



Grundlagen Firewall und NAT

- Was sind die Aufgaben einer Firewall?
- Welche Anforderungen sind zu definieren?
- Grundlegende Funktionsweise
- Technische Varianten
- NA[P]T
- Portmapping
- Übungsaufgabe



Quellen im WWW

- <http://www.netfilter.org>
- <http://iptables-tutorial.frozentux.net/iptables-tutorial.html>
- <http://www.tecChannel.de>
- <http://www.harry.homelinux.org/index.php>
- ... selber suchen! ...



Aufgaben einer Firewall

- Kontrolle des Netzwerkverkehrs zwischen einem „geschützten“ und einem „öffentlichen“ Datennetz
- Geschütztes oder inneres Netz ist oft identisch mit dem LAN oder Intranet
- Öffentliches oder äusseres Netz ist zumeist das Internet
- Bestimmte Dienste, Computer und / oder Benutzer werden „reguliert“



Fragen zur Einrichtung einer Firewall

- Welche Internet-Dienste dürfen vom Intranet aus genutzt werden?
- Welche Dienste darf die ganze Welt in Anspruch nehmen?
- Welche Dienste dürfen ausgewählte Benutzer oder Computer in Anspruch nehmen?
- Welche Dienste dürfen nur lokal genutzt werden?



Grundlegende Funktionsweise

- Zwischen den Teilnehmern im Netz werden Pakete ausgetauscht (IP: TCP, UDP, ICMP)
- Jedes IP-Paket hat eine „Source Address“ und eine „Destination Address“ sowie einen Protokoll-Typ (z.B. ICMP, TCP, UDP).
- Jedes TCP oder UDP Paket hat einen Source- und einen Destination-Port.
- ICMP Pakete haben einen Nachrichtentyp, der den „Dienst“ beschreibt.



Paketfilterung

- Aufgrund der IP-Adressen, der Portnummern, Flags und dem Verbindungs-Status kann bei jedem Paket entschieden werden, wie damit zu verfahren ist.
- Ausserdem muss unterschieden werden, welches Netz (angeschlossen über ein „Interface“) das innere und welches das äussere Netz und ggf. eine DMZ ist.
- Daraus kann man dann Regeln zur Filterung der Pakete ableiten.



Kriterien für Paketfilterung

- Netzwerkschnittstelle (innen / aussen),
- IP-Adressen (Quelle / Ziel),
- Portnummern bei TCP und UDP,
- Nachrichtentypen bei ICMP,
- SYN- und ACK-Flags im TCP-Header,
- Flussrichtung (ein- oder ausgehend).



Filter-Regeln (1)

- Wir betrachten hier Linux, siehe: www.netfilter.org
- netfilter kann Pakete filtern (sic!), manipulieren (*mangle*) und NAT/Masquerading betreiben.
- Aufgrund der vorgenannten Kriterien wird eine Menge von Regeln definiert.
- Diese werden als Regel-Listen (*chains*) im Linux-Kernel hinterlegt, die *chains* sind wiederum in Tabellen (*tables*) organisiert.



Filter-Regeln (2)

- Bei jedem IP-Paket wird bis zur ICMP-, TCP- und UDP-Ebene das Regelwerk im Kernel durchlaufen, bis eine passende Regel gefunden wird.
- Daraufhin wird das Paket durchgelassen oder verworfen („Policy“, siehe unten).
- Merke: Der Kernel macht die Arbeit, zur Regel-Eingabe benötigt man ein User-Interface!
- Auf shell-Ebene heißt dieses `iptables` gilt seit Kernel-Version 2.4 heute als Standard.



Filter-Policies

- Die Verhaltensweise, die aus einer Filter-Regel resultiert, heisst „Policy“ oder „Target“.
- Die wichtigsten Policies sind
 - „ACCEPT“ (annehmen)
 - „DROP“ (ablehnen und wegwerfen)
- Eine weitere Policy namens „REJECT“ lehnt das Paket ab und sendet es mit Fehlermeldung an die Quelle zurück.



Tables und Chains (1)

- Es gibt 3 Tables (*mangle*, *nat*, *filter*), die wiederum jeweils mehrere Chains enthalten.
- Die *filter* Tabelle enthält die Regel-Listen für die eigentliche Paketfilterung.
- Die Tabelle *nat* dient der NAT-Funktion, *mangle* der Manipulation von Paketen.
- Man unterscheidet für den Firewall-Rechner bestimmte Pakete (INPUT), von ihm selbst zu versendende (OUTPUT) und durchzuleitende (FORWARD) Pakete.
- Es gibt dazu je eine Chain in *filter*, sowie noch Chains für PREROUTING und POSTROUTING (*mangle*, *nat*).

Tables und Chains (2)

- Die Pakete durchlaufen je nach Source und Destination auf verschiedene Weise die Tables und Chains.
- Typische Reihenfolge der Chains:
 - PREROUTING – INPUT (-> Firewall)
 - OUTPUT – POSTROUTING (Firewall ->)
 - PREROUTING - FORWARD – POSTROUTING
(für durchzuleitende Pakete)
- Einzelheiten dazu siehe <http://iptables-tutorial.frozentux.net/iptables-tutorial.html>

Umgang mit den Regeln

- Die Regeln werden über Linux-Kommandos an den Kernel geschickt, zweckmässigerweise durch eine *Skript-Datei* mit `iptables`-Kommandos.
- Der Kernel speichert sie in der entsprechenden internen Struktur ab.
- Ansehen kann man sie mit `iptables -L`
- Tools: `yast` und `fwbuilder` (fortgeschritten)
- `iptables`-Generator: www.harry.homelinux.org



Firewall-Skript

- Kommando zum Handling der Regeln:
 - `Iptables <options> <functions>`
 - z.B.: `iptables --list`
- Da es hunderte von Regeln geben kann, schreibt man sie wie ein Programm in eine Skript-Datei.
- Siehe: `/etc/init.d/SuSEfirewall` oder ein entsprechendes eigenes Skript dort.



Network Address Translation

- ... oder auch NAT oder Masquerading.
- Dient dazu, ein ganzes Netzwerk aus internen (meist privaten) Adressen hinter einer einzigen öffentlichen IP-Adresse zu betreiben.
- Eigentlich eine Router-Eigenschaft, unter Linux aber in ipchains / iptables integriert.
- Funktioniert zusammen mit Port-Mapping.
- NAPT = Network Address Port Translation

NAT und Portmapping

- Beim Verbindungsaufbau von innen nach aussen legt der Router eine Tabelle an:
 - $IP_1/Port_1 \leftrightarrow IP_2/Port_2$
 - Dabei bedeutet Index 1 innen, Index 2 aussen.
 - Port 2 wird neu angelegt und dient als Verbindungs-Port zur äusseren IP-Adresse.
- Portmapping (oder Port Forwarding) definiert, was mit Verbindungswünschen geschieht, die von aussen für einen bestimmten Port eingeht.

Beispiel NAT

- IP 10.0.0.1 will mit Port 80 an 193.175.21.160 Verbindung aufnehmen. Der Verbindungswunsch wird von einem NAT-Router „abgefangen“.
- Der NAT-Router allokiert einen neuen Port an seiner eigenen Aussen-Adresse.
- Der NAT-Router trägt „10.0.0.1:xx/193.175.21.160:80“ unter der neuen Portnummer in seine NAT-Tabelle ein.
- Von dem neuen Port aus nimmt er Verbindung mit dem Ziel auf. Alle *eingehenden* Daten auf diesem Port werden an 10.0.0.1 geleitet.
- Für 10.0.0.2:yy -> 193.175.21.160:80 wird ein anderer „neuer Port“ an der Aussenadresse des NAT-Routers benutzt.



Aufgabenstellung

- Firewall einrichten mit yast2
- Regeln einrichten soweit mit yast möglich
- NAT einrichten
- Firewall aktivieren
- Iptables --list analysieren
- Skriptfile finden
- Skriptfile analysieren