



Erstellung von Regelsätzen für Paketfilter

Marco Thorbrügge

DFN-CERT GmbH
Oberstraße 14b
D-20144 Hamburg
thorbruegge@cert.dfn.de

Zusammenfassung Paketfilter werden zwischen (Teil-)Netzwerken mit unterschiedlichen Sicherheitsanforderungen eingesetzt, um anhand von Regeln unerwünschten Datenverkehr herauszufiltern. Vor dem Einsatz eines Paketfilters muß ein maßgeschneiderter, genau auf die Sicherheitsbedürfnisse des zu schützenden Netzwerks angepaßter Satz von Paketfilterregeln implementiert werden. Nach einer kurzen Übersicht über TCP/IP beschreibt der folgende Beitrag diesen Vorgang anhand von Fallstudien. Zur Veranschaulichung werden die Beschreibungen durch ihre Umsetzung in die Regelsyntax von Netfilter/iptables für Linux ergänzt. Abschließend wird die Kommunikationsmatrix als ein Werkzeug für die Erstellung von Paketfilter-Regeln vorgestellt.

1 Einleitung

Paketfilter (oder auch „Packet Screening Devices“) werden eingesetzt, um erwünschten und unerwünschten Netzwerkverkehr auf Ebene der Transport- und Netzwerkprotokolle zu trennen. Paketfilter stellen gleichsam ein Sieb dar: die richtige Konfiguration der Filterregeln vorausgesetzt, passieren erwünschte Datenpakete den Filter und unerwünschte werden verworfen.

Für die richtige Konfiguration eines Paketfilters läßt sich leider kein allgemeingültiges „Kochrezept“ aufstellen, zu unterschiedlich sind Netzwerktopologien, Strukturen und der Sicherheitsanspruch. Dennoch gibt es einige Grundregeln, die jeder Administrator eines Paketfilters in Erwägung ziehen sollte.

Der vorliegende Beitrag versucht, diese Grundregeln anhand von Beispielen plausibel zu machen. Die Umsetzung der Regeln für Linux Netfilter/iptables soll Anregung für eigene Experimente mit dem Open-Source Paketfilter bieten.

2 Die Internet Protokolle

Die meisten der im Internet eingesetzten Dienste wie EMail (SMTP) oder Web (HTTP) können mit Hilfe der TCP/IP Protocol Suite (siehe Abbildung 1) auf verschiedenen Ebenen abstrahiert werden. Scheinbar kontinuierliche Datenströme auf der obersten Schicht (Anwendungsschicht) stellen sich auf darunterliegenden Schichten als eine Aneinanderreihung von Datenpaketen dar.

Beim Durchlaufen der Schichten fügt jede Schicht weitere Informationen in Form sogenannter Header hinzu, welche dem Datenpaket der darüberliegenden Schicht vorangestellt



werden. So wird einem TCP-Datenpaket der Transportschicht mit TCP-Header und Datenteil auf der darunter liegenden Schicht ein weiterer Header, der IP-Header, vorangestellt.

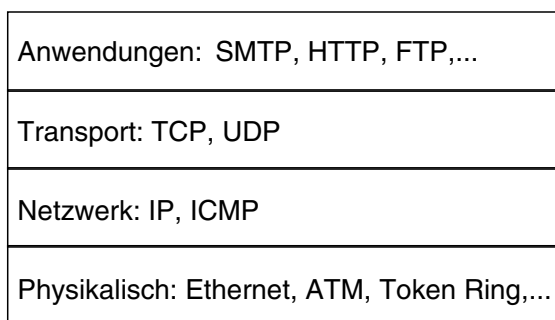


Abbildung 1: Die TCP/IP Protocol Suite

Beim Eintreffen auf dem Zielhost durchwandern die Datenpakete die Schichten in umgekehrter Reihenfolge, wobei auf jeder Schicht der zugehörige Header ausgewertet, entfernt und das resultierende Paket an die darüberliegende Schicht entsprechend der Informationen aus dem Header weitergereicht wird. Im letzten Schritt wird der Datenstrom auf dem Zielhost wieder zusammengesetzt.

Die wichtigsten Protokolle sind TCP (Transmission Control Protocol), UDP (User Datagram Protocol), ICMP (Internet Control Message Protocol) und IP (Internet Protocol).

Eine umfassende Darstellung der hier nur kurz umrissenen Protokolle bietet [Ste94].

2.1 TCP - Transmission Control Protocol

TCP arbeitet verbindungsorientiert und führt zu diesem Zweck Sequenz- und Acknowledge-Nummern im Header mit. So kann auf dem Zielhost der Verlust von Paketen bemerkt und die richtige Reihenfolge ermittelt werden. Um eine Verbindung von einem Client zu einem Server eindeutig identifizieren zu können, verwendet das TCP-Protokoll sogenannte Portnummern. Anhand von Source- und Destination Portnummer ordnen beide Seiten TCP-Pakete einem Dienst (SMTP) bzw. einer Anwendung (Mailclient) zu. TCP-Header haben noch eine Reihe weiterer Parameter wie Flags, die Informationen über den Verbindungszustand beinhalten. Reine TCP-Dienste sind zum Beispiel HTTP, SMTP oder SSH.

Typische Paketfilter-Regeln betrachten Source- und Destination-Port sowie die TCP-Flags.

2.2 UDP - User Datagram Protocol

UDP arbeitet verbindungslos, es fehlen Sequenz- und Acknowledge-Nummern. Der Verlust von Paketen kann nicht ohne weiteres festgestellt werden. Die Dienstzugehörigkeit wird bei UDP ebenfalls anhand von Portnummern festgelegt. Reine UDP-Dienste sind

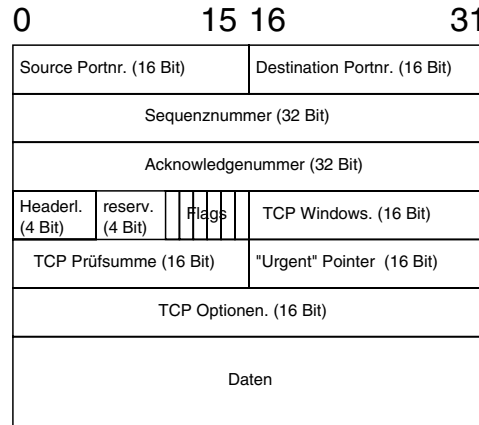


Abbildung 2: TCP Header

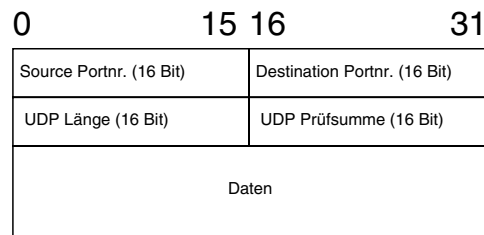


Abbildung 3: UDP Header

zum Beispiel SYSLOG oder BOOTP. Einige Dienste wie DNS nutzen situationsabhängig beide Protokolle.

Typische Paketfilter-Regeln betrachten Source- und Destination-Port.

2.3 ICMP - Internet Control Message Protocol

ICMP arbeitet ebenfalls verbindungslos und dient hauptsächlich der Übermittlung von Steuer- und Administrationsinformationen. Ein ICMP-Paket ist, je nach Service-Typ (`echo-request`, `echo-reply`, etc.) anders aufgebaut.

Typische Paketfilter-Regeln betrachten den Service-Typ.

2.4 IP - Internet Protocol

Unterhalb der Transportprotokolle liegt das IP-Protokoll. Der IP-Header enthält unter anderem Informationen über die IP-Adressen von Quell- und Zielhost.

Typische Paketfilter-Regeln betrachten Source- und Destination-IP-Adresse.

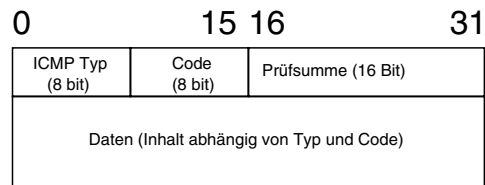


Abbildung 4: ICMP Message

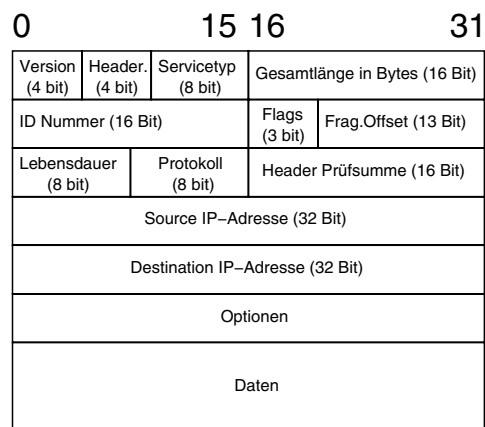


Abbildung 5: IP Header

3 Paketfilter oder Firewall?

Paketfilter werden im Sprachgebrauch häufig als Firewalls bezeichnet, was im Allgemeinen zu verschiedenen Mißverständnissen und Verwechslungen führt.

Paketfilter sind Einrichtungen, die die Header-Informationen eines Datenpaketes auswerten und anhand der gewonnenen Informationen regelbasiert Entscheidungen über die weitere Handhabung dieses Paketes treffen können. Im Allgemeinen bestehen die Optionen für einen Paketfilter aus Weiterleiten eines Paketes an den Zielhost bzw. den nächstgelegenen Routing-Point und dem Verwerfen eines Paketes.

Eine Firewall hingegen ist ein komplettes Sicherheitskonzept, das meist einen oder mehrere Paketfilter enthält. Zu einer Firewall gehören allerdings noch andere Komponenten wie Demilitarisierte Zonen (DMZ), Bastion Hosts, Proxies und Intrusion-Detection Systeme (IDS). Als DMZ wird meist ein von Paketfiltern geschütztes Subnetz bezeichnet, in welchem die Systeme untergebracht sind, die Dienste für das Extranet anbieten (vgl. Abbildung 6). Eine weiterführende Erläuterung von Firewall-Konzepten und ausführliche Beschreibungen der Begriffe bietet [ZCC00].

Im weiteren Verlauf bezeichnet der Begriff Paketfilter ein auf Basis von TCP, UDP, ICMP und IP filterndes Gerät. Hauptsächlich für die Filterung herangezogen werden Quell- und

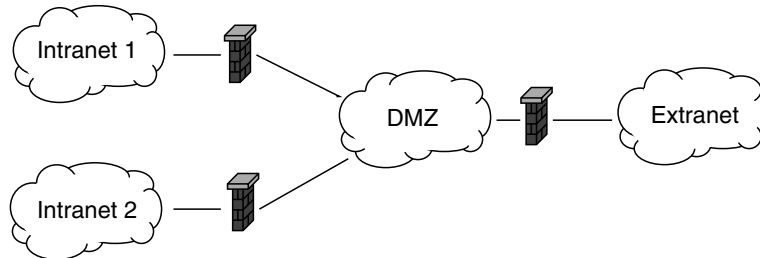


Abbildung 6: Platzierung von Paketfiltern

Zieladresse, Quell- und Ziel-Portnummer sowie die TCP-Flags. Prinzipiell können moderne Paketfilter aber auch die übrigen Headerfelder der Protokolle auswerten.

3.1 Stateful und Non-Stateful Paketfilter

Paketfilter teilen sich grob in 2 Kategorien: Stateful und Non-Stateful. Während Non-Stateful Paketfilter einzelne Pakete unabhängig von ihrem Kontext (also der Verbindung, zu der sie gehören) betrachten, haben Stateful Paketfilter eine Art Gedächtnis. Vergangene Ereignisse beeinflussen also zukünftige Entscheidung über die Handhabung von Datenpaketen durch den Stateful Paketfilter.

In der Praxis führt der Stateful Paketfilter eine interne Tabelle über alle aktiven (ESTABLISHED) Verbindungen. Um beispielsweise eine TCP-Verbindung in den Zustand ESTABLISHED zu versetzen, muss der 3-Way-Handshake zwischen 2 Hosts ordnungsgemäß abgeschlossen sein (siehe Abbildung 7). Pakete, die zu einer solchen Verbindung gehören, können von denjenigen, die keiner Verbindung zuordenbar sind, unterschieden werden und entsprechend anders gehandhabt werden.

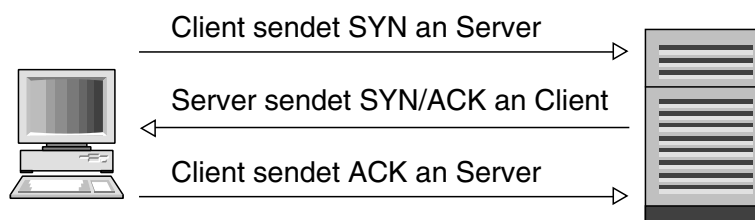


Abbildung 7: TCP 3-Way-Handshake

Filterregeln für Stateful Paketfilter können sehr viel effizienter, eleganter und engmaschiger erstellt werden, als für Non-Stateful Paketfiltern.

Kompliziertere Protokolle wie FTP, welches zwei Kommunikationskanäle (Kontroll- und Datenkanal) benutzt, können nur mittels stateful filtering sicher durch einen Paketfilter

geschleust werden. Nur durch die Information über eine gültige Verbindung auf dem Kontrollkanal (Port 21) zwischen 2 Hosts kann der Filter auf die Rechtmäßigkeit des Verbindungsaufbaus auf dem Datenkanal (Port 20 bei aktivem FTP) zwischen diesen Hosts schliessen.

4 Netfilter/iptables für Linux

Netfilter und das zugehörige Userspace-Tool iptables ist eine unter der „GNU General Public License“ (GPL, vgl. [gpl91]) frei verfügbare Software für Linux-Kernel ab Version 2.4.x. Die Features von Netfilter/iptables sind:

- Stateful Paketfiltering (durch Connection Tracking)
- verschiedene Arten von Network Address Translation (NAT) wie SNAT, DNAT und Masquerading
- weiterführende Manipulationen an Paketen durch Benutzerprogramme

Mit dem Tool iptables werden die eigentlichen Filterregeln implementiert. Die Syntax folgt dem Schema

```
iptables -<Befehl> <Chain-Name> <Pattern> \
-j <Target>
```

Befehle sind unter anderem **A** (fügt eine neue Regel am Ende des bestehenden Regelsatzes einer Chain ein), **F** (löscht alle Regeln einer Chain), **N** (erstellt eine neue benutzerdefinierte Chain) und **P** (setzt die Policy für eine der System-Chains fest).

Die sogenannten Chains sind Paketkanäle, in die Datenpakete eingeschleust werden, wenn sie bestimmten Parametern genügen (siehe Abbildung 8). Neben der Möglichkeit, benutzerdefinierte Chains zu erzeugen, gibt es eine Reihe von fest vorgegebenen System-Chains. Die wichtigsten für einen Paketfilter sind die Chains

- **FORWARD** (für Pakete, die weitergeroutet werden)
- **INPUT** und **OUTPUT** (für Pakete, die für das Paketfiltersystem selbst bestimmt sind).

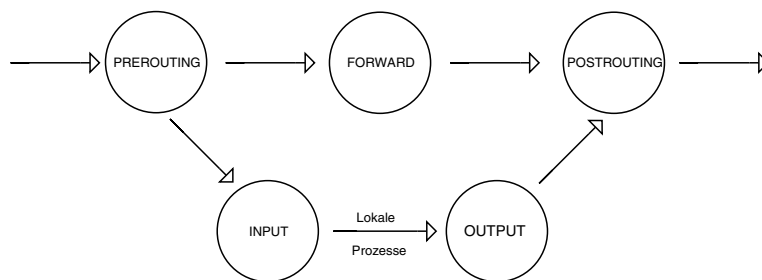


Abbildung 8: Weg eines Datenpaketes durch Netfilter-Chains

Die Pattern sind Eigenschaften eines Datenpaketes, bei deren Vorhandensein ein sogenannter Match ausgelöst und eine festgelegte Aktion ausgeführt wird.



Mittels Target wird die Aktion festgelegt, die bei Eintreten eines Matches ausgeführt wird. Targets können unter Anderem sein:

- **DROP** (das Paket wird verworfen)
- **ACCEPT** (das Paket wird akzeptiert)
- **LOG** (es erfolgt ein Eintrag in die Syslog-Datei)
- der Name einer anderen Chain, in die das Paket zwecks weiterer Bearbeitung wandert.

Netfilter ist im WWW unter [netfilter] erhältlich. Die Seite bietet darüberhinaus einen reichhaltigen Fundus an Dokumentation, FAQs, HOWTOs und Tutorien.

5 Policy: Erlauben oder Verbieten?

Unabhängig von der Auswahl des Paketfilter-Produktes ist vor der Einrichtung und der Inbetriebnahme des Gerätes eine grundlegende Entscheidung zu treffen, nämlich die der zugrundeliegenden Policy. Als Policy werden Richtlinien bezeichnet, die festlegen, wie mit eingehenden und ausgehenden Datenströmen umzugehen ist. Zur Auswahl stehen **DEFAULT ACCEPT** und **DEFAULT DENY**.

5.1 DEFAULT ACCEPT Policy

Mit dieser Policy wird festgelegt, daß grundsätzlich jeder Datenverkehr gestattet ist, es sei denn, er wird explizit durch Filterregeln verboten. Dieser Ansatz hat aus sicherheitstechnischer Sicht eine Reihe von Nachteilen und ist nur solchen Einrichtungen anzuraten, bei denen die Sicherheit eine untergeordnete, dafür die universelle Nutzbarkeit von Internet-Diensten die Hauptrolle spielen. Eine ganze Reihe von akademischen Einrichtungen verwenden diese Policy, indem sie einzelne Dienste (Ports) an ihren Routern sperren und ansonsten keine Paketfilter betreiben.

5.2 DEFAULT DENY Policy

Die aus sicherheitstechnischer und administrativer Sicht zu bevorzugende Policy bestimmt, daß jeglicher Datenverkehr verboten ist, außer er wird explizit freigeschaltet. Diese Policy hat neben der einfacheren Wartbarkeit den Vorteil, daß der Administrator bzw. Netzverantwortliche die vollständige Kontrolle über den ein- und ausgehenden Datenverkehr und die nach außen zur Verfügung gestellten Dienste hat. Viele Einrichtungen betreiben eine abgemildere DENY-Policy, indem sie den eingehenden Datenverkehr filtern, ausgehenden Datenverkehr jedoch nicht reglementieren.

Eines der geschilderten Bedrohungsszenarien stellt den wesentlichen Sicherheitsvorteil der DENY-Policy gegenüber der ACCEPT-Policy in anschaulicher Weise dar.

Bei Linux Netfilter/iptables spiegelt sich die Policy in der Aktion wider, die ausgeführt wird, wenn ein Paket den kompletten Regelstrang durchlaufen hat, ohne das es zu einem Match kam.

Gesetzt wird die Policy für eine bestimmte System-Chain mittels





114 Marco Thorbrügge

```
iptables -P FORWARD DROP
```

bzw.

```
iptables -P FORWARD ACCEPT
```

6 Bedrohungsszenarien

Die Erstellung von effektiven Regelsätzen für Paketfilter ist keine leichte Aufgabe. Die Kommunikationsmatrix (siehe Kapitel 7) ist ein zumindest für kleine und mittelgroße Netzwerke brauchbares Hilfsmittel, um die Kommunikationsbedürfnisse der zu schützenden Systeme zu ermitteln und in Filterregeln umzuwandeln.

Es gibt jedoch eine Reihe von Bedrohungen der Netzwerksicherheit, welche sich in der Kommunikationsmatrix nicht widerspiegeln, denen jedoch trotzdem durch den Einsatz von Paketfiltern begegnet werden kann.

Durch die Analyse von Bedrohungsszenarien werden im Folgenden die dazu notwendigen allgemeinen Filterregeln hergeleitet. Die Beispielszenarien lehnen sich an realen Vorfällen an, die dem DFN-CERT im Rahmen der Vorfallsbearbeitung gemeldet wurden.

6.1 Szenario 1: Der neue Rechner

Ziel

Ziel dieses Szenarios ist es, die Vorteile einer **DEFAULT DENY** Policy gegenüber der **DEFAULT ACCEPT** Policy aufzuzeigen.

Beschreibung

Ein Fachbereich einer Einrichtung hat einen neuen PC-Pool mit 64 PC beschafft. Die erste Veranstaltung soll ein SQL-Praktikum sein, für das Microsoft SQL-Server verwendet wird. In Vorbereitung dieser Veranstaltung installieren 2 Admins des Fachbereichs Windows 2000 Server auf den Systemen. Nach dem ersten Reboot, noch bevor die Systeme fertig konfiguriert und Sicherheitspatches eingespielt werden konnten, stellten die Admins eine sich rapide verschlechternde Antwortzeit der Windows 2000 Serversysteme fest. Ein zusätzlich ins Netz gehängter Sniffer (Gerät zum Protokollieren von Netzverkehr) stellt extrem hohen Datenverkehr zwischen den Systemen fest. Die Systeme waren zur Zeit der Installation bereits ans Netzwerk angeschlossen. Der Fachbereich sowie das netzverantwortliche Rechenzentrum der Einrichtung betreiben eine **DEFAULT ACCEPT** Policy und hat lediglich den RPC-Dienst (TCP Port 111) an den Routern gesperrt.

Analyse

Windows 2000 Server installiert verschiedene Dienste standardmässig, unter anderem den Internet Information Server (IIS), wenn dieser nicht explizit deaktiviert wird. Nach dem



ersten Neustart, noch bevor der Admin die Gelegenheit hatte, Konfigurationseinstellungen vorzunehmen oder Sicherheitsupdates einzuspielen, startet Windows bereits alle installierten Dienste inklusive des IIS. Durch DHCP (Dynamic Host Configuration Protocol) wird dem System eine IP-Adresse zugewiesen. Der Server ist somit voll funktionsfähig und von außen ansprechbar. Wie sich später herausstellte, genügte die wenigen Minuten Betrieb bereits, alle 64 PC des neuen PC Pools mit dem Code Red 2 Wurm zu infizieren (vgl. [Tho01]).

Abgesehen von dem Fehler, ein System während der Installation bereits am Netz zu haben, hätte diese Infektion durch einen Paketfilter mit DEFAULT DENY Policy verhindert werden können. Jeder neu im Netz in Betrieb genommene Host wäre so weder von außerhalb des Intranets ansprechbar gewesen, noch hätte er selber Kontakt zum Extranet aufnehmen können.

6.2 Szenario 2: Der experimentierfreudige Kollege

Ziel

Ziel dieses Szenarios ist, die Notwendigkeit der Egres-Filterung, also der Filterung von Datenpaketen aus dem Intranet, die eine Absendeadresse des Extranets tragen, aufzuzeigen. Egres-Filterung ist eine grundsätzliche Sicherheitsmaßnahme und ist prinzipiell jeder Einrichtung mit Anschluß ans Internet zu empfehlen.

Beschreibung

Ein Mitarbeiter der Verwaltung, durch anfängliche Erfolge bei der Installation von Linux auf seinem Arbeitsplatzrechner ermutigt, wird zunehmend experimentierfreudiger. Kürzlich hat er das Sicherheitstool nmap ([fy0]) entdeckt und beginnt, in seinen Pausen damit zu experimentieren. Das Programm nmap ist ein sehr mächtiger Portscanner. Ein Portscanner versucht, anhand der Antworten auf Datenpakete an einzelne Destination-Ports zu ermitteln, welche Dienste ein System im Netzwerk bereitstellt. Das Tool nmap verfügt unter anderem über eine Option, die eigene IP-Adresse während des Scannens zu verbergen (Decoy). Dazu muß der Benutzer eine Reihe von zufällig gewählten IP-Adressen angeben, in der dann die eigene IP-Adresse versteckt wird. Gestartet wird der Decoy-Scan durch

```
nmap <Opfer-IP> -D [IP, IP, . . . , IP, ]ME[, IP, IP, . . . , IP]
```

wobei ME die eigene IP-Adresse ist. Das Tool prüft das Opfersystem, wobei es ein Paket pro angegebener IP-Adresse versendet. Die Antwortpakete wandern dann an die gefälschten Absendeadressen, die entweder existieren und das Paket verwerfen oder nicht existieren. Nur ein einziges Antwortpaket wird an das System des Scanners zurückgeschickt. Wird der Scan auf dem Opfersystem bemerkt, so kann nur sehr schwer festgestellt werden, welche der vielen Absendeadressen die echte ist.

Der Mitarbeiter erzeugt eine Reihe zufälliger IP-Adressen und prüft damit den Webserver seiner Bank auf offene Ports. Die Einrichtung betreibt keine Egres-Filterung. Somit gelangen alle, auch die gefälschten Pakete, ungefiltert ins Extranet. Selbst wenn die Bank den Scan bemerkt, kann sie nur schwer Rückschlüsse auf den Urheber ziehen.

Analyse

Würde die Einrichtung Egres-Filterung betreiben, so würde lediglich das eine Datenpaket, welches die echte Absendeadresse trägt, zu dem Banksystem durchdringen und der Urheber wäre entlarvt. Selbst wenn der Angreifer so intelligent wäre, nur IP-Adressen seiner eigenen Einrichtung zu fälschen, könnten die Admins der Bank so zumindest das Subnetz des Urhebers feststellen und den Administratoren der Einrichtung entsprechende Informationen zukommen lassen.

Portscans an sich sind im Grunde noch kein Angriff, werden jedoch von vielen Einrichtungen als Maßnahme zur Angriffsvorbereitung angesehen und entsprechend behandelt. Selbst wenn der Urheber keine weiteren Aktionen wie einen Einbruch plant, so erleidet doch die Einrichtung, in der er arbeitet, einen Image-Schaden durch sein Handeln. Egres-Filterung hilft, seine Scan-Aktivitäten aufzudecken und ihn zur Rechenschaft zu ziehen.

Egres-Filterung ist auch sehr hilfreich für CERTs (Computer Emergency Response Teams), die DDoS (Distributed Denial of Service) Angriffe analysieren und verfolgen. DDoS-Agenten wie „Stacheldraht“ (vgl. [CER00]), werden auf kompromittierten Systemen installiert und fälschen während eines Angriffs ähnlich wie nmap die Absendeadresse. Durch Egres-Filterung kann zumindest das Subnetz, aus dem ein Angriff geführt wird, bestimmt und deren Administratoren gewarnt werden.

Mit Netfilter/iptables läßt sich Egres-Filterung folgendermaßen einrichten:

```
iptables -A FORWARD -s ! $intranet \
        -i $INT_INTERFACE -j DROP
```

Die Variablen (beginnend mit \$) müssen entsprechend belegt werden. **\$INT_INTERFACE** bezeichnet den Netzwerkadapter des Paketfilters für das Intranet. Die Variable **\$intranet** beinhaltet die Subnetz-Adresse des Intranets. Mit vorangestelltem Ausrufezeichen erfolgt dann ein Match, wenn die nachfolgende Bedingung nicht erfüllt ist (Negation).

6.3 Szenario 3: Das falsche Intranet

Ziel

Ziel dieses Szenarios ist, die Notwendigkeit der Ingres-Filterung aufzuzeigen, also der Filterung von Datenpaketen aus dem Extranet, die eine Absendeadresse des Intranets tragen.

Beschreibung

Der IRC-Server einer Einrichtung steht unter einem massivem Denial-of-Service-Angriff. Die Analyse des Netzverkehrs durch die Administratoren ergibt, daß der Server von ICMP Echo-Reply Paketen, also Antwortpakete auf das PING-Kommando, geflutet wird. Urheber sind ohne Ausnahme Hosts aus dem Intranet, die scheinbar auf sehr viele hintereinander gesendete PING-Kommandos antworten. Durch die Masse der eintreffenden Echo-Reply Pakete ist der IRC-Server komplett außer Betrieb gesetzt und kann auf keine weiteren Requests antworten. Die Einrichtung betreibt keine Ingres-Filterung, dies bedeutet,

daß auch Datenpakete mit einer Intranet Absendeadresse aus dem Extranet den Paketfilter passieren dürfen werden.

Analyse

Der IRC-Server der Einrichtung steht unter einer sogenannten Smurf-Attacke (vgl. [SMK01] und Abbildung 9). Ein Angreifer aus dem Extranet schickt zu diesem Zweck Echo-Requests Pakete (Pings) per ICMP Broadcast zeitgleich an alle Systeme des Intranets. Der Angreifer fälscht die Absendeadresse und trägt die IP-Adresse des Opfers ein, in diesem Fall die des IRC-Servers. Die Echo-Requests treffen durch den Broadcast gleichzeitig bei allen Systeme des Intranets ein. Diese antworten protokollgemäß mit einem Echo-Reply, senden die Datenpakete jedoch an den gefälschten Absender, den IRC-Server. Dieser wiederum wird durch die Masse der eintreffenden Pakete geflutet und kann auf keine anderen Anfragen mehr antworten.

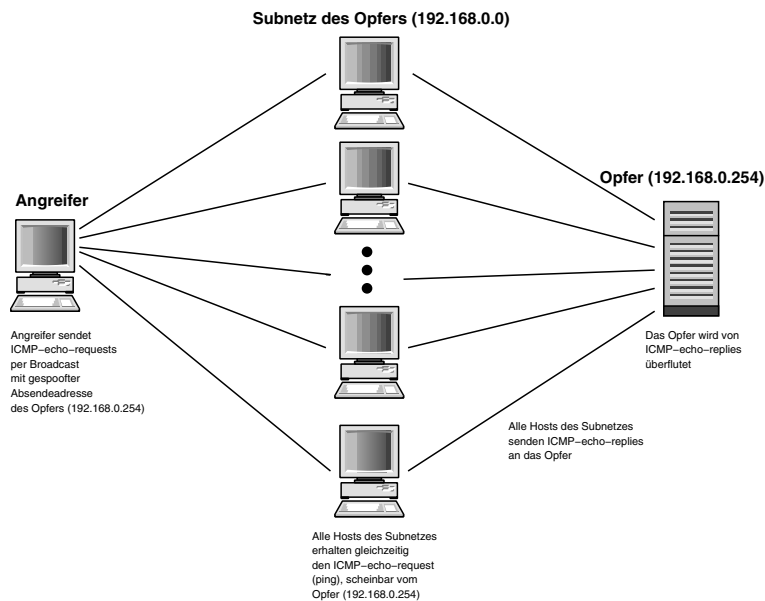


Abbildung 9: Ablauf einer Smurf-Attacke

Smurf-Attacken dieser Art lassen sich durch Filterung von ICMP-Broadcasts am einfachsten verhindern. Es gibt keinen vernünftigen Grund, diese in das Intranet gelangen zu lassen. Genauso gibt es keinen vernünftigen Grund, Pakete mit offensichtlich gefälschter Absendeadresse in das Intranet zu lassen. Es sind Szenarios denkbar, in denen ähnliche Angriffe ohne ICMP-Broadcasts möglich sind. Hinzu kommt, daß diese Pakete, wenn Sie erst ins Intranet gelangt sind, auch von anderen Sicherheitseinrichtungen wie IDS entsprechende behandelt und eventuell nicht erfasst werden.

Mit Netfilter/iptables läßt sich Ingres-Filterung folgendermaßen erreichen:

```
iptables -A FORWARD -s $intranet \  
-i $EXT_INTERFACE -j DROP
```

Auch hier müssen die Variablen entsprechend der lokalen Gegebenheiten belegt werden.

6.4 Szenario 4: Der Netzwerk-Kartograph

Ziel

Ziel dieses Szenarios ist zu zeigen, wie Hacker mit manipulierten Datenpaketen versuchen, Paketfilter auszutricksen, um Informationen über das dahinter liegende Netzwerk zu erlangen.

Beschreibung

Viele automatisierte Einbruchs-Tools überprüfen eine Reihe von mehr oder weniger zufällig ausgewählte Hosts auf das Vorhandensein bestimmter Dienste, die dann im Erfolgsfall angegriffen werden. Solche brachialen Einbruchsversuche fallen in der Regel auf und sind sehr leicht von Intrusion Detection Systemen aufzuspüren (vgl. [NN00]). Den Urhebern dieser Angriffe ist die Entdeckung meist gleichgültig, denn es gibt genügend Einrichtungen, denen die Attacks nicht auffallen.

Sehr viel gewiefter jedoch geht ein Hacker vor, der ein System gezielt angreifen will. Normalerweise läßt sich ein Hacker bei seinem Tun Zeit, denn er will die Gefahr, entdeckt zu werden, minimieren.

Die erste Phase eines gezielten Hackversuchs ist, soviel Informationen wie möglich über die Systeme des Opfers zu sammeln. Ein intelligenter Hacker verwendet für die Aufklärung zuerst legale, unauffällige Mittel wie DNS- und Whois-Abfragen. Erst danach tastet er sich schrittweise weiter an die Systeme heran. So hofft er, weitere, nicht ohne weiteres erhältliche Informationen wie beispielsweise angebotene Dienste auf einzelnen Hosts oder die Versionsnummern der Serverprogramme zu erhalten. Erst wenn der Hacker sich zu 100 Prozent sicher ist, alles über sein Zielsystem zu wissen, startet er den eigentlichen Angriff.

Ein gerne verwendetes Aufklärungsmittel sind sogenannte Stealth-Scans (verborgene Scans), welche neben der Informationsgewinnung das Ziel haben, von IDS und Administrator unentdeckt zu bleiben.

Ein beliebter Stealth-Scan ist der „Slow-Scan“, bei dem zwischen dem Absenden der Scan-Pakete längere Zeit gewartet wird in der Hoffnung, daß etwaige IDS in der Zeit schon wieder vergessen haben, daß bereits früher weitere Pakete vom System des Hackers geschickt worden sind.

Ein anderes Mittel sind Scans mit sogenannten „crafted packets“, also selbst erstellten Datenpaketen. Beliebte sind Manipulationen an den TCP-Flags eines Paketes in der Hoffnung, daß Paketfilter und IDS diese nicht protokollkonformen Pakete ignorieren.



Es gibt sogenannte Christmas-Scans, bei denen alle Flags gesetzt, und NULL-Scans, bei denen keine Flags gesetzt sind. Half-Open-SYN, ACK- und SYN-FIN-Scans sind weitere beliebte Scan-Methoden. Unter anderem beherrscht das Tool nmap ([fy0]) alle diese Scanmethoden und kann darüber hinaus durch sogenanntes „OS-Fingerprinting“ erstaunlich präzise das Betriebssystem auf dem Zielsystem ermitteln.

Diese und weitere Scan-Methoden werden genauer in [NN00] beschrieben.

Analyse

Datenpakete, die offensichtlich keiner bestehenden Verbindung zugeordnet werden können, sind mit Non-Stateful Paketfiltern nur sehr schwer auszufiltern, ohne den übrigen, zulässigen Datenverkehr zu beeinträchtigen. Stealth-Scans und teilweise OS-Fingerprinting funktionieren nur mit Hilfe solcher Datenpakete.

Das beste Mittel, diese Aufklärungstätigkeiten abzublocken, ist der Einsatz eines Stateful-Paketfilters mit einer DEFAULT DENY Policy, denn nur so kann ein Datenpaket auch anhand von früher behandelten Datenpaketen bewertet werden. Darüberhinaus hat es sich bewährt, einige Regeln an den Beginn des Paketfilter Regelsatzes einzufügen, welche gezielt TCP-Pakete mit nicht-protokollkonformen Flag-Kombinationen herausfiltern.

Mit Netfilter/iptables lassen sich solche Pakete folgendermaßen filtern:



```
iptables -A FORWARD -p tcp --tcp-flags ALL NONE \
-j DROP
```



```
iptables -A FORWARD -p tcp --tcp-flags ALL ALL \
-j DROP
```

```
iptables -A FORWARD -p tcp --tcp-flags \
SYN,ACK,FIN,RST SYN,RST -j DROP
```

```
iptables -A FORWARD -p tcp --tcp-flags \
SYN,ACK,FIN,RST SYN,FIN -j DROP
```

```
iptables -A FORWARD -p tcp --tcp-flags \
SYN,ACK,FIN,RST SYN,FIN,RST -j DROP
```

Erlaubte eingehende Verbindungen lassen sich mit Netfilter/iptables durch das STATE-Match realisieren (im Beispiel soll SSH zu einem bestimmten System erlaubt werden):

```
iptables -F FORWARD -p tcp --destination-port 22 \
-m state --state NEW,ESTABLISHED \
-d $host1 -j ACCEPT
```

```
iptables -F FORWARD -p tcp --source-port 22 \
-m state --state ESTABLISHED \
-s $host1 -j ACCEPT
```





7 Die Kommunikationsmatrix

Die **DEFAULT DENY** Policy setzt voraus, daß erwünschter Datenverkehr explizit mit Hilfe einer Filterregel freigeschaltet wird. Eines der Hauptprobleme bei der Erstellung von Filterregeln ist die Übersichtlichkeit, die abnimmt, je mehr Systeme involviert sind. Diese Unübersichtlichkeit wird noch gesteigert, wenn mehrere Paketfilter gestaffelt hintereinander Subnetze mit unterschiedlichen Sicherheitsbedürfnissen voreinander abschirmen.

Für kleine und mittlere Netzwerke ist eine als Kommunikationsmatrix bezeichnete Tabelle ein probates Mittel, die Kommunikation Intranet-Extranet und umgekehrt zu klassifizieren, um so die Erstellung geeigneter Paketfilter-Regeln zu erleichtern.

Im Folgenden wird anhand eines Beispielnetzwerks die Erstellung der Matrix, deren Umsetzung in die Meta- und schließlich in die endgültigen Filterregeln erläutert.

7.1 Das Beispielnetzwerk

Bevor der Administrator sich mit der Konfiguration der Paketfilter befassen kann, muß die zugrundeliegende Struktur des Netzwerks feststehen. Welche Systeme mit welcher IP-Adresse sollen durch einen Paketfilter geschützt werden? Welche Systeme müssen Benutzern aus dem Extranet Zugriff gewähren? Und welche Dienste müssen von diesen Systemen zur Verfügung gestellt werden? Dies sind drei der Fragen, die im Vorfeld zu stellen und zu klären sind.

Eng damit verknüpft ist die Frage, welche Systeme wie zu gruppieren sind, um möglichst einfache Administrationsbedingungen zu schaffen. So ist es beispielsweise sinnvoll, die Client-Systeme der Mitarbeiter einer Einrichtung, die zwar Dienste im Extranet in Anspruch nehmen müssen, jedoch selbst keine Dienste anbieten, in einem Subnetz zusammenzufassen und mit einem Paketfilter zu sichern.

Systeme, die selber Dienste für das Extranet anbieten, kommen in ein eigenes, vorgelagertes Subnetz (Demilitarisierte Zone, DMZ) und werden ebenfalls durch einen Paketfilter gesichert.

Das Beispielnetzwerk besteht aus zwei Client-Systemen (Client 1 und 2), zwei Server-Systemen (Mail- und Webserver) und zwei Paketfiltern. Die Netzwerkverantwortlichen haben sich entschieden, die beiden Server durch einen Paketfilter vor dem Extranet abzuschotten. Durch die Einrichtung eines zweiten Filters, der die Client-Systeme zusätzlich schützt, wird für die Server eine vorgelagerte DMZ geschaffen. Damit wollen die Netzwerkverantwortlichen im Falle der Kompromittierung eines der Server verhindern, daß der Angreifer vom Server aus weitere Angriffe gegen das Intranet verübt.

7.2 Kommunikationsbeziehungen

Die notwendigen Kommunikationsbeziehungen innerhalb des Beispielnetzwerks legen die Netzwerkverantwortlichen gemäß der folgender Abbildung fest.

An Client 1 arbeitet beispielsweise der Webmaster der Einrichtung. Er soll von seinem Arbeitsplatz SSH- und HTTP/HTTPS-Verbindungen zum Webserver aufbauen dürfen.



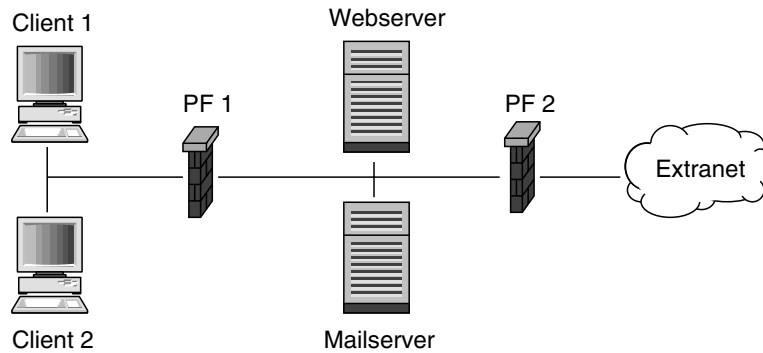


Abbildung 10: Netzwerkplan des Beispielnetzes

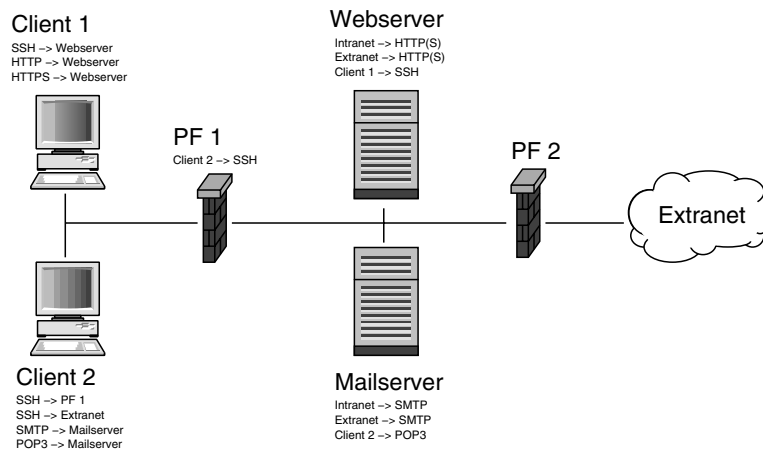


Abbildung 11: Kommunikationsbeziehungen im Beispielnetzwerk

Client 2 ist der Arbeitsplatz des Paketfilter Administrators. Dieser soll SSH-Verbindungen zu Paketfilter 1 aufbauen, Mail über den Mailserver verschicken und empfangen dürfen.

Darüberhinaus sollen beliebige Systeme aus dem Extranet Mail per SMTP an den Mailserver senden und HTTP/HTTPS-Verbindungen zum Webserver aufbauen können.

7.3 Die Kommunikationsmatrix

Die festgelegten Kommunikationsbeziehungen werden in eine Tabelle eingetragen. In den Spalten und den Zeilen sind jeweils die betroffenen Systeme aus Intranet und DMZ, die Paketfilter und das Extranet aufgetragen. Im Schnittpunkt zweier Systeme werden die Kommunikationsbeziehungen, also die Dienste, die von dem einen Host auf dem anderen in Anspruch genommen werden dürfen, eingetragen.

Für die Konfiguration eines einzelnen Paketfilter ist jeweils nur ein Teil der Matrix interessant (in den Abbildungen schraffiert dargestellt).

Die Informationen aus der Matrix können nun in einer Form extrahiert werden, die eine Umsetzung in die Regelsyntax des verwendeten Paketfilter-Produktes leicht machen:

Für Paketfilter 1:

Source	Protocol	Destination
Client 1	SSH, HTTP(S)	Webserver
Client 2	SSH	PF 1
Client 2	SMTP, POP3	Mailserver
Client 2	SSH	Extranet

und für Paketfilter 2:

Source	Protocol	Destination
Client 2	SSH	Extranet
Mailserver	SMTP	Extranet
Extranet	SMTP	Mailserver
Extranet	HTTP(S)	Webserver

Beispielsweise läßt sich die SSH-Verbindung von Client 1 in das Extranet mit Netfilter/iptables wie folgt darstellen:

```
iptables -A FORWARD -p tcp --destination-port 22 \
```



```

-s $client1 \
-m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -p tcp --source-port 22 \
-d $client1 \
-m state --state ESTABLISHED -j ACCEPT

```

8 Zusammenfassung

Die Erfahrung zeigt, daß Paketfilter mit einer DEFAULT DENY Policy (Kommunikation zwischen Systemen ist untersagt, wenn nicht explizit erlaubt) sicherer und einfacher zu konfigurieren sind als mit DEFAULT ACCEPT Policy (jede Kommunikation ist erlaubt, wenn nicht explizit verboten).

Die Erfahrung zeigt ebenfalls, daß mit Stateful Paketfiltern, die Informationen über den Zustand von Verbindungen besitzen, sehr viel engmaschigere Filterregeln möglich sind, als mit Non-Stateful Paketfiltern.

Kommunikationsbeziehungen zwischen einzelnen Systemen oder Systemgruppen können in kleinen und mittleren Netzwerken mit Hilfe einer Kommunikationsmatrix erfaßt und in Paketfilterregeln umgesetzt werden.

Einige Bedrohungen der Netzwerksicherheit spiegeln sich nicht in den Regeln wider, die mit Hilfe der Kommunikationsmatrix erstellt wurden, können aber trotzdem durch Paketfilter beseitigt werden. Dazu gehören Attacken mit gespoofen IP-Adressen (Smurf), Portscans mit gespoofen IP-Adressen und Slow-/Stealth-Scans. Besonders hervorzuheben ist die Wichtigkeit von Egres-Filterung. Durch das Filtern ausgehender Datenpakete, die keine Absendeadresse des Intranets tragen, sind Angriffe mit gefälschtem Absender sehr viel leichter zu verfolgen, was nicht nur die Arbeit von Computer Notfallteams erleichtert.

Literatur

- [CER00] DFN CERT. Distributed Denial of Service Angriffe. Informationsbulletin, 2000. <http://www.cert.dfn.de/infoserv/dib/dib-2000-01.html>.
- [fy0] Nmap Homepage. <http://www.insecure.org>.
- [gpl91] GNU general public license, 1991. <http://www.gnu.org/licenses/licenses.html#GPL>.
- [netfilter] Netfilter Homepage. <http://netfilter.samba.org>.
- [NN00] Stephen Northcutt and Judy Novak. *Network Intrusion Detection - An Analyst's Handbook*. New Riders Publishing, 2. edition, 2000.
- [SMK01] Joel Scambray, Stuart McClure, and George Kurtz. *Hacking exposed*, page 489 ff. Osborne/McGraw-Hill, 2. edition, 2001.
- [Ste94] W.Richard Stevens. *TCP/IP Illustrated*, volume 1. Addison-Wesley, 1994.
- [Tho01] Marco Thorbrügge. CodeRed - Analyse und Gegenmaßnahmen. Vortrag auf der 35. DFN Betriebstagung (Vortragsfolien), 2001. <http://www.cert.dfn.de/dfn/bt/2001/bt-2001-codered.pdf.gz>.
- [ZCC00] Elizabeth D. Zwicky, Simon Cooper, and D. Brent Chapman. *Building Internet Firewalls*. O'Reilly, 2. edition, 2000.