
5 Firewall und Masquerading

In diesem Kapitel lernen Sie

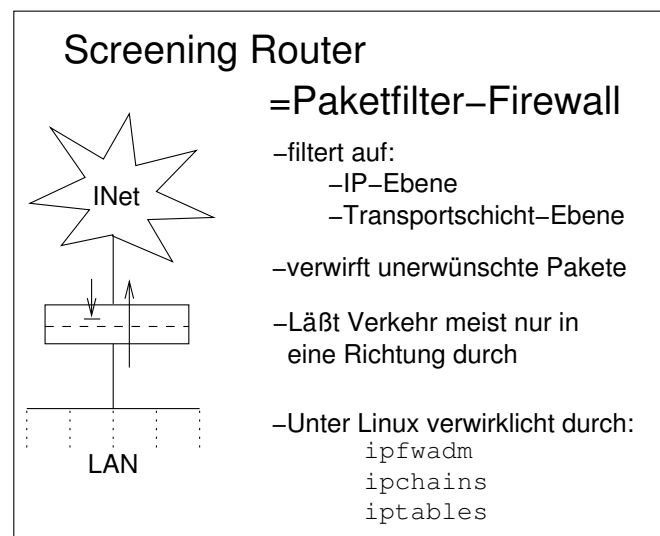
- Ⓛ ▶ verschiedene Firewall-Architekturen kennen (LPI 1: 110.1).
- ▶ den Paketfilter **ipchains** kennen.
- ▶ den Paketfilter **iptables** kennen.
- ▶ eine Beispiel-Firewall-Konfiguration kennen.
- Ⓛ ▶ die Kommandos **nmap** und **netstat** kennen (LPI 1: 110.1).

5.1 Übersicht über verschiedene Firewall-Architekturen

Um eine Firewall zu bauen, stehen im Wesentlichen zwei Hauptinstrumente zur Verfügung:

5.1.1 Screening Router oder Paketfilter

Ist ein Rechner oder ein Router, der sich auf IP-Ebene und Transportschichtebene die Header aller durchlaufenden Pakete anschaut, und anhand der Quell- und Ziel-IP-Adresse und des Quell- und Zielports, sowie anhand von Protokoll, Interface und Richtung des Verbindungsaufbaus entscheidet, ob ein Paket verworfen wird, oder nicht.

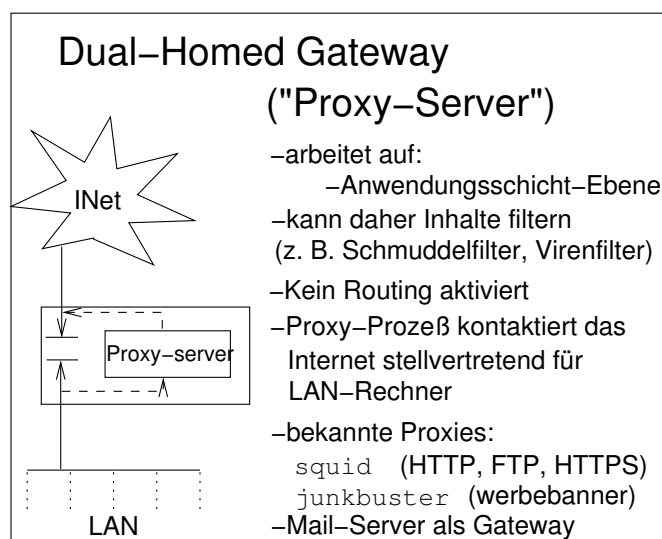


Firewall und Masquerading

Ziel eines Screening Routers ist es, alle unerwünschten Pakete auszusondern. Meist wird dieser so konfiguriert, dass er zwar Verbindungen vom Intranet ins Internet erlaubt, aber Verbindungen vom Internet ins Intranet nur auf wenige Rechner oder Dienste zulässt, etwa einem öffentlichen Web-Server.

5.1.2 Dual-Homed-Gateway oder Proxy-Server

Da auch viele Angriffe oder unerwünschte Daten sich *innerhalb* des Anwendungsschicht-Protokolls abspielen, ist dafür ein Paketfilter ungenügend. Man deaktiviert daher das IP-Forwarding und stattdessen müssen sich nun alle LAN-Client-Prozesse mit dem Proxyserver auf dem Gatewayrechner verbinden. Dieser wiederum verbindet stellvertretend für die LAN-Rechner mit den Rechnern im Internet.



Dadurch hat der Proxyserver nun die volle Kontrolle über alle durchfließenden Daten. Ferner kann er noch viel feinere Zugriffsmechanismen verfügen, zum Beispiel das Filtern von Meta-Tags im HTML-Protokoll (verbotene Inhalte filtern) oder die Gewährung von Zugriff auf Tageszeitbasis für bestimmte Hosts. Meist hat ein Proxy-Server auch noch einen eingebauten Cache, der den Zugriff aufs Web für häufig benutzte Seiten drastisch beschleunigt.

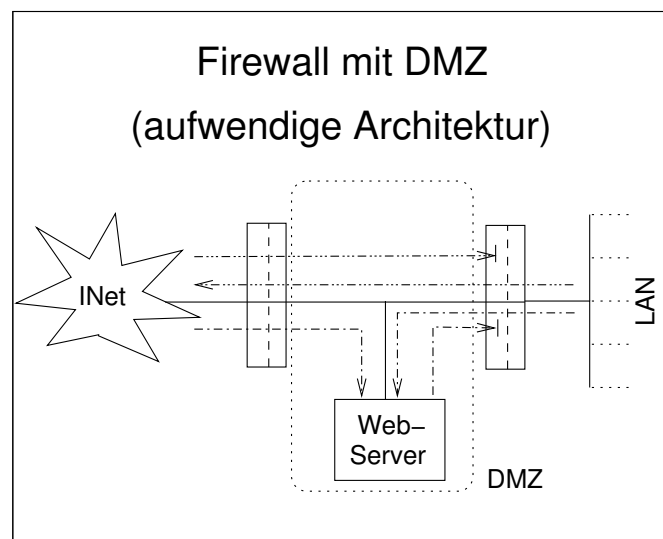
5.1.3 Kombinierte Architektur mit demilitarisierter Zone

Die obigen, simplen Architekturen haben einen gravierenden Nachteil: Ist der Firewall Rechner gekapert worden, so ist das lokale Netz schutzlos.

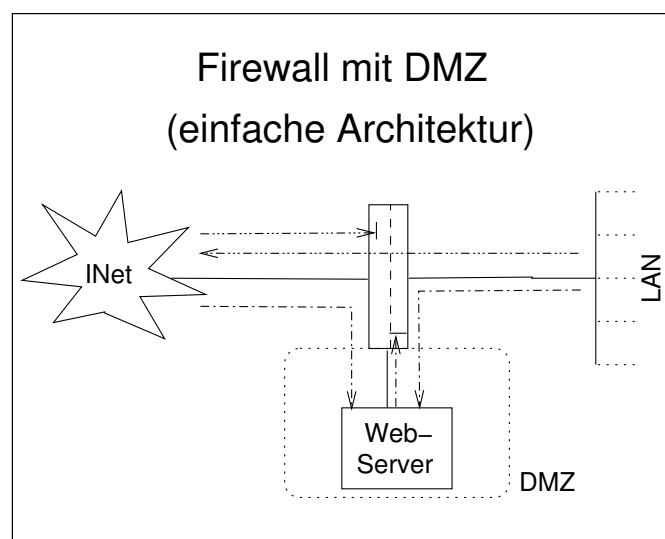
5.1 Übersicht über verschiedene Firewall-Architekturen

Dies spielt besonders eine Rolle, wenn öffentliche Dienste im Internet angeboten werden sollen, man also vom Internet aus auf gewisse Server zugreifen muss. Würden diese Server im lokalen Netz stehen, so könnte man wieder ans lokale Netz dadurch, dass man in den Server einbricht, denn der Verkehr zum Server muss ja gestattet sein.

Deswegen stellt man Server, die öffentliche Internet-Dienste anbieten in ein eigenes Physikalisches Netz, die so genannte *Demilitarisierte Zone*. Durch Paketfilterregeln schottet man die demilitarisierte Zone gegenüber dem Intranet ab. Damit kann ein Angreifer sich nur innerhalb dieser demilitarisierten Zone bewegen, hat jedoch auf das LAN keinen Zugriff.



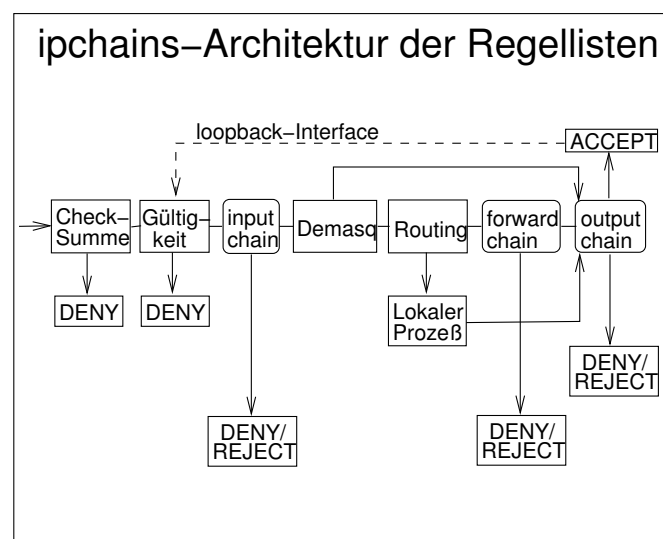
Diese Architektur ist aber sehr teuer, denn es stehen müssen gar zwei Paketfilter eingesetzt werden, von den notwendigen Hubs, Switches und Kabeln ganz zu schweigen. Daher verwendet man auch folgende einfachere (aber auch nicht mehr so sichere) Architektur. Dabei muss in den Screening-Router für die DMZ eine dritte Netzwerkkarte eingebaut werden:



5.2 Paketfilter mit `ipchains`

Um einen Paketfilter konfigurieren zu können, werden wir einen Überblick über Funktionsweise und Benutzung von `ipchains` gewinnen, das unter Linux für Masquerading und Paketfilterung zuständig ist. `ipchains` ist ein Shell-Kommando, das direkt mit dem Kernel kommuniziert. Ab Kernel 2.4 ist auch das neuere `iptables` verfügbar. Auch `ipchains` lässt sich ab Kernel 2.4 weiterverwenden, jedoch kann es nicht gleichzeitig mit `iptables` eingesetzt werden, so dass man in diesem Falle auf dessen neue Funktionen verzichten muss.

Der Name dieser Tools rührt daher, dass Regellisten – eben die so genannten *Chains* – der Reihe nach abgearbeitet werden:



Ein Paket durchläuft ipchains von links nach rechts:

Checksumme Zuerst wird überprüft, ob das Paket keine Übertragungsfehler enthält. Ist das Paket fehlerhaft, so wird es sang- und klanglos weggeworfen, ohne dem Absender davon etwas mitzuteilen (*DENY*).

Gültigkeit Manche verunstalteten Pakete bringen auch ipchains durcheinander, deswegen wollen wir auch diese wegwerfen (*DENY*).

Input Chain Diese erste Regelliste testet ein Paket, das gerade zu irgend einer Netzwerkschnittstelle herein gekommen ist. Je nach Regel kann dieses entweder weggeworfen (*DENY*), zurückgewiesen (*REJECT*), auf einen anderen Port umgeleitet (*REDIRECT*) oder angenommen werden (*ACCEPT*).

REJECT macht prinzipiell das selbe wie *DENY*, nur dass der Absender benachrichtigt wird. Normalerweise wird man nur vertrauten Hosts gegenüber so freundlich sein, denn für einen Cracker kann diese Antwort eine wertvolle Information sein. Überdies lässt sich der Host damit fast lahm legen, indem man einfach massenweise Pakete an diesen Port schickt; unser Rechner muss dann brav antworten, was durchaus einen Großteil der System- und Netzwerkressourcen konsumieren kann. Dieses nicht gerade freundliche Verfahren nennt man einen *denial of Service-Angriff*. Täuscht der Absender auch noch eine falsche IP-Adresse vor, so bekommt die Antwortkaskaden möglicherweise irgend ein anderer unschuldiger Rechner auf dem Internet ab, was dann einen *smurf-Angriff* darstellt. Zu derartigen Angriffen und anderen Unliebsamkeiten lese man die Web-Seite der Internet-Feuerwehr *CERT(Computer Emergency Response Team)*, <http://www.cert.org>.

Firewall und Masquerading

ACCEPT nimmt das Paket an und leitet es zur nächsten Instanz weiter.

REDIRECT leitet ein Paket, das eigentlich wo anders hin wollte, auf einen anderen Port des eigenen Rechners um. Dies ist praktisch, wenn man zum Beispiel allen Web-Verkehr durch einen Proxy-Server leiten will, ohne dass die Benutzer davon etwas merken.

Demasq Haben wir nur eine offizielle IP-Adresse zur Verfügung und wollen aber ein ganzes Netz ans Internet bringen, so kann unser Firewall-Rechner Adressübersetzung (*Masquerading*) durchführen. Dabei bekommen alle ins Internet gehenden Pakete die Absenderadresse der Firewall verpasst; gleichzeitig wird auch der Quellport verändert. Trifft nun eine Antwort auf ein solches Paket ein, erkennt die Firewall an der Portnummer des Pakets, an welchen Rechner im lokalen Netz dieses geschickt werden soll.

Dieses Vorgehen hat einige köstliche Vorteile:

- Der Rechner im lokalen Netz merkt davon überhaupt nichts, als Konfiguration ist lediglich die Firewall als Standard-Gateway einzustellen.
- Nach außen hin ist das lokale Netz vollkommen unsichtbar, denn scheinbar verschickt und empfängt nur der Firewall-Rechner Pakete. Dieser arbeitet dadurch als eine Art Proxy(Stellvertreter).
- Weil nur eine IP-Adresse verwendet wird, und sich mehrere Rechner einen gemeinsamen Internetzugang teilen, werden Kosten gespart.
- Auf dem Firewall-Rechner ist es sehr einfach zu bewerkstelligen.

Routing Hier entscheidet sich, wohin das Paket letztendlich geschickt wird. Dies kann entweder ein lokaler Prozess sein oder irgendein anderer Rechner. Im letzteren Falle wird das Paket an die `forward chain` weitergereicht. *Sendet* ein lokaler Prozess Pakete, so verlassen diese direkt über die `output chain` den Rechner.

Forward Chain Diese Regelliste überprüft alle Pakete, die versuchen, den Rechner von Schnittstelle zu Schnittstelle zu durchqueren. Dies ist *nur* möglich, wenn man vorher IP-Forwarding aktiviert hat. Dazu dient folgende Kommandozeile:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

Ist das IP-Forwarding deaktiviert, können keine Pakete vom Internet ins lokale Netz und umgekehrt. Hat sich jedoch im Firewall-Rechner ein ungebetener Gast eingenistet, so steht das lokale Netz offen.

Output Chain Jedes Paket, dass den Rechner durch irgendeine Netzwerkschnittstelle verlässt, muss letztendlich durch diese Regelliste laufen.

5.3 ipchains-Handwerkszeug für Einsteiger

Loopback Interface Spricht die Firewall mit sich selbst, so tut sie das über das loopback-interface `lo` mit der IP-Adresse `127.0.0.1`. Das loopback-Interface wirkt wie eine Drahtschleife: Kaum hat ein Paket die output chain verlassen, so wird es wieder an den Anfang der Kette geleitet.

Diese Konstruktion sieht auf den ersten Blick kompliziert aus, aber für die Praxis genügt es, zu wissen, welche Aufgaben die input-, output- und forward-chain erfüllen.

5.3 ipchains-Handwerkszeug für Einsteiger

Zur Verwaltung der Regellisten dient der Befehl `ipchains`.

5.3.1 Benutzerdefinierte Regellisten

Um die eingebauten Regellisten übersichtlicher zu gestalten, lassen sich benutzerdefinierte Regellisten einrichten. Die Namen dieser Chains dürfen bis zu acht Zeichen lang sein und sollen in Kleinbuchstaben gehalten werden. Ihre Aufgabe ist am ehesten mit *Unterprogrammen* in einer Programmiersprache vergleichbar.

-N chain Eine neue Regelliste erzeugen

-X [chain] löscht eine *leere*, benutzerdefinierte Regelliste. Die eingebauten Regellisten können nicht gelöscht werden.