Sascha Badstübner
Manuel Ecker

Aufbau eines Kommunikations-Servers unter LINUX

Proxy, Firewalling, DNS

Projektarbeit aus dem Fachbereich
Kommunikations- und Netzwerktechnik

Höhere Berufsfachschule Medien, Neustadt/Weinstraße
Projektbetreuung durch Alexander Scheib und Walter Schorr

1011110001011010010110101001111010101

Inhaltsverzeichnis

Aufgabenstellung

Die Schülerinnen und Schüler der Höheren Berufsfachschule Medien in Neustadt / Weinstraße erhielten durch den Klassenleiter Hans-Peter Marz den Auftrag zur Durchführung eines Projektes.

Dieses Projekt umfasst eine Gesamtlaufzeit von 3 Monaten. Die Durchführung betreffend gibt es einen Projektverlaufsplan, der von der Klassenkonferenz am 16. Oktober 2001 beschlossen wurde.

Der Projektverlauf (12 Wochen) ist in 6 zeitlich festgelegte Phasen unterteilt:

- **Orientierungsphase**
  22.10.2001 bis 16.11.2001
  Themenabgabe 09.11.2001

- **1. Arbeitsphase** (mit Unterricht)
  19.11.2001 bis 11.01.2002

- **Intensivphase**
  14.01.2002 bis 18.01.2002
  19.01.2002 (freiwillig)
  Intensives Arbeiten ohne Unterbrechnungen durch den „normalen“ Schulalltag soll in dieser unterrichtsfreien Zeit ermöglicht werden (praxsisorientiertes Arbeiten).

- **2. Arbeitsphase** (mit Unterricht)
  21.01.2002 bis 25.01.2002

- **eintägige Intensivphase**
  26.01.2002 (freiwillig)

- **Abschlussphase** (mit Unterricht)
  28.02.2002 bis 01.02.2002
  Diese letzte Projektphase ist frei von Klassenarbeiten.

- **Abgabe der Projektarbeit**
  Freitag, 01.02.2002, 12^{00} Uhr

  3 Exemplare der Dokumentation und gegebenenfalls des erstellten Produktes an den / die Projektbetreuer / in

Die Beurteilung soll die reine Projektarbeit, die Teamfähigkeit, das Endergebnis oder Produkt selbst, sowie die inhaltliche und gestalterische Form der Dokumentation betreffen; weiterhin die Organisation, Durchführung der Originalität der Präsentation (18.02.2002 bis 22.02.2002) berücksichtigen. Die Präsentation wird von einem kurzen Kolloquium abgerundet. Die Note wird einem Fach zugeordnet (themenabhängig, nach Absprache mit Projektbetreuer / in) und wird dort wie 2 Klassenarbeiten gewichtet.

(vgl. Hans-Peter Marz: Rahmenbedingungen und Vorgaben zur Durchführung von Projekten in der Höheren Berufsfachschule Medien und aktueller Projektverlauf BFHTM00 (Schuljahr 2001 / 2002).)

Einleitung und Problemstellung

1. Wozu benötigt man Kommunikations-Server?

In der Informationsgesellschaft, in der wir heute leben ist es für Unternehmen unverzichtbar geworden und aus vielen Privat-Haushalten nicht mehr wegzudenken - wovon ich spreche? - dem Internet natürlich! Keiner möchte mehr auf die Möglickeiten verzichten, die uns das Internet bieten. Deshalb steigt von Tag zu Tag die Zahl der Anschlüsse für Internetzugänge. Neben den Einzelrechnern, die über ein Modem oder eine ISDN-Karte mit der Telefon-Leitung verbunden, diesen Dienst in Anspruch nehmen, sind auch Anschlüsse von Firmen und Schulen zu realisieren, die über private Netzwerke mit bis zu hunderten von Computern verfügen. Ein Lösungsansatz für diesen Sachverhalt ist die Anbindung über einen lokalen Kommunikations-Server, der den Internetzugriff über einen Anschluss zur Verfügung stellt. Dazu werden häufig sog. Proxy-Server auf den Kommunikationsserver installiert. Durch solche Systeme können Unternehmen sehr viel Geld sparen.
Aber nicht nur das Internet ist von großer Bedeutung sondern auch ein firmen- oder schulinternes Intranet wird immer wichtiger. Beim Einsatz eines Intranets ist die Installation eines lokalen Web-Servers erforderlich, von dem die Seiten des Intranets aufgerufen werden. Ein Anwendungsbeispiel dafür ist der Zugriff des kompletten Unternehmens auf eine Datenbank oder für Internetagenturen, die Web-Seiten lokal testen können, bevor diese auf einen externen Web-Server upgeloadet werden. Bei der Realisierung einer solchen Infrastruktur besteht natürlich die Gefahr des Datenmissbrauchs und der -manipulation. Da der Einbruch in solche Firmennetze ein großes Risiko darstellt, müssen Maßnahmen ergriffen werden, um diese Netze vor unerlaubten Zugriffen zu schützen. Gefahrenpotenzial droht aber nicht nur von außen sondern auch durch Mitarbeiter, die Daten bewusst manipulieren oder vielleicht auch versehentlich löschen. Mit einem solchen Kommunikations-Server sind auch andere Dienste anzubieten und zu regeln z.B. lokale Mail-Server-Lösung werden so realisiert.

2. Zielsetzung

Wir wollen uns in unserem Projekt mit der Installation eines Kommunikations-Servers auseinandersetzen. Dieser Server soll auf der Basis des Betriebssystems SuSE LINUX 7.3 und den in der Distribution mitgelieferten Netzwerk-Software-Paketen aufgebaut werden. Dazu verwenden wir einen handelsüblichen aktuellen PC, der mit einem AMD-Athlon-1000 MHz-Prozessor und 256 MB Hauptspeicher ausgestattet ist. Um die Routerfunktion zwischen den verschiedenen logischen Netzen zu ermöglichen muss dieser Rechner mit zwei 100 MBit-Ethernet-Netzwerkkarten ausgestattet sein. Hierbei geht es auch um die Theorie von Sicherheitsaspekten und deren Umsetzung, die wir beleuchten möchten. Bei der Umsetzung von Sicherheitsaspekten können auf Grund der Zeit und der hochkomplexen Zusammenhänge nur Grundlagen behandelt und angewendet werden. Dabei soll eine einfache Paketfilter-Firewall zum Einsatz kommen. Es soll möglich sein, aus einem zweiten lokalen Netzwerk, Angebote eines lokalen Apache-Web-Servers in einem anderen hausinternen Netzwerk aufzurufen. Unerwünschte Zugriffe sollen über einen Proxy und eine Firewall abgeblockt werden. Hauptziel ist das Erweitern unseres Wissens und der Erfahrungen im Bereich der Netzwerktechnik und der Installation von Servern mit dem Betriebssystem LINUX sowie die Weitergabe der gemachten Erfahrungen an interessierte Kollegen unseres Bildungsganges und unsere Lehrer aus dem Bereich der Kommunikationstechnik.

Hardwarevoraussetzung

Linux ist, verglichen mit anderen Betriebssystemen, weitaus weniger anspruchsvoll die Hardware betreffend. Da auf der Linux Firewall außerdem auf eine grafische Benutzeroberfläche verzichtet werden kann, halten sich die Hardwareanforderungen sehr in Grenzen. Allerdings hängt die Dimensionierung von CPU, Hauptspeicher und Festplattenplatz vom Einsatzzweck ab. Eine Firewall, die nur Port-Filterung durchführt, ist wesentlich anspruchsloser als eine, die zusätzlich einen Webproxy beinhaltet. Natürlich braucht eine Firewall mindestens zwei Schnittstellen, über die Netzwerkpakete ausgetauscht werden können. Hierfür ist die Auswahl groß, denn fast alles, das Linux als Netzwerkschnittstelle unterstützt, lässt sich verwenden. Das wäre beispielsweise Ethernet- oder Token-Ring-Netzwerkkarten für das lokale Netzwerk und ISDN-Karten beziehungsweise Modems für WAN-Anbindungen.

Anbindung per Modem/ISDN

Ist der Server per Modem oder ISDN-Karte an das Internet angeschlossen, genügt eine Netzwerkkarte (NIC - Network Interface Card) für die Anbindung an das lokale Netz.

Anbindung per Kabelmodem, Router oder DSL

Bei Einsatz eines Kabelmodems, eines DSL-Routers oder eines vergleichbaren Anschlusses an das Internet, ist der Server mit zwei Netzwerkkarten auszustatten. Eine für den Anschluss an das Modem oder den Router und eine für die Verbindung zum lokalen Netz.
Bei Einsatz eines Kabelmodems, eines DSL Routers oder eines vergleichbaren Anschlusses an das Internet, ist die Firewall mit zwei Netzwerkkarten auszustatten. Eine für den Anschluß an das Modem oder den Router und eine für die Verbindung zum lokalen Netz.

Softwarevoraussetzungen

1. Warum wird Linux als Firewall-Plattform verwendet?

Was ist Linux?

Das Grundgerüst von Linux wurde vor etwa 10 Jahren vom Finnen Linus Torvalds gelegt und entwickelt. Im Gegensatz zu anderen Betriebssystemen liegt hier der "Quellcode offen" und kann von anderen eingesehen und weiterentwickelt werden. Deswegen nennt man es auch "Open Source". Heute arbeiten tausende von freiwilligen Entwicklern an diesem Betriebssystem, die laufend Verbesserungen und Erweiterungen an Linux durchführen. Verschiedene kommerzielle Entwickler wie Red Hat oder SuSE bieten das freie Betriebssystem an und erstellten einfach zu installierende Softwarepakete. Man spricht hier auch von Distributionen, obwohl es sich bei Linux nur um den Kernel handelt, der die grundsätzlichen Ein- und Ausgabefunktionen regelt. Im Laufe der Zeit bürgerte sich der Begriff ein, dass alle Bestandteile einer Distribution als "Linux" bezeichnet wurde.

Vorteile von Linux:

In den letzten Jahren hat sich das Betriebssystem Linux auf dem Markt durchgesetzt und weist im Serverbereich die höchsten Zuwachsraten auf. Immer mehr Unternehmen entscheiden sich heute deswegen für Linux.
Linux überzeugt die Benutzer durch hohe Sicherheit gegenüber Angriffen aus dem Internet, weiterhin durch seine hohe Stabilität, durch Geschwindigkeit und Belastbarkeit. Ein besonderer Vorteil von Linux besteht in seiner Modularität und Anpassungsfähigkeit an die Bedürfnisse des Anwenders.
Die hohe Effizienz bei der Ausnutzung vorhandener Hardwareressourcen macht Linux auch in Hinsicht auf die geringen Wartungskosten als Lösung interessant, darüber hinaus fehlen auch die Lizenzkosten. Linuxsysteme sind flexibel einsetzbar und für jede Aufgabe optimierbar. Ob es sich um einen Datei-, Datenbank-, Backup-, Internet oder Intranet-Server handelt, Linux übernimmt die Aufgabe optimal.

• **Verfügbarkeit**. Für kein anderes Betriebssystem existieren derart viele Kernel-Features und Tools wie für Linux.
• Verfügbarkeitsstudien zeigen die herausragende **Stabilität** von Linux.
• **ausgereifter Netzwerkteil**
• Keine weitere Software neben den Bordmitteln in Bezug auf Firewalling notwendig.
• Fast die gesamt **Hardware** ist unter Linux lauffähig. Lediglich die neueste Hardware macht hier eine Ausnahme. Manchmal sind die ersten Treiber nicht sofort verfügbar.
• Im Gegensatz zu kommerziellen Betriebssystemen entfällt bei Linux der **Support** des Herstellers. Dies wird jedoch durch eine Fülle anderer Supportwege, wie z.B. Newsgroups, Mailing Lists und Online-Dokumentation mehr als ausgeglichen.
• Das gesamte Betriebssystem, inklusive Dienstprogramme und wichtiger Anwendungen, unterliegt der **GPL** und sind somit kostenlos und frei im Quellcode zugänglich.

Der Kernel

Der Linux-Kernel ist der innerste und der an die Hardware angepaßte Teil des Betriebssystems. Er stellt ganz elementare Funktionen wie zum Beispiel Speicherverwaltung, Prozessverwaltung und Steuerung der Hardware zur Verfügung. In ihm integriert sind auch die Firewallfunktionen. Der Kernel ist je nach verwendeter Linux Distribution unterschiedlich konfiguriert.

2. Was ist squid?

Squid ist ein Programm, das unter Linux die Proxy-Funktionen übernimmt. Das Programm dient dazu auf dem lokalen Server einen Cache aufzubauen. Der Cache wiederum bewirkt eine Beschleunigung der meisten Zugriffe auf das Internet und ist eine zusätzliche Entlastung des Netzes.
Ein Cache, auch Webcache genannt, empfängt von den Browsern der Benutzer WWW-Adressen, die diese gerne downloaden möchten. Daraufhin untersucht der Cache seinen Datenbestand, um herauszufinden, ob er das Objekt schon einmal geladen hat. Falls dem so ist, werden die lokal auf dem Rechner liegenden Daten an den Browser ausgeliefert. Ansonsten werden die relevanten Daten von dem ursprünglich adressierten Webserver heruntergeladen und für eine weitere Verwendung lokal gespeichert.

Vorteile von Squid:

- ***Kürzere Wartezeit beim Download***
- ***Geldersparnis*** (Telefon, Volumengebühr)
- ***Bandbreitenschonung***

Die Zeitersparnis kommt dadurch zustande, dass Daten, die sich im Cache befinden, rasch über die schnellen Intranetleitungen transportiert werden, anstelle über langsame Internetleitungen. Weiterhin kann die zu übertragende Datenmenge zu überlasteten Servern minimiert werden. Hier bringt z.B. bei Newsseiten Erleichterung, dass die meist recht vielen grafischen Icons sich lange nicht ändern, direkt aus dem Cache kommen, und so nur Text übrigbleibt. So ist man in der Lage, nicht nur sich selbst und seinem Geldbeutel einen Gefallen zu tun, sondern z.B. auch den anderen Mitgliedern des Hausnetzes, die um jedes Byte, das mehr auf der Telefonleitung frei ist, froh sein werden.

- ***Beseitigung von Netscape-Hängern***

Lästig sind die häufigen Aussetzer von Netscape, bei denen alle Fenster des Browsers betroffen sind. Sobald eine DNS-Abfrage länger dauert, lässt sich der Seiteninhalt nicht mehr scrollen und wird auch nicht mehr aktualisiert. Dies führt dazu, dass man während der manchmal recht langen Zeit gar nichts mit Netscape anfangen kann, da es nicht einmal möglich ist, bereits fertig dargestellte Seiten zu lesen. Zumindest wenn man mehrere Fenster auf einer anderen Seite des Windowmanager liegen hat. Hier kann Squid weiterhelfen, denn Netscape stellt seine Anfrage nun an den lokal gut zu erreichenden Squid-Server und ist von der DNS-Problematik befreit.

- ***Freischaltung bestimmter Rechner zu genau definierten Zeiten***

Squid verfügt über sogenannte "Access Lists", kurz acl. In diese kann man Rechner, Subnetze, Protokolle, Ports, URLs etc. aufnehmen. Sobald eine acl definiert ist, kann diese benutzt werden, um den Zugang zu ermöglichen oder zu sperren. So ist es in einer Firma z.B. sehr einfach zu realisieren, dass die Rechner aller Mitarbeiter in der Mittagspause surfen können. Während der Arbeitszeit hingegen bleibt der Zugang denen vorbehalten, die ihn benötigen. Hierzu wird neben dem Webcache noch ein Firewall gebraucht, der verhindert, dass alle Rechner freien Zugang ins und aus dem Netz heraus besitzen.
Durch die Möglichkeit komplette URLs und reguläre Ausdrücke in acl-Listen aufzunehmen, können komplette WWW-Server gesperrt werden oder solche bei denen bestimmte Schlüsselwörter im URL auftreten.

- ***Nutzung eines Wählzugangs*** mit einer IP zum Surfen mit mehreren Rechnern

Eine weitere schöne Einsatzmöglichkeit liegt darin, ohne sich mit IP-Masquerading befassen zu müssen, weiteren Rechnern den Zugang zum WWW bereitzustellen. Hierzu wird Squid einfach auf dem Rechner installiert der für den Verbindungsaufbau zuständig ist. Die Konfiguration für Squid ändert sich dadurch nicht.

• ***Einfache Überprüfung der heruntergeladenen Inhalte möglich***
Durch die von Squid erstellten Logdateien lassen sich mittels diverser Skripte Übersichten erstellen, die einige interessante Statistiken erfassen. So können alle WWW-Server, auf die zugegriffen wurde, mit Anzahl, Menge und Art der Abfragen eingesehen werden.

• ***Zurückverfolgung*** der Übertragungen zu den einzelnen Rechnern ist möglich.

• ***Anonymisieren der User*** nach innen und außen hin möglich
Sobald Squid WWW-Anfragen der User weiterleitet, wird ein "forwarded for xyz" in den HTTP-Header eingefügt. Diese Informationen kann der WWW-Server speichern und verarbeiten. Um zu verhindern, dass Nutzerprofile angelegt werden, kann die Konfiguration so umgestellt werden, dass diese Informationen nicht verschickt werden bzw. einen fest definierten Wert erhalten. Je nach Konfiguration speichert Squid in seinen eigenen Logdateien aber noch, für welche Rechner er die Daten herausgegeben hat. Dies kann für alle diejenigen wünschenswert sein, die auf eine Kontrolle bzw. einen Beleg angewiesen sind. Dabei sollten aber alle Aspekte des Datenschutzes beachtet werden. Falls man andererseits die Privatssphäre seiner Benutzer schützen will, kann dies durch eine entsprechende Konfiguration auch gewährleistet werden.

Nachteile von Squid:

• ***Eventuell kommt es zur Auslieferung nicht mehr aktueller Daten***
Squid teilt gespeicherte Daten in die Klassen FRESH und STALE ein. Diese Einteilung wird mittels mehrere Regeln und dem Zeitpunkt der Übertragung sowie des Alters des Dokuments bestimmt. Solange ein Objekt FRESH ist, wird es bei einer erneuten Anfrage sofort aus dem Cache ausgeliefert. Ansonsten werden verschiedene Maßnahmen eingeleitet um zu überprüfen, ob das gespeicherte Objekt noch aktuell ist. Trotzdem kann es vorkommen, dass Objekte ausgeliefert werden, die nicht mehr dem letzten Stand entsprechen. Dies tritt vor allem bei Seiten ein, die sich täglich oder stündlich ändern. Wenn man den Verdacht hat, dass die dargestellte Seite nicht aktuell ist sollte man in Netscape, falls ein Reload erfolglos ist, mittels Shift-Reload einen neuen kompletten Download erzwingen.

• ***Eventuelle Probleme mit nur über Links zugänglichen Daten***
Leider kann mit Shift-Reload kein neues Laden eines Links erzwungen werden. Diese Designschwäche des Browsers kann zu ärgerlichen Ergebnissen führen. Beispiel dafür ist eine Liveübertragung eines Fußballspieles per Realaudio verfolgen. Die dazu nötige .ra-Datei ist über einen Link zugänglich und beim Anklicken wird das Realvideo-Plug-in aktiv. Weiterhin befindet sich vor der eigentlichen Übertragung ein Ankündigungstrailer unter diesem Link. Ist man nun etwas früh, hat man folgendes Problem: Der Trailer befindet sich im Cache und ist FRESH. Daher wird dieser, sobald man wieder auf den Link klickt, vom Cache ausgeliefert, obwohl die Übertragung jetzt läuft. Da man kein neues Laden über den Browser erzwingen kann, befindet man sich in der Situation, dass es nur möglich ist die Übertragung zu verfolgen, wenn man für alle Fenster des Browsers den Cache abschaltet. Um dem Ganzen aus dem Weg zu gehen, muss man dem Cache mitteilen, welche Dateitypen er nicht cachen soll. In der Standardkonfiguration befinden sich nur cgi und ? in dieser Liste. Sie sollte also nach Bedarf um all die Dateitypen erweitert werden, die nur per Links zugänglich sind.

(vgl. [7])

3. iptables - Die Entwicklung über ipfwadm und ipchains

Der Anfang:

Vor dem Kernel 2.0 gab es bereits zahlreiche Programme zur Administration von IPFW-Firewalls, die allerdings meist als Patch in den Kernel implantiert werden mußten, um diverse Effekte zu erreichen. Diese Programme waren zwar vom Code her bemerkenswert, doch bei weitem nicht so elegant wie die heute eingesetzte Architektur.

Ipfwadm:

Mit dem Kernel 2.0 tauchte ipfwadm auf. Es ist das erste Firewall Toolkit von JOS VOS (Niederlande) gewesen, welches LINUX mit Firewallfähigkeiten ausgestattet hat und dient zur Administration der IPFW-Firewall. Obwohl ipfwadm ein leistungsfähiges System war und der Transport von IP-Paketen auf Systemebene durch eine einfache Routing-Tabelle geregelt wird, gab es nur die Möglichkeit eingehende, geroutete oder ausgehende Pakete zu filtern. Angesichts der Entwicklung von ipchains wurde die Weiterentwicklung von ipfwadm eingestellt.

Ipchains:

Ab der Kernelversion 2.1.102 erschien ipchains, entwickelt von Paul „Rusty“ Russell. Hier gab es bereits mehrere "chains", die aus jeweils einzelnen Regeln bestanden, die der Reihe nach getestet wurden. Die erste erfolgreiche Regel verließ das Regelwerk und bestimmte den Werdegang des gerade getesteten Paketes. Darüber hinaus konnte man in ipchains erstmals eigene Ketten definieren, die den Charakter von Unterprogrammen hatten.

Iptables:

Paul Russell wollte die IP-Firewalling weiter vereinfachen und verfeinern und entwickelte das Netfilter-Konzept mit dem Tool iptables. Dies kam dann ab der Kernel-Version 2.4 zum Einsatz. Die Iptables-Architektur arbeitet mit einer im Gegensatz zum Vorgänger Ipchains wesentlich einfacheren Systemarchitektur. Der Kernel hält drei Listen von Filterregeln namens INPUT, OUTPUT und FORWARD vor. Man nennt sie auch Ketten, da sie jeweils aus einer Liste sequentiell abzuarbeitender Regeln bestehen.
Jede Regel bestimmt anhand des Paketheaders, was mit anfallenden Paketen zu geschehen hat. Entspricht der Paketheader nicht dem Regelkriterium, wird das Paket an die nächste Regel der Kette weitergereicht. Hat ein Paket die Kette ohne Zutreffen einer Regel bis zum Ende durchlaufen, greift die Policy der Kette. Sie wird ein solches nicht identifizierbares Paket im Regelfall verwerfen.

Zusammenfassung der Verbesserungen bei iptables:
[Auflistung einiger Schlagwörter, auf die hier nicht weiter eingegangen wird]

- konsistentere Namensgebung
- drei Tabellen, die aus mehreren "Chains" bestehen
- stateful inspection jetzt möglich
- Port fowarding mit dem selben Tool
- snat, dnat, masquerading
- DoS Limiting
- Ethernet MAC Adress-Filtering
- variables Logging
- rejects mit einstellbarem Verhalten
- bessere Routing-Kontrolle
- flexiblere Architektur

(vgl. [8])

Sicherheit - Die Firewall

1. Grundlagen der Firewalltechnik

Der Begriff „Firewall“ lässt sich mit „Brandschutzmauer“ übersetzten. Dieses Sinnbild kann in Ansätzen die Funktion einer Firewall beschreiben. Eine Brandschutzmauer soll das Übergreifen eines Brandes von einem Teil eines Gebäudes auf einen anderen angrenzenden Teil verhindern. Die Brandschutzmauer soll also nichts durch lassen. Solche Mauern können aber auch Löcher, z.B. in Form von Brandschutztüren, haben. Diese lassen den Verkehr so lange ungehindert passieren, bis ein Gefahrenfall (Brand) eintritt. In diesem Fall schließen die Türen hermetisch ab und lassen nichts mehr hindurch.
Anders formuliert könnte man sagen, dass es sich bei einer Firewall um ein Modell zur Absicherung eines Computernetzes gegenüber anderen - meist öffentlichen, also Internet - Netzen handelt.
Egal welches Firewall-System eingesetzt wird muss man sich im Vorfeld Gedanken über ein geeignetes Sicherheitskonzept machen. Man muss sich fragen, vor was will oder muss ich mich schützen? Welche Internetdienste stelle ich meinen Mitarbeitern zur Verfügung? Wie kann ich eventuelle Schwachstellen in meinem System vermeiden? Sicherheit um jeden Preis?
Man sollte sich darüber klar sein, dass es einen hundertprozentigen Schutz vor Gefahren aus dem Internet nur geben kann, wenn es keine Verbindung gibt.

Aber was ist denn nun eine Firewall eigentlich genau?

Wir möchten Daten über unser lokales Netz - das Intranet - als auch über das Internet austauschen. Gleichzeitig sollen aber keine ungewünschten Daten (z.B. Viren) auf unsere Rechner gelangen, irgendwelche Programme ausgeführt werden oder andere Daten in diese Netzwerke gelangen. Wenn man eine Firewall unter diesen Aspekten sieht, ist sie eher mit einer Porte mit strenger Zutrittskontrolle zu verstehen. Das Firewall-System trennt einen öffentlichen Bereich von einem privaten Bereich des Netzwerkes ab. Dort, wo Durchgangsverkehr gestattet ist, öffnet man ein Loch in der Brandschutzmauer und postiert einen Pförtner, der überwacht, dass nur der durch die Öffnung kommt, der autorisiert ist. Dieser Passant wird nach definierten Kriterien kontrolliert. Eine Firewall kann aus nur einem einzigen Stück Hardware bestehen oder auch aus einer ganzen Infrastruktur mit mehreren Servern, Routern und anderen Komponenten.

(vgl. [1] Seite 5-6 und [3] Seite 6)

Was kann eine Firewall?

Eine Firewall kontrolliert und registriert den Verkehr zwischen einem lokalen Netz und dem öffentlichen Netz.
- Durchsetzten von Sicherheitsbestimmungen
- Protokollierung

- und was nicht?

- eine Firewall schützt nur Verbindungen, die über sie laufen
- Angriffe von innen
- untreue Mitarbeiter
- Hacker im eigenen Netzwerk

Gefährdungspotential durch externe Angriffe

• ***Fälschung von Information***
Webseiten, Links, DNS-Daten werden manipuliert. Dadurch gelangen Surfer auf falsche Seiten, ungewünschte Skripte werden auf lokalen Rechnern ausgeführt , Überweisungen landen auf falschen Konten, Teile des Internets sind zeitweise nicht erreichbar etc.

• ***Diebstahl von Daten***
Klau von Kreditkartendaten, Zugangsdaten von Internet-Accounts und Online-Banking, die von Crackern missbraucht oder verkauft werden.

• ***DoS/DDoS***
Gezielte Angriffe auf Sicherheitslücken eines Servers, die diesen zum Absturz bringen könnnen. Auch "normale" Anfragen mit künstlich hohen Abfrageraten, deren Antwort viel Rechenzeit in Anspruch nehmen, können solche Abstürze provozieren. Bei einem DDoS-Angriff überflutet eine Vielzahl von Rechnern einen Server mit sinnlosen Netzwerkpaketen d.h. Durchführung eines gemeinsamen Angriffs auf einen Server.

• ***Überflutung mit überflüssigen Informationen***
z.B. Ping-Broadcast-Sturm -> Netzwerküberlastung
oder
Eintrag des Betroffenen in hunderte von Mailinglisten -> Postfach des Betroffenen ist wegen Überfüllung in der Regel nicht mehr nutzbar

• ***Rootkits***
Programme, mit denen ein Hacker die normalen Systemprogramme (netstat, ps, ls, inetd, ...) für eigene Zwecke benutzen kann. Einige Programme erhalten so eine Hintertüre, durch die der Angreifer jeder Zeit wieder in den Rechner zurückkommen kann um diesen für eigene Zwecke zu missbrauchen z.B. für weitere Angriffe. Er verwischt somit seine Spuren.

... durch interne Angriffe

Angriffe aus dem eigenen Netzwerk sind nicht zu unterschätzen. Im Gegensatz zu exteren Angriffen sind sie meist nicht sehr spektakulär sondern verlaufen oft lautlos und werden gar nicht entdeckt. Der Angreifer kennt die lokale Netzwerktopologie und ihm stehen sämtliche Tools zur Verfügung. Die Sicherheitsvorkehrungen sind in der Regel selten so streng wie im Firewallbereich. Meistens werden wichtige Infos in internen Netzen unverschlüsselt übertragen, die somit abgefangen werden können. So können auch Passwörter herausgefunden werden. Oder ein unerlaubter Netzwerkzugang über freie, aber konfigurierte Netzwerkdosen oder unverschlossene Server- und Netzwerkschränke stellen Sicherheitsrisiken dar, völlig abgesicherte Workstations sind also nutzlos. Für einen Angreifer ist es auch einfach Zugriffsrechte zu verändern und Protokolle zu Fälschen. Es ist somit sehr schwer nachzuvollziehen, wer sich unerlaubten Zugriff auf Teile des Netzes verschafft. Der größte Teil von Datenverlusten sind auf "versehentliches Löschen" zurückzuführen - Auch dies sollte in einem Sicherheitskonzept nicht vernachlässigt werden.

(vgl. [1] Seite 12-16)

2. Sicherheitspolitik

Es gilt als sehr schwierig eine Sicherheitspolitik - also Absichtserklärungen was man im Hinblick auf Sicherheit tun möchte - auszuarbeiten. Man muss darauf achten, dass man sich nicht in Details verstrickt, denn dann wäre man schon bei Dingen, die erst in den Ausführungsbestimmungen definiert werden. Die Formulierungen dürfen allerdings auch nicht zu viel Interpretationsspielraum zulassen. Es müssen Sicherheitsziele definiert werden.

Mögliche Sicherheitsziele:

- Alles, was nicht ausdrücklich verboten ist, ist erlaubt! oder
- Alles, was nicht ausdrücklich erlaubt ist, ist verboten!
- Bestmöglicher, zentraler Schutz der Anwender beim Surfen am Arbeitsplatz
- Schutz des eigenen Netzwerkes nach dem Stand der Technik
- Erlaube dem Anwender nur Web und FTP-Verbindungen
- Zugang zum Internet ausschließlich über einen zentralen Punkt
- generell keine Verbindung zwischen Internet und Iokalem Netzwerk
- Verhinderung von Schäden durch autonome Einheiten
- zentraler Schutz vor autonomen Einheiten
- verbiete alle anstößigen, sittenwidrigen oder gar kriminellen Webinhalte
- erlaube nur Webinhalte mit dienstlichen Aspekten
- schütze interne Anwender vor Werbe-E-Mails und E-Mails mit schädlichem Inhalt
- erlaube private E-Mails in geringem Umfang

Sicherheitskonzept

Die festgelegten Sicherheitsziele müssen nun in ein Sicherheitskonzept eingearbeitet werden. Dabei ist darauf zu achten, dass die Ausführungsbestimmungen die Gundelemente enthalten, die man in den vier Bereichen Organisation bzw. übergreifende Aspekte, Infrastruktur, IT-Systeme und Netzwerke zusammenfassen kann. Wenn man sich mit der Erstellung eines Sicherheitskonzeptes und dessen Umsetzung näher befassen möchte sollte man das IT-Grundschutzhandbuch des Bundesamtes für Sicherheit und Informationstechnik, BSI, das auch auf der Website http://www.bsi.de/ verfügbar ist, zu Hilfe nehmen.

(vgl. [1] Seite 19-24)

3. Paketfilter-Firewall

Definition:

„Ein Paketfilter sitzt zwischen zwei logischen oder physikalischen Netzwerken. Er überwacht und kontrolliert den Netzwerkverkehr, indem er jedes Paket analysiert und dann eine Entscheidung über dieses Paket trifft."
(Wolfgang Barth: Das Firewall Buch, SuSE Press, 2001, Seite 45)

Dieser Filter entscheidet darüber, ob das Paket durchgelassen oder verweigert wird. Falls das Paket verweigert wird, kann es verweigert werden oder es wird eine Fehlermeldung zurückgegeben. Dazu müssen sog. Annahme- und Ablehnungskriterien definiert werden. Dies geschieht bei LINUX ab Kernel Version 2.4 mit dem Programm iptables.

Paketfilterung mit iptables

Sowohl für die Flussrichtung von ankommenden Paketen (input) als auch abgehender Pakete (output) gibt es jeweils eigene Filter. Dieser Mechanismus besteht aus einer Vielzahl aufeinanderfolgender Regeln, die nacheinander abgearbeitet werden müssen. Alle ankommenden Pakete werden an die INPUT-Chain gesendet. Diese prüft, ob das Paket angenommen oder abgelehnt wird. Wird es durchgelassen gelangt es zu einem lokalen Prozess. Soll ein Paket von einem lokalen Rechner versendet werden, wird dieses in der OUTPUT-Chain geprüft und bei bestandener Prüfung direkt an das Output-Interface weitergeleitet. Es gibt noch eine weitere Regelkette für Pakete, die nicht für den lokalen Rechner bestimmt sind, die FORWARD-Chain. Damit ein Paket an diese Kette weitergeleitet wird, muss der Kernel „Forwarding" eingeschaltet haben und wissen wie das Paket geroutet (zum Paket passender Eintrag in der Routing-Tabelle) werden soll. Treffen diese Bedingungen zu, wird das Paket an die FORWARD-Chain weitergeleitet und verlässt den Rechner wieder. *[Abbildung 3.1]*
Diese Regeln sind in Paketfiltertabellen nacheinander aufgelistet. Diese Listen werden auch als "Firewall-Chain" oder auch nur "Chain" bezeichnet. Die Kette von Regeln wird solange abgearbeitet, bis eine Regel zutrifft oder das Ende der Tabelle erreicht ist. Dabei gibt es grundsätzlich zwei verschiedene Strategien, die man bei LINUX-Firewalls verfolgen kann - die sog. Drop-Policy und Accept-Policy.

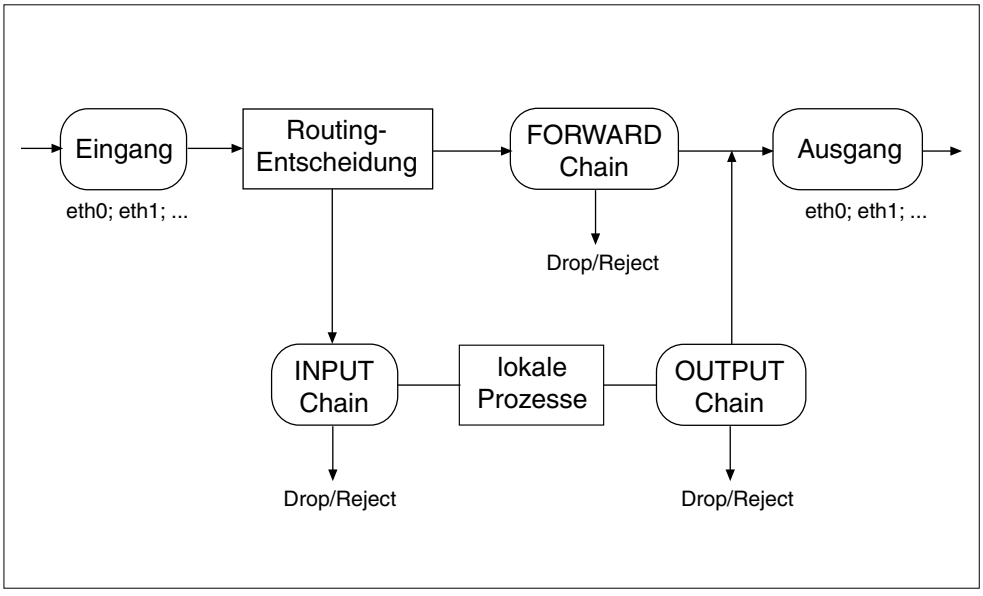

Abbildung 3.1: Architektur der IP-Tables-Regel-Ketten

• **DROP-Policy**
Grundsätzlich wird alles abgewiesen; nur explizit durch Regeln zugelassene Pakete werden akzeptiert. *[Abbildung 3.2]*
• **ACCEPT-Policy**
Grundsätzlich wird alles akzeptiert; nur explizit durch Regeln verbotene Pakete werden abgewiesen. *[Abbildung 3.3]*

Die DROP-Strategie bietet mehr Sicherheit und Kontrolle, jedoch auch eine Menge Arbeit bei der Einrichtung und Administration, da alle Pakete, die durchgelassen werden sollen bzw. müssen, definiert werden müssen.
Bei der Einrichtung ist die ACCEPT-Methode wesentlich einfacher, bietet allerdings auch mehr Angriffsfläche. Um eine ausreichend sichere Firewall zu bauen, müssen alle denkbaren Angriffstypen erkannt und abgewiesen werden. Diese Art erfordert also mehr Arbeit und die Gefahr, dass Sicherheitslücken entstehen bzw. nicht erkannt werden, ist um ein vielfaches größer als mit der DROP-Policy.
Wenn ein Paket abgewiesen wird, kann es entweder einfach verworfen werden (DROP) oder es kann eine ICMP-Fehlermeldung mit Begründung zurückgegeben werden. In diesem Fall spricht man dann von einem REJECT. Wenn ein System eine möglichst hohe Systemsicherheit erreichen soll ist es vorzuziehen, die Pakete mit DROP zu verwerfen. Dafür kann man drei Gründe nennen:

1. unnötige Vergrößerung des Netzwerkverkehrs
2. Gefahr Opfer einer Denial-of-Service-Attack zu werden
3. Fehlermeldungen können dem Angreifer Informationen geben, die er besser nicht haben sollte, z.B. die Existens einer Firewall oder ähnliches.

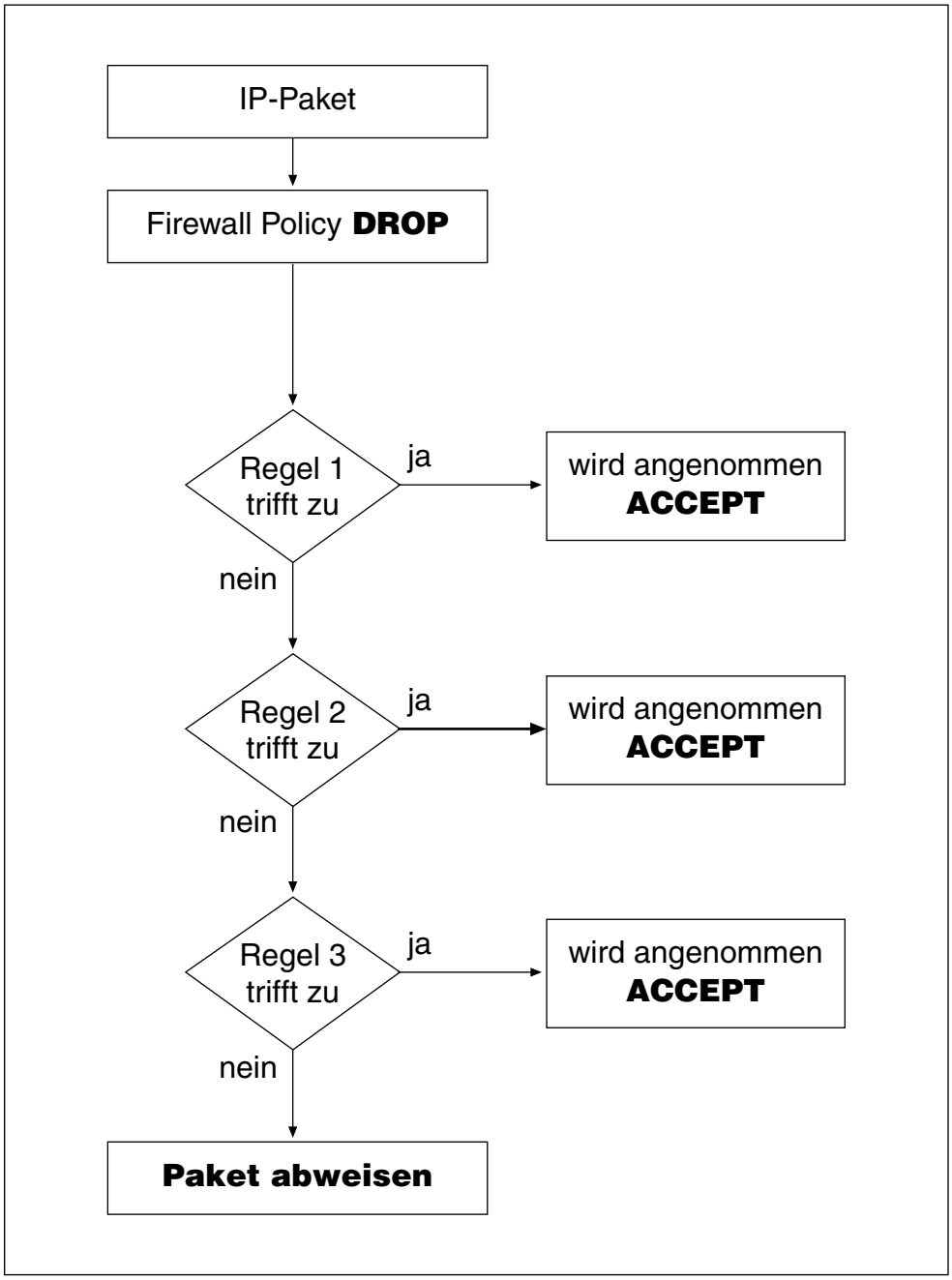

Abbildung 2.2: Die DROP-Policy

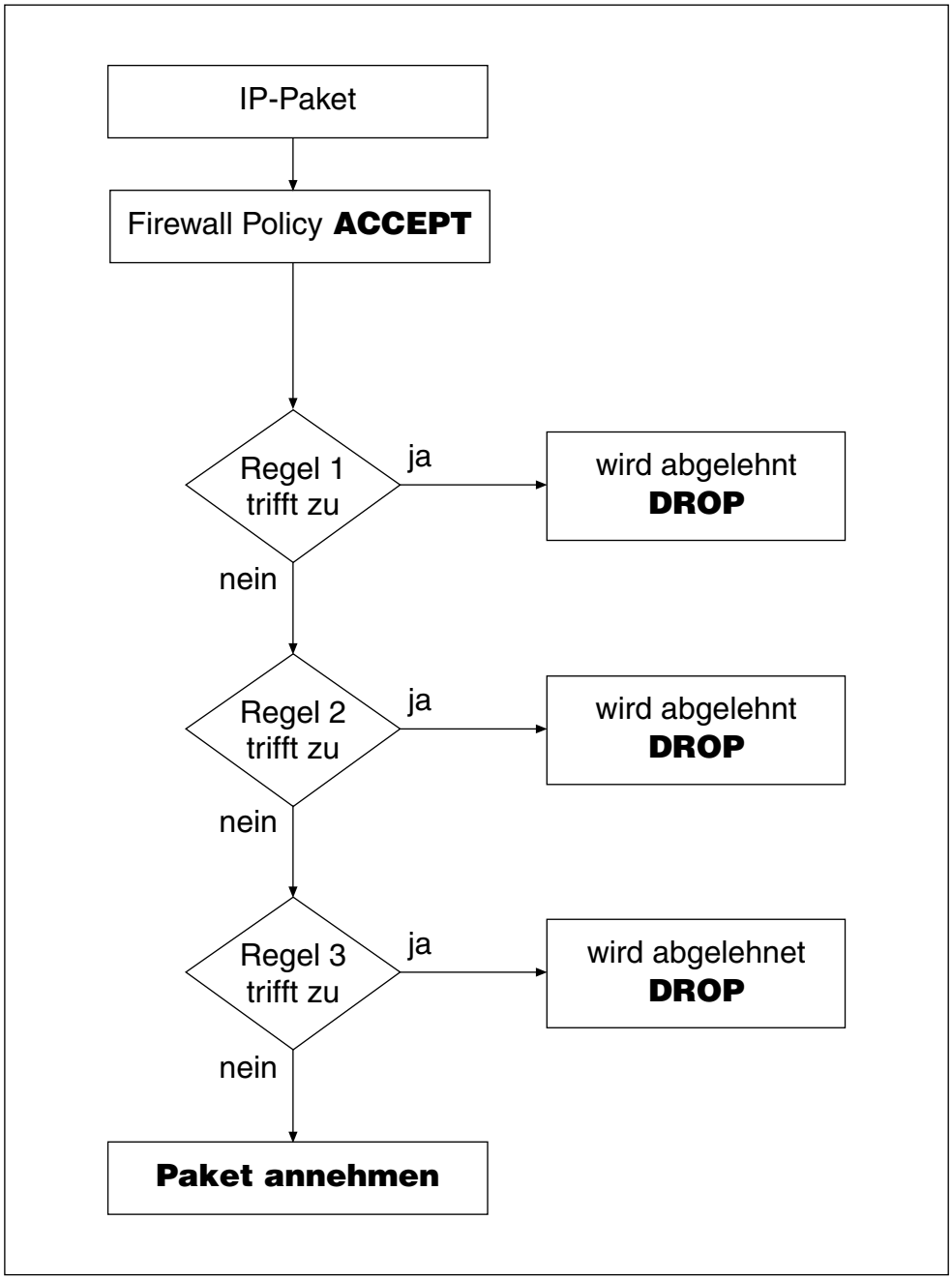

Abbildung 2.3: Die ACCEPT-Policy

Filtern eingehender Pakete

Das Filtern eingehender Pakete ist wichtig, um zu vermeiden, dass Angriffe von außen, also aus dem unsicheren Netz, auf unser zu schützendes, lokales Netzwerk stattfinden. Es gibt verschiedene Kriterien, nach denen Pakete gefiltert werden können.
Filtern nach IP-Adressen
Auf der Vermittlungs- und Transportschicht des OSI-Modells ist die IP-Adresse die einzige Möglichkeit, herauszufinden, wer das Paket geschickt hat. Grundstätzlich ist es aber fragwürdig, ob diese Absenderadresse echt ist, da es kein Problem ist, eine Absender-IP im IP-Header des Paketes zu fälschen. Sowohl Absender-IP als auch Empfänger-IP können als Firewall-Kriterien betrachtet werden.
Da es natürlich nicht einfach ist gefälschte IPs zu erkennen gibt es sechs Hauptgruppen von IPs, die generell abzulehnen sind:

1. Die **eigene IP-Adresse**, da es sich dabei um ein Paket vom eigenen Rechner handeln würde, das niemals regulär eingehen kann.

2. **Private-IP-Adressen**, da diese im Internet nicht weitergeleitet werden - sie existieren dort nicht.
   - Klasse A 10.0.0.0 bis 10.255.255.255
   - Klasse B 172.16.0.0 bis 172.31.255.255
   - Klasse C 192.168.0.0 bis 192.168.255.255

3. **Multicast-Adressen der Klasse D**, da es sich hierbei um reservierte Adressen handelt für sog. Multicast-Broadcasts, die für Audio- oder Video-Übertragungen an mehrere Empfänger verwendet werden. Diese dürfen niemals als Absender vorkommen. (224.0.0.0 bis 239.255.255.255)

4. **Klasse E-Adressen**, da diese für experimentelle Erweiterungen vorgesehen sind und nicht öffentlich vergeben sind. Diese Adressen werden höchstens von US-Militärs und -Geheimdiensten verwendet.
   (240.0.0.0 bis 247.255.255.255)

5. **Loopback-Adresse**, da es sich dabei um die reservierte Adresse 127.0.0.1 handelt, die immer den lokalen Rechner anspricht. Bei einem Paket aus dem Internet mit dieser IP kann es sich also nur um eine gefälschte Adresse handeln.

6. **Broadcast-Adressen** sind als Absender-Adresse illegal. Sie können nur als Empfänger-Adresse vorkommen um mehrere Rechner in einem Netz gleichzeitig anzusprechen.

Außerdem ist es möglich, bestimmte existierende Adressen, von denen man weiß, dass von ihnen oft Angriffe ausgehen, zu sperren. Umgekehrt ist es möglich einem konkretem bekannten Netzwerk-Zugriff auf einen Server zu gestatten und sonst niemand darauf zugreifen kann. Im Gegensatz dazu ist auch das Filtern nach Empfänger-Adressen möglich. So ist es möglich, dass eine bestimmte IP in einem lokalen Netz Pakete erhalten darf aber alle anderen gesperrt sind. Beispielsweise ein Web-Server, der für die Teilnehmer des lokalen Netzes erreichbar sein soll.

Filtern nach Portnummern

Bei den Portnummern handelt es sich um Informationen aus dem UDP- oder TCP-Header (Transportschicht). Diese geben an, für welches Programm das Paket bestimmt ist bzw. von welcher Anwendung es geschickt wurde.
Möchte ein Client von außen einen lokalen Service nutzen, schickt dieser seine Anfrage an eine der Portnummern zwischen 1024 und 65535. Die Empfänger-Portnummer eines eingehenden Paketes ist für eine Firewall wichtig, da dies die Information enthält auf welchen Dienst der User zugreifen möchte. So können Dienste, die nicht erreichbar sein sollen, gesperrt werden.

Filtern nach TCP-Status-Flags

Mit dieser Information der Transportschicht kann entschieden werden, ob ein Paket eine Anfrage von außen oder eine Antwort auf eine Anfrage von innen enthält. Hierbei handelt es sich um den Mechanismus des Handshakes beim Verbindungsaufbau.

Filtern von ausgehenden Paketen

Das Filtern von ausgehenden Paketen ist nicht so kritisch, wenn man davon ausgeht, dass man den Benutzern des lokalen Netzwerkes vertraut. Aber auch hier bestehen bei Bedarf die gleichen Möglichkeiten, nach denen man diese Pakete filtern kann.

(vgl. [3] Seite 24-26)

4. Modelle routender Firewalls

Eine Firewall, die zu zwei verschiedenen Netzen - dem Internet auf der einen Seite, das lokale Netzwerk auf der anderen Seite - gehört, d.h. auch als Router fungiert nennt man dual homed hosts. Mit dieser Technik kann man je nach Anforderung des zu schützenden, lokalen Netzes verschiedene Sicherheitsstufen realisieren. Die beiden wichtigsten Modelle dazu sind die Bastion-Host-Firewall und die Demilitarisierte Zone (DMZ).

Bastion-Host-Firewall

Die Bastion-Host-Firewall ist die einfachste zugleich aber auch die unsicherste Form einer routenden Firewall. Diese routet die Pakete von einem in das andere Netz. Die IP-Pakete müssen die drei Ketten der Paketfilterung (INPUT, OUTPUT, FORWARD) passieren. Der Begriff "Bastion" meint eine Verteidigungseinrichtung, wenn diese überwunden ist, ist das System schutzlos gegenüber Angriffen. Für kleinere Netzwerke ist diese Methode allerdings in der Regel völlig ausreichend, da keine bzw. nur wenige Dienste selbst im Internet angeboten werden.

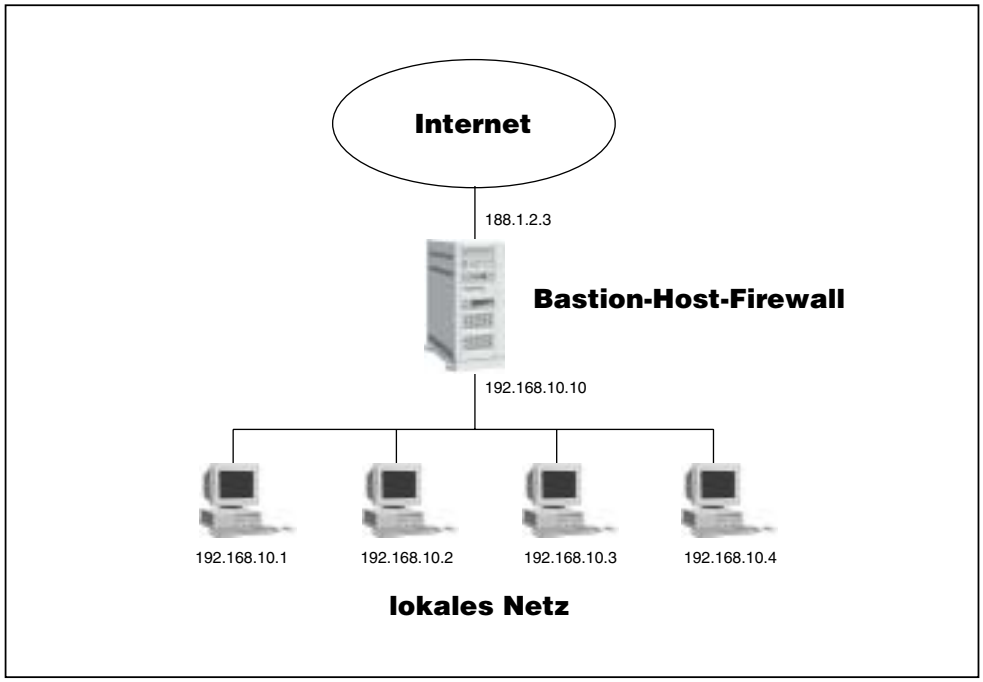

Abbildung 4.1: Bastion-Host-Firewall

Demilitarisierte Zone (DMZ)

Dieses wesentlich sichere System ist für größere Firmen besser geeignet. Dieses Sicherheitssystem verwendet zwei Firewall-Rechner. Das lokale Netz und das Internet werden in zwei Netze aufgetrennt, die am Übergang oder Gateway je einen Firewall-Rechner haben. Das damit entstandene zusätzliche Netz nennt man Demilitarisierte Zone. Diese DMZ kann nun Web-, Mail- oder FTF-Server - also öffentliche Server - enthalten, die von außen zugänglich sind. Die Einheit, die das Internet mit der DMZ verbindet ist wieder eine Bastion-Host-Firewall, die das lokale Netz mit der DMZ verbinden nennt man Choke-Firewall. Der große Vorteil dieser Methode ist, dass eine geknackte Komponente des Systems noch nicht zur Unsicherheit des zu schützenden Netzwerkes führt.

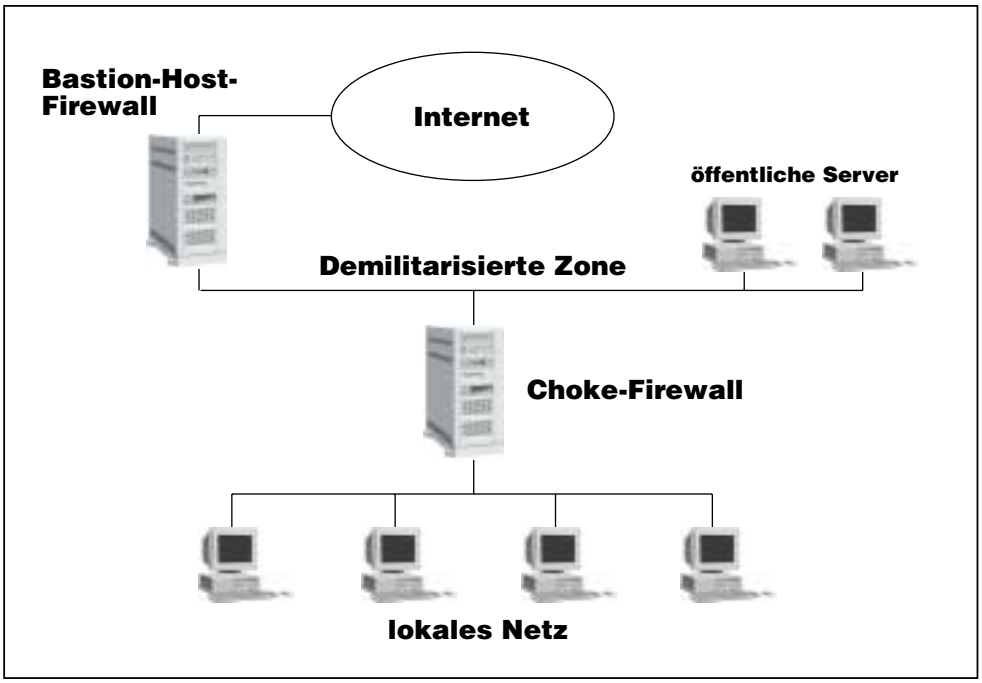

Abbildung 4.2: Demilitarisierte Zone (DMZ)

(vgl. [3] Seite 50-52)

Anbindung an das Internet - Router oder Proxy

Um den Computern eines lokalen Netzwerkes den Zugriff zum Internet zu ermöglichen, ohne dass dabei jeder Rechner ein eigenes Modem haben muss, gibt es verschiedene Methoden. Dabei handelt es sich um die sog. Router- bzw. Proxy-Technik.

1. Proxy-Technik

Proxies werden schon relativ lange verwendet, da man festgestellt hat, dass bestimmte Seiten sehr häufig aufgerufen werden. Der Proxy ist ein lokaler Web-Zwischenspeicher, der Web-Seiten lokal verfügbar macht und somit die Zeit und auch die Kosten für Internetverbindungen reduziert werden konnten. Der Client hat also keine direkte Internetverbindung. Er greift auf den Proxy zu, der die gewünschte Seite - falls vorhanden - aus den lokalen Cash zur Verfügung stellt. Hat dieser die Seite nicht gespeichert baut er eine Verbindung ins Internet auf und lädt sie aus dem Internet. Da ein Proxy auf der Anwendungsschicht arbeitet stehen nur wenige Protokolle zur Verfügung. Es ist also kein bedingungsloser Zugriff auf das Internet möglich. Dies kann von Vorteil sein, wenn man den Usern außer den Diensten die der Proxy ermöglicht, keine weiteren zur Verfügung stellen möchte. Für den Web-Server sieht die Verbindung so aus, als ob der Proxy - oder auf deutsch „Stellvertreter“ - die Verbindung aufgebaut hätte. Es gibt unterschiedliche Proxies für spezielle Internetdienste (z.B. Mail- oder FTP-Proxy). Diese können die benötigten Befehle und Kommandos dieser Dienste interpretieren d.h. auch einzelne Befehle erlauben bzw. ablehnen.

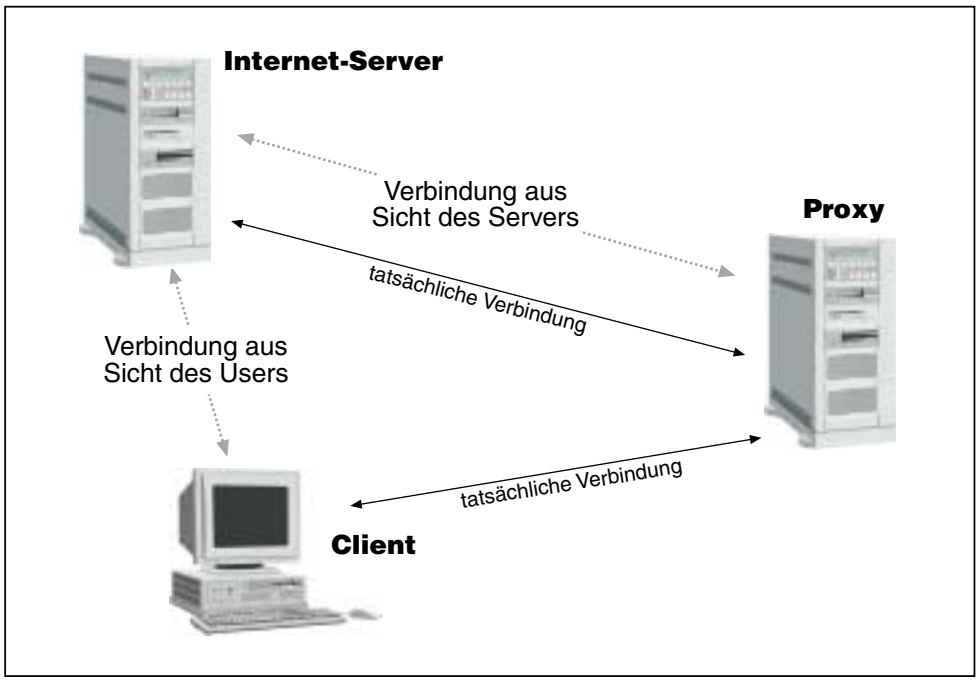

Abbildung 1.1: Dreiecksbeziehung der Wirkungsweise eines Proxy

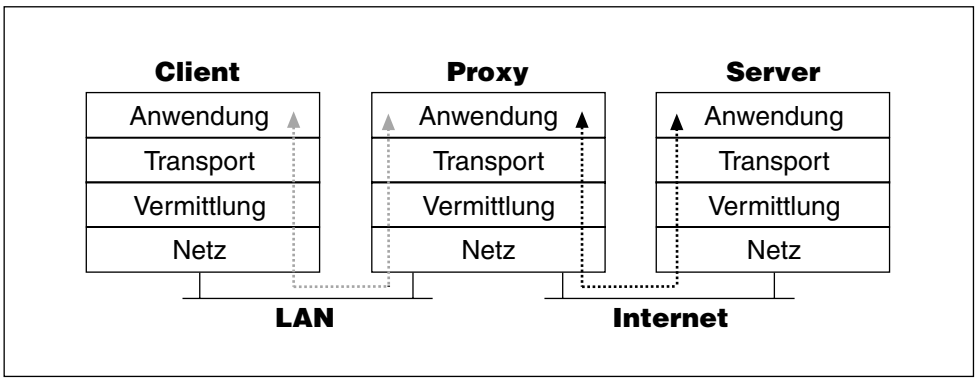

Abbildung 1.2: Die Proxy-Technik (Application-Layer-Gateway)

2. Router-Technik

Ein Router arbeitet auf der Verbindungsschicht des OSI-Modells. Er leitet also Pakete, die er erhält an das entsprechende Netzwerk-Interface weiter. Dabei ist es ihm egal welches Protokoll verwendet wird und auch was die IP-Pakete enthalten. Der große Unterschied zum Proxy ist, dass der Router tatsächlich IP-Pakete weiterleitet.

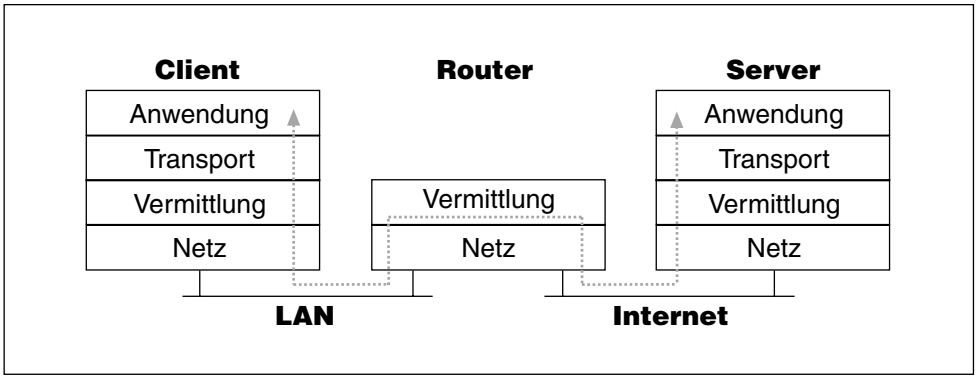

Abbildung 2.1: Die Router-Technik (Screening-Router)

(vgl. [3] Seite 46-47)

Konfiguration „unseres“ Kommunikations-Servers

1. Ergebnisse des Gesamtprojektes

In einem Großprojekt, das aus mehreren Einzel-Projekten bestand, wurde ein Netzwerk zum Schulungs- und Ausbildungszweck in die Infrastruktur des Netzwerkes der Berufsbildenden Schule Neustadt/Weinstraße implementiert. Die Projektteilnehmer der Höheren Berufsfachschule Medien simulierten dabei wichtige Komponenten eines Intranets, das in Teilen oder auch komplett in kleinen Firmen oder in Schulen eingesetzt werden kann. Dieses Netzwerk soll über einen lokalen Apache-Web-Server verfügen, auf den via FTP ein direktes Uploaden von neu erstellten Webseiten möglich ist. Das Aufrufen dieser Seiten mit einem Webbrowser soll von den Clients im Intranet möglich sein. Zusätzlich wurde ein Samba-File-Server installiert, auf den sowohl Windows98- als auch Windows2000-Clients Zugriff haben sollen. Er soll auch als Primary Domain Contoller (PDC) für Windows2000-Clients arbeiten. Zu dem bestehenden Netz mit Clients und dem Web- und Samba-Server (172.16.0.0/16) soll ein zweites Netz (192.168.210.0/24) eingerichtet werden, das über einen Kommunikations-Server mit diesem verbunden ist. Dieser Server fungiert als Router zwischen den beiden lokalen Netzen. Da im 172.16.0.0-Netz bereits ein Proxy-Server (172.16.0.2) läuft, der den Internetzugang dieses Netzes regelt, ermöglicht der neue K-Server den Zugang zum Internet für das 192.168.210.0-Netz. Diese IP-Pakete durchlaufen dabei eine Firewall in Form von Paketfiltern mit iptables, die im Kernel des Servers einkompiliert sind. Der Zugriff auf den lokalen Intranet-Server erfolgt über einen, auf dem K-Server installierten Proxy-Squid-Server. Mit Hilfe des Squids wird die Zugriffsberechtigung der Hosts gesteuert. Die Clients können mit der Proxyeinstellung (172.16.210.101:3128) auf den Apache zugreifen. Damit dies nicht nur mit der IP der angeforderten Intranet-Domain möglich ist, musste ein lokaler Domain-Name-Server (DNS) installiert werden, wodurch die Namensauflösung erreicht wurde.
Bei diesem Server-Projekt wurden PCs mit dem Betriebssytem SuSE LINUX 7.3 verwendet, sowie die mitgelieferten Servermodule und die neueste Version des Programmes „Webmin“, einem beliebten grafischen Administrations-Tool. Die komlette Server-Struktur dieses „Schulungs-Netzes“, das voll funktionsfähig ist, und auch für den täglichen Schulalltag genutzt werden kann, insbesondere vom Fachbereich Medien, ermöglicht es somit im Rahmen von Web-Programmierung mit HTML und Script-Sprachen (z.B. PHP) den Webserver zum Testen der gestalteten Webseiten zu benutzen.
[Auf der folgenden Seite ist ein Netzwerkplan der installierten Struktur abgebildet.]

Vielen Dank an dieser Stelle an unsere Projektbetreuer, besonders an unseren Netzwerkspezialisten Alexander Scheib, der uns unzählige Stunden bei der Lösung von Problemen behilflich war und immer an unseren Erfolg geglaubt hat.

Manuel Ecker

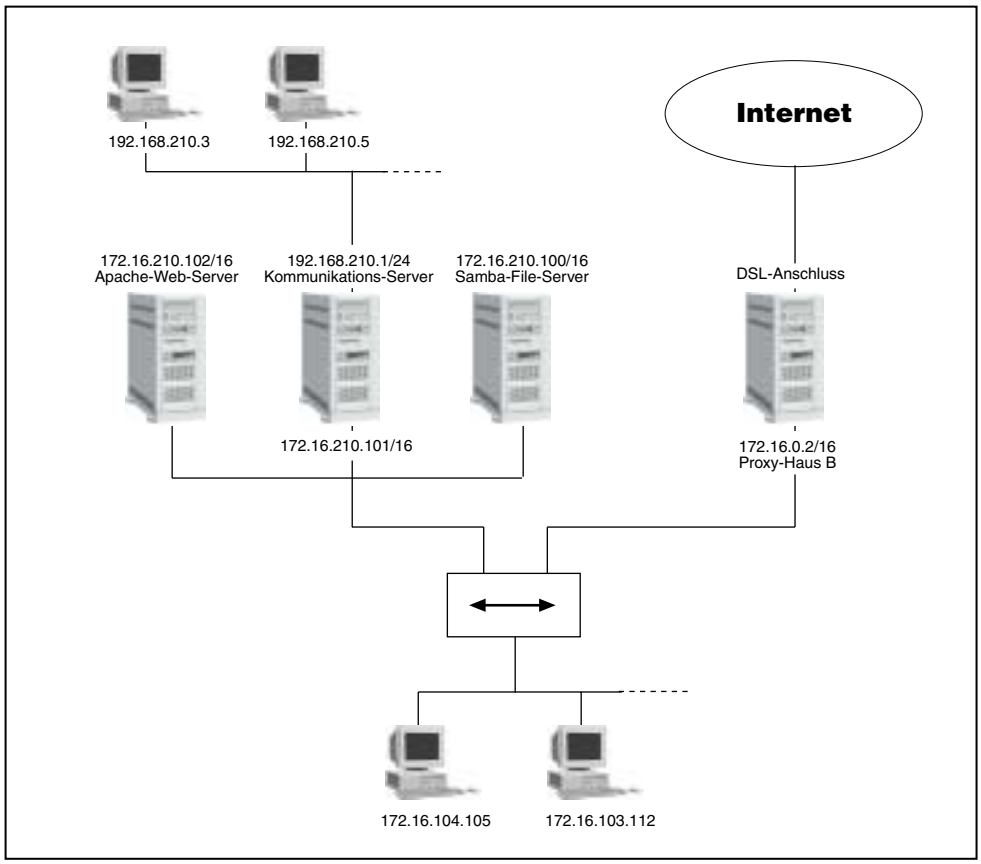

Abbildung 1.1: Topologie unserer Netzwerkstruktur *[schematisch]*

2. Paketfilter-Firewall - iptables

Der Befehl „iptables“ kann mit diversen Parametern und Filterangaben direkt auf der Konsole ausgeführt werden. Die auf diese Weise hinzugefügten, gelöschten oder geänderten Filterregeln werden sofort im laufenden System angewandt. Bei einem Neustart des Rechners sind die Firewall-Regeln allerdings gelöscht.

Firewall-Script

Deshalb sollte man diese Regeln fortlaufend in einem Script niederschreiben. Dieses Script kann in jedem Text-Editor geschrieben werden. Der Benutzer (root) muss das Ausführen-Dateirecht (execute bzw. x) für diese Datei erhalten. Eine Shell (*hier: tc-shell*) interpretiert die dort aufgelisteten Befehle, die nacheinander abgearbeitet werden.

Das **erste erstellte Script** soll alle evtl. vorhandenen Regeln löschen und alle IP-Pakete in alle Richtungen **erlauben**.

fw_clear

```
#!/bin/tcsh
# ------------ Firewall-Script von Manuel Ecker, 15.01.2002 ---------------
# --------Script zum Löschen aller Regeln + alles ERLAUBEN !!!! -----------

set IPTABLES = /usr/sbin/iptables

# ----------------------------------------
# Default Policy und flush

$IPTABLES -P INPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT
$IPTABLES -P OUTPUT ACCEPT

$IPTABLES -F         # flush aller chains (Tabelle Filter)
$IPTABLES -t nat -F  # flush aller chains (Tabelle NAT)
$IPTABLES -X         # delete all userdefined chains (Tabelle Fiter)

echo "done (default policy + flush)"
echo "done Firewall not activ"
```

In der ersten Zeile des Scriptes muss immer die Zeile `#!/bin/tcsh` stehen. Dieser Eintrag definiert, dass es sich hierbei um ein auszuführendes Script handelt und welches Konsolenprogramm (Shell) dieses ausführen soll.

Die Zeile `set IPTABLES = /usr/sbin/iptables` definiert die Variable `IPTABLES`. Sie gibt an, in welchem Verzeichnisses sich der Befehl befindet.

Im nächsten Block wird die **Default Policy** für die verschiedenen Regelketten festgelegt:

`$IPTABLES`	*Befehlsaufruf*
`-P`	*Parameter, der die Default Policy der Regelkette ändert*
`INPUT`	*Angabe für welche Kette (Chain) die Regel gilt*
`ACCEPT`	*Policy dieser Regel*

mögliche Policies:

`ACCEPT`	*Durchlassen des IP-Paketes*
`DROP`	*Verwerfen des Paketes, der Absender bekommt keine Info über den Verbleib des IP-Paketes*
`REJECT`	*Verwerfen des Paketes, der Sender erhält eine ICMP-Fehlermeldung („Port Unreachable“)*

Im nächsten Schritt sollen alle Filterregel-Tabellen gelöscht werden; dazu gibt es folgende Parameter:

`-F`	*Parameter, der alle Filterregeln (-t filter) löscht*
`-t nat -F`	*Parameter der Tabelle NAT, die alle Regeln dieser Tabelle löscht (default: -t filter)*
`-X`	*Parameter, der alle userdefinierten Regelketten löscht*

Ein anderes Script ermöglicht alle bestehenden und von innen kommende Verbindungen. Sie verbietet den Aufbau einer neuen Verbindung aus dem Internet.

fw_me3

```
#!/bin/tcsh
# ------------ Firewall-Script von Manuel Ecker, 16.01.2002 ---------------

echo "setting firewall rules "

set IPTABLES = /usr/sbin/iptables

# ----------------------------------------
# Default Policy und flush

$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
$IPTABLES -P OUTPUT DROP

$IPTABLES -F         # flush aller chains (Tabelle Filter)
$IPTABLES -t nat -F  # flush aller chains (Tabelle NAT)
$IPTABLES -X         # delete all userdefined chains (Tabelle Fiter)

echo "done (default policy + flush)"
# ----------------------------------------

# lokale Prozesse

$IPTABLES -A OUTPUT -o lo -j ACCEPT
$IPTABLES -A INPUT  -i lo -j ACCEPT

echo "done (lokale Prozesse)"
# ----------------------------------------


# Kette (Name: block) erstellen (-N), die neue Verbindungen blockt,
# es sei denn, sie kommen von innen
$IPTABLES -N block
$IPTABLES -A block -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A block -m state --state NEW -i ! eth0 -j ACCEPT
$IPTABLES -A block -j DROP


# von INPUT und FORWARD zu block springen
$IPTABLES -A INPUT -j block
$IPTABLES -A FORWARD -j block

# MASQUERADING
# Maskiere eth0 (externes Interface)
$IPTABLES -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

```
# Erlaube ESTABLISHED und RELATED ankommende Pakete von eth0
$IPTABLES -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT

# Verbiete NEW und INVALID ankommende Pakete von eth0
$IPTABLES -A INPUT -i eth0 -m state --state NEW,INVALID -j DROP



# Erlaube ESTABLISHED und RELATED weitergeleitete Pakete von eth0
$IPTABLES -A FORWARD -i eth0 -o 0 -m state --state ESTABLISHED,RELATED \
          -j ACCEPT

# Verbiete NEW und INVALID weitergeleitete Pakete von eth0
$IPTABLES -A FORWARD -i eth0 -o 0 -m state --state NEW,INVALID -j DROP


# IP-Forwarding aktivieren
echo 1 > /proc/sys/net/ipv4/ip_forward

echo "done (masquerading)"
echo "done Firewall fw_me3 is activ"
```

Im ersten Teil des Scriptes stehen die gleichen Anweisungen wie in der Script-Datei fw_clear. Es werden also alle bestehenden Filterregeln gelöscht und die Default Policy der Standardketten auf DROP gesetzt.

Im Abschnitt „lokale Prozesse“ werden alle Pakete (input und output) **lokaler Prozesse** d.h. interne Prozesse an der Schnittstelle localhost (lo) **erlaubt**.

Syntax:

```
$IPTABLES -A [Chain] -o lo -j ACCEPT
```

`-A`	*Regelkette anfügen*
`[Chain]`	*Name der Regelkette*
`-o`	*Output-Interface*
`-i`	*Input-Interace*
`lo`	*Interface Localhost*
`-j [target]`	*jump-to ... (Ziel bzw. Target einer Filterregel)*

Als nächstes wird eine **userdefinierte Kette** erstellt, die neue Verbindungen abblockt, also verwirft, es sei denn die Verbindung kommt von innen.

Syntax:

```
$IPTABLES -N [Name der User-Chain]
$IPTABLES -A ...
```

`-N`	*Userdefinierte Regelkette erstellen*

Ab der zweiten Zeile dieses Blocks werden dann die Regeln hinzugefügt, die für diese neu erstellte User-Kette gelten sollen.

Im Abschnitt „**Masquerading**“ wird das externe Interface (eth0; 172.16.210.101) maskiert.

Syntax:

```
$IPTABLES -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

`POSTROUTING` *Regelkette von SNAT, die erst durchlaufen wird, kurz bevor das Paket in das Netzwerk entlassen wird*

`MASQUERADE` *Spezialtarget von SNAT - Automatische Übernahme der jeweiligen IP-Adresse des Output-Interface als Source-IP-Adresse*

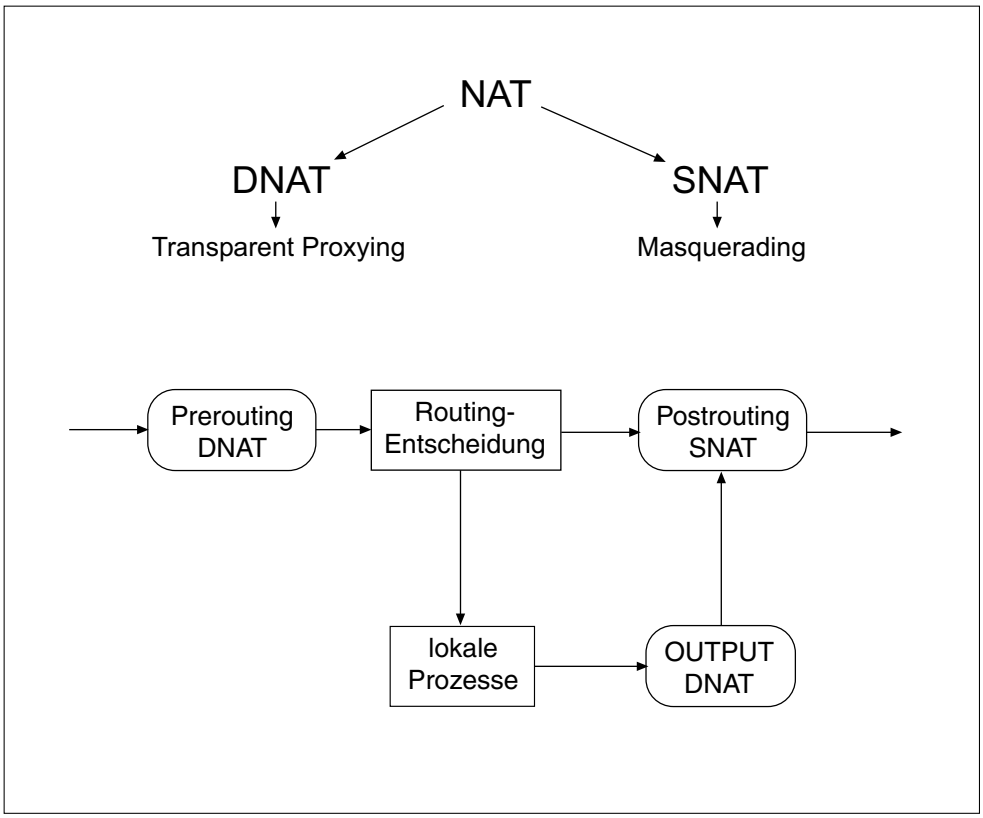

Abbildung 1.2: Skizze Post-, Prerouting und Masquerading

Erlauben und **Verbieten** für bestimmte ankommende und abgehende Pakete werden im folgenden geregelt.

Syntax:

```
$IPTABLES -A INPUT -i eth0 -m state --state ESTABLISHED, RELATED -j ACCEPT
```

`-m state ...`	*Kontrolle des Verbindungszustandes*
`ESTABLISHED`	*Paket, das zu einer bestehenden Verbindung gehört*
`RELATED`	*Pakete, die in Beziehung zu einer offenen Verbindung stehen, aber nicht direkt zur Verbindung gehören*
`NEW`	*Paket, das eine Verbindung initiiert*
`INVALID`	*Alle Pakete, die keinem der zuvor genannten Zustände zugeordnet werden können*

Zusätzlich können **Textmeldungen** beim Abarbeiten des Scriptes auf dem Monitor ausgegeben werden. Dies kann sinvoll sein, um nachzuvollziehen, welche Funktionen das Firewall-Script ausführt.

Syntax:

```
echo "[Ausgabe-Text]"
```

fw_me4

Dieses Script ist das um eine Zeile erweiterte fw_me3-Script. Es ermöglicht den Zugriff von 192.168.210.0-Clients den Zugriff über den Proxy auf den Apache (172.16.210.102).
[Auszug der Änderung gegenüber fw_me3]

```
#!/bin/tcsh
# ------------ Firewall-Script von Manuel Ecker, 18.01.2002 ---------------
# ----- Zugriff von Client über Poxy auf lokalen Apache-Web-Server --------

# von INPUT und FORWARD zu block springen
$IPTABLES -A INPUT -j block
$IPTABLES -A FORWARD -j block
$IPTABLES -A OUTPUT -j block        # ergänzt zu fw_me3
```

Hinweis:

Mit dem Befehl `IPTABLES -L [chain]` kann man sich alle gesetzten Filterregeln der genannten Kette auflisten lassen. - Kontrollfunktion.

Automatischer Script-Start beim Booten

Hat man nun ein lauffähiges Firewall-Script mit den benötigten Eigenschaften erstellt, muss man den Kommunikationsserver noch anweisen, diese Script-Datei bei jedem Booten des Rechners auszuführen, damit die Firewall auch aktiviert wird. Dazu muss ein symbolischer Link angelegt werden, der das Script in den Runleveln 2 und 3, in denen das Netzwerk aktiv ist, startet. Um zu verhindern, dass beim Betriebssystem-Update das Firewall-Script verloren geht, da das Verzeichnis `/etc/init.d`, das für Startscripte zuständig ist, evtl. „aufgeräumt“ wird, legen wir das Script im Verzeichnis `/root/fw` ab.

Der symbolische Link für unseren Firewall-Start (*z.B. fw_me3*):

```
ln -s /root/fw/fw_me3/ /etc/init.d/rc2.d/S04fw_me3
ln -s /root/fw/fw_me3/ /etc/init.d/rc3.d/S04fw_me3
```

(vgl. [1] Seite 224)

Unbenötigte Dienste deaktivieren

Auf einem Firewall-Rechner sollten alle unbenötigten Dienste abgeschaltet sein, da diese nur unnötige Sicherheitsrisiken darstellen. Diese Dienste müssen in der Datei `/etc/inetd.conf` deaktiviert werden.

Wir haben folgende Dienste beim Start deaktiviert:

- NFS Automount-Dämon (`amd`)
- Datenbank zur Zuordnung zwischen Ethernet- und IP-Adresseb (`arpwatch`)
- Automatisches Mounten von Netzwerkdateisystemen (`autofs`)
- Boot Parameter Server (`bootparamd`)
- Lokaler DHCP-Server (`dhcpd`)
- Gateway Routing Dämon (`gated`)
- Apache-Web-Server (`httpd`)
- Lokaler Usenet-News-Server (`innd`)
- Druck-Server (`lpd`)
- File- und Print-Server für Novell-Windows-Clients im lokalen Netz (`mars-nwe`)
- File-Server zum Midnight-Commander (`mcserv`)
- Dateisystem-Mount vom Netzwerk über NFS, Samba oder NetWare (`netfs`)
- NFS-Dienste (`nfs`)
- Name Switch Cache-Dämon (`nscd`)
- Portmap-Manager (`portmap`)
- SQL-Dantenbank-Server (`postgresql`)
- Routed-Dämon zum dynamischen aktualisieren der Routig-Tabellen des Kernels (`routed`)
- `rstatd`-Dämon liefert andern Geräten im Netzwerk Systeminformationen
- Remote-User-Local-Service (`ruserd`)

(liefert Infos über gerade auf Computern des Netzes angemeldete User)

- `rwalld`-Dämon
- `rwhod`-Dämon
- Sendmail-E-mail-Dienst (`sendmail`) *[wenn kein lokaler Mailserver eingesetzt wird]*
- Samba-Server (`smb`)
- Netzwerkadministration über SNMP (`snmpd`)
- NIS-Client (`ypbind`)
- NIS-Passwort-Server (`yppasswdd`)
- NIS-Master-Server (`ypserv`)

(vgl. [2] Seite 70-79)

3. Squid als Proxy-Server

Die Software **Squid Version 2.3** ist im Softwarepaket „n“ der LINUX-Distribution SuSE 7.3 enthalten. Nach der Installation der Software muss der Startparameter in der Datei `/etc/rc.config` für `start_squid` auf „`yes`“ gesetzt werden, damit der Proxy-Server nach jedem Neustart des Rechners automatisch gestartet wird. Dies kann man z.B. relativ einfach über YaST machen. Bei Neustart des Squid wird die Konfigurationsdatei `squid.conf` eingelesen. Wird bei laufendem Server diese Konfigurationsdatei geändert, muss - damit die Änderungen im laufenden System übernommen werden - der Squid mit dem Befehl `/etc/init.d/squid restart` oder `rcsquid restart` neu gestartet werden. Zum Ändern der Konfiguration gibt es auch die Möglichkeit, das komfortable Squid-Proxy-Modul des Webmin (*hier: Version 0.91*) zu benutzen.

In der **squid.conf** machten wir folgende Einträge *(Auszug der eingetragenen Zeilen)*:

```
http_port 3128

dns_children 20

# ACCESS CONTROL

acl RaumB210 src 172.16.210.100 - 172.16.210.200/255.255.255.255

http_access deny RaumB210   # bzw. http_access allow RaumB210
```

Die Einstellungen für Access Control sind am besten mit dem Webmin zu machen. Auf diese Weise ist eine einfache Regelung der Zugriffsrechte für bestimmte Hosts bzw. Hostbereiche (z.B. Hosts verschiedener Räume) möglich. Man kann so also ganz leicht festlegen, wer (in unserem Fall) auf das Intranet zugreifen darf und wer nicht.

Um den Zugriff auf die Proxydienste zu regeln, müssen Zugriffsregeln eine sog. Access List = acl definiert werden.

`acl [Name] src [x.x.x.x/x.x.x.x]` *Zugriffsregel für einen bestimmten Host*
`acl [Name] src [x.x.x.x - x.x.x.x/x.x.x.x]` *Zugriffsregel für einen Hostbereich*

Es ist auch möglich einzelne Rechner vollständig zu sperren. Zum Beispiel Domains mit unerwünschten Inhalten (Gewalt, Sex, Porno, ...)

`acl [Name] dstdomain [Domain]` *Zugriffsregel für eine bestimmte Domain*
z.B.

```
acl sex dstdomain www.sex.de www.sex.com
```

Es gibt auch eine etwas allgemeinere Regel, die einen Zugriff für bestimmte Schlüsselwörter, die in der Adresse und den Dateinamen sowie in Anfragen in Suchmaschinen vorkommen, regelt.

`acl [Name] browser [Schlüsselwörter]` *Zugriffsregel für Schlüsselwörter*
z.B.

```
acl gewalt browser Gewalt Folter Krieg
```

Es ist auch eine Freigabe bzw. Sperrung möglich, die einen Zugang ins andere Netz (Intranet bzw. Internet) über den Proxy nur zu einer bestimmten **Zeit** ermöglicht.

`acl [Name] time [Zeitangabe]` *Zugriffsregel für Zeitraum*

z.B. von Montag bis Freitag von 7:45 bis 16:00 Uhr

```
acl Zeit-Freigabe time MTWHF 7:45-16:00
```

Die so definierten Regeln müssen im folgenden noch **aktiviert** werden:

```
http_access allow RaumB210
```

Diese Freigabe-Zeilen müssen unbedingt **vor** folgender Zeile stehen:

```
http_access deny all
```

Wenn eine Sperrung aktiv ist (bei einer Änderung muss der Squid neu gestartet werden) zeigt der Browser beim Versuch gesperrte Seiten bzw. Seiten von einem gesperrten Client aus aufzurufen, eine Fehlermeldung.

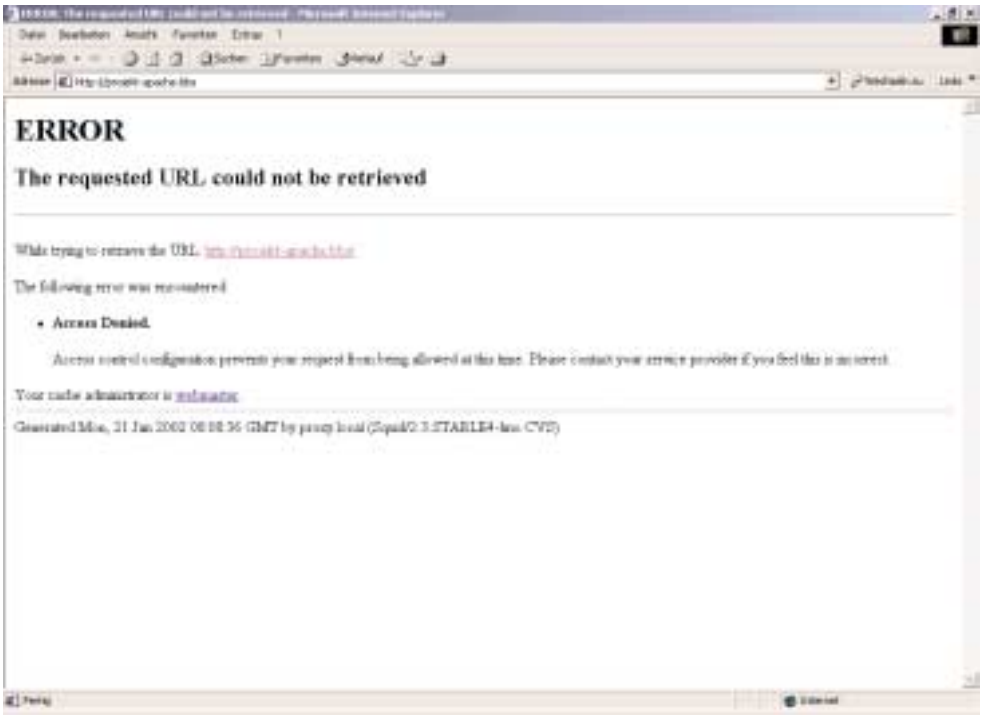

Abbildung 3.1: Zugriffsverweigerung bei Sperrung - Access Denied

Webmin - das Administrations-Tool

Das Programm Webmin ist ein **Fernadministrations-Tool**, mit dem man die wichtigsten Einstellungen alternativ zur Bearbeitung der Konfigurationsdateien in einem Editor erledigen kann. Für den Webmin gibt es eine ganze Reihe von Modulen für verschiedene Server- und Systemeinstellungen. Zugriff auf den Webmin hat man über einen beliebigen Browser von einem freigeschalteten Host aus. Der Webmin wird mit der Eingabe der Server-IP mit Angabe des Port 10000 gestartet (z.B. http://172.16.210.101:10000). Die aktuelle Version kann aus dem Internet (www.webmin.org) heruntergeladen werden.

Browser einstellen

In den Browser-Einstellungen muss der Proxy-Cache aktiviert werden. Dazu muss im entsprechenden Dialogfeld die IP des Proxy-Servers sowie die Nummer des verwendeten Ports eingetragen werden.

Achtung: *Wenn auf dem Kommunikations-Server* `IP-Masquerading` *aktiviert ist, können Anwender den Proxy umgehen, in dem sie im Browser die Proxy-Einstellung deaktivieren. Somit hätten sie einen direkten Internetzugriff.*

Logdatei des Squid

In der Logdatei `accesss.log` kann man nachschauen, wer welche Internetseiten (in unserem Fall Intranetseiten auf dem Apache) aufgerufen hat. So lassen sich alle Zugriffe aus unserem lokalen Netz nachvollziehen.

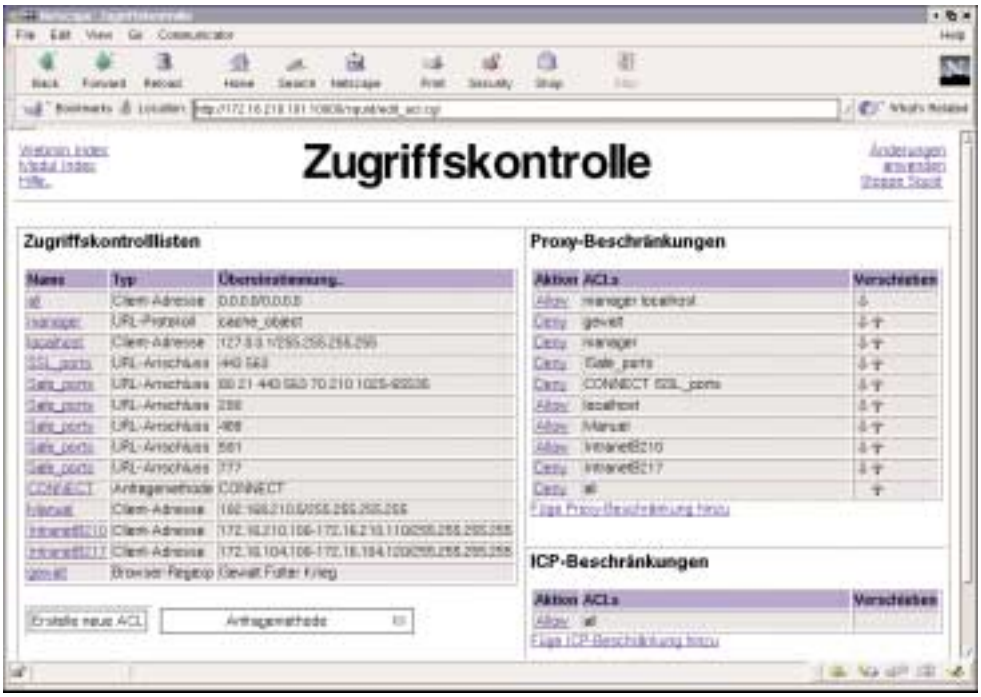

Abbildung 3.2: Fernadministration mit dem Webmin

Abbildung 3.3: Proxy-Regeln bearbeiten mit dem Webmin

4. BIND8 - der eigene Name-Server

Was ist ein Name-Server?

Der **Domain Name Service (DNS)** ist dafür zuständig, Adressen (z.B. www.bbs-nw.de) in eine gespeicherte Zahlenreihe, die IP-Adresse umzusetzen. Dies ist notwendig, da es im Internet keine Namen sondern nur IPs gibt. Damit ein User die uns bekannten Internet-adressen in seinem Browser eingeben kann und dann die entsprechende Webseite vom Webserver übertragen wird, muss also eine **Namensauflösung** stattfinden.

Warum brauchen wir in unserem Intranet einen DNS?

In unserem Projekt stellte sich das Problem, dass der lokale Proxy-Server (192.168.210.1) für die Namensauflösung auf seine lokale HOSTS-Datei (`/etc/hosts`) nicht zugreift. (*Zur Erinnerung:* In unserem System greifen die Hosts (192.168.210.0) über den lokalen Proxy auf den Apache-Web-Server (172.16.210.102) zu.) Deshalb entschlossen wir uns, das Projekt um die Installation eines lokalen Name-Servers zu erweitern. Bei dieser Konfiguration handelt es sich aber um eine lauffähige Minimal-Konfiguration. Die weitergehende Untersuchung der Funktionsweise eines DNS konnte aus Zeitgründen in diesem Projekt nicht durchgeführt werden. Es wurde lediglich der Versuch unternommen, eine Konfigurationsdatei von Julian Pawlowski (info@thepenguin.de) zu ändern und auf unsere Gegebenheiten anzupassen.

BIND8

Bei dem hier verwendeten **DNS-Implementierung** handelt es sich um die Software BIND8 (**B**erkley **I**nternet **N**ame **D**omain Software), die wohl den bekanntesten DNS zur Verfügung stellt und in den Netzwerkpaketen von LINUX enthalten ist.

Konfiguration

Bei unserem Server handelt es sich um einen Master-Server, einen sog. Primary-Server, der die Anfragen direkt aus Dateien beantwortet, die auf diesem Rechner angelegt wurden. Er ist ein authorativer Server, d.h. er verfügt über alle Daten seiner Domain. Die zentrale Konfigurationsdatei des BIND8 ist die `/etc/named.conf`. Bei der hier gezeigten Konfiguration handelt es sich um eine Standardkonfiguration die noch zu optimieren ist. Dieser Konfigurationsdatei verweist auf die sog. Zonefiles (`var/named`).

Unsere Konfiguration umfasst folgende Dateien:

- `/etc/named.conf`
- `/var/named/127.0.0.rev`
- `/var/named/192.168.210.rev`
- `/var/named/bbs.hosts`
- `/var/named/localhost.hosts`
- `/var/named/root.hint`

Die zwei Dateien, die für uns am wichtigsten für die lokale DNS-Installation waren und dazu angepasst und geändert werden mussten sind auf den folgenden Seiten zu sehen. Die wichtigsten Einstellungen sind kommentiert. Auf Details und einzelne Einstellungen kann nicht eingegangen werden, da wir uns damit nicht beschäftigen konnten und nur eine funk-tionsfähige Basis-Konfiguratiuon für einen lauffähigen DNS in unserem Netzwerk erreichen wollten.

named.conf

```
# BIND-Konfigurationsdatei

# Erstellt von Julian Pawlowski, info@thepenguin.de
# Angepasst und geändert von Manuel Ecker, 23.01.2001

acl internal {                      //Access-Liste für INTERNAL
      127.0.0/24;
      192.168.210/24;
      172.16/16;
      };

options {                           //Grundoptionen, die festzulegen sind
        directory "/var/named";
      check-names master warn;
      recursion yes;
      files unlimited;
      transfer-format many-answers;
      allow-query { internal; };
      forward first;
      forwarders {
             194.25.2.129;          //Einrichten exterer Nameserver
             212.185.248.180;
             };
};

logging {
        category lame-servers { null; };
      category cname { null; };
};
                                    //Zonen-Deklaration
zone "bbs" in {                     //BBS-Zone wird definiert
      type master;
      file "bbs.hosts";             //Zuweisung des entsprechenden
      check-names fail;             //Konfigurations-Files
      allow-update { none; };
      allow-query { any; };
      allow-transfer { none; };
      notify yes;
};

zone "localhost" in {
      type master;
      file "localhost.hosts";
      check-names fail;
      allow-update { none; };
      allow-query { any; };
      allow-transfer { none; };
      notify yes;
};

zone "210.168.192.in-addr.arpa" in {
      type master;
      file "192.168.210.rev";
      check-names fail;
      allow-update { none; };
      allow-query { any; };
      allow-transfer { none; };
      notify yes;
};

zone "0.0.127.in-addr.arpa" in {
      type master;
      file "127.0.0.rev";
      check-names fail;
      allow-update { none; };
};

zone "." in {
        type hint;
      file "root.hint";
};
```

bbs.hosts

```
@     IN      SOA     k-server.bbs.  20001111703 (
                      18000
                      900
                      432000
                      36000
                       )
;
; Adressen fuer kanonische Namen:
;
localhost                     IN      A       127.0.0.1
felix.bbs.            5W      IN      A       172.16.210.204
projekt-apache.bbs    5W      IN      A       172.16.210.102
@                     5W      IN      A       172.16.210.102
```

In dieser Datei wird die Zuweisung der Namen auf die entsprechenden IP-Adressen festgelegt, die der DNS bei einer Anfrage ausliest. In der ersten Spalte wird der Name, in der letzten die IP-Adresse angegeben. Dazwischen sind noch einige benötigte Parametereinstellungen zu machen, wie auch in den ersten Zeilen der Datei. Diese Einstellungen sind erforderlich - es kann aber nicht näher darauf eingegangen werden.

Start des Name-Servers

Der BIND8 sollte beim Systemstart automatisch gestartet werden. Dazu muss der Startparameter in der Datei `/etc/rc.config` für `start_named` auf „yes“ gesetzt werden. Der BIND8 lässt sich nach dem Ändern der Konfigurationsdateien mit dem Befehl `rcnamed restart` neu starten.
Nun muss der lokale DNS noch im System definiert werden, damit er bei der Namensauflösung um Hilfe gebeten wird. Dazu muss in der Datei `/etc/resolv.conf` die Zeile `nameserver 127.0.0.1` eingetragen werden. Alternativ kann hierzu auch YaST *(Administration des Systems - Netzwerk konfigurieren - Konfiguration Nameserver)* verwendet werden.

Funktionstest

Man kann die Funktion des DNS mit dem Befehl `nslookup server` testen.

In der Ausgabe ist zu erkennen, dass der lokale DNS (localhost 127.0.0.1) den eingegebenen Namen (projekt-apache.bbs) in die IP 172.16.210.102 auflöst.

Projektmanagement

Aufgabenverteilungsplan

Aufgaben:	**Verantwortlicher:**
1. **Theorieteil** (bis 14.01.2002) *Text zu folgenden Themengebieten verfassen:*	
• Aufgabenstellung	Manuel Ecker
• Problemstellung und Einführung	Manuel Ecker
• Zielsetzung	Manuel Ecker
• Hardware	Sascha Badstübner
• Software	Sascha Badstübner
• Sicherheit - Die Firewall	Manuel Ecker
• Anbindung an das Internet - Router oder Proxy	Manuel Ecker
• Glossar	Sascha Badstübner
2. **Praxisteil** *Intensivphase (14.01.2002 bis 18.01.2002)*	
• Konfiguration unseres Kommunikationsservers	Manuel Ecker
• Dokumentation der Konfiguration	Manuel Ecker
3. **Sonstige Arbeiten**	
• Dokumentation Projektmanagement	Sascha Badstübner / Manuel Ecker
• Satz, Druck und Bindung der Dokumentation	Manuel Ecker

Protokolle

Datum: 02.11.2001
Uhrzeit: 12^{15} - 13^{15} Uhr
Protokollführer: Sascha Badstübner

• Festlegung des Themengebietes „Proxy und Firewalls“
• Projektbetreuung: Alexander Scheib und Walter Schorr
• Projektteilnehmer: Sascha Badstübner, Manuel Ecker

• Überlegung: Installation eines Kommunikations-Servers mit Firewall-Funktion

• Ziel bis zur nächsten Projektsitzung ist die Konkretisierung des Themas, die Strukturierung und das Festlegen der Ziele.

Datum: 07.11.2001
Uhrzeit: 13^{00} - 14^{30} Uhr
Protokollführer: Sascha Badstübner

In dieser Projektsitzung, in der die Herren Scheib und Schorr anwesend waren, wurden die Ideen, wie ein konkretes Projektthema aussehen könnte zusammengetragen. Daraus folgte die vorläufige Themenformulierung:

Proxy und Firewalling unter Linux - Aufbau eines Kommunikations-Servers zur Anbindung eines lokalen Netzwerkes an das Internet unter Berücksichtigung von Sicherheitsaspekten.

Denkanstöße zum Inhalt des Projektes:
• was?, wozu?, mit was?, wie?, warum?
• Warum wird ein solches System installiert? Notwendigkeit?
• Zusammenhang mit Apache- und SAMBA-Projekten erläutern (Netzwerktopologie)
• Was werden bzw. haben wir konkret gemacht? Konzept des Aufbaus
• Welche Dienste soll unser Kommunikationsserver zur Verfügung stellen? - Verwendung bzw. Einsetzbarkeit in anderen Netzwerken?

Ziel bis zur nächsten Projektsitzung ist die genauere Ausarbeitung des Themas, die Suche nach Informationen und Einlesen ins Thema.

Datum: 13.11.2001
Uhrzeit: 15^{15} - 18^{00} Uhr
Protokollführer: Sascha Badstübner / Manuel Ecker

Einrichtung des Projektraumes B210 :
(in Zusammenarbeit mit den anderen Projektgruppen und den Herren Scheib und Schorr)
• Aufbau und Installation der Hardware
• Vernetzung der Rechner
• Installation des Netzwerkanschlusses an das hausinterne Netz
• Installation des Betriebssystems SuSE LINUX 7.3 (mit Netzwerkpaket (n))
• Einrichten des Internetzuganges

Datum: 14.11.2001
Uhrzeit: 15^{15} - 16^{00} Uhr
Protokollführer: Manuel Ecker

Absprache mit den anderen in Raum B210 tätigen Projektgruppen über Rechnerzuteilung und Arbeitsweise

Die bei der Installation nicht anwesenden Gruppenmitglieder der anderen Gruppen wurden über die erfolgte Installation und Vorgehensweise informiert. Die Teilnehmer der Kommunikationstechnik-Projekte sahen kein Problem oder Konfliktpotenzial für die Zusammenarbeit in diesem Projekt-Raum. Bei Bedarf werde man sich aber zusammensetzen und Gespräche führen, um zu Lösungen zu gelangen, so dass jede Gruppe möglichst ungestört arbeiten kann.

Datum: 09.01.2002
Uhrzeit: 13^{00} - 14^{15} Uhr
Protokollführer: Sascha Badstübner

- Besprechung über die bisher gesammelten und bearbeiteten Materialien.

- **Aufgabenverteilung des Theorieteils:**

  Sascha Badstübner:
  - Hardware
  - Verwendete bzw. benötigte Hardwarekomponenten
  - Software
  - Wieso Linux, Squid, iptables? (Begründung, Geschichtliches, Aufbau)

  Manuel Ecker:
  - Die Firewall
  - Grundlagen der Firewalltechnik (Definition, Funktion, Anwendung)
  - Paketfilter-Firewall
  - Modelle routender Firewalls (Erläuterung, Schema, Vor- und Nachteile)
  - Anbindung an das Internet - Router oder Proxy

Datum: 14.01.2001 bis 18.01.2002 (Intensivphase)
Uhrzeit: 8^{00} - 20^{00} Uhr

Wir legten fest, dass Manuel die Konfiguration des Servers durchführt und Sascha seinen noch nicht komplett ausgearbeiteten Dokumentationsteil fertigstellt und sich weiterhin über die technischen Aspekte informiert. Tägliche Besprechnungen mit Alexander Scheib über die weitere Vorgehensweise und den Inhalt der Dokumentation sowie gemeinsames Erarbeiten praktischer Problemlösungen am Rechner wurden durchgeführt.

Datum: 25.01.2002
Uhrzeit: 13^{30} - 14^{00} Uhr
Protokollführer: Manuel Ecker

Gemeinsame Besprechnung mit Alexander Scheib mit Sichtung des aktuellen Stands der Dokumentation. Der Projektbetreuer war der Ansicht, dass die Art der Dokumentation und die darin abgehandelten Aspekte in Ordnung sind und keine wesentliche Dinge fehlen. Wir besprachen auch die Art der Dokumentation der Serverinstallation. Diese sollte so gestaltet sein, dass er ,als Netzwerkspezialist, die von uns durchgeführte Installation nachvollziehen kann.

„W-Fragen“

Was?

Proxy, Firewalling und DNS - Aufbau eines Kommunikations-Servers unter Linux zur Anbindung eines lokalen Netzwerkes an das Internet unter Berücksichtigung von Sicherheitsaspekten.

Wer?

Sascha Badstübner
Manuel Ecker

Warum?

Projekt wird im Rahmen der Ausbildung zum staatl. geprüften Medienassistenten an der Höheren Berufsfachschule Medien Neustadt / Weinstraße durchgeführt.

Wozu?

In dem Projekt sollen die Teilnehmer wertvolle Erfahrung in der Planung und Durchführung von Projekten sammeln. Das Wissen und die Fähigkeiten sollen im entsprechenden Fachbereich erweitert werden und die Teamfähigkeit verbessert werden. Des weiteren sollen die Projektbetreuer von der Projekt-Arbeit profitieren d.h. Erfahrungen und Wissen können anderen Schülerinnen und Schülern sowie Lehrerinnen und Lehrern zu gute kommmen.

Mit wem?

Dieses Projekt wird in Zusammenarbeit mit dem Netzwerkspezialisten Alexander Scheib und Walter Schorr sowie den anderen Projektgruppen aus dem Bereich der Kommunikations- und Netzwerktechnik durchgeführt.

Für wen?

Auftraggeber ist die Berufsbildende Schule Neustadt / Weinstraße

Wie?

- Definieren von Zielen
- Informationsrecherche
- Einarbeitung ins Thema
- Praktische Umsetzung
- Regelmäßige Treffen der Projektteilnehmer und gegebenenfalls mit den Projektbetreuern

Womit?

Als Hilfsmittel dienen Fachliteratur, Internet und das Wissen der Projektteilnehmer und Betreuer.

Wann?

Projektsitzungen werden nach Bedarf im Projektraum oder einem anderen Informatiklabor abgehalten.

Wo?

Der Arbeitsraum dieses Projektes ist der extra dafür eingerichtete Raum B210.

Projekt-Ziele

- Erweiterung des Wissens im Bereich der Kommunikations- und Netzwerktechnik
- Erfahrungen im Arbeiten mit dem Betriebssystem LINUX und der Installationen von Servern zu sammeln
- Arbeiten im Team trainieren
- Fähigkeit zu entwickeln innerhalb kurzer Zeit sich in ein neues Thema einzuarbeiten
- Erstellen einer soliden Projektarbeit, die als Referenz für Bewerbungen verwendet werden kann

Persönliche Erfahrungen von Sascha Badstübner

Durch meine geringen Vorkenntnisse über das Betriebssystem Linux und das konfigurieren eines Kommunikationsserver mit Firewall-Eigenschaften, wurde dieses Projekt für mich zu einer großen Herausforderung. Da es eine riesige Auswahl an Fachliteratur gab und ich mir erst das Fachwissen aneignen musste, ging sehr viel Zeit für die eigentliche Ausarbeitung verloren. Abgesehen davon fand ich es wichtig in diesem Themengebiet eine Projektarbeit durchzuführen, da meine Interessen in der Computertechnik liegen, und meine spätere Berufslaufbahn in diese Richtung gehen soll.

In der heutigen Zeit sind Computer am Arbeitsplatz nicht mehr weg zu denken. An Rechnern werden Daten erarbeitet und diese können zu verschiedensten Orten auf der Welt versand werden, deswegen rückt der Schutz vor Eindringlingen von außen immer mehr in den Mittelpunkt. Für mich war es deswegen wichtig zu sehen, wie man sich wirksam und sinnvoll vor Angriffen von außerhalb schützen kann und welche Methoden man zur Verteidigung anwendet.

Die Themenbereiche des Projektes waren klar definiert und jeder wusste was er zu tun hatte. Dadurch kam meiner Meinung nach die Kommunikation zwischen Manuel und mir zu kurz. Jeder ging seinen eigenen Aufgaben nach und keiner wusste so recht, was der andere gerade macht. Wegen kleineren Auseinandersetzungen hatte ich manchmal Schwierigkeiten in der Gruppe klar zu kommen. Nichts desto trotz waren das eher kleinere Probleme in der Projektarbeit. Kritischer war dagegen die knapp bemessene Zeit, die uns zur Verfügung stand.

Im großen und ganzen war es eine erfahrungsreiche Zeit, dass mein Leben positiv beeinflusste. Dank der guten Verbindung zu unserem Projektleiter Herr Alexander Scheib hatten wir einen guten Ansprechspartner für Fragen und Problemen. Durch die theoretische und praktische Ausarbeitung des Projektes kam es hin und wieder zu Hindernissen, die zu bewältigen waren, was einen höheren Lerneffekt zur folge hatte. Am Ende ist nur noch zu sagen, dass mir die Arbeit, trotz zeitweiligem Stress und großer Belastung, spaß gemacht hat und ich nicht vor neuen Projekten zurück schrecken werde.

Persönliche Erfahrungen von Manuel Ecker

Ich bin mit dem Erreichen der Ziele und somit mit der Projektarbeit sehr zufrieden. Für mich ist diese Projektarbeit sehr wichtig, sie stellt als Referenzprojekt mein Können, meinen Arbeitseinsatz und die aufgebrachte Energieleistung im Bereich der Kommunikations- und Netzwerktechnik dar. Seit Beginn der Projektphase beschäftigte ich mich intensiv mit der Materie. Dennoch muss ich sagen, dass es sich bei der im Projekt erarbeiteten Konfiguration nur um Grundkonfigurationen handelt, die optimierbar und erweiterbar sind. Im Anbetracht dessen, dass es sich bei diesem Thema um ein riesiges Gebiet handelt, war eine Optimierung im Rahmen dieses Projektes leider nicht möglich, da neben dem Projekt noch zahlreiche weitere Verpflichtungen im Rahmen der Ausbildung an der BFHTM in Neustadt / Weinstraße zu erfüllen waren, und somit ein Großteil des Projektes unter großem Stress bewältigt werden musste.

Die Zeit der Intensivphase war für mich ein ständiges auf und ab - an einen Tag lief das System, am nächsten machte es mit uns was es wollte! Ich musste die Erfahrung machen, dass man nicht alles erklären kann, auch wenn man noch so intensiv nach Lösungsansätzen sucht und sämtliche Konfigurationseinstellungen durchprobiert. Leider lag diese unterrichtsfreie Zeit, in der man intensiv an einer solchen Konfiguration arbeiten und „tüfteln“ konnte, im Projektverlauf zeitlich erst ziemlich spät. So drängte die Zeit, da die Konfiguration auch noch dokumentiert und erläutert werden musste.

Die Zusammenarbeit mit meinem Projektpartner war nicht immer einfach, da er nicht die Fähigkeiten im Umgang mit dem Betriebssystem LINUX und den Erfahrungen im Bereich der Server-Konfiguration hatte wie ich. In einem Praktikum im Rahmen dieser Ausbildung konnte ich in der Cyberbureau Interntagentur, Grevenbroich-Kapellen im Oktober 2001 bereits Erfahrungen sammeln. Dort erarbeitete ich in einem Projekt zusammen mit dem Geschäftsführer Paul Linßen eine Konfiguration eines Samba-Servers mit Anbindung von Windows-Clients und Apple Macintosh-Clients sowie eines Apache-Web-Servers zum Einsatz in der Agentur.

Für mich und meine weitere berufliche Zukunft konnte ich wichtige und hilfreiche Erfahrungen sammeln. Meine vorhandenen Kenntnisse der Netzwerktechnik konnte ich anwenden und vertieften. Des weiteren sammelte ich wichtige Erfahrungen mit dem Betriebssystem LINUX und im Aufbau von Servern und den damit verbundenen Problemen. Das Projekt sensibilisierte mich außerdem für den Umgang mit Sicherheitsaspekten in lokalen Netzwerken. Hierzu trug auch das gute Verhältnis zu Alexander Scheib bei, der sich viel Zeit für uns und unser Projekt genommen hat und großes Engagement zeigte.

Glossar

Squid

Ein Programm, dass unter Linux die Proxy-Funktionen übernimmt. Es baut auf einem lokalen Server einen Cache auf , der Daten aus dem Internet zwischen speichert.

Ipfwadm

Ist ein Firewall Toolkid zur Administration der IPFW-Firewall. Durch eine einfache Routing-Tabelle regelt es den Transport von IP-Paketen auf Systemebene.

Ipchains

Aktuelleres Programm zur Administration der IPFW-Firewall. Durch definierte Regeln in einer Kette werden Pakete gefiltert.

Iptables

Neuste Programmversion zur Administration von IPFW-Firewalls. Iptables enthält drei Listen mit Filterregeln die entscheiden, was mit den Datenpaketen geschied.

Kernel

Der Linux-Kernel ist der innerste und der an die Hardware angepaßte Teil des Betriebssystems.

Linux

Ein Betriebssystem, dass von dem Finnen Linus Torvalds initiiert wurde und inzwischen von vielen Programmierern weiterentwickelt wird. Linux wird als GPL (General Public License) vertrieben, d.h. nur der Vertrieb selber kostet Geld, die eigentliche Lizenz des Betriebsystems bzw. der Software ist kostenlos.

Proxy-Server

Proxy bedeutet soviel wie "Stellvertreterdienst". Proxies nehmen Anforderungen von einem Client (z.B. einem WWW-Browser) entgegen und geben sie, gegebenenfalls modifiziert, an das ursprüngliche Ziel (z.B. eine WWW-Site) weiter. Proxies können die durchgeschleusten Daten lokal ablegen und beim nächsten Zugriff direkt liefern.

LAN

(Local Area Network) Lokal angelegtes Netzwerk, das sich in diesem Sinne auf einen gemeinsamen Standort, wie beispielsweise ein Firmengelände oder einen Raum, bezieht.

Firewall

Englische Bezeichnung für "Feuermauer" oder "Brandmauer". Das ist eine Technik, in Form von Hard- und/oder Software, die den Datenfluss zwischen einem privaten und einem ungeschützten Netzwerk (also LAN und Internet) kontrolliert bzw. ein internes Netz vor Angriffen aus dem Internet schützt.

Ethernet

Von den Firmen INTEL, DEC und Xerox entwickelter Netzwerktyp für LAN-Netzwerke, die für die Übertragung Koaxialkabel oder Twisted Pair verwenden.

IP

Abkürzung für "INTERNET Protocol" o IP gehört zur TCP / IP Protokollfamilie, einem anerkannten Industriestandard für die Kommunikation zwischen offenen Systemen. Das Übertragungsprotokoll definiert die Regeln und Vereinbarungen, die den Informationsfluss in einem Kommunikationssystem steuern. Hauptaufgabe des IP ist die netzübergreifende Adressierung. Das Protokoll arbeitet nicht leitungs-, sondern paketvermittelt: Sogenannte Datagramme suchen sich über die jeweils verfügbaren Verbindungen ihren Weg zum Empfänger.

Token-Ring

Ist eine von IBM entwickelte Netzwerk-Architektur. In der Netzwerk-Technik versteht man unter einem Token ein Bitmuster von drei Byte Länge, das ständig in einer Richtung ein Token-Ring-Netzwerk durchläuft. Ein Netzwerk-Token besteht aus dem
• Starting-Delimiter-Byte,
• dem Ending-Delimiter-Byte sowie
• dem dazwischen liegenden Zugriffskontrollfeld.
Delimiter
Ist ein vereinbartes Abgrenzungszeichen. Beim Austausch von Daten zwischen verschiedenen Datenbank-Programmen gibt es nicht immer die Möglichkeit, dass die Fremddaten direkt gelesen werden können. Dann werden diese als ASCII-Datei ausgegeben, wobei die einzelnen Sätze in der Regel durch eine Zeilenschaltung voneinander getrennt werden und die einzelnen Felder entweder eine festgelegte Länge haben oder durch ein vereinbartes Delimiter abgetrennt werden - z.B. "," oder ";".

WAN

(Wide Area Network) Ein Netzwerk, dass weltumspannend angelegt sein kann (das aber zumindest die eigenen vier Wände verlässt).

NIC

(Network Interface Card). Ist die englische Bezeichnung für Netzwerkkarte.

GPL

Ist die Abkürzung für "General Public License" und ist ein alternatives Vertriebskonzept, dass im weitesten Sinne mit "Shareware" vergleichbar ist. Das heißt, nur der Vertrieb selber kostet Geld, die eigentliche Lizenz der Software ist kostenlos.

Cache

Ein schneller Puffer, der Daten zwischenspeichert und diese immer wieder sehr schnell zur Verfügung stellen kann.

Netscape

Hersteller der WWW-Browser Navigator und Communicator.

DNS

Ist die Abkürzung für "Domain Name Service", "-Server" oder "-System". DNS ist ein dezentraler Dienst, der Rechner-Namen bzw. INTERNET-Adressen im Klartext (z.B. www.google.de) und IP-Adressen (z.B. 209.204.209.212) einander zuordnet. Für jeden Server beziehungsweise für jedes LAN mit Internet-Anschluss, muss ein DNS-Server diese Informationen verwalten. Sobald eine Seite im Internet angewählt wird, fragt der Browser zuerst einen Domain Name Server. Dieser meldet die entsprechende numerische Adresse zurück, worauf der Browser eine direkte Verbindung zu IP-Adresse aufbauen kann.

DDoS

Ist die Abkürzung für "Distributed Denial of Service" ("denial": Ablehnung, Leugnung). Im Februar 2000 wurden verschiedene, große Internet-Dienste (wie z.B. Yahoo, CNN, Amazon, eBay, usw.) durch DDoS-Attaken lahm gelegt. Hierbei hatten sich die Angreifer Zugang zu hunderten von Rechnern im Internet verschafft (darum das "distributed"), um die Wirksamkeit ihrer Attaken durch die Vielzahl der gleichzeitig angreifenden Rechner stark zuerhöhen.

WWW

World Wide Web.

Header

Im allgemeinen ist das der Bereich am Anfang (am Kopf) von Dateien, in dem grundsätzliche Informationen über die Datei gespeichert sind.

Link

Ist die englische Bezeichnung für Verknüpfung oder Verbindung und ist eine Verbindung zu Daten, die sich in einem anderen Programm oder Dokument befinden. Diese interne Verknüpfung der Daten sorgt dafür, dass die Daten mit der Anwendung, in der sie ursprünglich erzeugt wurden, weiterbearbeitet oder automatisch aktualisiert werden könnnen.

IPFW

IP-Firewall-Mechanismus.

Patch

Ein Patch (englische Bezeichnung für "Flicken", manchmal auch "Bug fix" genannt) ist ein kleines Programm, das z.B. Bugs (Fehler) von in der Regel großen Anwendungsprogrammen repariert. Die meisten Patches werden von den Software- Herstellern auf ihren Web-Sites kostenlos zum Download angeboten. Oftmals werden Patches aber auch in die nächsten Versionen eines Programmes eingebaut, damit fehlerhafte Programme, die nicht gepatcht wurden auch repariert werden.

Paket

Ein IP-Datagramm.

IP-Datagramm

Ein Paket der Netzwerkschicht von IP.

Policy

Die Policy einer Firewall-Chain, unabhängig ob input, output oder forward, gibt an, was mit einem Paket geschied, wenn keine der Regeln aus der Liste zutrifft.

NAT

Abkürzung für "Network Address Translation" - Methode zur Umsetzung der (meist privaten) IP-Adressen eines Netzes auf andere (meist öffentliche) IP-Adressen eines anderen Netzes. NAT ermöglicht damit mehreren PCs in einem LAN, einerseits die IP-Adresse des Internet-Access-Routers für den Internet-Zugang zu nutzen, und andererseits versteckt es das LAN hinter der im Internet registrierten IP-Adresse des Routers.

Masquerading

Ist ein Vorgang, bei dem die Absenderadresse eines Paketes mit der IP der Firewall bzw. des Gateways ersetzt wird, dadurch bleibt die IP-Adresse des LANs verborgen. Nach außen hin wirkt es so, als käme das Paket vom Gateway, nicht von einem Computer im LAN. Bei ankommenden Antworten von einem fremden Server wird der Vorgang umgekehrt: Die Empfängeradresse des Paketes - die IP-Adresse der Firewall - wird durch die Adresse des Clients im internen LAN ersetzt. IP-Masquerading bezeichnet man allgemeiner auch als NAT (Network Adress Translation).

MAC Adresse

Ist die Abkürzung für Media Access Control Address und wird zur eindeutigen Bezeichnung einer Hardware (z. B. einer Netzwerkkarte) eingesetzt.

Literaturverzeichnis, Quellenangaben

[1] Das Firewall Buch - Grundlagen, Aufbau und Betrieb sicherer Netzwerke mit Linux,
Wolfgang Barth, SuSE-Press, Nürnberg, 2001
ISBN 3-934678-40-8

[2] linux firewalls - Konzeption und Implementierung für kleine Netzwerke und PCs,
Robert Ziegler, Markt+Technik Verlag, München, 2001
ISBN 3-8272-5849-9

[3] Crashkurs - Firewallaufbau unter Linux,
Florian Kalhammer, 21. März 2001
http://www.linux-praxis.de/downloads/Firewall.pdf

[4] Linux 2.4 Packet Filtering HOWTO,
Rusty Russell, Mai 2000

[5] Linux 2.4 NAT HOWTO,
Rusty Russell, Mai 2000

[6] Firewalling unter Linux
http://www.gehrical.de/computer/linux/ipchains

[7] Der Webcache Squid
Michael Manternach
http://www.unix-ag.uni-kl.de/~linux/linuxtag99/squid.html

[8] Linux&Unix / Know-how / Workshop: Desktop-Firewall mit Linux 2.4
http:// www.tecchannel.de/betriebssysteme/751/0.html#, 10.09.2001

[9] Centrum für Produktionstechnik, Universität Kaiserslautern
http://www.uni-kl.de/

Abschlusserklärung

Wir erklären, dass die Projektarbeit selbstständig und ohne fremde Hilfe verfasst wurde und keine anderen als die angegebenen Mittel verwendet wurden.

Wir versichern, dass alle wörtlich und sinngemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht wurden.

Neustadt / Weinstraße, 01. Februar 2002

_____
Sascha Badstübner

_____
Manuel Ecker