration ermöglicht, bei der kein Paket vom externen in das interne Netz bzw. umgekehrt gelangen kann, ohne den Paket-Filter zu passieren (Siehe Abbildung 4.1).

Abbildung 4.1: Verbindung zweier Netze mit Hilfe eines Paket-Filters

Die allgemeine Arbeitsweise eines Paket-Filters besteht darin, jedes eintreffende Paket in Hinblick auf seine Quell- und Zieladresse sowie auf Quell- und Zielport und Transportprotokoll zu untersuchen. Diese Informationen werden mit einer Reihe von Regeln verglichen, die eine Entscheidung ermöglichen, ob das Paket der Sicherheitspolitik des Unternehmens entspricht oder nicht. Sollte dies der Fall sein, so wird das Paket weitergeleitet, andernfalls verworfen. Es wird kein Kontext über die Verbindung aufrechterhalten, weshalb die Entscheidung über den Transport eines Paketes nur aufgrund der Header-Informationen des jeweiligen Paketes erfolgen kann.

4.2.1 Programmierung von Paket-Filtern

Um einen Paket-Filter zu programmieren, geht man in drei Schritten vor:

- Erstellen einer Sicherheitspolitik

  Wie bei jedem anderen Firewall-Konzept auch, muß zu Beginn entschieden werden, welche Dienste zwischen internem und externem Netz angeboten werden sollen, bzw. welche Einschränkungen bei einzelnen Diensten gemacht werden müssen.

- Erstellen logischer Terme aus den Bestandteilen der Pakete

  Man muß nun versuchen, die vorher festgelegte Sicherheitspolitik nur mit Hilfe der im Paket-Filter zur Verfügung stehenden Informationen zu formulieren. Hier kann man bereites auf ernsthafte Schwierigkeiten treffen, da sich nicht alle gewünschten Einschränkungen der Dienste mit Hilfe von Paket-Filtern realisieren lassen. Auch ist es keine leichte Aufgabe, diese Terme richtig festzulegen, da sich Abhängigkeiten zwischen den einzelnen Termen ergeben können, die auf den ersten Blick nicht auffallen.

- Formulierung der Terme in der entsprechenden Syntax

  Nachdem die Terme aufgestellt sind, müssen sie in die vom jeweiligem Hersteller angebotene Syntax überführt werden, damit die Sicherheitspolitik aktiviert werden kann.

Erstellung logischer Terme

Es ist relativ einfach, mit Hilfe von Paket-Filtern Verbindungen zwischen zwei bestimmten Hosts zu erlauben oder auch zu verbieten. Da aber die Sicherheitspolitik es meist erforderlich macht, daß eine genauere Kontrolle ausgeübt wird, werden normalerweise zusätzlich zu den IP-Adressen zumindest noch die Ports sowie das Transportprotokoll überprüft. Auch kann mit Hilfe des ACK-Flags untersucht werden, ob es sich um einen Verbindungsaufbau oder die Fortsetzung einer bestehenden TCP-Verbindung handelt (vgl. Kap. 2.1.2).

Dies soll an einem kleinen Beispiel verdeutlicht werden. Angenommen, man möchte *mail* von und zu einem bestimmten *mailserver* innerhalb des Firmennetzes von allen externen Hosts aus erlauben. Jeder andere Verkehr durch den Firewall soll verboten sein. Ein Filter der dies implementiert könnte wie in Tabelle 4.2 gezeigt aussehen. Bei den folgenden Beispielen wurde jeweils auf eine Angabe des *protocol*-Feldes verzichtet, um die Tabellen nicht zu unübersichtlich werden zu lassen. Mit Ausnahme der letzten Regel handelt es sich immer um TCP, die letzte Regel sollte für beliebige Protokolle gelten.

Nr.	Aktion	Source	Port	Dest.	Port	Flags	Bemerkung
1	allow	mailserver	*	*	*	*	Der *mailserver* darf beliebige Pakete versenden
2	allow	*	*	mailserver	25	*	externe Hosts dürfen nur Verbindungen zum SMTP-Port unseres *mailservers* aufbauen
3	allow	*	*	mailserver	*	ACK	externen Hosts wird die Fortsetzung beliebiger Verbindungen mit dem *mailserver* gestattet
4	block	*	*	*	*	*	Alles andere ist verboten

Tabelle 4.2: Einsatz des ACK-Flags bei der Filterprogrammierung

Regel 1 dient dazu, daß der *mailserver* Verbindungen zu externen Rechnern aufbauen kann. Da auf eine Untersuchung des ACK-Flags verzichtet wird, erlaubt diese Regel gleichzeitig die Fortsetzung bereits bestehender Verbindungen. Regel 2 hingegen erlaubt es externen Rechnern, Kontakt zum *mailserver* aufzunehmen. Der Verbindungsaufbau wird dabei aber auf Verbindungen zum SMTP-Port (Port 25) beschränkt. Um es den externen Rechnern zu ermöglichen, auf von innen initiierte Verbindungen zu antworten, wurde Regel 3 eingefügt.

Sie gestattet jedes Paket, daß an einen beliebigen Port des *mailservers* gerichtet ist, es sei denn, dieses Paket hat das ACK-Flag nicht gesetzt und dient somit also zum Neuaufbau einer Verbindung. Eine Regel in der Art von Regel 4 sollte grundsätzlich den Schluß der Filterregeln bilden, um alle Pakete, die nicht explizit durch die anderen Filterregeln gestattet werden, zu verwerfen. Da die Regeln in der angegebenen Reihenfolge abgearbeitet werden und abgebrochen wird, wenn eine Regel auf das entsprechende Paket angewandt werden konnte, kommt diese Regel nur dann zum Einsatz, wenn keine andere Regel auf das eingetroffene Paket paßt.

Ein nächstes Beispiel (Tabelle 4.3) zeigt, daß es möglich ist, mehrere Rechner zu einer Gruppe zusammenzufassen, um somit nicht für jeden Rechner eigene Regeln aufstellen zu müssen. In diesem Beispiel soll die Behandlung der *mail* analog zum vorherigen Beispiel erfolgen, während es zusätzlich allen internen Rechnern möglich sein soll, *telnet*-Verbindungen zu externen Rechnern aufzubauen.

Nr.	Aktion	Source	Port	Dest.	Port	Flags.	Bemerkung
1	allow	mailserver	*	*	*	*	Der *mailserver* darf beliebige Pakete versenden
2	allow	*	*	mailserver	25	*	externe Rechner dürfen Verbindungen zum *mail*-Port unseres *mailservers* aufbauen
3	allow	*	*	{our hosts}	*	ACK	externen Rechnern ist die Fortsetzung bereits bestehender Verbindungen zu allen internen Rechnern erlaubt
4	allow	{our hosts}	*	*	23	*	interne Rechner dürfen *telnet*-Verbindungen zu beliebigen Rechnern aufbauen
5	block	*	*	*	*	*	Alles andere ist verboten

Tabelle 4.3: Zusammenfassen mehrerer Rechner zu einer Gruppe

Die Regel 1 und 2 erfüllen in diesem Beispiel die gleichen Aufgaben wie im vorhergehenden. Regel 3 wurde dahingehend abgeändert, daß nun externe Hosts Verbindungen, die von beliebigen internen Rechnern initiiert wurden, fortsetzen können. Dies schließt natürlich insbesondere den *mailserver* mit ein. Um *telnet*-Verbindungen von innen nach außen zu erlauben, wurde Regel 4 eingefügt. Sie erlaubt es allen internen Rechnern, Verbindungen zu Port 23 eines beliebigen Rechners aufzubauen. Die Fortsetzung dieser Verbindungen durch den externen Rechner erfolgt dann wiederum mit Hilfe von Regel 3. Hierbei ist allerdings zu beachten, daß man sich nicht darauf verlassen kann, daß auf einem externen Rechner Port 23 für den *telnet-server* verwendet wird. Dies ist zwar der *well-known port* für *telnet*, eine Garantie hierfür besteht aber nicht. Regel 5 sorgt wieder dafür, daß keine Pakete den Filter passieren können, die nicht ausdrücklich dazu berechtigt wurden.

Ebenso wie es möglich ist, Gruppen von Hosts zu bilden, mit deren Hilfe sich einfachere Regeln aufstellen lassen, kann man ganze Portbereiche zulassen oder ausschließen, ohne für jeden Port eine eigene Regel verwenden zu müssen. Das in Tabelle 4.4 dargestellte Beispiel soll dies verdeutlichen.

Nr.	Aktion	Source	Port	Dest.	Port	Flags	Bemerkung
1	allow	*	*	{out hosts}	*	ACK	Externe Hosts dürfen bestehende Verbindungen fortsetzen
2	allow	{our hosts}	*	*	*	*	Interne Hosts dürfen Verbindungen auch initiieren
3	block	*	*	{our hosts}	6000-6100	*	Hier sind die X-Server angesiedelt
4	allow	*	*	{our hosts}	$\geq$1024	*	Verbindungen zu nicht-privilegierten Ports sind erlaubt
5	block	*	*	*	*	*	Alles andere ist verboten

Tabelle 4.4: Zulassen bzw. Ausschließen ganzer Portbereiche

Mit Hilfe von Regel 1 und 2 wird es wieder allen internen Hosts ermöglicht, Verbindungen aufzubauen, auf die alle externen Hosts antworten können. Da die Regeln in der Reihenfolge abgearbeitet werden, in der sie aufgeschrieben wurden, wurde die Regel, die vermutlich am häufigsten zutrifft, an den Anfang gesetzt. Dies führt zu einer Verbesserung der Performance, da weniger Regeln untersucht werden müssen, ist aber auch gefährlich, da gewisse Abhängigkeiten zwischen den Regeln bestehen können, so daß die Reihenfolge nicht unbedingt geändert werden kann. Regel 4 erlaubt es externen Hosts, Verbindungen zu Ports im nicht-privilegierten Bereich aufzubauen. Dies kann zum Beispiel bei FTP-Verbindungen notwendig sein, da hierbei der *server* einen Datenkanal zum *client* auf einem nicht-privilegierten Port eröffnet. Da aber auch im nicht-privilegierten Bereich einige wichtige *server* angesiedelt sein können, werden vor diese Regel noch weitere Regeln gesetzt, die diese Ports ausdrücklich ausschließen. Im Beispiel wurde mit Hilfe der Regel 3 der ganze Bereich ausgeschlossen, in dem *X-server* angesiedelt sein können. Da die Regeln der Reihe nach abgearbeitet werden, wird Regel 4 gar nicht mehr beachtet, wenn Regel 3 bereits zutrifft. Regel 5 blockt wie üblich alle Pakete ab, auf die keine andere Regel zutrifft.

Weiterhin von Bedeutung ist die Möglichkeit, das Interface festzulegen, auf dem ein Paket von einem bestimmten Host eintreffen muß. Eine beliebte Angriffsmethode ist es, als Absender eine Adresse des internen Netzes anzugeben, auch wenn das Paket aus dem externen Netz stammt. Legt man nun fest, daß Pakete aus dem firmeneigenen Netz nur an einem bestimmten Interface erscheinen können, so können solche Versuche unterbunden werden (siehe Abbildung 4.2/Tabelle 4.5).

Die Regel sorgt dafür, daß jedes Paket, dessen Absenderadresse einen Host des internen Netzes bezeichnet, verworfen wird, sofern es an Interface 1, also dem Interface zum externen Netz ankommt.

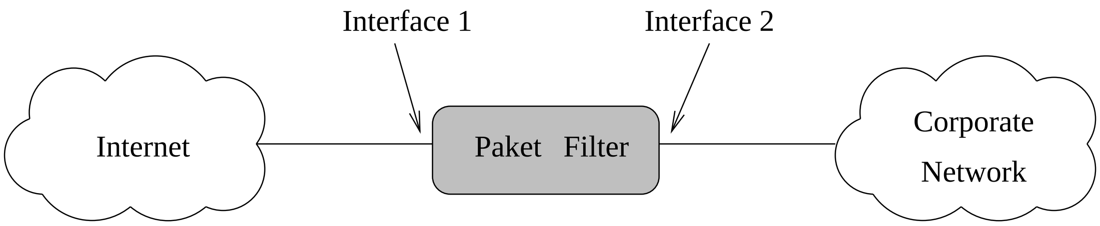

Abbildung 4.2: Paket-Filter mit Unterscheidung des empfangenden Interfaces

Nr.	Aktion	Source	Port	Dest.	Port	Interface	Bemerkung
1	block	{our hosts}	*	*	*	Interface1	Alle Pakete, die als Absender einen internen Host haben aber auf Interface 1 eintreffen sind verboten

Tabelle 4.5: Unterscheidung nach dem empfangenden Interface

Formulierung der Terme in der entsprechenden Syntax

Hier soll nun gezeigt werden, wie die aufgestellten Terme tatsächlich implementiert werden könnten. Dies soll am Beispiel der Programmierung eines CISCO-Routers dargestellt werden [Cisco]. Insbesondere interessant sind hierbei die Möglichkeiten, ganze Portbereiche mit einer einzelnen Regel abzudecken, sowie mehrere Hosts mit einer Regel anzusprechen.

Ein CISCO-Router verfügt über die Operatoren `lt` (less than) und `gt` (greater than), mit deren Hilfe alle Ports, die größer oder kleiner als der angegebene sind, angesprochen werden können. Es besteht aber keine Möglichkeit, einen sowohl nach oben als auch nach unten begrenzten Portbereich anzugeben. Außerdem kann ein CISCO-Router nicht den Port des Absenders überprüfen. Dies ist aber auch nur selten erforderlich, wie aus den vorher angegeben Beispielen ersichtlich wird.

Möchte man darauf verzichten, für jeden einzelnen Host eine Regel aufzustellen, so kann man durch die Verwendung von Subnetz-Masken die jeweilige Regel auf ein ganzes Subnetz anwenden. Hierzu gibt man zusätzlich zur Adresse eine Maske an, die angibt, welche Bits der Adresse untersucht werden sollen. Es werden nur die Bits der Adresse untersucht, die in der Maske nicht gesetzt sind.

Als Beispiel soll nun die in der Tabelle 4.4 dargestellte Politik in CISCO-Syntax überführt werden. Abbildung 4.3 zeigt das Netz, daß für dieses Beispiel verwendet wird, während die Implementierung in Tabelle 4.6 dargestellt ist.

Es werden zwei sogenannte *access-lists* definiert, eine für das Interface zum Internet und eine für das Interface zum privaten Netz. Diese werden mit den Regeln 6–10 den entsprechenden Interfaces zugeordnet.

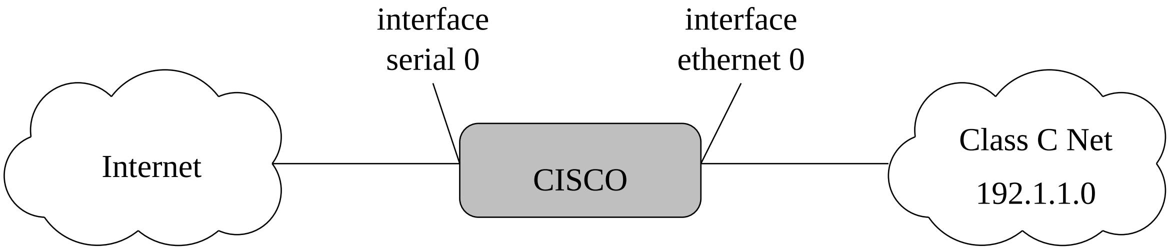

Abbildung 4.3: Beispielsszenario

```
1 access-list 101 permit tcp 192.1.1.0 0.0.0.255 0.0.0.0 255.255.255.255
2 access-list 102 permit tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 established
3 access-list 102 block tcp 0.0.0.0 255.255.255.255 192.1.1.0 0.0.0.255 lt 1024
4 access-list 102 permit tcp 0.0.0.0 255.255.255.255 192.1.1.0 0.0.0.255 lt 6000
5 access-list 102 permit tcp 0.0.0.0 255.255.255.255 192.1.1.0 0.0.0.255 gt 6100
6 interface serial 0
7 ip access-group 101
8 interface ethernet 0
9 ip access-group 102
```

Tabelle 4.6: Filterregeln beim CISCO-Router

- access-list 101

  Die *access-list* 101 dient dazu, daß jeder Rechner des privaten Netzes Verbindungen zu jedem beliebigen externen Rechner aufbauen und fortsetzen kann. Sie entspricht der Zeile 1 der Tabelle 4.4.

- access-list 102

  Die Zeilen 2–5 definieren die *access-list* 102. Sie entspricht den Zeilen 1, 3 und 4 der Tabelle 4.4. Zeile 2 erlaubt die Fortsetzung bestehender Verbindungen, während die anderen drei Zeilen notwendig sind, um den Portbereich unter 1024 sowie zwischen 6000 und 6100 zu verbieten. Zeile 3 verbietet zunächst jede Verbindungseröffnung zu einem nicht-privilegierten Port. Die Fortsetzung von Verbindungen zu diesen Ports ist aber trotzdem aufgrund von Regel 2 möglich. Regel 4 bzw. Regel 5 gestatten daraufhin Verbindungen zu den Ports, die kleiner als 6000 oder größer als 6100 sind.

Da ein CISCO-Router automatisch eine letzte Regel einfügt, die alles andere verbietet, implementieren diese Regeln die gewünschte Politik. Insbesondere wird hierdurch der Bereich zwischen Port 6000 und Port 6100 ausgeschlossen, der nicht ausdrücklich erlaubt wurde.

4.2.2 Bewertung des Paket-Filter Konzepts

Im folgenden soll nun versucht werden, das Paket-Filter Konzept mit Hilfe der im vorangegangenen Kapitel aufgestellten Kriterien zu bewerten [Ches94].

Bewertung aufgrund der Kriterien aus den Dienstspezifikationen

- Behandlung von TCP-basierenden Client/Server-Diensten

  Da man bei Eintreffen eines Paketes im Paket-Filter anhand des ACK-Flags probemlos unterscheiden kann, ob es sich bei diesem Paket um einen neuen Verbindungsaufbau oder um die Fortsetzung einer bereits bestehenden Verbindung handelt, ist die übliche Art und Weise, TCP-basierende *client/server*-Dienste mit Paket-Filtern zu sichern, die folgende: Interne Benutzer dürfen sämtliche Dienste nach außen hin nutzen, während externe Benutzer keine Verbindungen nach innen initiieren dürfen. Somit ist gewährleistet, daß kein Angreifer in der Lage ist, eine Verbindung zu einem internen *server* herzustellen.

- Behandlung von UDP-basierenden Client/Server-Diensten

  Da UDP ein verbindungsloses Protokoll ist, beinhalten UDP-basierende Pakete keinerlei Informationen darüber, ob es sich bei einem Paket um die Fortsetzung einer Verbindung handelt oder nicht. Eine Politik, wie sie bei den TCP-basierenden *client/server*-Diensten beschrieben wurde, ist hier nicht anwendbar, da man nicht erkennen kann, ob ein eintreffendes Paket die Antwort auf eine Anfrage eines internen Benutzers ist oder ein versuchter Angriff. Die einzige Möglichkeit, die bleibt um die Gefahren, die UDP-basierende *client/server*-Dienste mit sich bringen, ist das vollständige Verbot derartiger Dienste.

- Behandlung von TCP-basierenden Peer-to-Peer-Diensten

  Die Behandlung TCP-basierender *peer-to-peer*-Dienste gestaltet sich ähnlich wie die der *client/server*-Dienste. Wiederum werden nur Antworten des externen Partners gestattet, wobei diesmal zusätzlich noch der Port bekannt ist, auf den diese Antworten gesendet werden.

  Da ja bereits im vorhinein feststeht, welche Rechner über welche Ports miteinander kommunizieren werden, kann auch die Sicherheitspolitik dahingehend gelockert werden, daß bestimmte externe Rechner Verbindungen aufbauen können. Knackt dann aber ein Angreifer den externen Rechner oder nimmt er auf andere Art und Weise dessen Identität an, so besteht die Gefahr, daß über diese Lücke Angriffe durchgeführt werden können.

- Behandlung von UDP-basierenden Peer-to-Peer-Diensten

  Möchte man UDP-basiernde *peer-to-peer*-Dienste erlauben, so besteht keine andere Möglichkeit, als Pakete zwischen den entsprechenden Rechnern, die an die entsprechenden Ports adressiert sind, zu gestatten. Im Gegensatz zu den *client/server*-Diensten ist es hier ja nicht erforderlich, Antworten zu beliebigen Ports zuzulassen, da sämtliche Antworten an einen vorbestimmten Port adressiert werden. Dies kann aber, wie oben bereits beschrieben, Probleme ergeben, wenn es ein Angreifer schafft, die Identität des externen Partners anzunehmen (siehe Seite 32). Die Sicherheitspolitik eines Unternehmens muß festlegen, ob dieses Risiko eingegangen werden soll oder nicht.

- Behandlung von Sonderfällen
  - Behandlung von FTP

    Die Sicherung von FTP mit Hilfe eines Paket-Filters gestaltet sich relativ schwierig, da FTP normalerweise eine Datenverbindung erfordert, die vom *server* zum *client* aufgebaut wird (siehe Kap. 2.2.4). Somit ist eine Politik, wie sie bei den normalen *client/server*-Diensten beschrieben wurde, nicht durchführbar. Es bestehen im wesentlichen zwei Möglichkeiten, die im folgenden beschrieben werden sollen:

    Da die Datenverbindung auf *client*-Seite einen Port aus dem nicht-privilegierten Bereich verwendet, kann man einfach sämtliche Verbindungen zu nicht-privilegierten Ports gestatten. Dies beinhaltet allerdings große Gefahren, da auch im Bereich dieser Ports *server* angesiedelt sein können, die ein Angreifer benutzen kann, um Angriffe durchzuführen. Diese würde auch dem Grundsatz 'Alles was nicht ausdrücklich erlaubt ist, ist verboten' widersprechen.

    Die zweite Möglichkeit ist deutlich sicherer, ist aber auch nicht so leicht zu realisieren. Wie bereits erwähnt (siehe Kapitel 2.2.4), kann man ein sog. *PASV*-Kommando an den Server schicken, was diesen veranlaßt eine Portnummer mitzuteilen, zu der der *client* den Datenkanal öffnen kann. Somit handelt es sich wieder um eine Verbindung von innen nach außen, die mit Hilfe des ACK-Flags gesichert werden kann. Hierzu ist es allerdings erforderlich, die FTP-*clients* dahingehend zu verändern, daß sie das *PASV*-Kommando verwenden, wobei aber nicht sicher ist, daß jeder angesprochene *server* auch in der Lage ist, ein derartiges Kommando zu akzeptieren.
  - Behandlung von SMTP

    Paket-Filter sind in der Lage, den Austausch von Nachrichten nur zwischen einem speziellen *mailserver* und dem externen Netz zu gestatten. Somit kann man den Großteil der Rechner vor Angriffen schützen, setzt aber den *server* allen Gefahren des *SMTP*-Protokolls aus. Hier ist dann unbedingt darauf zu achten, daß ein sicheres Programm als *mailserver* eingesetzt wird.
  - Behandlung von X11

    Es ist nicht möglich, *X11* mit Hilfe eines Paket-Filtes ausreichend abzusichern, da da Protokoll zu viele Gefahren enthält (vgl. Kap. 2.2.8). Es bleibt nur die Möglichkeit, den gesamten Bereich, in dem *x-server* angesiedelt sein können (TCP-Ports 6000-6100) zu verbieten. Man sollte aber nur Verbindungen von außen unterbinden, da man andernfalls Probleme bekommen kann, wenn einem beliebigen *client* zufällig ein Port aus diesem Bereich zugewiesen werden sollte.
  - Behandlung von RPC-basierenden Diensten

    Auch die sichere Behandlung von RPC-basierenden Diensten ist mit einem Paket-Filter unmöglich. Da nicht bekannt ist, welcher Dienst auf welchem Port läuft, ist es nicht möglich, einzelne Dienste zu erlauben. Aus diesem Grund sollten sämtliche RPC-basierenden Dienste verboten werden.

Bewertung aufgrund der Kriterien aus den möglichen Angriffen

- Kenntnis des Authentifikationsschlüssels

  Mit Hilfe eines Paket-Filters kann man keine zusätzlichen Maßnahmen zur Authentifizierung der Identität des Angreifers vornehmen. Ist einem Angreifer also ein für den entsprechenden *server* gültiger Authentifikationsschlüssel bekannt, so stehen sämtliche Möglichkeiten dieses *servers* zur Verfügung, es sei denn, der entsprechende Dienst wird völlig verboten bzw. von außen nicht zugänglich gemacht.

- Umgehen des Authentifikationssystems

  Versucht ein Angreifer die Privilegien eines bestimmten *servers* zu seinen Zwecken zu mißbrauchen, so stehen je nach Ursache der Angriffsmöglichkeit die folgenden, recht eingeschränkten Möglichkeiten zur Absicherung zur Verfügung:

  - Fehlfunktion eines Servers

    Da *bugs* und *trapdoors* in jedem Programm enthalten sein können, gibt es keinen hundertprozentigen Schutz gegen Angriffe, die dies ausnutzen. Die einzige Möglichkeit ist es, *server*, die bekanntermaßen über Fehler verfügen, durch andere zu ersetzen bzw. die Dienste, für die keine hinreichend sicheren *server* existieren nicht von außen zugänglich zu machen.

  - Fehlkonfiguration eines Servers

    Da es meist unmöglich ist, daß der Betreiber des Firewalls gleichzeitig die Konfiguration sämtlicher *server* innerhalb des zu sichernden Netzes übernimmt, ist es nicht auszuschließen, daß hier Fehler auftreten können. Hier bleibt nur die Möglichkeit, die entsprechenden Systemadministratoren auf die hierdurch entstehenden Gefahren aufmerksam zu machen und Dienste, die aufgrund einer schwierigen Konfiguration eine Gefahr darstellen, ausschließlich von innen nach außen zu gestatten.

    Handelt es sich um Dienste, bei denen der Benutzer Einfluß auf die Konfiguration nehmen kann, so sollten sie verboten werden, da man sich keinesfalls darauf verlassen kann, daß die Benutzer ihre Konfiguration im Hinblick auf die Sicherheit des Netzes wählen. Benutzer legen im allgemeinen mehr Wert auf Bequemlichkeit als auf Sicherheit bzw. verfügen meist gar nicht über das erforderliche Fachwissen, um eine entsprechend sichere Konfiguration vorzunehmen.

    Resultiert die Fehlkonfiguration aus einem bereits erfolgten Angriff, so besteht natürlich keine Möglichkeit, diesen Dienst mit Hilfe eines Paket-Filters zu sichern.

  - Schwächen in der Protokollspezifikation

    Handelt es sich um einen Dienst, der über keine ausreichenden Sicherheitsmaßnahmen verfügt, so sollte er verboten werden, da auch mit Hilfe eines Paket-Filters keine Möglichkeiten zur Verfügung stehen, derartige Dienste hinreichend abzusichern.

  Versucht ein Angreifer hingegen durch das Ausnutzen von Benutzerprivilegien seinen Angriff durchzuführen, so stellt ein Paketfilter keine Möglichkeiten zur Verfügung dies zu verhindern. Gelingt es einem Angreifer, ein *trojan horse* zu installieren, so kann mit keiner Art von Firewall mehr etwas dagegen unternommen werden, da die Programme ja daraufhin von autorisierten Benutzern ausgeführt werden. Beim Versenden gefälschter

Mails kann ein Paket-Filter ebenfalls nichts unternehmen, da er nicht in der Lage ist, den Absender einer Mail zu überprüfen und zu verifizieren.

- Überlastung eines Servers

  Ein Paket-Filter ist nicht in der Lage, einen *server* vor Überlastung zu schützen, da er zustandslos ist und somit bei Untersuchung eines Paketes nicht weiß, ob bereits eine große Anzahl gleichartiger Pakete an diesen *server* gesendet wurde.

- Maskerade

  Maskerade mittels *IP-Spoofing* ist ein Thema, daß speziell bei der Verwendung von Paket-Filtern besondere Bedeutung genießt. Da Paket-Filter sich bei der Identifizierung des Absenders eines Paketes nur auf die angegebene IP-Adresse verlassen können, eröffnet es einem Angreifer große Möglichkeiten, dies zu hintergehen. Paket-Filter können im einzelnen folgende Maßnahmen gehen *IP-Spoofing* ergreifen:

  - Source-Routing
    Die wohl einfachste Art des *IP-Spoofing*. Ebenso einfach ist aber auch die Verteidigung gegen derartigen Angriffe. Da es praktisch keinen sinnvollen Grund gibt, Pakete mit *source-routing* Option zuzulassen, können derartige Pakete einfach verworfen werden.
  - Attacken mit Hilfe der Routing-Protokolle
    Verwendet ein Paket-Filter statische Routingtabellen, d. h. verläßt er sich nicht auf Informationen, die ihm durch andere Router (oder aber durch den Angreifer) mitgeteilt werden, so kann ein solcher Angriff gegen den Paket-Filter selber keinen Erfolg bringen. Gelingt es einem Angreifer aber, einen anderen Router auf dem Weg des Paketes zu seinem Ziel umzukonfigurieren, so können die Pakete doch an ein verkehrtes Ziel geleitet werden, wodurch der Angreifer eine falsche Identität annehmen kann.

- Überlastung des Netzes

  Paket-Filter können einer Überlastung des Netzes entgegenwirken, indem sie *broadcasts* zurückweisen.

- Abhören des Netzverkehrs

  Ein Paket-Filter kann das Abhören des Netzverkehrs nicht verhindern. Er kann weder die Adressen des internen Netzes geheimhalten, noch kann er eine Verschlüsselung des Verkehrs vornehmen, um somit die Inhalte der Pakete vor eventuellen Angreifern zu verbergen.

Bewertung aufgrund weiterer Kriterien

1. **Audit**

   Paket-Filter verfügen meist nicht über ausreichende Audit-Mechanismen. Wird ein Router als Paket-Filter eingesetzt, so ist keinerlei Audit vorhanden, während ein als Paket-Filter eingesetzter Host natürlich sämtliche Logging- und Alerting-Möglichkeiten anbieten kann. Da aber die Pakete aufgrund der Zustandslosigkeit des Paket-Filters einzeln geloggt werden, ist eine Weiterbearbeitung der Log-Einträge meist unumgänglich.

2. sicherheitsrelevante Kriterien

- Sicherheit der Firewall-Komponenten

  Die Sicherheit der Firewall-Komponenten kann nicht anhand eines Konzeptes aufgezeigt werden. Hierzu ist ein konkretes Produkt notwendig. Die einzig allgemeingültige Aussage, die hier gemacht werden kann, ist die, daß eine Existenz von User-Accounts auf dem Firewall bei Verwendung des Paket-Filter-Konzeptes nicht erforderlich ist.

- Domain Name Sevice

  Paket-Filter erlauben es nicht, die Adressen des internen Netzes geheimzuhalten, sodaß jeder Angreifer, der in der Lage ist, das externe Netz abzuhören, Informationen über mögliche Angriffsziele innerhalb des Unternehmensnetzes erhalten kann.

3. Administration

- Installation und Konfiguration

  Die Installation und Konfiguration eines Paket-Filters ist im allgemeinen nicht sehr aufwendig, sie muß allerdings mit sehr großer Sorgfalt geschehen, da zwischen den einzelnen Filterregeln Abhängigkeiten bestehen können, die nur durch ausführliches Testen entdeckt werden können. Auch erfordert die Syntax, in der die Regeln formuliert werden müssen, häufig eine gewisse Eingewöhnungszeit (siehe Seite 48).

- Behandlung proprietärer bzw. neuer Dienste

  Bei Verwendung eines Paket-Filters fügt man neue bzw. proprietäre Dienste durch einfaches Hinzufügen einiger weniger Filterregeln ein. Aufzupassen ist hierbei wieder auf die Abhängigkeiten, die sich zu den bereits existierenden Regeln ergeben können.

Die übrigen Kriterien aus dem Bereich der Administration lassen sich nur anhand eines konkreten Produktes untersuchen, sodaß hier nicht weiter darauf eingegangen werden kann.

4. Benutzerfreundlichkeit

- Transparenz

  Paket-Filter sind sowohl für den Benutzer als auch für die beteiligten Anwendungen transparent. Ein Benutzer bemerkt die Existenz eines Paket-Filters einzig beim Versuch, einen nicht gestatteten Dienst zu benutzen.

- Performance

  Die Performance eines Paket-Filters wird durch die Anzahl der zu untersuchenden Regeln bestimmt. Solange die meisten Pakete mit der Untersuchung einiger weniger Regeln behandelt werden können, leidet die Performance nicht. Deshalb ist es sehr wichtig, bei der Erstellung der Regeln darauf zu achten, daß die am häufigsten greifenden Regeln am Anfang stehen und somit die Untersuchung der übrigen Regeln in der Vielzahl der Fälle entfallen kann.

5. **Kosten**

   Die Kosten für einen Paket-Filter sind meist relativ gering. Da in Verbindung mit einem Internet-Anschluß meist sowieso ein Router vorhanden sein wird, können die Regeln in diesem implementiert werden. Ist dies nicht der Fall, so besteht sogar die Möglichkeit, Regeln beim Service-Provider zu implementieren, wobei man sich natürlich auf die Sicherheit der Komponenten des Providers verlassen muß.

4.3 Das TCP-Relay

TCP-Relays (oder auch *circuit-level gateways*) dienen ausschließlich der Sicherung von TCP-Verbindungen. Sie benutzen ähnlich wie der Paket-Filter nur die Informationen der Ebenen eins bis vier. Der Unterschied zum Paket-Filter ist der, daß der Benutzer nicht eine direkte Verbindung zu dem gewünschten *server* herstellt, sondern eine Verbindung zum Gateway. Das Gateway untersucht daraufhin, ob die gewünschte Verbindung zulässig ist und stellt die Verbindung her. Die Untersuchung der Zulässigkeit der Verbindung kann im Gegensatz zum Paket-Filter benutzerbezogen vorgenommen werden. Nachdem die Verbindung aufgebaut wurde, untersucht das Gateway die Pakete, die zu dieser Verbindung gehören, nicht mehr. Es reicht die Pakete nur noch durch, so daß es für den Benutzer erscheint, als ob das Gateway nicht vorhanden wäre [Ches94].

4.3.1 Programmierung von TCP-Relays

Da TCP-Relays ebenfalls auf Informationen der Ebenen eins bis vier zurückgreifen, erfolgt ihre Programmierung ähnlich wie die eines Paket-Filters. Da hier aber ein Kontext über eine bestehende Verbindung aufrechterhalten wird, ist es nur erforderlich, das erste Paket der Verbindung zu untersuchen. Das bedeutet, daß eine Untersuchung z. B. des ACK-Flags unnötig ist, da es ohnehin beim ersten Paket einer Verbindung nicht gesetzt sein kann.

Erstellung logischer Terme

Als Beispiel soll hier eine Konfiguration gezeigt werden, die die Benutzung des *finger*-Dienstes durch den Firewall völlig verbietet, *telnet*-Verbindungen durch den Firewall nur internen Hosts erlaubt und *mail* sowohl von innen als auch von außen ungehindert passieren läßt. Alles andere ist verboten (siehe Tabelle 4.7)).

Nr.	Aktion	Source	Port	Dest.	Port	Bemerkung
1	block	*	*	*	finger	Vollständiges Verbot von Finger
2	allow	{our hosts}	*	*	telnet	interne Hosts dürfen Telnet nutzen
3	allow	*	*	*	smtp	Verbindungen zum Mail-Port sind immer erlaubt
4	block	*	*	*	*	Alles andere ist verboten

Tabelle 4.7: Filterregeln für ein TCP-Relay

Man erkennt, daß die einzelnen Regeln nur aus fünf Elementen bestehen, der Aktion sowie der Adresse und Portnummer des Senders bzw. des Empfängers. Die Regeln werden wiederum in der vorgegebenen Reihenfolge auf das eintreffende Paket angewandt, bis daß eine Regel zutrifft. Regel 1 verhindert *finger*-Verbindungen durch den Firewall zwischen beliebigen Sende- bzw. Empfangsadressen. Die Regel 2 dient nun dazu, den Hosts des internen Netzes die Möglichkeit zu geben, *telnet*-Verbindungen zu externen Hosts zu eröffnen, die von diesen natürlich ungehindert fortgesetzt werden können. Regel 3 erlaubt uneingeschränkte Nutzung des *mail*-Dienstes, während Regel 4 alle Verbindungen unterbindet, auf die keine der vorherigen Regeln angewandt werden konnte.

Auch beim TCP-Relay ist es natürlich wieder möglich, Gruppen von Rechnern zu bilden und ganze Portbereiche mit einer einzigen Regel zu gestatten oder zu verbieten.

Formulierung der Terme in der entsprechenden Syntax

Wie die oben beschriebene Politik in eine konkrete Implementierung überführt werden könnte ist in Tabelle 4.8 am Beispiel des frei erhältlichen TCP-Relays *Socks* dargestellt [Koblas92].Für die hier dargestellte Implementierung wurde wiederum ein internes Klasse-C-Netz mit der Adresse 192.1.1.0 angenommen, wie es in Abbildung 4.4 skizziert wird.

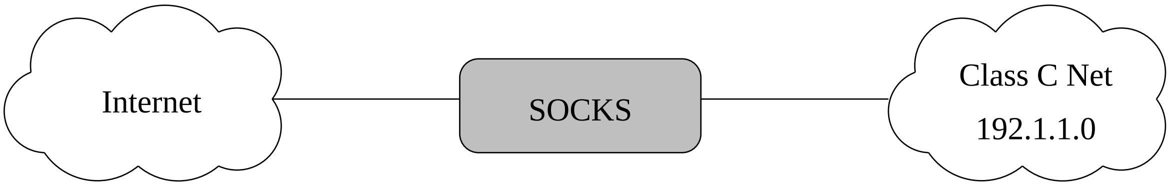

Abbildung 4.4: Beispielsszenario

Die Adressen von Sender und Empfänger werden mit Hilfe von Netzmasken eingegeben, um ganze Netze mit einer einzelnen Konfigurationszeile behandeln zu können. Eine Portnummer läßt sich bei *Socks* nur für die Empfängerseite angeben, weshalb jede Zeile nur aus den vier Elementen Aktion, *source adress*, *destination adress* und *destination port* besteht.

Nr.	Aktion	Source	Mask	Dest.	Mask	Port
1	deny	0.0.0.0	255.255.255.255	0.0.0.0	255.255.255.255	eq finger
2	allow	192.1.1.0	0.0.0.255	0.0.0.0	255.255.255.255	eq telnet
3	allow	0.0.0.0	255.255.255.255	0.0.0.0	255.255.255.255	eq mail

Tabelle 4.8: Programmierung von SOCKS

Die Zeilen entsprechen denen der Tabelle 4.7. Durch die Verwendung der Netzmaske 0.0.0.255 in Regel 2 wird die letzte Zahl der Adresse ignoriert. Deshalb dürfen alle Hosts des Subnetzes 192.1.1.0 den Telnet-Dienst nutzen. Die Netzmasken 255.255.255.255 in den Regeln 1 und 3 dienen dazu, jeden beliebigen Host in die Regel einzuschließen. Auf eine abschließende

Regel, die alle weiteren Verbindungen unterbindet, kann verzichtet werden, da dies von *Socks* automatisch so gehandhabt wird.

4.3.2 Bewertung des TCP-Relay Konzepts

Bewertung aufgrund der Kriterien aus den Dienstspezifikationen

- Behandlung von TCP-basierenden Client / Server-Diensten

  TCP-Relays sind hervorragend geeignet, um TCP-basierende *client/server*-Dienste abzusichern. Üblich ist hierbei wieder das Verbot des Verbindungsaufbaus von außen bei gleichzeitigem Gestatten von innen initiierter Verbindungen. Verbindungen von außen können nur dann zugelassen werden, wenn das Relay über zusätzliche Authentifikationsmöglichkeiten verfügt.

- Behandlung von TCP-basierenden Peer-to-Peer-Diensten

  Noch einfacher gestaltet sich natürlich die Sicherung von TCP-basiernden *client/server*-Diensten. Je nach Sicherheitspolitik werden entweder wieder nur von innen nach außen gehende Verbindungen zwischen den entsprechenden *peer*-Prozessen gestattet oder es werden Verbindungen in beiden Richtungen erlaubt, was aber zu den bereits in Zusammenhang mit dem Paket-Filter beschriebenen Problemen führen kann, wenn es einem Angreifer gelingt, die Identität eines anderen Rechners anzunehmen.

- Behandlung von UDP-basierenden Diensten

  TCP-Relays wurden ursprünglich zur Sicherung von TCP-basierenden Diensten entwickelt, neuere Versionen sollen aber in der Lage sein, auch UDP-basierende Dienste mit einem ähnlichen Maß an Sicherheit auszustatten.

- Behandlung von Sonderfällen

  - Behandlung von FTP

    Bei der Sicherung von FTP ergeben sich die gleichen Probleme wie bei der Verwendung eines Paket-Filters. Die Datenverbindung, die von außen ausgehend zu einem im vorhinein unbekannten Port aufgebaut werden muß, kann nicht gestattet werden, da man zu diesem Zweck den gesamten nicht-privilegierten Portbereich zugänglich machen müßte. Was bleibt, ist wiederum die Umgehung des Problems durch Abänderung der *clients*, sodaß diese PASV-Kommados absetzen und damit eine von außen initiierte Verbindung überflüssig machen.

- Behandlung von SMTP

  Auch bei der Behandlung von *SMTP* ähneln sich die Konzepte Paket-Filter und TCP-Relay. Es besteht wiederum nur die Möglichkeit, Verbindungen nur zu einem speziellen *mailserver* zu erlauben, sodaß die übrigen Rechner vor Angriffen über *SMTP* geschützt sind. Der *mailserver* ist aber wieder allen Angriffen schutzlos ausgesetzt.

- Behandlung von X11

  Es besteht keine Möglichkeit, *X11* sicher mit Hilfe eines TCP-Relays anzubieten.

- Behandlung RPC-basierender Dienste

  Die meisten RPC-basierenden Dienste benutzen ohnehin UDP als Transportprotokoll, sind also mit einem TCP-Relay auch theoretisch derzeit nicht zu sichern. Aber auch die TCP-basierenden Dienste können nicht angeboten werden, da sie keine festen Portnummern besitzen und es somit nicht möglich ist, bestimmte Pakete bestimmten Diensten zuzuordnen.

Bewertung aufgrund der Kriterien aus den möglichen Angriffen

- Kenntnis des Authentifikationsschlüssels

  TCP-Relays sind in der Lage, Verbindungen nur dann zuzulassen, wenn sich der Benutzer über einen *one-time password* Mechanismus authentifizieren kann. Somit kann kein Angreifer Kenntnis eines gültigen Authentifikationsschlüssels erlangen, da jeder Schlüssel im Moment seiner Benutzung für weitere Einsätze unbrauchbar wird. Deshalb ist es möglich, auch von außen initiierte Verbindungen zu gestatten, ohne dadurch die Sicherheit des Unternehmensnetzes zu gefährden (vgl. Anhang A).

- Umgehen des Authentifikationssystems

  Versucht ein Angreifer die Privilegien eines internen *servers* für seinen Angriff zu nutzen, so ist es gleichgültig, welche Ursache für diese Angriffsmöglichkeit er auszunutzen versucht. Bei Verwendung eines TCP-Relays bleiben nur die beiden folgenden Möglichkeiten, derartige Angriffe zu unterbinden:

  - Verbot von außen initiierter Verbindungen

    Ein TCP-Relay kann natürlich jegliche Verbindung von außen abblocken und somit einem Angreifer keine Chance geben, mit einem internen *server* zu kommunizieren. Kann ein Angreifer keine Verbindung zu einem *server* aufbauen, so kann er ihn natürlich auch nicht für seine Zwecke mißbrauchen.

  - Beschränkung auf autorisierte Benutzer

    Ist es erforderlich, Verbindungen von außen zuzulassen, so sollten nur wenige Benutzer in der Lage sein, diese zu initiieren. Diese Benutzer müssen sich über *one-time passwords* authentifizieren, sodaß ebenfalls kein Angreifer die Möglichkeit hat, mit einem internen *server* zu kommunizieren.

  Auch das Ausnutzen von Benutzerprivilegien läßt sich analog zum Paket-Filter mit einem TCP-Relay nicht verhindern.

- Überlastung eines Servers

  Es besteht keine Möglichkeit, einen *server* vor Überlastung zu schützen, außer man beschränkt die Zahl der autorisierten Benutzer so weit, daß die Gefahr einer Überlastung nicht besteht.

- Maskerade

  Viele TCP-Relays verlassen sich bei der Authentifizierung des Absenders ausschließlich auf die IP-Adresse. Ist dies der Fall, so ist ein TCP-Relay genauso anfällig für Angriffe mittels Maskerade wie es ein Paket-Filter ist.

Verfügt er aber über weitere Authentifikationsmaßnahmen, so läßt sich das Problem der Maskerade relativ leicht in den Griff bekommen, da es dann nicht mehr erforderlich ist, sich auf nicht zu überprüfende Angaben im Paket-Header zu verlassen.

- Abhören des Netzverkehrs

  Ein Abhören des Netzverkehrs ist auch mit Hilfe eines TCP-Relays nicht zu verhindern, da keinerlei Verschlüsselung der übertragenen Daten vorgenommen wird. Allerdings enthalten alle Pakete, die über das ungesicherte Netz transportiert werden, die Adresse des Relays statt des entsprechenden internen Rechners. Somit können Komunikationsbeziehungen bis zu einem gewissen Grad geheimgehalten werden, da es einem Angreifer bedeutend schwerer fällt, ein Paket einem bestimmten internen Rechner zuzuordnen.

Bewertung aufgrund weiterer Kriterien

1. **Audit**

   - Logging

     Im Gegensatz zu den zustandslosen Paket-Filtern, die nur ein Logging jedes einzelnen Paketes durchführen können, kann ein TCP-Relay mit einigen wenigen Log-Einträgen die gesamte Verbindung beschreiben. Meist wird ein Eintrag bei Aufbau der Verbindung angelegt, der Informationen über den Zeitpunkt, den benutzen Dienst und den verwendeten Authentifikationsmechanismus enthalten kann. Bei Ende der Verbindung wird dann ein weiterer Eintrag vorgenommen, der genaue Informationen über die Dauer der Verbindung sowie über die übertragene Datenmenge enthalten kann. Dies reduziert die Anzahl der erforderlichen Einträge natürlich erheblich, sodaß meist auf eine weitere Formatierung verzichtet werden kann.

   - Alerting

     Ebenso könnten verschiedene Alerting-Mechanismen in ein TCP-Relay integriert werden, die bei Eintreffen eines bestimmten Ereignisses in irgendeiner Form einen Verantwortlichen verständigen.

2. **sicherheitsrelevante Kriterien**

   - Authentifikationsmöglichkeiten

     Wie bereits mehrfach erwähnt, besteht die Möglichkeit, bei Verwendung eines TCP-Relays die Identität eines Benutzers über einen beliebigen *one-time password* Mechanismus zu überprüfen.

   - Sicherheit der Firewallkomponenten

     Hier verhält es sich genau wie beim Paket-Filter. Die Sicherheit der Firewallkomponenten muß anhand eines konkreten Beispiels untersucht werden. Die einzige Aussage, die sich allgemein in diesem Zusammenhang über TCP-Relays treffen läßt ist, daß die Existenz von User-Accounts auf dem Firewall nicht erforderlich ist.

- Domain Name Service

  Es besteht die Möglichkeit einer Zweiteilung des *domain name services*. Da externe Rechner nur mit dem Firewall kommunizieren, der wiederum die Verbindung zu den internen Rechnern aufrechterhält, ist es ausreichend, wenn im externen Netz die Adresse des Firewalls bekannt gemacht wird.

3. Administration

- Installation und Konfiguration

  Die Installation und Konfiguration ist nicht sehr aufwendig. Die Konfiguration gestaltet sich etwas leichter als beim Paket-Filter, da weniger Informationen zu untersuchen sind und somit die Möglichkeiten, daß sich Regeln gegenseitig beeinflussen, geringer sind. Die Gefahr besteht aber trotzdem, sodaß größte Sorgfalt bei der Aufstellung der Regeln angebracht ist.
- Behandlung proprietärer bzw. neuer Dienste

  Sofern es sich um TCP-basierende Dienste handelt, ist die Integration dieser Dienste sehr leicht durch das Einfügen von Filterregeln zu bewerkstelligen.

Weitere Aussagen über die Administration von TCP-Relays lassen sich nicht treffen, hierzu müßte ein konkretes Produkt herangezogen werden.

4. Benutzerfreundlichkeit

- Transparenz

  Bei der Untersuchung der Transparenz eines TCP-Relays zeigt sich der wesentliche Nachteil dieses Konzeptes gegenüber dem Paket-Filter: Ein TCP-Relay ist nicht transparent. Denkbar wäre ein zweistufiger Verbindungsaufbau, der den Firewall für die Benutzer nicht transparent erscheinen lassen würde, meist wird aber eine Änderung der Anwendungen vorgenommen, sodaß die Transparenz für die Benutzer gewahrt bleibt.
- Performance

  Die Performance eines TCP-Relays dürfte im allgemeinen etwas besser sein als die eines Paket-Filters, da eine Untersuchung der Regeln nur beim Eintreffen des ersten Paketes einer Verbindung durchgeführt werden muß. Da die Pakete im Firewall nur bis zur Schicht 4 bearbeitet werden müssen, ist der Performance-Verlust durch den Firewall relativ gering zu erwarten.

5. Kosten

TCP-Relays erfordern einen dedizierten Rechner, auf dem die Firewallsoftware läuft. Das bedeutet, daß zusätzlich zum Preis für die Software noch die Anschaffung eines leistungsfähigen Rechners einkalkuliert werden muß.

4.4 Das Application-Gateway (Proxy)

Die dritte Möglichkeit, einen Firewall einzurichten, ist die Verwendung sogenannter Proxies [Ches94]. Unter einem Proxy versteht man einen *server*, der auf dem Firewall installiert wird und der das entsprechende Protokoll in allen sieben Schichten implementiert. Ein Benutzer, der mit einem *server* kommunizieren möchte, baut eine Verbindung zum Proxy auf, der wiederum die Verbindung zum eigentlichen *server* unterhält. Dies bedeutet aber, daß für jeden Dienst, der gesichert werden soll, ein eigener Proxy existieren muß. Deshalb werden Proxies meist nur für einige wenige Dienste eingesetzt.

4.4.1 Bewertung des Proxy-Konzeptes

Bewertung aufgrund der Kriterien aus den Dienstspezifikationen

- Behandlung von Client / Server-Diensten

  Mit Hilfe eines Proxies läßt sich jeder *client/server*-Dienst sichern. Selbst UDP-basierende Dienste stellen kein Problem dar, da der Proxy das Protokoll ja bis zur Schicht 7 abwickelt und somit genug Informationen zur Verfügung stehen, die eine Zuordnung von Paketen zu bestehenden Verbindungen ermöglichen. Da Proxies es ermöglichen, *one-time passwords* zur Authentifikation von Benutzern einzusetzen, können Verbindungen von außen ebenfalls gestattet werden.

- Behandlung von Peer-to-Peer-Diensten

  Peer-to-peer-Dienste werden duch den Proxy einfach transparent an den entsprechenden Kommunikationspartner weitergeleitet. Es besteht die Möglichkeit, sowohl TCP- als auch UDP-basiernde Dienste nur von innen nach außen zu gestatten. Häufig wird auch der Proxy selber als der interne Partner einer *peer-to-peer*-Verbindung eingesetzt, sodaß auf eine Weiterleitung verzichtet werden kann.

- Behandlung von Sonderfällen

  - Behandlung von FTP

    Die Behandlung von *FTP* ist mit einem Proxy relativ leicht zu bewerkstelligen. Er kann problemlos erkennen, ob eine eintreffende Verbindung der Datenkanal einer bestehenden Kontroll-Verbindung ist oder nicht. Außerdem kann er bestimmte Kommandos, die z. B. den Export von Daten erlauben, unterbinden.

  - Behandlung von SMTP

    Es besteht die Möglichkeit, einen sicheren *mailserver* auf dem Firewall zu installieren, der die Weiterverteilung der Nachrichten an die einzelnen Benutzer übernimmt. Dies schützt die einzelnen Rechner vor Angriffen und bietet gute Sicherheit, solange der Proxy keine Fehler enthält, die von Angreifern ausgenutzt werden können.

  - Behandlung von X11

    Neuere Entwicklungen versuchen X11 zu sichern, indem jeder Verbindungswunsch von dem jeweiligen Benutzer ausdrücklich bestätigt werden muß. Somit läßt sich sicherstellen, daß jede Verbindung dem Benutzer zumindest bekannt ist. Die Sicherheit liegt damit aber wiederum zum Teil in den Händen der Benutzer, was nicht unbedingt wünschenswert sein kann.

  - Behandlung RPC-basierender Dienste

    Auch Proxies haben Schwierigkeiten bei der Behandlung RPC-basierender Dienste. Das Problem ist wiederum, daß nicht bekannt ist, welcher Dienst der untersuchten Verbindung zuzuordnen ist. Die im Verlauf dieser Arbeit untersuchten Firewalls stellen keine Proxies zur Verfügung, die die Sicherung RPC-basierender Dienste erlauben würden.

Bewertung aufgrund der Kriterien aus den möglichen Angriffen

- Kenntnis des Authentifikationsschlüssels

  Proxies stellen Authentifikationsmechanismen über *one-time passwords* zur Verfügung. Somit besteht keine Gefahr, daß ein Angreifer in den Besitz eines gültigen Authentifikationsschlüssels gelangen könnte.

- Umgehen des Authentifikationssystems

  Proxies besitzen recht umfangreiche Möglichkeiten um zu verhindern, daß ein Angreifer die Privilegien eines *servers* mißbraucht, um einen Angriff durchzuführen:

  - Fehlfunktion eines Servers

    Möchte ein Benutzer über einen Proxy mit einem Server kommunizieren, so erhält er keinen direkten Kontakt zum Server sondern nur zum Proxy. Das bedeutet, daß eventuell im Server enthaltenen *bugs* oder *trapdoors* keine Bedeutung zukommt, da ein Angreifer den Server nicht direkt ansprechen kann. Proxies sind üblicherweise sehr einfach geschrieben, sodaß sie relativ schnell auf Fehlerfreiheit überprüft werden können. Man kann also davon ausgehen, daß *bugs* und *trapdoors* bei Verwendung von sicheren Proxies keine Gefahr darstellen können. Enthält natürlich der Proxy selber einen Fehler, so ist nicht nur der entsprechende *server* angreifbar sondern der Firewall selber. Deshalb muß unbedingt auf absolute Fehlerfreiheit bei der Verwendung von Proxies geachtet werden.

  - Fehlkonfiguration eines Servers

    Da der Proxy der einzige *server* ist, mit dem externe Benutzer kommunizieren können und da er von einem sicherheitsbewußten Firewall-Administrator konfiguriert wird, sollte es keine Probleme mit Konfigurationsfehlern durch System-Administratoren geben. Auch durch Benutzer hervorgerufene Konfigurationsfehler können bei entsprechender Konfiguration des Proxies behoben werden.

  - Protokollschwächen

    Mit Hilfe eines Proxies ist es möglich, einzelne Kommandos eines Dienstes zu unterbinden bzw. zu gestatten, um somit eventuell vorhandene Schwächen der einzelnen Protokolle ausgleichen zu können. Ebenso kann man natürlich Dienste, die über keine oder nur unzureichende Authentifizierungsmaßnahmen verfügen dadurch sichern, daß die Authentifizierung der Benutzer bereits im Proxy stattfindet.

  Auch bei der Ausnutzung von Benutzerprivilegien gibt es Ansätze, mit Proxies zur Sicherung beizutragen. So können neuere Versionen von Mail-Proxies den Absender der

Mail mit Hilfe des *identd*-Daemons überprüfen. Dies bietet aber keinen hundertprozentigen Schutz, da ein Angreifer den Daemon ebenfalls verändert haben könnte, sodaß dieser falsche Informationen verbreitet.

- Überlastung eines Servers

  Ein Proxy könnte natürlich so programmiert werden, daß er nur eine bestimmte Anzahl von Verbindungen zu einem bestimmten Server pro Zeiteinheit gestattet. Somit würde der Server für die internen Benutzer stets verfügbar bleiben, auch wenn die Anzahl der von außen eintreffenden Verbindungswünsche sehr groß wird.

- Maskerade

  Da Proxies im allgemeinen eine benutzerbezogene Authentifizierung vornehmen, besteht nur eine geringe Gefahr, daß durch Maskerade ein Angreifer unerlaubten Zugang zum System erhalten könnte. Bei den *peer-to-peer*-Diensten verläßt sich natürlich auch ein Proxy auf die angegebene IP-Adresse, sodaß hier die Möglichkeit besteht, einen Angriff zu starten. Verbietet man hier aber wieder Verbindungen von außen, so kann auch diese Gefahr unterbunden werden.

- Abhören des Netzverkehrs

  Natürlich kann mit Hilfe eines Proxies eine Verschlüsselung der übertragenen Daten vorgenommen werden. Dies setzt aber voraus, daß der Kommunikationspartner in der Lage ist, diese Daten wieder zu entschlüsseln. Da dies im allgemeinen nicht der Fall ist, muß auf eine Verschlüsselung meist verzichtet werden. Dann bietet der Proxy nur die Möglichkeit, das Erkennen von Kommunikationsbeziehungen zu erschweren, da jedes Paket auf dem ungesicherten Netz die Adresse des Proxies statt des eigentlichen Zieles enthält.

Bewertung aufgrund weiterer Kriterien

1. **Audit**

   Proxies verfügen über hervorragende Audit-Mechanismen. Es können sowohl einzelne Pakete als auch Verbindungen geloggt werden. Auch wiederholte Authentifikationsfehler können im Log-File abgelegt werden. Es ist sowohl möglich, Angaben über das transportierte Datenvolumen zu erhalten als auch darüber, welcher Benutzer die Verbindung initiiert hat. Auch die Alerting-Mechanismen, die von Proxies zur Verfügung gestellt werden können, sind sehr umfangreich.

2. **sicherheitsrelevante Kriterien**

   - Authentifikationsmöglichkeiten

     Bei Verwendung von Proxies steht das gesamte Spektrum an Authentifikationsmöglichkeiten zur Verfügung (siehe Anhang A).

- Sicherheit der Firewall-Komponenten
  - Existenz von User-Accounts
    Es gibt zwei unterschiedliche Arten von Proxies:
    * Proxies mit User-Accounts
      Bei dieser Art von Proxy muß auf dem Firewall ein Account für jeden berechtigten Benutzer existieren. Der Benutzer loggt sich erst in den Firewall ein, von wo aus er den entsprechenden Proxy wie einen *client* startet. Meist erhält der Benutzer auf dem Firewall nur eine eingeschränkte Shell, mit der er nur einige wenige Kommandos ausführen kann.
    * Proxies ohne User-Accounts
      Die andere Art von Proxies kommt ohne User-Accounts aus. Hier baut der Benutzer eine Verbindung zum Proxy auf, der daraufhin nach dem gewünschten Ziel fragt. Der Proxy baut die Verbindung dorthin auf und der Benutzer hat nie die Möglichkeit, ein Kommando auf dem Firewall auszuführen.
  - periodische Prüfsummenbildung
    Proxies können die eingesetzten Konfigurationsfiles einer periodischen Prüfung unterziehen, um somit festzustellen, ob Änderungen vorgenommen wurden.
- Domain Name Service
  Bei Verwendung von Proxies ist es nicht erforderlich, die Adressen des internen Netzes im externen Netz bekannt zu machen. Deshalb kann eine Zweiteilung des *domain name services* vorgenommen werden, die internen Rechnern Informationen über sämtliche externen Rechner bietet, umgekehrt aber externen Rechnern nur die Adresse des Firewalls anbietet.

3. Administration

- Installation und Konfiguration
  Die Installation und Konfiguration von Proxies ist relativ aufwendig, da für jeden Dienst ein eigener Proxy notwendig ist, der natürlich auch eine spezielle Konfiguration erfordert. Da aber keine Abhängigkeiten zwischen den einzelnen Diensten bestehen, wie es bei den Filterregeln eines Paket-Filters oder TCP-Relays der Fall ist, ist die Konfiguration nicht so fehleranfällig wie bei den anderen Konzepten.
- Behandlung proprietärer bzw. neuer Dienste
  Da für jeden Dienst ein eigener Proxy notwendig ist, ist dieses Konzept nicht so flexibel in der Behandlung von proprietären bzw. neuen Diensten. Es ist erforderlich, einen neuen Proxy zu programmieren, bevor ein derartiger Dienst durch den Firewall angeboten werden kann.

4. Benutzerfreundlichkeit

- Transparenz
  Proxies sind nicht transparent. Die meisten Proxies erfordern zwar keine Änderungen der *client*-Anwendungen aber einen zweistufigen Verbindungsaufbau, sodaß die Benutzer eine Umstellung ihrer Arbeitsgewohnheiten in Kauf nehmen müssen.

- Performance

  Da im Proxy der gesamte Protokollstack zweimal abgearbeitet werden muß, kann es natürlich zu Leistungseinbußen kommen.

5. **Kosten**

   Ähnlich wie beim TCP-Relay erfordert das Proxy-Konzept mindestens einen dedizierten Rechner, der ausschließlich zur Realisierung des Firewalls eingesetzt wird. Das bedeutet, daß sowohl Kosten für Hardware als auch für Software entstehen.

Kapitel 5

Vorstellung existierender Firewall-Lösungen

5.1 IBM NetSP

5.1.1 Beschreibung

Der IBM-Firewall NetSP verwendet eine Kombination der drei bereits vorgestellten Firewall-Konzepte Paket-Filter, TCP-Relay sowie Application-Gateway [IBM94]. Abbildung 5.1 stellt dar, wie diese Konzepte kombiniert sind.

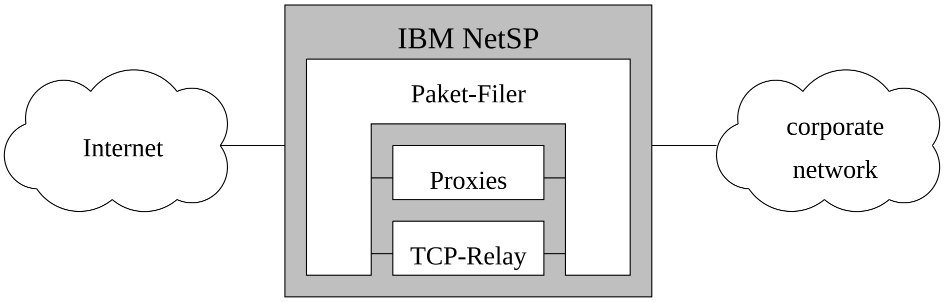

Abbildung 5.1: Kombination der Firewall-Konzepte beim IBM NetSP

Jedes Paket, das am Firewall eintrifft oder diesen verläßt, wird durch den Paket-Filter untersucht. Zusätzlich kann noch eine Weiterbearbeitung des Paketes durch Proxies bzw. durch das TCP-Relay erfolgen. Möchte man auf die Verwendung des Paket-Filters verzichten, so läßt sich dieser deaktivieren und die entsprechenden Pakete erreichen die Proxies bzw. das TCP-Relay ohne vorhergehende Untersuchung. Alle anderen Pakete werden automatisch verworfen.

Der IBM NetSP unterscheidet die Interfaces, auf denen die Pakete eintreffen in sichere (secure) und unsichere (non-secure). Deshalb benötigt man als Firewall einen Rechner, der über mindestens zwei Schnittstellen verfügt. Abbildung 5.2 stellt die Verbindung eines sicheren Netzes mit einem unsicheren über den Firewall dar.

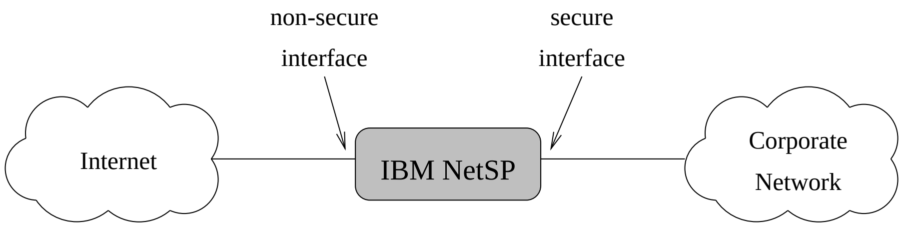

Abbildung 5.2: Verbindung eines sicheren und eines unsicheren Netzes mit dem NetSP

Die Verwendung unterschiedlicher Schnittstellen für das sichere und das unsichere Netz garantiert, daß kein Paket von einem Netz ins andere gelangen kann, ohne den Firewall passieren zu müssen.

Der Paket-Filter

Der Firewall enthält einen Paketfilter, der die Pakete sowohl beim Eintreffen als auch beim Verlassen des Firewalls überprüft. Es werden aber jeweils dieselben Regeln angewandt, d.h. man kann nicht zwei unterschiedliche Regelsätze aufstellen, die je nach Richtung des Datenverkehrs angewandt werden. Es kann aber unterschieden werden, ob das Paket den Firewall gerade verläßt oder ob es eintrifft, sowie ob es sich um ein Interface zum sicheren oder zum unsicheren Netz handelt. Somit ist man in der Lage, unterschiedliche Filter für Pakete, die eintreffen und Pakete, die den Firewall verlassen, aufzustellen.

Das TCP-Relay

Der IBM NetSP verwendet als TCP-Relay das frei erhältliche Produkt *SOCKS*. Die eingesetzte *SOCKS*-Version ist in der Lage, einzelnen Benutzern Zugang zu gewähren, sowie die Identität dieser Benutzers mit Hilfe des *identd-servers* des entsprechenden Rechners zu überprüfen. Die derzeit eingesetzte Version ist aber noch nicht in der Lage, UDP-basierende Dienste zu sichern.

Die Application-Gateways

Der Firewall verfügt über drei Proxies, einen Telnet-Proxy, einen FTP-Proxy sowie einen Proxy für E-Mail.

- Telnet

  Der *telnet*-Proxy erfordert einen zweistufigen Verbindungsaufbau. Ein Benutzer muß sich zuerst über *telnet* in den Firewall einloggen, wo er sich, eventuell über ein *one-time password*, authentifizieren muß. Daraufhin erhält der Benutzer eine sogenannte *restricted shell*, d. h. eine *shell*, mit der nur einige bestimmte Kommandos ausgeführt werden können. Von hier aus baut er eine weitere *telnet*-Verbindung zum eigentlichen Zielrechner auf.

- FTP

  Ähnlich verhält es sich beim *FTP*-Proxy. Auch hier wird erst eine *FTP*-Verbindung zum Firewall initiiert und eine Authentifikation des Benutzers durchgeführt. Mit Hilfe eines Kommandos der Form `quote site ftp.host` wird dann die Verbindung zum Zielrechner erzeugt. Der *FTP*-Proxy kann nur für Verbindungen eingesetzt werden, die von einem Rechner des internen Netzes stammen, d. h. nur für Verbindungswünsche, die an einem *secure interface* eintreffen.

- E-Mail

  Die Dokumentation des IBM NetSP spricht zwar bei der Behandlung von E-Mail nicht von einem Proxy, es besteht aber prinzipiell kein Unterschied zum Proxy-Konzept. Der Firewall enthält eine veränderte Version des *sendmail*-Programms, daß so konfiguriert wird, daß es jede ankommende *mail* sofort an einen internen *mailserver* weiterleitet, der für die richtige Verteilung der *mail* inenerhalb des Unternehmensnetzes sorgt.

5.1.2 Bewertung

Von der Firma IBM wurde der Firewall NetSP in einer Testinstallation in der Version 1.2 zur Verfügung gestellt, so daß er einer eingehenden Untersuchung unterzogen werden konnte.

Bewertung aufgrund der Kriterien aus den Dienstspezifikationen

- Behandlung von TCP-basierenden Client/Server-Diensten

  TCP-basierende *client/server*-Dienste lassen sich mit dem IBM NetSP gut sichern. Zum einen verfügt er über den *SOCKS*-Server, mit dessen Hilfe sich alle von innen nach außen laufenden Verbindungen sichern lassen, zum anderen besitzt er einen Telnet-Proxy, der auch von außen initiierte Verbindungen ermöglicht.

  - Verwendung des Telnet-Proxies

    Der *telnet*-Proxy bietet eine ausreichende Sicherung von *telnet*-Verbindungen, gleichgültig ob sie von einem Rechner des internen oder des externen Netzes initiiert wurden. Er kann auch verwendet werden, um andere Dienste zu sichern. Ein Benutzer kann sich über den *telnet*-Proxy auf den Firewall einloggen und dort beliebige *clients* starten, sofern die *restricted shell* dies erlaubt. Man kann also beispielsweise *mosaic-clients* auf dem Firewall installieren, die von Benutzern, die sich über *telnet* einloggen, genutzt werden können.

  - Verwendung des SOCKS-Servers

    Da der SOCKS-Server nicht über ausreichende Authentifikations-Maßnahmen verfügt, um einem externen Benutzer Zugang zum internen Netz zu gestatten, wird er eingesetzt, um Dienste zu sichern, die nur von innen nach außen angeboten werden sollen. Hier bietet er aber ausreichenden Schutz sowie Konfigurationsmöglichkeiten, die es erlauben, nur bestimmten Benutzern die Nutzung bestimmter Dienste zu gestatten.

  - Verwendung des Paket-Filters

    Der Paket-Filter ist in der Lage, daß ACK-Flag zu untersuchen, kann also unterscheiden, wer eine Verbindung initiiert hat. Deshalb könnte man ihn einsetzen, um TCP-Dienste, die nur von innen nach außen benötigt werden, zu sichern. Dies kann erforderlich sein bei Diensten, für die keine *socksified clients*, d. h. Client-Programme, die mit dem *SOCKS-server* zusammenarbeiten können, exisitieren. Im Normalfall sollte aber das TCP-Relay für TCP-basierende *client/server*-Dienste verwendet werden.

- Behandlung von UDP-basierenden Client/Server-Diensten

  Da für UDP-basierende *client/server*-Dienste keine Proxies zur Verfügung stehen und auch der *SOCKS-server* keine UDP-Dienste sichern kann, bleibt nur die Möglichkeit der Sicherung mit Hilfe des Paket-Filters. Wie bereits erwähnt, ist ein Paket-Filter aber nicht in der Lage, diese Dienst hinreichend sicher zu behandeln. Aus diesem Grund sollten sie völlig verboten werden. Auch hier gibt es die Möglichkeit, entsprechende *clients* auf dem Firewall zu installieren und den Zugang über *telnet* auf die Nutzung dieser *clients* auszudehnen.

- Behandlung von TCP-basienden Peer-to-Peer-Diensten

  Diese Dienste lassen sich wiederum sowohl über *SOCKS* als auch über den Paket-Filter sichern. Beide Methoden sind aus Sicherheitssicht problemlos einsetzbar.

- Behandlung von UDP-basierenden Peer-to-Peer-Diensten

  Hierfür steht wiederum nur der Paket-Filter zur Verfügung, der, wie vorher bereits erwähnt, ausreicht, um derartige Dienste abzusichern. Die Gefahr eines Angriffs durch Annahme einer falschen Identität bleibt aber bestehen.

- Behandlung von Sonderfällen

  - Behandlung von FTP

    Um den Schwierigkeiten, die mit *FTP* auftreten können, entgegenzuwirken, existiert ein Proxy, der *FTP*-Verbindungen von innen nach außen ermöglicht. Dieser Proxy führt sowohl eine Authentifikation des Benutzers als auch eine Überprüfung seiner Autorisation durch. Eine Einschränkung der verwendbaren Kommandos ist nicht möglich. Möchte ein externer Benzutzer *FTP* nutzen, so muß er sich zuerst über den *telnet*-Proxy in einen internen Rechner einloggen und von dort aus die gewünschte *FTP*-Verbindung aufbauen.

  - Behandlung von SMTP

    Der IBM Firewall leitet eintreffende *mail* direkt weiter an einen *Mail-server* innerhalb des internen Netzes. Dieser ist mit der Verteilung der Nachrichten an die eigentlichen Empfänger betraut.

  - Behandlung von X11

    Es sind keine speziellen Möglichkeiten vorgesehen, *X11* durch den Firewall zu gestatten. Aufgrund der Gefahren, die dieser Dienst beinhaltet, sollte er also verboten werden.

- Behandlung von RPC-basierenden Diensten

  Der IBM NetSP beinhaltet keine Möglichkeiten, mit denen RPC-basierende Dienste abgesichert werden können. Es bleibt nur das vollständige Verbot dieser Dienste.

Bewertung aufgrund weiterer Kriterien

1. **Audit**

   - Logging

     Das Logging des IBM NetSP erfolgt über das *syslog*-Programm. Man gibt eine Datei an, in die bis zu 512 Einträge geschrieben werden können. Wird die Zahl von 512 Einträgen überschritten, so wird als letzter Eintrag eine Zeile eingefügt, die angibt, wieviele Einträge nicht mehr geloggt werden konnten. Je nachdem, ob man Paket-Filter, TCP-Relay oder Proxy verwendet, unterscheiden sich die erzeugten Log-Files zum Teil erheblich:

     - Logging beim Paket-Filter

       Der Paket-Filter erzeugt Log-Einträge bei jeder Änderung seines Zustandes, d. h. bei Änderungen der Filterregeln oder bei Aktivierung bzw. Deaktivierung des Filters. Weiterhin ist er in der Lage, für jedes zurückgewiesene Paket einen Log-Eintrag anzulegen. Diese Einträge setzen sich z. B. aus folgenden Informationen zusammen:

       * inbound/outbound

         Gibt an, ob das Paket bereits beim Eintreffen am Firewall oder erst bei dessen Verlassen zurückgewiesen wurde.
       * adapter

         Gibt das Interface an, auf dem das Paket angekommen ist bzw. auf das das Paket geschrieben werden sollte.
       * source adress / port

         Die Absender-Adresse und Port, wie sie im Header des zurückgewiesenen Paketes angegeben waren.
       * destination adress / port

         Die Zieladresse und Port, an die das Paket adressiert war.
       * rule number

         Gibt die Nummer der Regel an, die die Zurückweisung des Paketes veranlaßt hat.

     - Logging beim TCP-Relay

       SOCKS stellt hervorragende Logging-Möglichkeiten zur Verfügung, die sogar ausreichen könnten, um die Kosten für den Internet-Anschluß auf die einzelnen Benutzer umlegen zu können. Im einzelnen werden folgende Daten zur Verfügung gestellt:

       * Verbindungsaufbau

         Beim Verbindungsaufbau wird ein Eintrag im Log-File vorgenommen, der den Zeitpunkt des Eintrages, die Adressen von Quell- und Zielrechner, den Benutzer sowie den verwendeten Dienst enthält.

    * Verbindungsabbau
      Beim Verbindungsabbau werden zwei Zeilen im Log-File angelegt. Die erste enthält Informationen, die mit denen beim Verbindungsaufbau vergleichbar sind, während die zweite Zeile die tatsächlich in beide Richtung übertragene Anzahl von Bytes angibt.
    * zurückgewiesener Verbindungswunsch
      Selbstverständlich werden auch Verbindungswünsche angezeigt, bei denen keine Verbindung zustande kam, da der entsprechende Benutzer nicht zum Aufbau einer derartigen Verbindung autorisiert war.
  - Logging beim Proxy
    Das Logging der Proxies ist in der derzeitigen Version des Firewalls noch unzureichend ausgeprägt. Es wird ein Eintrag in ein Log-File vorgenommen, wenn ein Benutzer eine Verbindung aufbaut bzw. wenn ein Verbindungswunsch zurückgewiesen wird. Das Ende einer Verbindung ist allerdings nicht über das Logging erkennbar. Ebenso fehlt jegliche Angabe über das innerhalb dieser Verbindung übertragene Volumen.
- Alerting
  Einzig der *SOCKS-server* enthält eine Möglichkeit, jeder Filterregel ein Kommando hinzuzufügen, das bei Eintreffen eines Paketes, das diese Regel erfüllt, ausgeführt wird. Somit lassen sich eigene Alerting-Mechanismen entwickeln, die dann vom *SOCKS-server* gestartet werden können. Weder der Paket-Filter noch die Proxies verfügen über derartige Möglichkeiten.

2. **sicherheitsrelevante Kriterien**

- Authentifikationsmöglichkeiten
  Der IBM NetSP verfügt über folgende Möglichkeiten, einen Benutzer zu authentifizieren:
  - Standard UNIX-Paßwort
  - Security Dynamics SecureID-Card
  - Digital Pathways SecureNet Key
- Sicherheit der Firewall-Komponenten
  - Fehlerfreiheit der Firewall-Software
    Der IBM NetSP verwendet als Proxy für *e-mail* eine veränderte Version des frei verfügbaren Programms *sendmail*. In dieser Version sollten alle derzeit bekannten Sicherheitsmängel beseitigt sein. Das Problem ist aber, daß neu entdeckte Fehler des *sendmail*-Programms, ebendso in der veränderten Version enthalten sein werden. Da in den letzten Jahren regelmäßig *sendmail*-Fehler bekannt geworden sind, ist es keinesfalls auszuschließen, daß weitere Fehler hinzukommen. Zur Verdeutlichung des Problems sind im folgenden zwei Beispiele angegeben, welche Fehler des *sendmail*-Programms in der Vergangenheit bereits zu Einbrüchen geführt haben:
    * Ausführbare Kommandos als Absender
      Gibt man als Absender einer Mail ausführbare Kommandos an und schickt diese Mail an einen nicht existierenden Benutzer, so werden diese bei älteren Versionen ausgeführt, wenn *sendmail* versucht, eine Nachricht über die

Unzustellbarkeit der Mail an den Absender zu übermitteln. Da *sendmail* unter *root*-Berechtigung läuft, werden auch diese Kommandos mit *root*-Privilegien ausgeführt.

  * Verwendung des *identd*
    Neuere Versionen von *sendmail* enthalten die Möglichkeit, die Identität des Absenders über das *identd*-Programm zu überprüfen. Hierzu wird eine Verbindung zum *identd*-Daemon des entsprechenden Rechners aufgebaut, der die Identität des Absenders bestätigen soll. Handelt es sich aber um eine gefälschte Version des *identd*-Programms, so kann man hier statt der Bestätigung der Identität Kommandos übergeben, die aufgrund eines Fehlers von *sendmail* ausgeführt werden.

- Existenz von User-Accounts
  Möchte man die Proxies einsetzen, so ist dies nur möglich, wenn für jeden Benutzer ein Account auf dem Firewall eingerichtet wird. Da Benutzer häufig Paßworte wählen, die keinen ausreichenden Schutz bieten, besteht hier die große Gefahr, daß ein Angreifer Zugang zum Firewall erlangt. Da die Möglichkeit besteht, den Benutzern nur Shells zu geben, die einige wenige Kommandos ausführen können, kann man hier noch relativ sicher sein, daß ein Angreifer keine Möglichkeit hat, mit einer derartigen Shell Angriffe durchzuführen. Gelingt es ihm aber, einen Account einzurichten (z. B. über einen der oben beschriebenen Fehler des *sendmail*-Programms), der ihm zu einer normalen Shell verhilft, so stehen ihm alle Möglichkeiten offen. Bei der großen Anzahl von Accounts, die auf dem Firewall existieren müssen, ist es denkbar, daß ein solches Eindringen über lange Zeit unbemerkt bleibt.
- hierarchische Sicherheitsabstufungen
  Der IBM Firewall verfügt über keine Möglichkeit, bestimmte *server* oder Subnetze speziell zu schützen. Bei Verwendung des TCP-Relays oder des Paket-Filters ist es natürlich möglich, mit Hilfe der entsprechenden Filterregeln, den Zugang von und zu bestimmten Adressen völlig zu verbieten.
- periodische Prüfsummenbildung
  Der IBM Firewall stellt keine Möglichkeiten zur Verfügung, die Konfigurationsfiles auf Veränderungen hin zu überwachen. Allerdings verfügt das AIX-Betriebssystem über tools, die eine Überwachung dieser Dateien ermöglichen.
- administrativer Zugang über das Netz
  Der Firewall gestattet ein Einloggen als *root* über das Netz mit selbst konfigurierbaren Authentifikationsmöglichkeiten. Hier sollte unbedingt ein sicheres Authentifikationsschema verwendet werden, um einem Angreifer nicht die Möglichkeit zu geben, über das Netz die Konfiguration des Firewalls zu ändern.

• Domain Name Service
Der IBM NetSP bietet eine Zweiteilung des Domain Name Service an, wobei der externe Name Server vom Firewall selber erstellt wird, während der interne Server natürlich vom Betreiber konfiguriert werden muß. Externe Hosts haben somit nicht die Möglichkeit, Adressen interner Rechner zu erfahren, interne hingegen haben Kenntnis über alle externen Rechner.

3. Administration

- Installation und Konfiguration

  Zur Installation und Konfiguration des Firewalls bieten sich zwei unterschiedliche Möglichkeiten an: Einerseits kann er mit Hilfe des *System Maintenance and Installation Tool (SMIT)* des IBM-Betriebssystems AIX/6000 konfiguriert werden. Es handelt sich hierbei um eine Benutzeroberfläche, die menügesteuert die Installation und Konfiguration vornimmt. Andererseits steht eine Kommandosprache zur Verfügung, die eine schnellere Konfiguration erlaubt, allerdings schwierig zu lernen und weniger komfortabel in der Benutzung ist.

  Die Installation gestaltet sich schnell und einfach. Nach einigen wenigen Eingaben wie z. B. dem Laufwerk, von dem aus installiert werden soll, findet die gesamte Installationsprozedur automatisch statt. Dies findet in weniger als fünf Minuten statt. Nach der Installation ist ein Reboot der Firewall-Machine notwendig.

  Ähnlich verhält es sich mit der Konfiguration. Probleme ergeben sich nur bei der Konfiguration der Proxies, da bei deren Verwendung für jeden berechtigten Benutzer ein Account eingerichtet werden muß. Der hieraus resultierende Aufwand ist bei Größenordnungen von mehreren tausend Benutzern nicht tragbar, sodaß eine Verwendung der Proxies nur für einige wenige Benutzer, die Zugang von außen benötigen, in Frage kommt.

- Support

  Im Kaufpreis des Firewalls ist der Support für das erste Jahr enthalten. Danach muß ein entsprechender Wartungsvertrag geschlossen werden, über dessen Konditionen aber keine Informationen vorliegen.

- Dokumentation

  Die Dokumentation ist ausreichend, um die Installation und Konfiguration des Firewalls selbstständig vornehmen zu können. Auch ist ein eigenes Kapitel dem Testen der erstellten Konfiguration gewidmet.

- SNMP-Unterstützung

  Ein SNMP-Agent ist im IBM Firewall weder vorhanden noch für zukünftige Versionen geplant.

- Behandlung proprietärer bzw. neuer Dienste

  Durch die Verwendung von *SOCKS* und des Paket-Filters, stehen zwei allgemeine Hilfsmittel zur Verfügung, mit denen weitere Dienste in den Firewall integriert werden können. Außerdem besteht die Möglichkeit, *clients* auf dem Firewall zu installieren, die dann über den *telnet*-Proxy benutzt werden können. Hierzu muß sich ein Benutzer über *telnet* in den Firewall einloggen und dort den entsprechenden *client* starten. Da Benutzer nur über eine eingeschränkte *shell* verfügen, muß die *shell* natürlich noch dahingehend geändert werden, daß der Start des *clients* ermöglicht wird.

4. **Benutzerfreundlichkeit**

   - Transparenz

     Mit Ausnahme des Paket-Filters handelt es sich um eine nicht-transparente Lösung. Bei der Verwendung von *SOCKS* müssen Änderungen an sämtlichen *client*-Programmen vorgenommen werden, was bei einem Netz mit einer fünfstelligen Anzahl von Endgeräten große Probleme bereiten kann. Die Proxies sind natürlich ebensowenig transparent, da sie einen echten zweistufigen Verbindungsaufbau erfordern.
   - Performance

     Im Testbetrieb bei der BMW AG traten keine Beinträchtigungen der Performance durch den Firewall auf.

5. **Kosten**

   - Kaufpreis

     Die NetSP-Software ist zu einem Preis von ca. 30.000 DM erhältlich.
   - Hardware-Voraussetzungen

     Für den IBM NetSP benötigt man eine Workstation vom Typ IBM RS/6000 mit dem IBM-Betriebssystem AIX 3.2.5. Die Workstation sollte über mindestens 32 MB Hauptspeicher verfügen und benötigt mindestens zwei Netz-Interfaces, um Anschlüsse zum sicheren und zum unsicheren Netz realisieren zu können.

5.2 Sun FW-I

5.2.1 Beschreibung

Der von der Firma SunSoft vertriebene und von der Firma CheckPoint Software entwickelte Firewall-I verwendet das Konzept des Paket-Filters, wobei dieses Konzept aber in erheblichem Maße erweitert wurde [Sun94a]. Insbesondere verfügt der FW-I über Möglichkeiten, *FTP* und UDP-basierende Dienste, also Dienste, die mit traditionellen Paket-Filtern nicht zu sichern sind, ohne Gefahren für das interne Netz zu gestatten.

Der Firewall besteht aus einem *control module*, das in der Workstation des Firewall-Verantwortlichen installiert wird, sowie einem oder mehreren *packet filter modules*, die auf jedem Rechner oder Router des Netzes installiert werden können [Sun94b]. Üblich ist hierbei eine Sicherung des Zugangs zum Internet mit einem *packet filter module* sowie eventuell eine spezielle Sicherung besonders wichtiger *server* durch eigene Module. Die Kommunikation zwischen *control module* und *packet filter module* erfolgt, sofern sie nicht auf ein und demselben Rechner residieren, über ein *one-time password*-Schema authentifiziert, sodaß keine Möglichkeit besteht, daß ein unberechtigter Benutzer die Konfiguration der *packet filter modules* verändern kann. Abbildung 5.3 zeigt ein typisches Einsatzszenario für den Sun Firewall. Die Version 1.2 des Firewalls gestattet außerdem eine Authentifikation der Benutzer mittels *one-time passwords* [Sun95a].

Je nach Größe des zu sichernden Netzes und der gewünschten Funktionalität kann der Firewall in verschiedenen Verisonen erworben werden:

- FW-I Internet Gateway Security Center

  Dieses Paket besteht aus einem *control module* sowie einem *packet-filter module*.

- FW-I 'Light' Internet Gateway Security Center

  Bei diesem Paket müssen das *control module* und das *packet-filter module* auf ein und demselben Rechner residieren. Außerdem ist die Maximalzahl zu sichernder Rechner auf 50 begrenzt. Es handelt sich also um eine kostengünstige Lösung für kleine Netze.

- FW-I Router Security Center

  Mit diesem Paket ist es möglich, statt eines *packet-filter module* einen Router einzusetzen, der ebenso vom *control module* aus gesteuert werden kann.

- FW-I Network Security Center

  Dies ist die größte Ausbaustufe des FW-I. Hier werden die Funktionen des *Internet Gateway Security Center* und des *Router Security Center* miteinander kombiniert.

Darüberhinaus ist es noch möglich weitere *packet-filter modules* zu erwerben, die dann in Verbindung mit dem *Internet Gateway Security Center* oder dem *Network Security Center* eingesetzt werden können. Auch kann mit Hilfe der sogenannten *FW-I Single Router Security Extension* das *Internet Gateway Security Center* um die Funktionalität zum Steuern eines einzigen Routers erweitert werden.

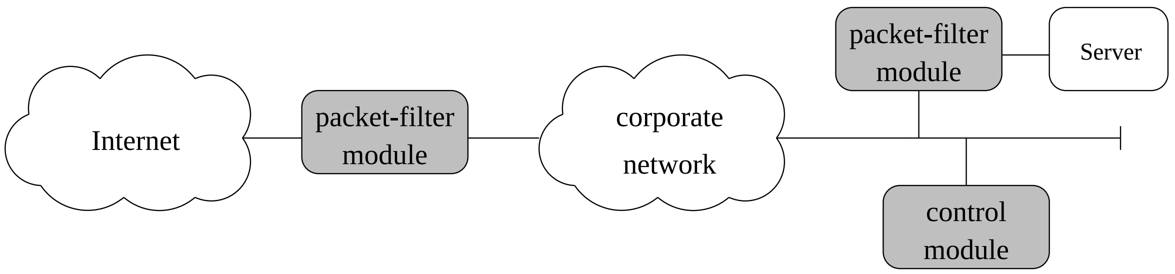

Abbildung 5.3: Beispielsszenario für den Sun Firewall-I

5.2.2 Bewertung

Auch der Firewall-I wurde von der Firma Sun zu Testzwecken in der Version 1.0 zur Verfügung gestellt und konnte somit einer genauen Untersuchung unterzogen werden.

Bewertung aufgrund der Kriterien aus den Dienstspezifikationen

- Behandlung von TCP-basierenden Client/Server-Diensten

  Wie mit jedem anderen Paket-Filter auch, können TCP-basierende *client/server*-Dienste mit dem FW-I derart gesichert werden, daß man Verbindungen nur von innen nach außen gestattet. Verwendet man die Version 1.2 des Firewalls, so ist es sogar möglich, diese Dienste von außen zugänglich zu machen, da hier Authentifikation über *one-time passwords* unterstützt wird.

- Behandlung von UDP-basierenden Client/Server-Diensten

  Eine der wesentlichen Erweiterungen gegenüber dem herkömmlichen Paket-Filter Konzept ist die Möglichkeit der Sicherung UDP-basierender *client/server*-Dienste. Gewöhnlich möchte man internen Benutzern die Möglichkeit bieten, UDP-Dienste zu nutzen, ohne dies auch externen Benutzern zu gestatten. Da man bei einem von außen eintreffenden UDP-Paket aber nicht unterscheiden kann, ob es sich um eine Anfrage von außen, also einen versuchten Angriff, oder um eine Antwort auf eine Anfrage eines internen Benutzers handelt, werden *client/server*-Dienste über UDP üblicherweise von Paket-Filtern völlig verboten.

  Der Firewall-I begegnet diesem Problem folgendermaßen: Sendet ein interner Benutzer eine Anfrage an einen externen *server*, so wird dies in eine Tabelle eingetragen. Trifft nun ein Paket von außen ein, so kann in der Tabelle nachgesehen werden, ob dieses Paket als Reaktion auf eine Anfrage eines internen Benutzers verschickt wurde oder nicht. Ist dies der Fall, so wird das Paket weitergeleitet, andernfalls verworfen. Es läßt sich ein Zeitraum von 0 bis 300 Sekunden einstellen, in dem das betreffende Antwortpaket eintreffen muß, andernfalls wird die Verbindung als beendet betrachtet.

  Bei Verwendung von Version 1.2 ist natürlich ebenfalls die Verwendung von *one-time passwords* möglich, sodaß selbst UDP-basierende *client/server*-Dienste gefahrlos von außen zugelassen werden können.

- Behandlung von TCP-basierenden Peer-to-Peer-Diensten

  Diese Dienste können wie üblich mit einem Paket-Filter gesichert werden, sodaß entweder Verbindungen nur von innen nach außen aufgebaut werden können oder aber daß auch externe Rechner Verbindungen nach innen initiieren können.

- Behandlung von UDP-basierenden Peer-to-Peer-Diensten

  Im Unterschied zu normalen Paket-Filtern, die, wenn sie derartige Dienste erlauben, diese in beide Richtungen gestatten müssen, kann der Sun FW-I dies auch auf Verbindungen beschränken, die von internen Rechnern ausgehen. Dies geschieht wiederum über die Tabelle ausstehender UDP-Antwortpakete.

- Behandlung von Sonderfällen

  – Behandlung von FTP

    Traditionelle Paket-Filter haben große Schwierigkeiten, *file transfer* über *FTP* angemessen zu sichern, da *FTP* es erfordert, die Daten auf einer zweiten, vom Server initiierten Verbindung zu übertragen. Da nicht bekannt ist, welcher Port auf *client*-Seite zum Empfang dieser Daten verwendet wird, muß man entweder den gesamten nicht-privilegierten Bereich öffnen oder aber *FTP* verbieten.

    Auch hierfür bietet der FW-I eine Lösung: Der Firewall überwacht die Kontroll-Verbindung und trägt ein gesendetes PORT-Kommando, also ein Kommando, daß dem *server* mitteilt, zu welchem Port er die Datenverbindung aufbauen soll, in eine Tabelle ein. Somit kann bei einer von außen initiierten TCP-Verbindung immer untersucht werden, ob es sich um eine Datenverbindung einer bestehenden *FTP*-Verbindung handelt oder nicht. Ist dies der Fall, so ist die Verbindung natürlich zuzulassen, andernfalls handelt es sich vermutlich um einen versuchten Angriff und das entsprechende Paket ist zu verwerfen.

- Behandlung von SMTP

  Mail wird bei diesem Firewall zu einem internen *mail-server* durchgelassen, wo die Verteilung der Nachrichten an die entsprechenden Empfänger stattfinden kann.

- X11

  Der Sun Firewall besitzt keine Möglichkeiten, *X11* zu sichern, die über die normalen Möglichkeiten eines Paket-Filters hinausgehen. Deshalb sollte *X11* verboten werden.

- Behandlung RPC-basiernder Dienste

  Selbst die Behandlung RPC-basierender Dienste ist mit dem Sun Firewall möglich. Hierzu erfrägt er dynamisch und transparent die Portnummern der verschiedenen Dienste bei den *portmappern* der einzelnen Rechner und legt eine Tabelle an. Somit kann erkannt werden, um welchen Dienst es sich handelt und gewisse Dienste können gestattet werden. In Verbindung mit der Tabelle für die ausstehenden UDP-Verbindungen ergibt sich also eine Möglichkeit, RPC-basierende Dienste zu sichern.

Bewertung aufgrund weiterer Kriterien

1. **Audit**

   - Logging

     Der Firewall-I verfügt über ausgiebige Logging-Möglichkeiten. Folgende Ereignisse werden vom FW-I mitgeloggt:

     - Kontroll-Operationen

       Hierbei handelt es sich z. B. um die Veränderung oder Neuinstallation von Regeln auf einem der *packet filter modules.*

     - eintreffende Pakete

       Auf sämtliche eintreffenden Pakete werden die vom Administrator angegebenen Regeln angewandt. Zu jeder Regel kann mitangegeben werden, ob ein Paket, das aufgrund dieser Regel transportiert oder verworfen wurde, einen Eintrag im Log-File verursachen soll oder nicht. Die Log-Einträge verfügen über alle wichtigen Informationen mit folgenden Ausnahmen:

       * auslösende Regel

         Es wäre wünschenswert zu wissen, aufgrund welcher Regel das jeweilige Paket transportiert oder verworfen wurde. Dies ist leider in der derzeitigen Version nicht möglich.

       * transportiertes Volumen

         Die Anzahl der übertragenen Bytes wäre wichtig, um die Log-Einträge gleichzeitig im Rahmen des Accounting einsetzen zu können.

Leider werden sämtliche Pakete unabhängig voneinander geloggt, anstatt einen Eintrag bei Beginn der Verbindung und einen Eintrag bei Verbindungsende zu erzeugen. Dies würde einerseits die Anzahl der Einträge deutlich verringern, andererseits die Übersichtlichkeit der Log-Files erhöhen.

Zur Betrachtung der Log-Files steht der sogenannte *Log Viewer* zur Verfügung, mit dessen Hilfe die Log-Einträge sehr übersichtlich angezeigt werden können. Verwendung des *Log Viewers* bietet folgende Vorteile:

- farbliche Unterscheidungen
  Der *Log Viewer* stellt verworfene Pakete in rot, weitertransportierte Pakete in grün und Kontroll-Operationen in grau dar. Somit wird es für einen menschlichen Betrachter einfacher, die jeweils relevanten Informationen zu erkennen.
- Verbergen nicht-relevanter Informationen
  Es besteht die Möglichkeit, nur die Informationen anzeigen zu lassen, die für die jeweilige Betrachtung relevant erscheinen. Bei der großen Menge an Informationen ist dies natürlich für den Betrachter ein angenehmer Vorteil, da er nicht durch nicht-relevante Informationen gestört wird.
- Suchmöglichkeiten
  Der *Log Viewer* verfügt über die Möglichkeit, daß Log-File nach bestimmten Kriterien zu durchsuchen. Gibt man einen Zeitraum an, in dem gesucht werden soll, sowie Kriterien, die auf die zu suchenden Pakete zutreffen sollen, so bekommt man sämtliche zutreffenden Einträge geliefert.
- Verbergen ähnlicher Einträge
  Da aufeinanderfolgende Log-Einträge, die sich nur in Datum und Uhrzeit unterscheiden, dem Betrachter keinerlei neue Informationen liefern, da sie offensichtlich zu ein und derselben Verbindung gehören, können diese verborgen werden, um die Übersichtlichkeit des betrachteten Log-Files zu erhöhen.

• Alerting
  Auch im Bereich des Alerting stehen umfangreiche Möglichkeiten zur Verfügung. Wiederum kann bei jeder Regel angegeben werden, ob eine Benachrichtigung bei Ausführung dieser Regel durchgeführt werden soll oder nicht. Ist dies der Fall, so existieren folgende Möglichkeiten:

  - Verschicken einer *mail*
  - Öffnen eines Fensters
  - Ausführen eines benutzerdefinierten Kommandos

2. sicherheitsrelevante Kriterien

• Authentifikationsmöglichkeiten
  In der hier untersuchten Version 1.0 bietet der Firewall keine Authentifikationsmöglichkeiten, die eine Nutzung interner *server* für externe Benutzer zulassen würden. Für die Version 1.2 ist allerdings die Integration folgender Produkte angekündigt:

  - Security Dynamics SecureID Cards
  - Bellcore S/Key

Man kann dann beim Aufstellen der Regeln für einen bestimmten Dienst angeben, nach welcher Methode der Benutzer authentifiziert werden soll. Trifft dann ein Paket ein, das zur Eröffnung einer entsprechenden Verbindung geeignet ist, so wird zuerst eine Authentifikation des Benutzers durchgeführt, bevor das Paket weitergeleitet wird. Die Authentifikation durch den eigentlichen *server* bleibt hiervon unberührt. Zusätzlich zur Sicherung über die genannten *one-time password*-Mechanismen kann auch eine Authentifizierung über Standard-UNIX-Paßworte erfolgen.

- Sicherheit der Firewall-Komponenten
  - Fehlerfreiheit der Firewall-Software
    Über die Anfälligkeit der Software für Fehler lassen sich keine genauen Aussagen machen. Man kann aber sagen, daß es sich um neuentwickelte Software handelt, wobei natürlich Wert auf Fehlerfreiheit gelegt wurde, was bei Verwendung von Standard-Software nicht unbedingt gesagt ist. Insbesondere ist der Firewall nicht von *sendmail* abhängig, was ihn anderen Firewalls gegenüber deutlich sicherer erscheinen läßt.
  - Existenz von User-Accounts
    Das *control module* des Firewalls befindet sich üblicherweise auf der Workstation des Firewall-Verantwortlichen. Normale Benutzer haben hierfür normalerweise keine Accounts. Die *packet filter modules* befinden sich hingegen zum Teil auf Rechnern, auf denen auch normale Benutzer über Accounts verfügen. Da Änderungen an der Konfiguration aber nur vom *control module* aus vorgenommen werden können, besteht hier keine Gefahr.
  - hierarchische Sicherheitsabstufungen
    Es besteht die Möglichkeit, mehrere *packet-filter modules* einzusetzen und somit einzelne Subnetze oder Rechner gesondert zu schützen. Selbst wenn dann ein Einbruch in einen anderen Rechner geglückt sein sollte, ist der zusätzlich geschützte Rechner weiterhin sicher.
  - periodische Prüfsummenbildung
    Eine Bildung von Prüfsummen über die Konfigurationsfiles ist nicht vorgesehen.
  - administrativer Zugang über das Netz
    Die Workstation, auf der das *control module* läuft, läßt sich ganz normal über das Netz erreichen. Die Verbindung zwischen *control module* und *packet filter modules* wird alledings über ein *one-time password*-Schema abgesichert, sodaß kein Angreifer die Möglichkeit hat, die Konfiguration eines *packet filter modules* zu verändern, indem er sich für das *control module* ausgibt.
- Domain Name Service
  Der Sun Firewall besitzt in der derzeitigen Version 1.0 keine Möglichkeit, die Adressen des internen Netzes nach außen hin zu verbergen. Stattdessen werden Anfragen zu oder von einem *name server* explizit gestattet. Die Version 1.2 ermöglicht das Verbergen der internen Adressen bei von innen initiierten Verbindungen, indem die Adresse des Firewalls als Absenderadresse eingetragen wird. Die Zuordnung der Antwortpakete zu der entsprechenden Verbindung erfolgt mit Hilfe unterschiedlicher Portnummern. Darüberhinaus kann der Firewall auch eine Adreßumsetzung

durchführen, wenn intern keine offiziellen IP-Adressen verwendet werden, der Aufwand einer Umkonfiguration aber zu groß wäre. Mit Hilfe einer Tabelle wird hier jedem internen Rechner eine offizielle Adresse zugeordnet, sodaß bei Verbindungen mit dem externen Netz diese Adresse als Absender bzw. Ziel eingetragen werden kann.

3. Administration

- Installation und Konfiguration
  Die Installation des Firewall-I bereitet keine Probleme und kann innerhalb weniger Minuten erfolgen.
  Die Konfiguration ist ebenfalls relativ einfach, kann aber speziell bei größeren Netzen einige Zeit in Anspruch nehmen. Es existiert eine graphische Benutzeroberfläche, die die Konfiguration wesentlich erleichtert. Für die Konfiguration sind die folgenden drei Fenster relevant:
  - Network Object Manager
    Mit Hilfe des *Network Object Managers* definiert man die Objekte, die bei der Aufstellung der Regeln verwendet werden sollen. Dies können Hosts, Gateways, Router oder ganze Subnetze sein. Man kann hierarchische Gruppen von Objekten bilden, um leichter verständliche Regeln zu ermöglichen. Für jedes Objekt müssen Daten wie z. B. Name, Adresse und angeschlossene Interfaces angegeben werden. Verfügt das Objekt über einen SNMP-Agenten, so können einige Informationen über diesen beschafft werden, um die Konfiguration zu vereinfachen.
  - Services Manager
    Als nächstes müssen mit dem *Services Manager* sämtliche verwendeten Dienste beschrieben werden. Die Standard-Dienste sind bereits enthalten, sodaß nur einige wenige Dienste konfiguriert werden müssen. Hier muß man z. B. den Namen und den Port des Dienstes angeben. Der FW-I versucht aufgrund des Namens, weitere Informationen wie die Portnummer aus Systemfiles wie `/etc/services` oder über NIS zu ermitteln, um die Konfiguration zu vereinfachen.
  - Rule Base Editor
    Mit dem *Rule Base Editor* werden die Filterregeln festgelegt, die auf den verschiedenen *packet filter modules* installiert werden sollen. Hierzu gibt man jeweils eines der mit dem *Network Object Manager* definierten Objekte als Quell- bzw. Zieladresse an, sowie einen mit dem *Services Manager* definierten Dienst. Weiterhin muß nur noch eine Aktion festgelegt werden, die mit einem entsprechendem Paket ausgeführt werden soll, also ob es transportiert oder verworfen werden soll. Die verbleibenden Einstellungen dienen dazu, festzulegen ob ein Log-Eintrag angefertigt werden soll, wenn die Regel auf ein Paket zutrifft, sowie um zu bestimmen, in welchem der *packet filter modules* die Regel aktiviert werden soll.

  Nachdem die Filterregeln definiert wurden, können sie in den entsprechenden *packet filter modules* bzw. Routern installiert werden. Bevor dies geschieht wird die Konsistenz der Regeln durch den Firewall überprüft, um so bereits einen ersten Schutz vor Fehlkonfigurationen zu erhalten.

- Support

  Im Kaufpreis sind drei Updates sowie der technische Support für die ersten drei Monate enthalten [Farrow92]. Danach sind für den Support pro Jahr 20% des Kaufpreises des Firewalls zu entrichten.

- Dokumentation

  Die Dokumentation ist leicht verständlich und beschreibt die Installation und Konfiguration des Firewalls sehr gut. Das Testen der installierten Filterregeln bzw. die Erstellung sinnvoller Filterregeln wird aber überhaupt nicht behandelt, sodaß der Verantwortliche bereits über ausgiebige Erfahrung in der Erstellung von Filteregeln verfügen sollte.

- SNMP-Unterstützung

  Der FW-I verfügt über zwei Arten von SNMP-Unterstützung. Zum einen kann er, wie bereits erwähnt, bei der Definition der im Netz enthaltenen Objekte gewisse Informationen über den SNMP-Agenten des jeweiligen Objektes einholen, zum anderen besitzt er selber einen SNMP-Agenten, der es ermöglicht, Status-Informationen über eine Managementanwendung abzufragen.

- Behandlung proprietärer bzw. neuer Dienste

  Die Integration von proprietären bzw. neuen Diensten stellt keinerlei Problem dar, da mit Hilfe des *Services Managers* die Eigenschaften dieser Dienste beschrieben werden können, woraufhin sie beliebig in Filteregeln verwendet werden können. Die Integration eines weiteren Dienstes ist innerhalb kürzester Zeit zu realisieren.

4. Benutzerfreundlichkeit

- Transparenz

  Der Firewall-I ist sowohl für die Benutzer als auch für die Anwendungen völlig transparent. Die Pakete werden weiterhin an die tatsächliche Zieladresse adressiert und werden nur auf dem Weg dorthin einer eingehenden Untersuchung unterzogen. Somit ist weder eine Änderung der verwendeten *client*-Software erforderlich, noch müssen sich die Benutzer auf neue Arbeitsgewohnheiten einstellen. Selbsverständlich kann bei zusätzlicher Authentifizierung eines Benutzers in der Version 1.2 die Transparenz nicht völlig aufrechterhalten werden.

- Performance

  Ein Performance-Test einer Computerzeitschrift konnte keine Einbußen der Performance bei Verwendung des Sun FW-I feststellen [Farrow92].

5. Kosten

- Kaufpreis

  Je nachdem, ob nur die Version für kleine Netze oder das *Network Security Center* benötigt wird, bewegen sich die Preise zwischen $9900 und $39900.

- Hardware-Voraussetzungen

  Der Sun Firewall läuft auf Sun SPARC oder x86-basierenden Systemen mit dem Betriebssystem SunOS 4.1.3 oder Solaris 2.3. Der für das *control module* eingesetzte Rechner muß über mindestens 16MB Hauptspeicher verfügen, während die Rechner für die *packet filter modules* keine bestimmten Voraussetzungen erfüllen müssen. Da die Module auf bereits vorhandene Rechner bzw. Router installiert werden können, entstehen keine weiteren Kosten für Hardware-Anschaffungen.

5.3 TIS FWTK

5.3.1 Beschreibung

Der TIS Firewall Toolkit (FWTK) setzt sich aus einem TCP-Relay sowie diversen Proxies zusammen [Ranum94]. Wie der Name bereits sagt, handelt es sich dabei nicht um einen Firewall, der mit wenigen Handgriffen zu installieren ist, sondern vielmehr um eine Ansammlung von Werkzeugen, die ganz nach den speziellen Anforderungen des jeweiligen Unternehmens zusammengestellt werden können. Es können folgende Firewall-Architekturen realisiert werden [TIS94b]:

- Dual-homed Gateway

  Beim *dual-homed gateway* stellt der Rechner, auf dem der Firewall läuft, die einzige Verbindung zwischen dem externen und dem internen Netz dar. Es findet kein Routing zwischen den beiden Netzen statt, sodaß eine Weiterleitung eines eintreffenden Paketes nur über einen der Proxies oder das TCP-Relay möglich ist. Eventuell kann noch ein Router eingesetzt werden, um zusätzlich einen Paket-Filter einzubauen, der bestimmten Angriffen, wie z. B. dem *IP-Spoofing* entgegenwirken kann. Abbildung 5.4 stellt diese Architektur dar.

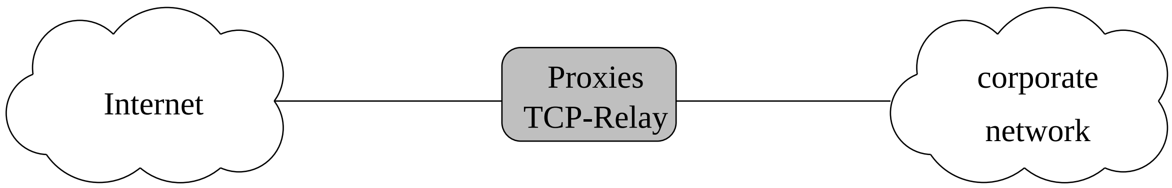

Abbildung 5.4: TIS FWTK als *dual-homed gateway*

- Screened Host Gateway

  Bei dieser Architektur, die in Abbildung 5.5 skizziert ist, befindet sich der Firewall-Rechner innerhalb des internen Netzes, wobei ein vorgeschalteter Router nur Pakete zwischen dem externen Netz und dem Firewall-Rechner gestattet. Es ist also keine direkte Verbindung zwischen dem sicheren und dem unsicheren Netz möglich.

- Screened Subnet Gateway

  Diese Lösung besteht aus einem eigenen Subnetz, in dem der Firewall-Rechner liegt und das über jeweils einen Router mit dem sicheren und dem unsicheren Netz verbunden ist. Dies ist in Abbildung 5.6 dargestellt.

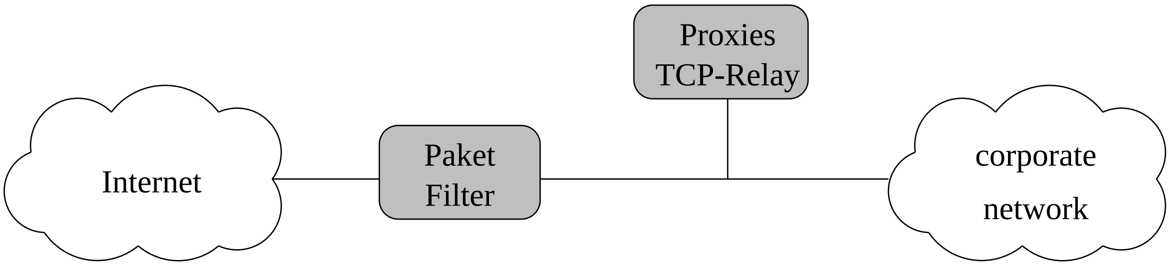

Abbildung 5.5: TIS FWTK als *screened-host gateway*

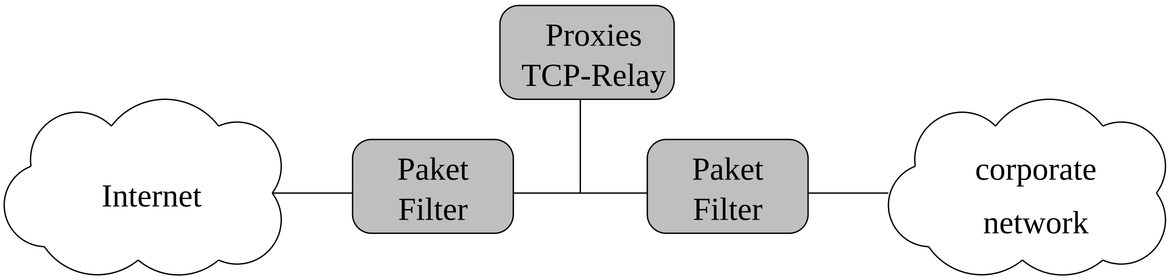

Abbildung 5.6: TIS FWTK als *screened-subnet gateway*

Im folgenden wird beispielhaft die *screened host* Lösung beschrieben. Wo sich große Unterschiede bei Verwendung einer anderen Architektur ergeben würden, wird natürlich gesondert darauf hingewiesen. Die Kombination der einzelnen Firewall-Konzepte zeigt Abbildung 5.7. Man erkennt, daß die Proxies parallel zu dem TCP-Relay existieren, während der Paket-Filter vorgeschaltet ist. Ebenfalls ist eine ausschließliche Verwendung des Paket-Filters denkbar.

Der Paket-Filter

Der Toolkit verfügt selber nicht über einen Paket-Filter. Allerdings besteht natürlich die Möglichkeit, im vorgeschalteten Router Filterregeln zu installieren, die bestimmte Pakete abblocken, sodaß sie gar nicht erst den eigentlichen Firewall-Rechner erreichen können.

Das TCP-Relay

Im Umfang des Firewall ist ein Programm namens *plug-gw* enthalten, daß Funktionen ähnlich denen eines TCP-Relays erfüllen kann [TIS93]. Der *inetd*-Daemon wird so konfiguriert, daß er bei Eintreffen eines Verbindungswunsches, der über *plug-gw* behandelt werden soll, dieses Programm startet und den jeweiligen Port als Parameter übergibt. *Plug-gw* untersucht nun sein Konfigurationsfile nach einer Regel, die die Adresse des Absenders sowie den entsprechenden Port enthält. Diese Regel gibt dann an, zu welchem Rechner die Verbindung weitergeleitet werden soll. Hier kann eventuell auch noch ein anderer Port angegeben werden.

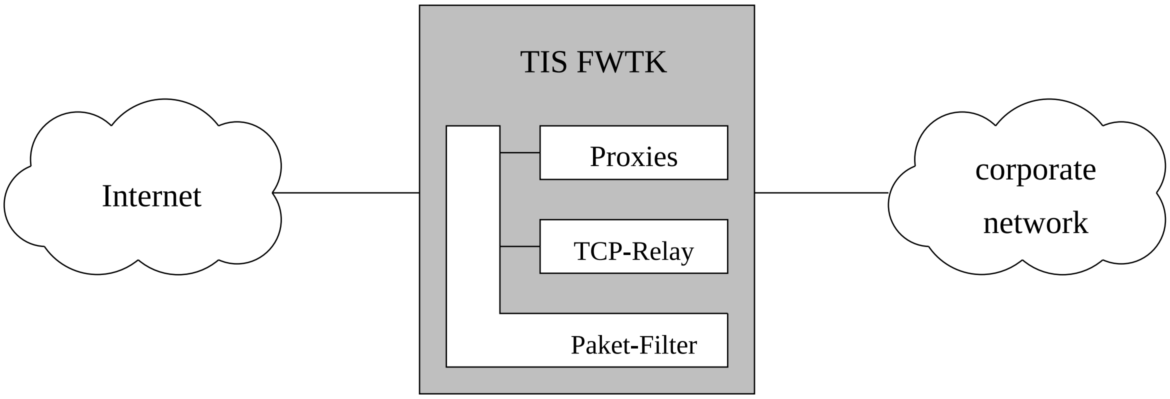

Abbildung 5.7: Kombination der Firewall-Konzepte beim *screened-host gateway*

Es ist somit nicht möglich, Verbindungen zu beliebigen, nicht vorher bekannten, Rechnern mit diesem Programm zu sichern. Stattdessen können nur Verbindungen zugelassen werden, bei denen im voraus bekannt ist, zwischen welchen Rechnern die Verbindung bestehen soll.

Die Proxies

Der *TIS Toolkit* enthält folgende Proxies:

- Telnet

  Der *telnet*-Proxy ermöglicht die sichere Benutzung des *telnet*-Protokolls durch den Firewall. Der Verbindungsaufbau erfolgt in zwei Schritten: Der Benutzer eröffnet eine *telnet*-Verbindung zum Firewall, wo der *telnet*-Proxy über den *inetd*-Daemon gestartet wird. Je nach Konfiguration des Proxies findet dann eine Authentifikation des Benutzers statt. Nach erfolgter Authentifikation gibt der Benutzer einen Rechner an, der das eigentliche Ziel der Verbindung sein soll. Die Entscheidung über die Zulässigkeit der Verbindung wird aufgrund der IP-Adressen von Quell- und Zielrechner sowie aufgrund der Identität des Benutzers getroffen. Jeder Verbindungswunsch, egal ob gestattet oder zurückgewiesen, erzeugt einen Eintrag im Log-File.

- Rlogin

  Um die Notwendigkeit einer zweifachen Authentifizierung, einerseits am Firewall und andererseits am jeweiligen Zielrechner, zu umgehen, exisitiert ein Proxy für *rlogin*. Der *rlogin*-Proxy läßt sich genau wie der *telnet*-Proxy konfigurieren und ist darüberhinaus in der Lage, die entsprechende Verbindung in nur einem Schritt aufzubauen. Hierbei wird ausgenutzt, daß das *rlogin*-Protokoll einen alternativen Benutzername beinhalten kann, der beim *login* verwendet werden soll. Statt diesem Benutzernamen gibt man eine Kombination aus gewünschtem Zielrechner und Benutzernamen in der Form `user@host` an. Die Verbindung wird dann automatisch erzeugt.

- FTP

  Um *file transfer* durch den Firewall zu ermöglichen, enthält der *TIS Toolkit* einen *FTP*-Proxy. Der Verbindungsaufbau erfolgt wiederum zweistufig, d.h. ein Benutzer eröffnet eine *FTP*-Verbindung zum Firewall und gibt dort seinen Namen in der Form `user@host` an. Nach eventueller Authentifizierung des Benutzers wird die Verbindung zu dem angegebenen Rechner aufgebaut, wenn das Konfigurationfile eine Regel enthält, die eine derartige Verbindung gestattet. Nach Aufbau der Verbindung werden die übertragenen Kommandos überwacht und es kann für jedes Kommando einzeln festgelegt werden, ob ein derartiges Kommando übertragen werden soll, ob bei Eintreffen dieses Kommandos ein Log-Eintrag vorgenommen werden soll, ob das Kommando nur nach vorhergehender Authentifizierung des Benutzers gestattet werden soll oder ob das Kommando ganz verboten werden soll.

- HTTP

  Weiterhin existiert ein *HTTP*-Daemon, der Verbindungen zu *HTTP-servern* sowie zu *gopher-servern* gestattet. Wiederum ist es möglich, den Benutzer zu authentifizieren, nur Verbindungen zwischen bestimmten Rechnern oder Subnetzen zu gestatten sowie nur bestimmte Kommandos zuzulassen. Auch ein selektives Logging einzelner Kommandos kann erfolgen. Der *HTTP*-Proxy arbeitet sowohl mit *clients*, denen die Existenz eines Proxies bekannt ist als auch mit den Standard-Versionen.

- X11

  Der *X11*-Proxy wird vom *telnet*-Proxy oder von *rlogin*-Proxy aus gestartet. Somit beinhaltet er keine eigenen Authentifizierungsmaßnahmen. Ein *client*, der auf einen *X-server* auf der anderen Seite des Firewalls zugreifen möchte, leitet seine Ausgaben auf ein virtuelles *display* auf dem Firewall, von wo aus die Verbindung zum tatsächlichen *display* des Benutzers aufgebaut wird. Bevor dies geschieht, muß aber der entsprechende Benutzer bestätigen, daß es sich um eine rechtmäßige Verbindung zu seinem *X-server* handelt. Hierzu wird auf seinem Bildschirm ein Fenster geöffnet, daß um eine Bestätigung der Verbindung bittet. Ist dies nicht der Fall, so wird die Verbindung abgebrochen.

- SMTP

  Electronic mail wird bei Verwendung des *TIS FWTK* durch einen dreigeteilten Proxy weitergeleitet. Eine eintreffende Nachricht wird zuerst von einem Programm namens *smap* empfangen, das eine minimale Version des *SMTP*-Protokolls implementiert. Dieses Programm legt die *mail* in einem Verzeichnis ab, das in regelmäßigen Abständen von einem zweiten Programm, dem sogenannten *smapd*-Daemon untersucht wird. Dieser Daemon gibt alle in diesem Verzeichnis gefundenen Nachrichten an das *sendmail*-Programm weiter, das die Weiterleitung der *mail* übernimmt. *Sendmail* läuft hier nicht als Daemon sondern wird von *smapd* jeweils neu gestartet.

5.3.2 Bewertung

Da der *TIS FWTK* ein Produkt ist, das kostenlos zur Verfügung steht, konnte er natürlich auch einer eingehenden Prüfung unterzogen werden.

Bewertung aufgrund der Kriterien aus den Dienstspezifikationen

- Behandlung TCP-basierender Client / Server-Dienste

  Da das hier eingesetzte TCP-Relay für die Behandlung von *client/server*-Diensten nicht in Frage kommt, bleiben zwei Möglichkeiten, diese Dienste zu sichern:

  - Verwendung von Proxies

    Telnet, *rlogin* und *HTTP/gopher* können mit Hilfe der Proxies gut gesichert werden, wobei von außen initiierte Verbindungen aufgrund der enthaltenen Authentifikationsmöglichkeiten keine Probleme bereiten.

  - Verwendung des Paket-Filters

    Ebenso lassen sich TCP-basierende Dienste natürlich mit dem Paket-Filter sichern. Dies bringt natürlich wieder die Einschränkung mit sich, daß die Dienste nur von innen nach außen angeboten werden können.

- Behandlung UDP-basierender Client / Server-Dienste

  Es existieren weder Proxies für UDP-basierende Dienste, noch kann das TCP-Relay hierfür eingesetzt werden. Da der Paket-Filter bekanntermaßen nicht in Frage kommt, um diese Dienste zu sichern, bleibt nur das totale Verbot als Alternative.

- Behandlung TCP-basierende Peer-to-Peer-Dienste

  Diese Dienste können nun hervorragend mit dem TCP-Relay abgesichert werden. Es ist möglich, Verbindungen nur zwischen zwei bestimmten Rechnern zu gestatten, wobei der Rechner, der die Verbindung wünscht, gar nicht wissen muß, mit welchem Rechner er verbunden wird. Weiterhin bietet sich die Möglichkeit, mit Hilfe entsprechender Kofigurationsregeln die Eröffnung der Verbindung nur in einer Richtung zu erlauben. Selbstverständlich kommt auch der Paket-Filter zum Schutz dieser Dienste in Frage, die Verwendung des TCP-Relays ist aber z. B. wegen des besseren Loggings anzuraten.

- Behandlung UDP-basierende Peer-to-Peer-Dienste

  Wiederum kommt nur der Paket-Filter in Betracht, um diese Dienste zuzulassen. Dabei ist natürlich wieder zu beachten, daß ein Angreifer die Identität eines anderen Rechners relativ leicht übernehmen kann.

- Behandlung von Sonderfällen

  - Behandlung von FTP

    Der *FTP*-Proxy bietet gute Möglichkeiten, *file transfer* durch den Firewall abzusichern. Er läßt sich auf die individuellen Wünsche jedes Unternehmens einrichten und bietet auch ausreichende Authentifikationsmöglichkeiten, um von außen initiierte Verbindungen zuzulassen.

  - Behandlung von SMTP

    Electronic mail wird transparent durch den Firewall weitergeleitet, wobei natürlich in Kauf genommen werden muß, daß das empfangende *smap*-Programm nicht die gesamte Funktionalität von *sendmail* zur Verfügung stellen kann. Dies ist aber aus Gründen der Sicherheit auch gar nicht erwünscht. Eine Möglichkeit, den Absender

der Nachricht zu überprüfen oder *mail*-Inhalte auf eventuelle Angriffsversuche zu untersuchen ist nicht vorgesehen.

- Behandlung von X11

  Der *X11*-Proxy stellt einen vernünftigen Ansatz dar, um das schwierig abzusichernde *X11*-Protokoll durch den Firewall anbieten zu können. Die ausdrückliche Bestätigung jeder Verbindung durch den entsprechenden Benutzer läßt sich aus Gründen der Sicherheit nicht vermeiden.

- Behandlung RPC-basierender Dienste

  Zur Behandlung RPC-basierender Dienste stellt der Firewall keine Möglichkeiten zur Verfügung.

Bewertung aufgrund weiterer Kriterien

1. **Audit**

   Das Audit des *TIS Toolkit* erfolgt mit Hilfe des Standard-Programms *syslogd*, wobei einige Änderungen vorgenommen wurden, die ein Alerting erlauben.

   - Logging
     - Logging bei den Proxies

       Die Proxies erstellen einen Log-Eintrag bei jedem Versuch, eine Verbindung zum Proxy aufzubauen, egal ob sie gestattet oder zurückgewiesen wurde. Dieser Eintrag enthält z. B. folgende Informationen:

       * Datum
       * Uhrzeit
       * Proxy

         Sowohl der Name als auch die Prozeß-ID des kontaktierten Proxies wird geloggt.
       * Aktion

         Je nachdem, ob die Verbindung akzeptiert oder zurückgewiesen wird, wird hier *permit* oder *deny* eingetragen.
       * Host

         Hier steht sowohl der Name als auch die IP-Adresse des Rechners, der die Verbindung wünschte.

       Versucht ein Benutzer nun die Verbindung zum eigentlichen Zielrechner aufzubauen, so erfolgen weitere Einträge. Der erste Eintrag gibt an, ob der Benutzer die Berechtigung zu dieser Verbindung besitzt, der zweite Eintrag gibt an, ob der Verbindungsaufbau geglückt ist oder nicht und ein dritter Eintrag wird bei Verbindungsabbau vorgenommen. Dieser enthält dann folgende Informationen:

       * Aktion

         Im Falle des Verbindungsabbaus steht als Aktion *exit* eingetragen.
       * Host

         Name und IP-Adresse des Quellrechners
       * Destination

         Name und IP-Adresse des Zielrechners

    * in / out
      Anzahl der übertragenen Bytes in der entsprechenden Richtung.
    * user
      Wurde der Benutzer authentifiziert, so steht hier der Name des Benutzers, andernfalls *noauth*.
    * duration
      Dauer der Verbindung.

  Bestimmte Proxies, wie z. B. der *FTP*-Proxy verfügen darüberhinaus noch über Konfigurationsmöglichkeiten, die es erlauben, bei Auftreten bestimmter Kommandos, einen Log-Eintrag zu erzeugen.
  - Logging beim TCP-Relay
    Beim TCP-Relay gestaltet sich das Logging analog zu dem der Proxies. Zusätzlich wird noch jeweils der Port angegeben, auf dem die Verbindung eintraf und zu dem die Verbindung weitergeleitet wurde. Es ist ebenfalls möglich, die Dauer und das Übertragungsvolumen der Verbindung zu erkennen.
  - Logging beim Paket-Filter
    Über das Logging des Paket-Filters lassen sich keine konkreten Aussagen machen, da es sich nicht um einen Teil des *Toolkits* handelt, sondern um einen beliebigen Router. Im allgemeinen verfügen Router aber nicht über ausreichende Logging-Mechanismen.

  Zusätzlich werden noch besondere Ereignisse, wie z. B. jeder Authentifikationsvorgang, gleichgültig ob gelungen oder fehlgeschlagen, mitgeloggt.

  Das Logging des *TIS FWTK* ist also gut. Insbesondere ist es möglich, ein Accounting durchzuführen, das auf den Logging-Informationen des Firewalls basiert, sofern der gesamte Verkehr durch den Firewall-Rechner läuft und nicht der Router so konfiguriert ist, daß er bestimmte Dienste gestattet, ohne einen Proxy oder das TCP-Relay zu verwenden.

- Alerting

  Der verwendete *syslogd*-Daemon verfügt über Möglichkeiten, reguläre Ausdrücke zu definieren, bei deren Auftreten in einer Zeile des Log-Files ein Kommando ausgeführt werden kann. Somit kann das Log-File in Echtzeit überwacht werden und der Administrator ist in der Lage, selber zu bestimmen, bei welchen Ereignissen er benachrichtigt werden möchte und auf welche Weise dies geschehen soll.

2. sicherheitsrelevante Kriterien

- Authentifikationsmöglichkeiten

  Der *TIS FWTK* besitzt einen Authentifikationsserver, der von den Proxies verwendet wird, um Benutzer zu authentifizieren. Dieser *server* stellt derzeit folgende Möglichkeiten zur Verfügung, die Identität eines Benutzers festzustellen:

  - Standard UNIX-Paßworte
  - Security Dynamics SecureID Cards
  - Digital Pathways SNK
  - Bellcore S/Key

Der Authentikikationsserver verfügt über eine Schnittstelle, mit der sehr einfach Benutzer hinzugefügt oder mit einem Paßwort versehen werden können. Es existiert ebenfalls ein *client*, der verwendet werden kann, um die Konfiguration von einem entfernten Rechner aus durchzuführen. Hierbei wird eine Verschlüsselung des Verkehrs vorgenommen, um einem Angreifer keine Möglichkeit zu geben, Paßworte mitzuhören.

- Sicherheit der Firewall-Komponenten
  - Fehlerfreiheit der Firewall-Software
    Alle Komponenten des *TIS Toolkit* wurden mit Blick auf die Sicherheit erstellt. Es handelt sich um sehr einfache Programme, deren Fehlerfreiheit zum Teil innerhalb kürzester Zeit zu überprüfen ist. Außerdem laufen alle Komponenten, die in direktem Kontakt zu eventuellen Angreifern stehen, in einer Umgebung, die mit `chroot` ein leeres Verzeichnis als *root*-Verzeichnis besitzt und es sind keinerlei *root*-Privilegien erforderlich. Man kann also davon ausgehen, daß die Software kaum Fehler enthält und wenn doch ein Fehler auftritt, daß er nur sehr schwer auszunutzen sein wird.
  - Existenz von User-Accounts
    Es ist nicht erforderlich, User-Accounts auf dem Firewall-Rechner einzurichten.
  - hierarchische Sicherheitsabstufungen
    Es sind keine Maßnahmen vorgesehen, bestimmten Rechnern oder Subnetzen einen besonderen Schutz zukommen zu lassen. Einzig die Möglichkeit, die Konfiguration so zu wählen, daß bestimmte Rechner gar keine Verbindungen aufbauen können, besteht natürlich.
  - periodische Prüfsummenbildung
    Der Firewall selber enthält keine Möglichkeit, die Konfiguration mittels einer Prüfsumme zu überwachen. Bestimmte Betriebssysteme besitzen aber *tools*, die hierfür verwendet werden können.
  - administrativer Zugang über das Netz
    Es besteht die Möglichkeit, *telnet* und *FTP* zu administrativen Zwecken zu gestatten. Hierzu steht ein Programm namens *netacl* zur Verfügung, das über den *inetd*-Daemon gestartet wird und eine Zugriffskontrolle auf Basis der IP-Adresse des Absenders durchführen kann. Somit wird es also möglich, den Zugang zu den Standard-*servern* nur von einigen wenigen Rechnern aus zu gestatten. Die Standard-*server* müssen aber auf anderen Ports laufen als normal, da die *well-known ports* bereits durch die Proxies belegt sind. Als Alternative bietet es sich an, *netacl* zu verwenden, um je nach Ursprung eines Verbindungswunsches, den Proxy oder den jeweiligen Standard-*server* zu starten. Der Administrator muß dann zuerst eine Verbindung zum Proxy aufbauen, von wo aus er eine Verbindung zu *localhost* initiiert. Trifft eine derartige Verbindung ein, so entscheidet *netacl*, daß der Standard-*server* verwendet werden soll.
- Domain Name Service
  Da es sich um eine reine Proxy-Lösung handelt, sollte es problemlos möglich sein, nach außen hin nichts weiter bekanntzugeben als die Adresse des Firewalls sowie Informationen zum Empfang von *mail*. Der *domain name service* wird aber in der gesamten Dokumentation nicht erwähnt.

3. Administration

- Installation und Konfiguration

  Die Installation gestaltet sich relativ schwierig. Da der *Toolkit* für viele verschiedene Plattformen entwickelt wurde, sollte man über Kenntnisse in der Installation von Software mittels `make` verfügen. Auch sind umfangreiche Kenntnisse in der Administration eines UNIX-Systems von Vorteil, um nicht Fehler zu begehen, die wiederum Angriffsmöglichkeiten darstellen können.

  Die Konfiguration erfolgt mittels eines einzigen Konfigurationsfiles, das Einträge für sämtliche Komponenten enthält. Jede Zeile des Files beginnt mit dem Namen einer Komponente, für die die jeweilige Zeile dann gültig ist. Da keine Abhängigkeiten zwischen den einzelnen Komponenten bestehen und die Syntax des Konfigurationsfiles leicht verständlich ist, ist die Gefahr, Fehler zu begehen, relativ gering.

- Support

  Es handelt sich um ein kostenlos erhältliches Produkt, für das natürlich keinerlei Support besteht.

- Dokumentation

  Der Firewall verfügt über eine umfangreiche Dokumentation. Hier wird die Installation und Konfiguration der wichtigsten Komponenten ausführlich beschrieben. Zusätzlich existiert zu jeder Komponente eine *manpage*, sodaß die wichtigsten Informationen auch *online* zur Verfügung stehen.

  Einige Programme, wie z. B. das *x-gw* sind aber zu knapp beschrieben, sodaß eine Installation und Konfiguration sehr erschwert wird. Auch fehlen Informationen über die Konfiguration des *domain name service* völlig. Wünschenswert wäre es auch noch gewesen, wenn das Testen der Konfiguration auf richtige Funktion in der Dokumentation besser behandelt worden wäre.

  Da es aber ohnehin unumgänglich ist, daß ein Betreiber des *TIS Toolkit* über ausgiebige UNIX-Erfahrung verfügt, ist die Dokumentation durchaus angemessen.

- SNMP-Unterstützung

  Es besteht keinerlei SNMP-Unterstützung.

- Behandlung proprietärer bzw. neuer Dienste

  Dienste, die über das TCP-Relay zu sichern sind, stellen keinerlei Problem dar, während die anderen Dienste gewisse Schwierigkeiten bereiten können. Mit Hilfe des Paket-Filters ist es natürlich möglich, fast alle Dienste zu sichern, man verzichtet aber auf Logging-Möglichkeiten und die Möglichkeit des Zugangs von außen.

  Die beste Möglichkeit wäre natürlich die Entwicklung weiterer Proxies. Für neue Standard-Dienste ist davon auszugehen, daß innerhalb relativ kurzer Zeit Proxies entwickelt werden, die diesen Dienst absichern können, während bei proprietären Diensten die Entwicklung des Proxies selber übernommen werden muß. Hierzu steht eine Bibliothek von C-Funktionen zur Verfügung, mit denen der Authentifikationsserver problemlos angesprochen werden kann. Es dürfte also keine größeren Probleme bereiten, einfache Dienste mit einem Proxy zu versehen.

4. Benutzerfreundlichkeit

- Transparenz

  Der *TIS FWTK* wurde so entwickelt, daß keine Änderungen an den *client*-Programmen erforderlich sind. Dies bedeutet aber, daß die Benutzer in Kauf nehmen müssen, ihre Gewohnheiten zu ändern, da meist ein zweistufiger Verbindungsaufbau erforderlich ist [TIS94a]. Der *HTTP*-Proxy erfordert trotzdem noch leichte Änderungen an der Konfiguration der *clients*.

- Performance

  Über die Performance des *TIS Toolkit* sind keine Informationen vorhanden.

5. Kosten

- Kaufpreis

  Der *TIS Toolkit* ist kostenlos über *anonymous ftp* von `ftp.tis.com` im Verzeichnis `pub/firewalls/toolkit` erhältlich.

- Hardware-Voraussetzungen

  Es ist keine spezielle Hardware erforderlich. Der Firewall kann auf vielen unterschiedlichen Plattformen installiert werden. Natürlich ist ein dedizierter Rechner notwendig, der ausschließlich den Firewall enthält. Sinnvoll ist auch noch die Verwendung eines Routers, um Pakete, die z. B. die *source-routing* Option verwenden abzublocken.

5.4 DEC SEAL

5.4.1 Beschreibung

Der von der Firma DEC entwickelte Firewall SEAL verwendet eine Mischung aus allen drei Firewall-Konzepten, Proxy, TCP-Relay und Paket-Filter [DEC94c]. Im Gegensatz zum IBM Firewall befindet sich der Paket-Filter aber auf einem separaten Host. Abbildung 5.8 stellt dies dar.

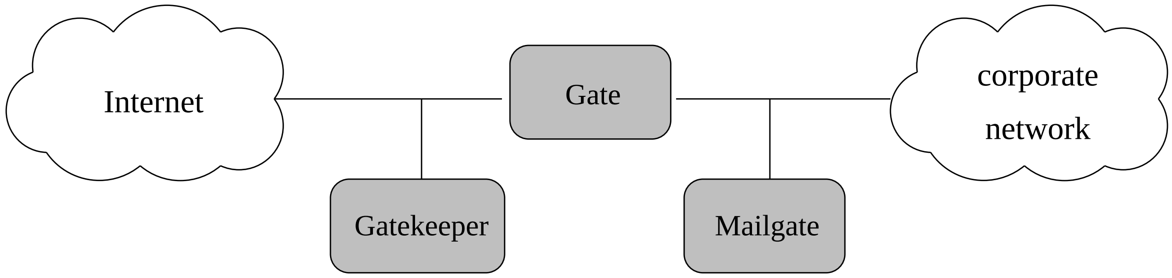

Abbildung 5.8: Der DEC SEAL Firewall

Der Firewall besteht aus drei Komponenten:

- Gatekeeper

  Gatekeeper ist ein Rechner im unsicheren Netz, auf dem sämtliche Proxies sowie das TCP-Relay laufen.

- Gate

  Gate ist der Host, auf dem die Paket-Filter Software installiert ist und der als einziger Zugang zu beiden Netzen hat. Jedes Paket, das also vom sicheren ins unsichere Netz möchte oder umgekehrt, muß *gate* passieren. *Gate* ist normalerweise so konfiguriert, daß nur *gatekeeper* die Möglichkeit hat, mit Rechnern des internen Netzes zu kommunizieren.

- Mailgate

  Bei *mailgate* handelt es sich um einen *mail*-Server, der eingesetzt wird, um die Umsetzung von SMTP-basierender *mail* auf DECNet-basierende *mail* durchzuführen. Außerdem dient er als Speicherungsort für sämtliche Log-Einträge.

Diese drei Komponenten werden normalerweise auf drei dedizierten Hosts eingesetzt, wie es in Abbildung 5.8 dargestellt ist[DEC94a]. Es gibt aber auch Möglichkeiten, sie auf nur zwei bzw. nur auf einem Rechner einzusetzen. Die weitere Untersuchung des Firewalls bezieht sich aber immer auch die Lösung mit drei Rechnern. Abbildung 5.9 stellt die Kombination der verschiedenen Firewall-Konzepte durch den DEC SEAL dar.

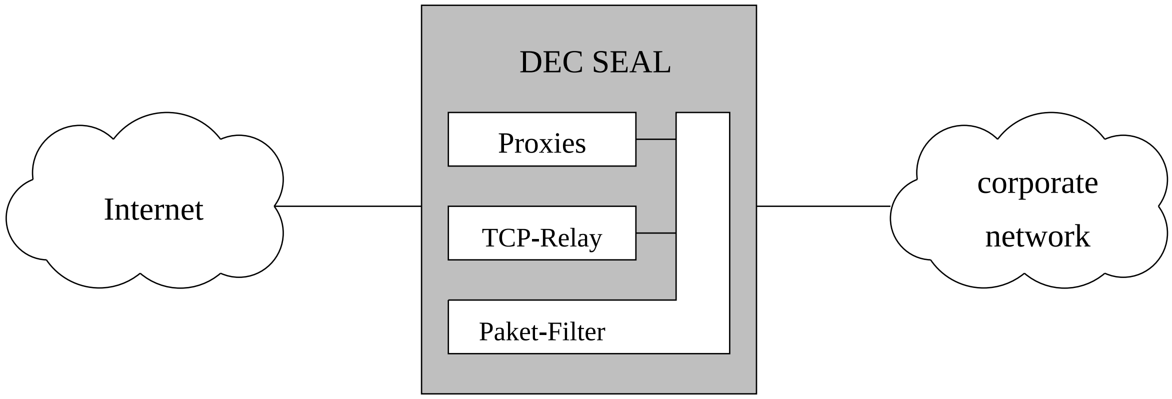

Abbildung 5.9: Kombination der Firewall-Konzepte beim DEC SEAL

Der Paket-Filter

Der Paket-Filter dient normalerweise ausschließlich dazu, den Verkehr nur zwischen *gatekeeper* und dem internen Netz zu gestatten. Es wird das Standard-Tool *screend* eingesetzt, das die Quell- und Zieladressen, die Quell- und Zielports sowie das benutzte Transportprotokoll untersuchen kann. Offensichtlich ermöglicht es nicht die Untersuchung weiterer wichtiger Informationen wie z. B. des ACK-Flags.

Das TCP-Relay

Als TCP-Relay wird das frei erhältliche Programm SOCKS eingesetzt. Die verwendete Version dient ausschließlich zur Sicherung von TCP-basierenden Diensten und ist bereits in der Lage, die Identität eines Benutzers mit Hilfe des *identd*-Daemons zu überprüfen. Am SOCKS-*server* wurden leichte Änderungen vorgenommen, sodaß er nicht in der Lage ist, mit normalen SOCKS-*clients* zu kommunizieren.

Die Proxies

Der DEC SEAL beinhaltet drei Proxies, für *telnet*, für FTP sowie für *news*. Die Proxies werden über den *inetd*-Daemon gestartet, wenn ein entsprechender Verbindungswunsch eintrifft und besitzen folgende Konfigurationsmöglichkeiten:

- Telnet

  Der *telnet*-Proxy läßt sich so konfigurieren, daß nur bestimmte Benutzer von bestimmten Rechnern aus *telnet* durch den Firewall nutzen können. Die Benutzer können dabei über *one-time passwords* authentifiziert werden.

- FTP

  Darüberhinaus bietet der FTP-Proxy noch die Möglichkeit, bestimmte Kommandos des FTP-Protokolls auszuschließen bzw. festzulegen, welche Kommandos bei ihrer Verwendung einen Eintrag im Log-File hervorrufen sollen.

- News

  Der *news*-Proxy dient ausschließlich dazu, eine Verbindung zwischen dem internen *news-server* und einem vorherbestimmten externen *news-server* zu ermöglichen.

5.4.2 Bewertung

Leider stand der DEC Firewall nicht zu Testzwecken zur Verfügung. Die Bewertung dieses Produktes muß also aufgrund der Informationen erfolgen, die von der Firma DEC zur Verfügung gestellt werden.

Bewertung aufgrund der Kriterien aus den Dienstspezifikationen

- Behandlung von TCP-basierenden Client / Server-Diensten

  Die Behandlung von TCP-basierenden *client/server*-Diensten kann wiederum auf drei verschiedene Arten erfolgen:

  - Verwendung von Proxies

    Es steht ein Proxy zur Behandlung von *telnet* zur Verfügung. Dieser wird auf *gatekeeper* installiert und *gate* wird so konfiguriert, daß *telnet* nur zwischen *gatekeeper* und dem privaten Netz gestattet ist. Ein Konfigurationsfile ermöglicht es, Verbindungen nur von bestimmten Rechnern aus zu erlauben sowie eine Authentifikation der Benutzer mittels *one-time passwords* vorzunehmen.

- Verwendung des TCP-Relays

  Die übrigen Dienste werden über das kostenlos verfügbare TCP-Relay SOCKS gesichert. Wie bereits erwähnt, kann SOCKS derzeit nur eingesetzt werden, um Dienste von innen nach außen anzubieten, da die umgekehrte Richtung zusätzliche Authentifikation erfordern würde. Allerdings ist es möglich, daß sich Benutzer über den *telnet*-Proxy in einen Rechner des internen Rechners einloggen und den entsprechenden Dienst von diesem Rechner aus starten. Da die verwendete SOCKS-Version, wie bereits erwähnt, nicht mit normalen *socksified clients* zusammenarbeitet, sind folgende *clients* im Lieferumfang enthalten:

  * Finger
  * FTP
  * Telnet
  * Whois
  * Gopher
  * Xmosaic

  Andere Dienste sind somit nicht mit Hilfe des TCP-Relays zu sichern, es sei denn, es stehen weitere modifizierte *clients* zur Verfügung.

- Verwendung des Paket-Filters

  Natürlich kann auch der Paket-Filter so konfiguriert werden, daß er direkte Verbindungen zwischen dem sicheren und dem unsicheren Netz erlaubt. Da er aber keine Untersuchung des ACK-Flags gestattet, kann nicht zwischen Paketen zum Eröffnen und zum Fortsetzen einer Verbindung unterschieden werden. Deshalb sollte davon abgesehen werden, TCP-basierende *client/server*-Dienste mit dem Paket-Filter zu sichern.

• Behandlung von UDP-basierenden Client / Server-Diensten

Da keine Proxies für UDP-basierende Dienste existieren und auch das TCP-Relay nicht für diese Dienste einsetzbar ist, bleibt als einzige Möglichkeit die Verwendung des Paket-Filters. Da ein Paket-Filter aber ebenfalls keine zufriedenstellende Sicherheit für derartige Dienste bieten kann, muß auf diese Dienste vollständig verzichtet werden.

• Behandlung von TCP-basierenden Peer-to-Peer-Diensten

Auch hier stehen wieder drei Möglichkeiten der Behandlung zur Verfügung:

- Verwendung von Proxies

  Es existiert ein Proxy, der es ermöglicht, *news* sicher zwischen einem externen und einem internen *news-server* auszutauschen. Der Proxy verläßt sich auf die IP-Adresse der *server*, sodaß ein Angreifer, der die Identität eines dieser *server* annehmen kann, in der Lage ist, den Proxy zu täuschen.

- Verwendung des TCP-Relays

  Da derzeit keine modifizierten SOCKS-*clients* für *peer-to-peer*-Dienste existieren, kommt eine Verwendung des TCP-Relays für diese Dienste nicht in Frage.

- Verwendung des Paket-Filters

  Zur Sicherung von *peer-to-peer*-Diensten ist auch dieser Paket-Filter geeignet, allerdings nur unter der Voraussetzung, daß Verbindungen auch von außen initiiert werden dürfen. Da aber das TCP-Relay nicht eingesetzt werden kann, muß dies in Kauf genommen werden, wenn man derartige Dienste erlauben möchte.

- Behandlung UDP-basierender Peer-to-Peer-Dienste

  Wie oben bereits erwähnt, bleibt für UDP-Dienste nur die Verwendung des Paket-Filters. Er kann eingesetzt werden, um diese Dienste ausreichend abzusichern, wenn die Tatsache, daß dann eine Eröffnung einer Verbindung auch von außen möglich ist, keine Rolle spielt.

- Behandlung von Sonderfällen

  - Behandlung RPC-basierender Dienste

    Der DEC-Firewall enthält keine Möglichkeiten, mit denen RPC-basierende Dienste sicher angeboten werden können.

  - Behandlung von FTP

    Zur Behandlung von FTP steht ein Proxy zur Verfügung, der umfangreiche Konfigurationsmöglichkeiten bietet. In einem Konfigurationsfile kann genau festgelegt werden, wie Benutzer verschiedener Rechner behandelt werden sollen. So kann man bestimmen, daß bestimmte Kommandos völlig verboten werden, während andere einen Eintrag im Log-File hervorrufen sollen. Auch kann man angeben, welche Kommandos nur ausgeführt werden dürfen, nachdem sich der entsprechende Benutzer hinreichend authentifiziert hat.

Bewertung aufgrund weiterer Kriterien

1. **Audit**

   - Logging

     Der Firewall bietet ausgiebige Logging-Möglichkeiten. Auf jedem der drei beteiligten Rechner werden sämtliche interessanten Ereignisse geloggt. Dies umfaßt jede Verbindung zum Firewall, wiederholte Fehlversuche beim *login*, sämtliche übertragene *mail* sowie die Benutzung der Proxies und des TCP-Relays. Hierzu wird eine veränderte Version des *syslog*-Programms verwendet, daß zusätzlich noch alle Log-Einträge von *gate* und *gatekeeper* an *mailgate* weiterleitet. Da *mailgate* innerhalb des sicheren Netzes liegt, ist es sehr schwierig für einen Angreifer, nach einem geglückten Angriff auf einen der beiden anderen Rechner, die Log-Einträge zu verändern und somit seine Spuren zu verwischen.

     Auf *mailgate* werden die Einträge nach ihrem Typ sortiert, also es werden getrennte Files z. B. für alle *mail*-Transaktionen oder für alle FTP-Verbindungen angelegt. Außerdem besteht die Möglichkeit, in regelmäßigen Abständen eine Zusammenfassung der Logs zu erzeugen, die dann längerfristige Aussagen über die Nutzung des Firewalls erlaubt.

- Alerting

  Der Administrator kann reguläre Ausdrücke definieren, bei deren Auftreten in einem Log-Eintrag, er den gesamten Eintrag in einer *mail* zugestellt bekommen möchte. Somit kann genau festgelegt werden, welche Ereignisse zu einem Alert führen sollen und welche nicht.

  Darüberhinaus enthält natürlich SOCKS die Möglichkeit, wie bereits mehrfach erwähnt, zu jeder Filterregel ein Kommando zu definieren, das bei deren Anwendung ausgeführt werden soll.

2. sicherheitsrelevante Kriterien

- Authentifikationsmöglichkeiten

  Die Proxies für *telnet* und FTP stellen die folgenden Authentifikationsmöglichkeiten zur Verfügung:

  - Security Dynamics SecureID Card
  - Digital Pathways SNK

  Mit Hilfe dieser Authentifikation ist es möglich, auch Verbindungswünsche von außen zu gestatten, ohne daß ein Angreifer die Möglichkeit hat, mitgehörte Paßwörter zu Angriffen einzusetzen.

- Sicherheit der Firewall-Komponenten

  - Fehlerfreiheit der Firewall-Software

    Aussagen über die Fehlerfreiheit der Firewall-Software gestalten sich relativ schwierig, da die Untersuchung des Firewalls sich, wie bereits erwähnt, nur auf Informationen stützen kann, die von der Firma DEC zur Verfügung gestellt werden. Ein Test des Firewalls konnte leider nicht durchgeführt werden.

    Der Firewall verwendet zu großen Teilen Standard-Software, die gewisse funktionelle Erweiterungen erfahren hat, sodaß sie den Ansprüchen eines Firewalls gerecht werden kann. Dies deutet aber normalerweise auf eine erhöhte Anfälligkeit für Fehler hin, da Standard-Software sehr stark von Angreifern auf mögliche Fehler untersucht wird und häufig auch nicht mit besonderem Augenmerk auf die Sicherheit entwickelt wurde. Die Proxies hingegen sind neuentwickelte Programme, die demnach relativ sicher vor Fehlern sein sollten, die von Angreifern ausgenutzt werden können.

  - Existenz von User-Accounts

    Der DEC SEAL erfordert es nicht, daß für Benutzer, die Firewall-Dienste nutzen wollen, Accounts auf dem Firewall eingerichtet werden. Einzig administrative Accounts sind notwendig, wobei natürlich davon auszugehen ist, daß für diese Accounts hinreichend sichere Paßworte gewählt werden.

  - hierarchische Sicherheitsabstufungen

    Da der Paket-Filter dazu dient, jeden Verkehr zwischen dem internen und dem externen Netz zu verbieten und nur Verkehr vom sicheren Netz zu *gatekeeper* und umgekehrt zu ermöglichen, ist es relativ leicht, diesen so zu konfigurieren, daß gewisse Rechner oder Subnetze des internen Netzes überhaupt keine Kommunikation mit dem unsicheren Netz führen können. Diese Rechner sind allerdings trotzdem von internen Rechnern aus ungehindert angreifbar. Wenn also ein Einbruch auf einen internen Rechner geglückt ist, so genießt auch ein derartig gesicherter Rechner keinen besonderen Schutz mehr.

- periodische Prüfsummenbildung
  Der DEC Firewall verfügt über keinerlei Möglichkeiten, die Konfiguratonsfiles auf nichtautorisierte Veränderungen zu überprüfen.
- administrativer Zugang über das Netz
  Um den Zugang über das Netz zu ermöglichen, ohne gleichzeitig Angreifern die Möglichkeit zu Einbrüchen zu geben, existiert das Programm *netaccess*. Es handelt sich um eine Erweiterung des *inetd*-Daemons und kann, bevor es den entsprechenden *server* startet, überprüfen, ob der Rechner, von dem der Verbindungswunsch stammt, autorisiert ist, Verbindungen zum Firewall aufzubauen. Es können sogar unterschiedliche *server* gestartet werden, je nachdem, von welchem Rechner aus die Verbindung initiiert wurde. Das ermöglicht es, nur bestimmten Rechnern den *telnet*-Zugang zu gestatten und allen anderen eine Nachricht zu präsentieren, daß eine Verbindung zum Firewall untersagt ist.
  Weiterhin erlaubt das *netaccess*-Programm, daß jeder Verbindungswunsch, gleichgültig ob er gestattet oder zurückgewiesen wurde, einen Eintrag im Log-File verursacht. Somit kann niemand unbemerkt Zugang zum Firewall erlangen.

- Domain Name Service
  Es besteht die Möglichkeit der Zweiteilung des *name service*. *Gatekeeper* agiert als *name server* für das externe Netz, während *mailgate* diese Aufgabe für das interne Netz erfüllt. *Mailgate* wird so konfiguriert, daß es Anfragen, die es nicht selber beantworten kann, an *gatekeeper* weiterleitet, der wiederum die Möglichkeit hat, sämtliche *name server* des externen Netzes zu kontaktieren. Umgekehrt darf *gatekeeper* natürlich keine Anfragen an *mailgate* weiterleiten, um die Adressen des internen Netzes geheim zu halten. *Gatekeeper* selber ist natürlich in der Lage, Adressen des internen Netzes von *mailgate* zu erfahren, da er ja ungehindert durch den Paket-Filter mit allen internen Rechnern kommunizieren kann.

3. Administration

- Installation und Konfiguration
  Die Installation des Firewalls gestaltet sich relativ umständlich. Es ist erforderlich jede Firewall Komponente einzeln zu installieren, wobei natürlich bereits Fehler passieren können.
  Die Konfiguration gestaltet sich ähnlich umstänlich. Da keinerlei Hilfsmittel wie z. B. eine graphische Benutzeroberfläche zur Verfügung stehen, muß die Konfiguration über das manuelle Editieren von Konfigurationsfiles erfolgen. Auch hier verbergen sich natürlich Gefahren, da die Unübersichtlichkeit eines derartigen Files relativ schnell zu kleinen Fehlern führen kann, die aber große Wirkungen auslösen können.
- Support
  Um den technischen Support der Firma DEC zu erhalten, ist es erforderlich, einen speziellen Vertrag abzuschließen. Die Kosten hierfür belaufen sich auf 328 DM/Monat.

- Dokumentation

  Der DEC SEAL verfügt über sehr umfangreiche Dokumentation, die die Installation und Konfiguration jeder Komponente detailliert beschreibt. Somit dürfte es für einen erfahrenen UNIX-Administrator kein Problem sein, den Firewall trotz der umständlichen und fehleranfälligen Konfiguration einzurichten.

- SNMP-Unterstützung

  Es ist keinerlei SNMP-Unterstützung enthalten.

- Behandlung proprietärer bzw. neuer Dienste

  Die Behandlung proprietärer bzw. neuer Dienste stellt ein großes Problem dar, da weder das TCP-Relay noch die Proxies hierfür problemlos eingesetzt werden können. Es wäre notwendig, entweder einen *client* so umzuschreiben, daß er in der Lage ist, mit der hier verwendeten Version von SOCKS zu kommunizieren oder einen Proxy zu programmieren, der die gewünschte Funktion erfüllt. Mit Hilfe eines Proxies würden sich dann auch UDP-basierende Dienste sichern lassen. Aus bereits genannten Gründen kommt ein Einsatz des Paket-Filters höchstens für *peer-to-peer*-Dienste in Frage.

4. Benutzerfreundlichkeit

- Transparenz

  Es handelt sich um eine nicht-transparente Lösung [DEC94b]. Zur Verwendung von SOCKS müssen die mitgelieferten *clients* auf sämtliche Rechner des internen Netzes verteilt werden, während die Proxy-Lösung einen Verbindungsaufbau in zwei Schritten erfordert. Erst wird der Kontakt zum Firewall hergestellt, dem man daraufhin den gewünschten Zielrechner mitteilt. Dann erst kann die Verbindung zum eigentlichen Ziel aufgebaut werden.

- Performance

  Da der Firewall leider nicht zu Tests zur Verfügung stand, können über die Performance kaum Aussagen gemacht werden. Zu Erwarten sind Performace-Einbußen bei starker Nutzung der Proxies, da diese jeweils den gesamten Protokollstack abarbeiten müssen.

5. Kosten

- Kaufpreis

  Der DEC SEAL kostet ohne Hardware ca. 105.000 DM.

- Hardware-Voraussetzungen

  In der Maximalkonfiguration erfordert der DEC SEAL drei dedizierte Workstations, die also ausschließlich für den Firewall eingesetzt werden. Dies stellt natürlich einen erheblichen Kostenfaktor dar, weswegen auch Möglichkeiten existieren, den Firewall mit zwei oder einem Rechner zu installieren.

Die verwendeten Rechner müssen als Betriebssystem über eines der folgenden Systeme verfügen:

– OSF/1
– ULTRIX

Aussagen über den erforderlichen Speicherausbau oder erforderliche Festplattenkapazität können leider nicht getroffen werden.

5.5 Zusammenfassender Vergleich

Die ermittelten Vor– und Nachteile der verschiedenen Firewall-Loesungen werden in den Tabellen 5.1 und 5.2 zusammenfassend dargestellt.

	IBM NetSP	SUN FW-I	DEC SEAL	TIS FWTK
TCP-client/server	+ +	+ +	+ +	+ +
UDP-client/server	–	+ +	– –	– –
TCP-peer-to-peer	+ +	+ +	+ +	+ +
UDP-peer-to-peer	o	+ +	o	o
FTP	+ +	+ +	+ +	+ +
SMTP	o	+	o	+ +
X11	– –	– –	– –	– –
RPC	– –	+ +	– –	– –

Tabelle 5.1: Behandlung der unterschiedlichen Klassen von Diensten

	IBM NetSP	SUN FW-I	DEC SEAL	TIS FWTK
Authentifikationsmög–lichkeiten	+ +	– – (+ +)[1]	+ +	+ +
Verbergen interner Ad–ressen	+ +	– – (+ +)[1]	+	+
benutzerbezogene Fil–termöglichkeiten	+ +	– – (+ +)[1]	+ +	+ +
Installation	+ +	+ +	–	–
Konfiguration	–	+ +	o	o
Transparenz	– –	+ +	– –	– –
Preis	+	o	– –	+ +

[1] Version 1.2

Tabelle 5.2: Vergleich weiterer Kriterien

Kapitel 6

Auswahl und Konfiguration einer geeigneten Lösung

6.1 Auswahl einer geeigneten Lösung

In Anhang B wird beschrieben, wie aus der großen Menge an unterschiedlichen Firewalls die geeignete Lösung für ein bestimmtes Unternehmen auszuwählen ist. Dies wird hier am Beispiel der BMW AG durchgeführt.

- Aufstellen einer Sicherheitspolitik

  Welche Dienste durch den Firewall angeboten werden sollen, ist in Tabelle 6.1 dargestellt.

Dienst	innen → außen	außen → innen
Telnet	nur für autorisierte Benutzer	Nur nach Authentifikation über one-time passwords
Rlogin	verboten	verboten
E-Mail	nur für autorisierte Benutzer	uneingeschränkt
FTP	nur für autorisierte Benutzer	Nur nach Authentifikation über one-time passwords
News	nur für autorisierte Benutzer	nur bestimmte Newsgruppen sichtbar
Finger	verboten	verboten
X11	verboten	verboten
WWW	nur für autorisierte Benutzer	Zugriff nur auf einen bestimmten Server

Tabelle 6.1: Sicherheitspolitik der BMW AG

  Um die Kosten für den Internet-Zugang auf die Benutzer umlegen zu können, ist es erforderlich, daß die Benutzer einzeln für die Nutzung bestimmter Dienste autorisiert werden können. Weiterhin ist eine Zweiteilung des *domain name service* erwünscht, um Angreifern keine Ansatzpunkte für eventuelle Angriffe zu geben.

- Auswahl möglicher Lösungen

  Hier werden nun die vier im Rahmen dieser Arbeit vorgestellten Firewall-Lösungen daraufhin untersucht, ob sie zur Sicherung des BMW-Internet-Zugangs in Frage kommen oder nicht.

  - gefordertes Dienstangebot

    Das hier geforderte Dienstangebot kann von allen vier Lösungen bereitgestellt werden.
  - Authentifikationsmöglichkeiten

    Sowohl *telnet* als auch FTP sollen von außen zugänglich sein. Deshalb muß der ausgewählte Firewall über entsprechende Authentifikationsmechanismen verfügen (siehe Anhang A). Die vier vorgestellten Lösungen verfügen, mit Ausnahme des Sun FW-I, über ausreichende Authentifikationsmöglichkeiten. Der Sun Firewall wird aber ab Version 1.2 über vergleichbare Mechanismen verfügen.
  - Domain Name Service

    Eine Zweiteilung des *domain name service*, wie sie von BMW gefordert ist, ist mit dem Sun Firewall erst ab Version 1.2 möglich. Die anderen Lösungen können ebenfalls so konfiguriert werden, daß interne Adressen nicht nach außen hin sichtbar werden.
  - Filtermöglichkeiten

    Die Sicherheitspolitik sieht vor, daß nur die Benutzer Zugang zum Internet erlangen können, die hierzu ausdrücklich autorisiert wurden. Das bedeutet, daß der entsprechende Firewall in der Lage sein muß, benutzerbezogene Transportentscheidungen treffen zu können. Wiederum kann der Sun Firewall erst in der Version 1.2 diese Forderungen erfüllen.

  Nach Untersuchung der Sicherheitspolitik muß eine Entscheidung für ein konkretes Produkt aus folgender Auswahl getroffen werden:

  - IBM NetSP
  - TIS Firewall Toolkit
  - DEC SEAL
  - Sun Firewall Version 1.2

- Auswahl einer geeigneten Lösung

  Wie bereits dargestellt, sind alle vier untersuchten Lösungen in der Lage, die geforderte Sicherheitspolitik zu erfüllen. Da die BMW AG hauptsächlich über IBM-Plattformen verfügt, kann eine Entscheidung zugungsten des IBM NetSP getroffen werden, der sich zwar nicht als die beste Lösung herauskristallisiert hat, der aber ausreicht, um die geforderte Politik zu realisieren. Gewisse Nachteile dieser Lösung können über entsprechende Konfiguration ausgeglichen werden (siehe Kap. 6.2).

6.2 Konfigurationshinweise

In diesem Kapitel sollen Hinweise gegeben werden, wie der IBM NetSP zu konfigurieren ist, sodaß er die geforderte Sicherheitspolitik erfüllen kann.

- Plazierung und Konfiguration der Server
  - Der WWW-Server

    BMW möchte sich im Internet mit Hilfe eines WWW-*servers* präsentieren. Dieser *server* sollte vor dem eigentlichen Firewall plaziert werden, da die große Gefahr besteht, daß er erfolgreich angegriffen werden kann. Befindet er sich dann innerhalb des privaten Netzes, so ist das gesamte Netz gefährdet, während bei Plazierung vor dem Firewall nur der *server* selber gefährdet ist. Bei der Konfiguration dieses *servers* muß unbedingt darauf geachtet werden, daß dieser Rechner keine weiteren Dienste zur Verfügung stellt, da diese von Angreifer ausgenutzt werden könnten.
  - Der Mail- und der News-Server

    Diese beiden *server* enthalten Informationen über Benutzer sowie z. T. Daten, die nicht nach außen gelangen sollten. Deshalb sollten sie innerhalb des Firewalls plaziert werden. Der *mail-server* sollte so konfiguriert werden, daß nur berechtigte Benutzer Nachrichten verschicken können, während der *newsserver* so konfiguriert werden muß, daß bestimmte *news*-Gruppen nicht nach außen bekannt gemacht werden.

- Szenario

  Aufgrund der bisherigen Untersuchungen bietet sich das in Abbildung 6.1 dargestellte Szenario an.

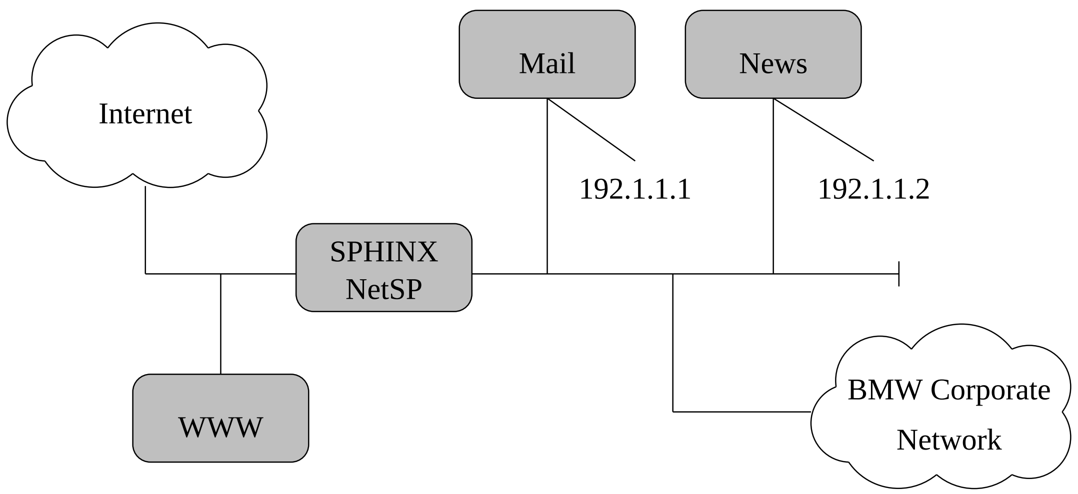

Abbildung 6.1: Firewall-Szenario bei der BMW AG

Der WWW-*server* verfügt über keinen Schutz durch den Firewall. Eventuell wäre es möglich, einen Router vor den *server* zu schalten, um so die möglichen Verbindungen auf Verbindungen zu Port 80 beschränken zu können. Bei entsprechender Konfiguration des *server* erscheint dies aber überflüssig. Der *mailserver* hat im Beispiel die IP-Adresse 192.1.1.1, während der *newsserver* zu Beispielszwecken die Adresse 192.1.1.2 erhält.

- Konfiguration der Proxies
  - Telnet, FTP

    Die beiden im Umfang des Firewalls enthaltenen Proxies werden so konfiguriert, daß kein externer Benutzer Zugang erhält, ohne sich über ein *one-time password* authentifiziert zu haben. Interne Benutzer erhalten Zugang über normale Paßworte.
  - E-Mail

    Der *mail*-Proxy wird vom Firewall automatisch so konfiguriert, daß eintreffende *mail* an den internen *newsserver* weitergeleitet wird, während Nachrichten aus dem internen Netz an den entsprechenden Zielrechner übergeben werden.
  - News

    Es sollte ein *news*-Proxy eingesetzt werden, der eine sofortige Weiterleitung eintreffender Artikel an den internen *news-server* durchführt. Ebenso verhält es sich in der umgekehrten Richtung.
  - WWW

    Der WWW-Proxy vermittelt zwischen einem *client* und dem gewünschten *server*. Der *client* muß hierzu aber in der Lage sein, über einen Proxy zu kommunizieren. Eventuell müssen die Benutzer mit neuen *clients* versorgt werden oder gewisse Änderungen an der Konfiguration der *clients* vorgenommen werden.

- Konfiguration des Paket-Filters

  Abschließend soll noch eine Beispielskonfiguration für den Paket-Filter des IBM NetSP angegeben werden. Die einzelnen Felder der in Tabelle 6.2 dargestellten Regeln haben folgende Bedeutung:

  - Aktion

    Hier läßt sich festlegen, ob die entsprechende Regel dazu dient, bestimmte Pakete zu gestatten oder zu verwerfen.
  - Source Adress

    Hier läßt sich die IP-Adresse des Absenders eingeben.
  - Source Mask

    Die *source mask* gibt an, welche Bits der *source adress* untersucht werdne sollen. Zu beachten ist hierbei, daß beim NetSP im Gegensatz zum CISCO-Router ein logisches UND verwendet wird, um die gültigen Bits zu bestimmen. Das bedeutet, daß nur die Bits untersucht werden, die in der Maske den Wert 0 enthalten.
  - Destination Adress / Destination Mask

    Diese zwei Felder geben die Zieladresse an.

- Protocol

  Das Protokoll, auf das sich diese Regel beziehen soll. Außer den bekannten Protokollen TCP, UDP, und ICMP ist hier der Eintrag 'all' möglich, um sich auf beliebige Protokolle zu beziehen. Enthält dieses Feld den Wert TCP/ACK, so sind nur TCP-Pakete gestattet, die das ACK-FLAG gesetzt haben (vgl. Kap. 2.1.2).

- Source Port / Destination Port

  Diese Felder geben den Port des Absenders bzw. des Empfängers an. Hierzu stehen verschieden Operatoren sowie der Eintrag 'any', der sich auf beliebige Ports bezieht, zur Verfügung.

- Adapter

  Hier kann festgelegt werden, ob sich die betreffende Regel auf das zum Internet gehende (non-secure) Interface beziehen soll, ob es sich um das zum privaten Netz gehende (secure) Interface handeln soll oder ob die Regel für beide Interfaces gelten soll.

- Routing

  Dieses Feld dient dazu, festzulegen, ob es sich um ein an den Firewall selbst adressiertes Paket handelt (local) oder um ein an einen anderen Rechner gerichtetes (route). Verwendet man hier 'local', so kann man auf die Angabe einer Zieladresse verzichten. Ebenso verhält es sich bei vom Firewall stammenden Paketen.

- Direction

  Mit Hilfe dieses Feldes kann man bestimmen, ob die Regel für Pakete gelten soll, die auf dem entsprechenden Interface eintreffen (inbound) oder von dem entsprechenden Interface versendet werden sollen (outbound).

Die Regeln 1 bis 8 beziehen sich ausschließlich auf das Interface zum Internet. Hierbei dienen die Regeln 1 bis 7 für Pakete, die aus dem Internet an den Firewall adressiert wurden, während Regel 8 der umgekehrten Richtung dient. Die Aufgaben der Regeln sind im einzelnen die folgenden: Regel 1 erlaubt es beliebigen Rechnern, auf vom Firewall initiierte Verbindungen zu antworten. Regel 2 bis 4 erlauben *telnet*, *mail* sowie DNS aus dem externen Netz zum Firewall, wo diese Dienste von den entsprechenden Proxies weiterbehandelt werden. Die Regel 5 verbietet daraufhin jeden Zugang zu nicht-privilegierten Ports. Aufgrund der Reihenfolge der Regeln ist natürlich weiterhin der Zugang zu den vorher explizit gestatteten Ports möglich. Um FTP durch den Firewall zu ermöglichen, ist es erforderlich, den gesamten nicht-privilegierten Portbereich von außen zugänglich zu machen. Einige, besonders gefährdete Ports können natürlich ausdrücklich ausgeschlossen werden. Dies ist im Beispiel bei Regel 6 der Fall, die den Zugang zu Port 6000, dem üblichen Port für *X-server*, verwehrt. Hier können noch beliebige weitere Ports gesperrt werden. Die abschließende Regel 7 gestattet alle anderen TCP-Pakete zum Firewall. Alle weiteren Pakete, also insbesondere nicht an den Firewall selbst adressierte Pakete, werden automatisch verworfen. Regel 8 dient ausschließlich dazu, vom Firewall ausgehende Pakete zu gestatten.

Die Regeln 9 bis 16 dienen nun dazu, Verbindungen über das sichere Interface zu ermöglichen. Wiederum dienen die Regeln 9 bis 15 für Verbindungen zum Firewall, während Regel 16 nur dazu dient, daß der Firewall Pakete in das interne Netz verschicken darf.

1	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 TCP/ACK any 0 any 0 non-secure local inbound
2	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 TCP any 0 eq 23 non-secure local inbound
3	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 TCP any 0 eq 25 non-secure local inbound
4	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 UDP any 0 eq 53 non-secure local inbound
5	deny 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 TCP any 0 lt 1024 non-secure local inbound
6	deny 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 TCP any 0 eq 6000 non-secure local inbound
...	...
7	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 TCP any 0 any 0 non-secure local inbound
8	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 any any 0 any 0 non-secure local outbound
9	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 TCP/ACK any 0 any 0 secure local inbound
10	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 TCP any 0 eq 23 secure local inbound
11	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 TCP any 0 eq 20 secure local inbound
12	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 TCP any 0 eq 80 secure local inbound
13	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 UDP any 0 eq 53 secure local inbound
14	permit 192.1.1.2 255.255.255.255 0.0.0.0 0.0.0.0 TCP any 0 eq 119 secure local inbound
15	permit 192.1.1.1 255.255.255.255 0.0.0.0 0.0.0.0 TCP any 0 eq 25 secure local inbound
16	permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 any any 0 any 0 secure local outbound

Tabelle 6.2: mögliche Filterregeln für den IBM NetSP

Die Regel 9 erlaubt es jedem internen Rechner auf vom Firewall ausgehende Verbindungen zu antworten, während die Regeln 10 bis 14 die Dienste *telnet*, FTP, WWW sowie DNS erlauben. Allerdings sind nur Verbindungen zum Firewall möglich, wo die Weiterleitung der Verbindungswünsche mit einem Proxy erfolgt. Die Regel 14 wurde eingefügt, um nur dem *mailserver* die Möglichkeit zu geben *e-mail* zu versenden. Hiermit wird erreicht, daß der *mailserver* zum Versenden von Nachrichten verwendet werden muß, und somit eine Beschränkung auf autorisierte Benutzer ermöglicht. Regel 15 gestattet dem *newsserver* den Aufbau einer Verbindung zum Firewall, der über einen Proxy die Verbindung zu einem externen *newsserver* herstellt. Alle weiteren Pakete werden auch hier vom Firewall automatisch verworfen.

- weitere Konfigurationshinweise

  Um den Problemen, die durch das fehlende Alerting auftreten, entgegenzuwirken, erscheint es sinnvoll, einige Skripts zu entwerfen, die die erstellten Log-Files in regelmäßigen Abständen auf verdächtige Einträge untersuchen.

  Zu beachten ist weiterhin, daß das automatisch erstellte Konfigurationsfile für den *name server* keinen Eintrag für den WWW-*server* erhält. Dieser Eintrag muß von Hand eingefügt werden, da der WWW-*server* im externen Netz bekannt sein muß.

Kapitel 7

Zusammenfassung

Die im Rahmen dieser Arbeit durchgeführten Untersuchungen haben ergeben, das fast alle Internet-Dienste Gefahren beinhalten, die einen ungesicherten Einsatz dieser Dienste verbieten. Da aber ein sehr großer Bedarf an der Nutzung von Internet-Diensten besteht, müssen diese Dienste über Firewalls abgesichert werden.

Hierzu stehen drei verschiedene Konzepte zur Verfügung, der Paket-Filter, das TCP-Relay und der Proxy. Sie alle können die wichtigsten Dienste relativ problemlos absichern. Die wesentlichen Unterschiede sind im folgenden kurz erwähnt:

- Paket-Filter und TCP-Relay
  - Probleme mit UDP-basierenden Diensten
  - Probleme mit Sonderfällen wie FTP und X11
- Proxy
  - alle Dienste können gesichert werden
  - auch Einschränkungen der Dienste sind möglich

Die Untersuchung von vier existierenden Lösungen aufgrund des ebenfalls im Rahmen dieser Arbeit erstellten Kriterienkataloges hat ergeben, daß sich die derzeit angebotenen Lösungen weniger im angebebotenen Dienstangebot unterscheiden sondern eher in anderen Kriterien. Beispiele hierfür sind die zur Verfügung stehenden Logging-Möglichkeiten, die Administration des Firewalls oder auch die anfallenden Kosten.

Für die BMW AG wurde der Firewall NetSP der Firma IBM ausgewählt, der zwar nicht das eindeutig beste Untersuchungsergebnis erlangte, für die speziellen Anforderungen von BMW aber sehr gut geeignet ist. Daran kann man erkennen, daß es nicht möglich ist, einen Firewall als die beste Lösung herauszustellen, sondern daß immer untersucht werden muß, welcher Firewall den ganz speziellen Ansprüchen eines Unternehmens am ehesten gerecht werden kann.

Anhang A

Authentifikationsmechanismen

Da bei Verwendung eines herkömmlichen Paßwortes über eine ungesicherte Leitung die Gefahr besteht, daß ein Angreifer die Leitung abhört und somit in den Besitz des Paßwortes gelangt, existieren Authentifikationsmechanismen, bei denen jedes Paßwort nur zur einmaligen Verwendung bestimmt ist. Gelingt es einem Angreifer dann, dieses Paßwort mitzuhören, so kann er es nicht für einen Einbruch verwenden. Es existieren verschiedene Methoden, um zu gewährleisten, daß ein Paßwort nur einmal verwendet werden kann. Drei der wichtigsten sind im folgenden beschrieben:

- Berechnung eines Paßwortes als Funktion der Zeit

  Bei dieser Methode besitzt der Benutzer ein Gerät, daß aus der momentanen Uhrzeit sowie einem geheimen Schlüssel Paßworte berechnet, die nur in einem bestimmten Zeitintervall zu verwenden sind. Dieses Intervall liegt üblicherweise zwischen 30 Sekunden und 2 Minuten. Aufgrund des kleinen Intervalls kann es zu Problemen kommen, wenn die Uhren des Rechners und des vom Benutzer verwendeten Gerätes nicht genau übereinstimmen. Da die Gefahr besteht, daß ein Angreifer innerhalb des vorgegebenen Zeitintervalls das mitprotokollierte Paßwort wiederverwendet, muß der Rechner eine zweimalige Verwendung eines Paßwortes innerhalb des Gültigkeitszeitraumes verhindern. Um die Verwendung gestohlener Geräte zu verhindern, kann jedes Gerät mit einer Tastatur versehen werden, über die vor Benutzung ein geheimer Schlüssel eingegeben werden muß.

  Security Dynamics SecureID Cards

  Ein Beispiel für diese Methode sind die *SecureID Cards* der Firma *Security Dynamics*. Auf scheckkartengroßen Geräten wird alle 60 Sekunden ein neues Paßwort ausgegeben, daß in Verbindung mit einem geheimen Schlüssel an den Rechner weitergegeben werden muß. Ein Problem hierbei könnte die Klartextübertragung des geheimen Schlüssels darstellen, die es einem Angreifer ermöglicht, diesen mitzuhören. Kann er dann in den Besitz der Karte gelangen, so hat er die Möglichkeit, sich in den entsprechenden Rechner einzuloggen.

- Challenge / Response-Mechanismen

  Um dem Problem der Synchronisation zweier Uhren entgegenzuwirken, gibt es ein anderes Verfahren, bei dem dem Benutzer vom gewünschten Rechner eine Zeichenfolge mitgeteilt wird, aus der dieser das entsprechende Paßwort berechnen muß. Um den Benutzer eindeutig zu identifizieren, wird wiederum ein geheimer Schlüssel in die Berechnung mit einbezogen. Die Berechnung erfolgt meist wieder durch Geräte, die eventuell noch mit einem geheimen Schlüssel versehen sind, um bei Diebstahl nicht mißbraucht werden zu können. Problem dieser Technik ist, daß es für den Benutzer relativ umständlich ist, die angegebene Zeichenfolge abzutippen und daraufhin das berechnete Paßwort einzugeben.

Digital Pathways SecureNet Key

Dieses Produkt verwendet ein *challenge/response*-Schema. Der Rechner gibt eine 8-stellige Zeichenkette an den Benutzer, der aus dieser das Paßwort bestimmen muß. Hierzu muß er sich in seine Karte einloggen, die durch einen PIN-Code geschützt ist. Er gibt die Zeichenfolge in die Karte ein, die daraus das gewünschte Paßwort berechnet.

- vorherberechnete Liste von Paßworten

  Ein dritter Mechanismus ist die Verwendung von vorherberechneten Paßworten [Ches94]. Hierzu verwendet man eine nicht-invertierbare Funktion F. Möchte sich ein Benutzer z. B. hundertmal einloggen, so wählt er einen geheimen Schlüssel `x`, und wendet die Funktion F hundertmal auf `x` an. Dieser Wert $F^{100}(x)$ wird gespeichert. Wenn sich der Benutzer dann einloggen möchte, berechnet er den Wert von $F^{99}(x)$, der vom Rechner dadurch überprüft wird, daß er $F(F^{99}(x))=F^{100}(x)$ überprüft. Daraufhin wird $F^{99}(x)$ der neue gespeicherte Wert. Mit diesem Schema kann man beliebig viele Paßworte erzeugen, wobei aber immer nur ein bereits ungültiges Paßwort gespeichert werden muß.

  Bellcore S/Key

  Das kostenlos erhältliche Produkt *S/Key* verwendet ein derartiges Schema. Um aber dem Benutzer den Aufwand der Berechnung des Paßwortes zu ersparen, erhält er hierbei bereits eine komplette Liste von Paßworten. Diese können dann jeweils einmal verwendet werden, wobei natürlich die große Gefahr besteht, daß die Liste abhanden kommt. Es besteht nämlich kein zusätzlicher Schutz durch irgendwelche geheimen Schlüssel.

Der Einsatz der unterschiedlichen Methoden richtet sich nach unterschiedlichen Kriterien. Natürlich ist *S/Key* die billigste Alternative, während die *SecureID Cards* die benutzerfreundlichste Alternative darstellen. Es wird also eine Mischung der verschiedenen Methoden geben, wobei Mitarbeiter, die häufig auf Reisen sind, eine Karte erhalten, die einen der ersten beiden Mechanismen implementiert, während Benutzer, die nur einmalig oder kurzfristig Zugang von außen benötigen mit einer Methode wie *S/Key* vorlieb nehmen müssen.

Anhang B

Vorgehen bei der Auswahl eines Firewalls

B.1 Auswahl einer geeigneten Lösung

Das folgende Kapitel soll beschreiben, was bei der Auswahl eines Firewalls zu beachten ist und welche Kriterien eines Firewalls besondere Bedeutung genießen sollten:

- Aufstellen einer Sicherheitspolitik

  Der erste Schritt muß immer die Erstellung einer Sicherheitspolitik sein, die genau festlegt, welche Dienste erforderlich sind, welche Dienste verboten sein sollen und welche Dienste mit gewissen Einschränkungen versehen werden müssen. Hier muß auch festgelegt werden, wer die jeweiligen Dienste nutzen darf, d. h. ob die Dienste auch externen Benutzern zur Verfügung stehen sollen.

- Auswahl möglicher Lösungen

  Aufgrund der Sicherheitspolitik können dann verschiedene Lösungen in die engere Wahl genommen werden. Die wichtigsten Kriterien, die hier zu untersuchen sind, sind die folgenden:

  - gefordertes Dienstangebot

    Selbstverständlich kommen nur Firewalls in Frage, die auch in der Lage sind, die geforderten Dienste sicher anzubieten. Hier zeigen sich bereits erhebliche Unterschiede zwischen den einzelnen Lösungen.

  - Authentifikationsmöglichkeiten

    Sollte es erforderlich sein, daß gewisse Dienste auch aus dem unsicheren Netz in Anspruch genommen werden, so muß der Firewall über ausreichende Authentifikationsmöglichkeiten verfügen. Insbesondere dürfen keine Paßworte verwendet werden, die von einem Angreifer durch Abhören des Netzes ermittelt und später wiederverwendet werden können.

- Domain Name Service

  Viele Unternehmen wünschen oder benötigen eine Zweiteilung des *name servers*, sodaß den externen Rechnern keine Informationen über die Adressen der internen Rechner mitgeteilt werden. Dies kann gewünscht sein, um einem Angreifer keine Ansatzpunkte für eventuelle Angriffe zu bieten, kann aber auch erforderlich sein, wenn intern keine offiziellen Adressen verwendet werden können. Der Firewall muß dann in der Lage sein, die entsprechende Adreßumsetzung durchzuführen [RFC1631].

- Filtermöglichkeiten

  Wenn die Sicherheitspolitik bestimmte Kriterien vorschreibt, wann ein Paket bzw. eine Verbindung zugelassen werden soll, dann muß der Firewall diese Kriterien auch untersuchen können. Ein typisches Beispiel ist eine Politk, die vorschreibt, daß nur bestimmte Benutzer bestimmte Dienste nutzen dürfen. Trifft der Firewall seine Entscheidungen einzig aufgrund der IP-Adresse des Absenders, so kommt er für die Implementierung einer solchen Politik nicht in Frage.

Nachdem nun eine gewisse Auswahl von Firewalls vorgenommen wurde, die alle in der Lage sind, die geforderte Sicherheitspolitik zu erfüllen, sollte nun anhand der Untersuchung weiterer Kriterien eine Entscheidung getroffen werden. Die hier besonders zu beachtenden Kriterien sind die folgenden:

- Audit-Möglichkeiten

  Der Firewall sollte über ausgiebige Logging-Möglichkeiten verfügen, sodaß es dem Administrator möglich ist, bereits versuchte Angriffe zu erkennen und Gegenmaßnahmen einzuleiten, bevor der Angreifer größeren Schaden anrichten kann. Hier sind natürlich auch die Alerting-Möglichkeiten des Firewalls von Bedeutung, da ein Administrator unmöglich den Firewall ständig überwachen kann. Insbesondere ist zu untersuchen, wo die Log-Daten gespeichert werden, d. h. ob ein Angreifer, nachdem ihm ein Einbruch in den Firewall gelungen ist, in der Lage ist, die Log-Files zu ändern.

- administrativer Aufwand

  Hier muß betrachtet werden, ob zur Installation und Konfiguration des Firewalls ein erfahrener UNIX- und TCP/IP-Administrator notwendig ist oder ob es mit relativ geringem Aufwand möglich ist, sich das erforderliche Wissen anzueignen. Die Unterschiede, die sich hier bei den einzelnen Lösungen zeigen, sind nämlich erheblich, sodaß gewisse Firewalls auch von weniger erfahrenen Administratoren problemlos betrieben werden können.

  Weiterhin ist es von Bedeutung, ob es notwendig ist, daß jeder Benutzer, der einen Dienst durch den Firewall nutzen möchte, einzeln dazu autorisiert werden muß. Ist dies durch die Sicherheitspolitik nicht gefordert, im Firewall aber vorgesehen, so ergibt sich ein unnötiger Zusatzaufwand, der bei großen Netzen zu beträchtlichen Problemen führen kann.

  Ebenso ist die Transparenz gegenüber den Anwendungen von Bedeutung. Wenn es um die Sicherung eines großen Netzes geht und jede Endstation mit neuen Versionen der *clients* versorgt werden muß, so ist dies ebenfalls ein enormer Aufwand, der sich vermeiden läßt, wenn man eine transparente Lösung wählt oder die Änderungen auf die Gewohnheiten der Benutzer abwälzt.

Aufgrund dieser Kriterien sollten sich einige wenige Lösungen herauskristallisiert haben. Die Entscheidung zwischen diesen Firewalls kann dann aufgrund weiterer Kriterien wie z. B. dem Preis getroffen werden. Hierbei ist aber zu beachten, daß man aufgrund eines günstigeren Preises nie Zugeständnisse an die Sicherheit machen sollte.

B.2 Konfigurationshinweise

Nachdem man sich für einen Firewall entschieden hat, muß er natürlich noch installiert und konfiguriert werden. Obwohl sich beträchtliche Unterschiede zwischen den einzelnen Lösungen ergeben, können doch einige allgemeingültige Überlegungen angestellt werden:

- Plazierung der Server

  Möchte man beispielsweise einen *anonymous FTP*- oder *WWW-server* einrichten, so ergeben sich prinzipiell drei Möglichkeiten, wo der *server* plaziert werden soll:

  - Plazierung vor dem Firewall

    Bei dieser Lösung gehört der entsprechende *server* zum ungesicherten Netz, es handelt sich um einen sogenannten *sacrificial host*. Dieser Rechner ist natürlich allen Angriffsversuchen aus dem ungesicherten Netz ausgesetzt, weshalb der Firewall so konfiguriert sein muß, daß er keinesfalls Verbindungen von diesem Rechner aus akzeptiert. Die erforderlichen Daten erhält der *server* dann in regelmäßigen Abständen über aus dem internen Netz initiierte Verbindungen. Diese Lösung ist für das zu sichernde Netz natürlich die sicherste, setzt aber den *server* beliebigen Angriffsversuchen aus.
  - Plazierung auf dem Firewall

    Diese Lösung ist natürlich nur bei hostbasierenden Lösungen möglich. Aufgrund des hohen Ressourcenverbrauchs dieser *server* und aufgrund potentieller Sicherheitslücken ist diese Lösung allerdings nicht empfehlenswert. Wird sie trotzdem eingesetzt, so ist darauf zu achten, daß der *server* in einer durch `chroot` geschützten Umgebung möglichst ohne besondere Privilegien läuft, um nicht die Sicherheit des Firewalls zu gefährden.
  - Plazierung hinter dem Firewall

    Bei der Plazierung im internen Netz, hat man natürlich die Möglichkeit, den Zugriff nur bestimmten Benutzern zu gestatten bzw. zu verbieten. Auch kann anhand der Log-Einträge erkannt werden, wenn Angriffe gegen den *server* unternommen wurden. Es gibt zwei Möglichkeiten, wie der *server* dann aus dem Internet erreicht werden kann: Entweder wird der Zugang mit Hilfe eines Paket-Filters oder TCP-Relays zu diesem bestimmten Rechner erlaubt. Dann ist der *server* aber wieder gewissen Angriffen ausgesetzt, die nur über das Logging des Firewalls unterbunden werden können. Die andere Möglichkeit ist die Verwendung eines sicheren Proxies auf dem Firewall, der nur die Anfragen, die gestattet werden sollen, an den eigentlichen *server* weiterleitet.

- Auswahl der Server

  Es sollten wenn möglich keine *public-domain server* verwendet werden, da diese meist Fehler enthalten, die von Angreifern ausgenutzt werden können. Außerdem steht für diese Programme meist der *source code* zur Verfügung, sodaß die Angreifer in der Lage sind, immer neue Fehler aufzudecken.

- Konfiguration der Server

  Allgemein kann man sagen, daß jeder *server*, egal ob er ein Teil des Firewalls ist oder zum internen Netz gehört, so sicher wie möglich konfiguriert werden sollte. Dies bedeutet insbesondere, daß vor Start des *servers* mit Hilfe eines `chroot`-Kommandos eine Umgebung geschaffen wird, in der auch bei Ausnutzen eines Fehlers des *servers* keine wichtigen Systemdateien zu erhalten oder zu verändern sind. Außerdem sollten *server* so wenige Privilegien wie möglich erhalten, also wenn möglich nicht mit *root*-Berechtigungen laufen.

- Konfigurationsgrundsätze

  Hier werden jetzt noch einige Grundsätze erläutert, die bei der Konfiguration eines Firewalls beachtet werden sollten:

  - Alles, was nicht ausdrücklich erlaubt ist, ist verboten

    Dies ist der oberste Grundsatz, der beim Aufbau eines Firewalls Beachtung finden sollte. Anstatt nur die Dienste zu verbieten, von denen bekannt ist, daß sie eine Gefahr darstellen können, sollte alles verboten werden, mit Ausnahme der Dienste, von denen man sicher sein kann, daß sie keine Gefahr darstellen.

  - Nicht auf die anderen Administratoren vertrauen

    Man kann sich nicht darauf verlassen, daß die Administratoren der anderen Rechner in der Lage sind, diese sicher zu konfigurieren. Genaugenommen kann man nicht einmal sicher sein, daß man selber den Firewall sicher konfigurieren kann. Es sollten also keine Rechner von außen ereichbar sein, mit Ausnahme derer, die vom Firewall-Administrator selbst verwaltet werden.

  - Nicht auf die Benutzer verlassen

    Noch weniger als bei den Administratoren kann man sich bei den Benutzern darauf verlassen, daß sie mit Blick auf die Sicherheit des Unternehmens handeln. Fehler der Benutzer dürfen also nicht zu einer Gefährdung des Netzes führen können.

  - Zugang von außen nur mit nicht-wiederverwendbaren Paßworten

    Erlaubt man den Zugang von außen über Paßworte, die abgehört und wiederverwendet werden können, so setzt man das Netz einer großen Gefahr aus. Für einen Angreifer ist es normalerweise ein leichtes, diese Paßwörter mitzuprotokollieren und für einen Angriff zu mißbrauchen.

  - Ausgiebiges Testen jeder Konfigurationsänderung

    Oft ergeben sich Abhängigkeiten zwischen verschiedenen Änderungen, die erst beim Testen des Firewalls entdeckt werden können. Es sollte also nach einer Änderung an der Konfiguration unbedingt eine ausgiebige Testphase erfolgen, bevor der Firewall wieder seine Arbeit aufnehmen kann.

Literaturverzeichnis

[Bello89] Steven M. Bellovin. Security Problems in the TCP/IP Protocol Suite. *Computer Communication Review*, Vol. 19, No. 2, pages 32-48, April 1989.

[Bisch94] Hans-Peter Bischof. Der große Lauschangriff. *unix/mail*, Seiten 62–66, Dezember 1994.

[Ches94] William R. Cheswick and Steven M. Bellovin. *Firewalls and Internet Security.* Addison–Wesley Publishing Company, Reading, Massachusetts, first edition, 1994

[Cisco] CISCO Systems Inc. *Router Products Configuration and Reference.* Release 9.1., Menlo Park.

[DEC94a] Digital Equipment Corporation. *Digital‘s Firewall Service, Administrator‘s Guide.* Maynard, Massachusetts, third draft, 1994.

[DEC94b] Digital Equipment Corporation. *Digital‘s Firewall Service, User‘s Guide.* Maynard, Massachusetts, second draft, 1994.

[DEC94c] Digital Equipment Corporation. *Digital‘s Firewall Service, Introductory Guide.* Maynard, Massachusetts, V 1.0, 1994.

[Ellis94] James Ellis, Barbara Fraser and Linda Pesante. Keeping Internet Intruders Away. *Unix Review*, pages 35–44, September 1994.

[Farrow92] Rik Farrow. Improve Your Security. *UNIXWORLD*, pages 59–62, April 1992.

[Farrow94] Rik Farrow. Protecting Your Network. *Open Computing*, pages 83–84, October 1994.

[Garf92] Simson Garfinkel and Gene Spafford. *Practical UNIX Security.* O‘Reilly and Associates, Inc. Sebastopol, 1992.

[High94] Harold Joseph Highland. Random Bits & Bytes. *Computers & Security*, pages 192–201, Vol. 13, No.3, 1994.

[Hoover95] Alton Hoover. Securing the Enterprise. *Internet World*, pages 39–43, February 1995.

[IBM94] IBM Corporation. *Secured Network Gateway Version 1.2, Installation, Configuration and Administration Guide.* Research Triangle Park, North Carolina, 1994.

[Kienle94] Michael Kienle. Standhafte Mauern. *iX*, Seiten 130–139, Juli 1994.

[Koblas92] David Koblas and Michelle R. Koblas. *Socks.* In *UNIX Security III Symposium,* pages 77-83, Baltimore, MD, September 14–17, 1992. USENIX.

[Lynch93] Daniel C. Lynch and Marshall T. Rose. *Internet System Handbook.* Addison–Wesley Publishing Company, Reading, Massachusetts, first edition, 1993.

[Morin94] Richard Morin. Security Resources. *Unix Review*, pages 91–92, August 1994.

[Mui92] Linda Mui. Improving X-Windows Security. *UNIXWORLD*, pages 115–122, December 1992.

[Ranum92] Marcus J. Ranum. *A Network Firewall.* In *Proc. World Conference on System Administration and Security,* Washington, D.C., July 1992. `ftp://DECUAC.DEC.COM/pub/docs/firewalls/firewall.ps`

[Ranum94] Marcus Ranum. Internet firewall protection. *Open Computing*, pages 95–99, September 1994.

[RFC742] K. Harrenstein. *Name/Finger Protocol*, 1977.

[RFC768] J. Postel. *User Datgram Protocol*, 1980.

[RFC791] J. Postel. *Internet Protocol*, 1981.

[RFC792] J. Postel. *Internet Control Message Protocol*, 1981.

[RFC793] J. Postel. *Transmission Control Protocol*, 1981.

[RFC821] J. Postel. *Simple Mail Transfer Protocol*, 1982.

[RFC854] J. Postel and J. Reynolds, *Telnet Protocol Specification*, 1983.

[RFC959] J. Postel and J. Reynolds, *File Transfer Protocol*, 1985.

[RFC977] Brian Kantor and Phil Lapsley. *Network News Transfer Protocol, a Proposed Standard for the Stream-Based Transmission of News*, 1986.

[RFC1034] P. Mockapetris. *Domain names – concepts and facilities*, 1987.

[RFC1035] P. Mockapetris. *Domain names – implementation and specification*, 1987.

[RFC1057] Sun Microsystems Inc. *RPC: Remote Procedure Call Protocol*, 1988.

[RFC1059] D. Mills. *Network Time Protocol (Version 1), Specification and Implementation*, 1988.

[RFC1157] J. Case, M. Fedor, M. Schoffstall and J. Davin. *A Simple Network Management Protocol*, 1990.

[RFC1258] B. Kantor. *BSD Rlogin*, 1991.

[RFC1579] Steven M. Bellovin. *Firewall-friendly FTP*, 1994.

[RFC1631] K. Egevang, P. Francis. *The IP Network Address Translator (NAT)*, 1994.

[Schwart95] Winn Schwartau. Beyond the Firewall. *Internet World*, pages 44–48, February 1995.

[SecDyn94] Security Dynamics Technologies. *Securing the Information Age Minute by Minute.* Swallowfield, Reading, 1994.

[Shel92] Dennis Sheldrick. Security and the X-Windows System. *UNIXWORLD*, pages 103–110, January 1992.

[Smoot92] Carl-Mitchell Smoot and John S. Quarterman. Building Internet Firewalls. *UNIXWORLD*, pages 93–102, February, 1992.

[Stempel94] Steffen Stempel und Hans-Joachim Knobloch. Netwerksicherheit durch Authentifikationsverfahren. *Spektrum der Wissenschaft*, Seiten 70–71, Mai 1994.

[Sun94a] Sun Microsystems, Inc. *FireWall-I, A White Paper.* Mountain View, California, 1994.

[Sun94b] Sun Microsystems, Inc. *Firewall-I, Installation and User's Guide, Release 1.0* Mountain View, California, 1994.

[Sun95a] Sun Microsystems, Inc. *Solstice FireWall-I, Principles of Operations, Version 1.2* Mountain View, California, 1995.
`http://www.sun.com`

[Sun95b] Sun Microsystems, Inc. *Solstice FireWall-I, Configuration in a Nutshell, Version 1.2* Mountain View, California, 1995.
`http://www.sun.com`

[Taylor95] Dave Taylor and Rosalind Resnick. Better Safe. *Internet World*, pages 32-93, February 1995.

[TIS93] Trusted Information Systems, Inc. *Manpages.* 1993.
`ftp://FTP.TIS.COM/pub/firewalls/toolkit/fwtk-doc-only.tar.Z.`

[TIS94a] Trusted Information Systems, Inc. *Firewalls User's Overview.* 1994.
`ftp://FTP.TIS.COM/pub/firewalls/toolkit/fwtk-doc-only.tar.Z.`

[TIS94b] Trusted Information Systems, Inc. *TIS Firewall Toolkit, Configuration and Administration.* 1994.
`ftp://FTP.TIS.COM/pub/firewalls/toolkit/fwtk-doc-only.tar.Z.`

[Wall94] Paul Wallich. Piraten im Datennetz. *Spektrum der Wissenschaft*, Seiten 64–70, Mai 1994.

[Weiss95] Aaron Weiss. Unlawful Entry. *Internet World*, pages 58–62, February 1995.

[X.800] CCITT *Recommendation X.800: Security Architecture for Open Systems Interconnection for CCITT Applications.* Geneva, 1991.