

# **Firewalls**

**Seminararbeit zur Veranstaltung 18.415:**

**Sicherheit in vernetzten Systemen**

**Ausgearbeitet von Oliver Hardt und Michael Tümmel**

**Betreut durch Dr. H.-J. Mück**

1	Klassische Firewalls .....	3
1.1	Was ist eine Firewall .....	3
1.1.1	Angriffsarten auf das zu schützende Netz .....	3
1.1.2	SYN-Flooding (Denial-of-service-Attacke) .....	3
1.1.3	IP Spoofing .....	4
1.2	Schwächen des IP-Protokolls .....	5
1.3	Firewalltypen .....	6
1.3.1	Packet Filtering Firewalls .....	6
1.3.2	Proxies / Application Gateways .....	7
1.3.3	Statefull Inspection Firewalls .....	9
1.4	Firewallarchitekturen .....	9
1.4.1	Bastion Host Architektur .....	10
1.4.2	Screening Router Architektur .....	10
1.4.3	Dual Homed Host / Gateway Firewall .....	11
1.4.4	Screened Host .....	12
1.4.5	Screened Subnet .....	12
1.5	Die modernen Firewalls von heute .....	13
2	Hochgeschwindigkeits- Firewalls .....	14
2.1	Überblick .....	14
2.2	Lösungsmöglichkeiten .....	14
2.3	Anforderungen .....	14
2.4	Paketparallele Verarbeitung .....	15
2.5	Lösungsansatz für einen paketparallelen Packetscreen .....	16
2.6	Performance .....	16
2.7	Parallele Proxyserver .....	16
3	Desktop Firewalls .....	17
3.1	Überblick .....	17
3.2	Nachteile, Gefahren und Probleme .....	17
3.3	Beispiel : Tiny Personal Firewall .....	18
4	Grenzen von Firewalls .....	18
5	Referenzen .....	19

# 1 Klassische Firewalls

## 1.1 Was ist eine Firewall

Eine Firewall ist eine Komponente oder eine Menge von Komponenten (bestehend aus Hard- / und oder Software), Regeln und Protokollen, welche den Zugriff zwischen geschütztem Netzwerk und dem Internet (oder anderen Netzwerken) kontrollieren und regeln.

Dabei gilt es nicht nur das zu sichernde Netzwerk zu schützen, sondern auch die Firewall selbst. Die Gefahren, vor denen es zu schützen gilt, sind vielfältig. Zum einen sollen sie das Ausspionieren von Daten auf Systemen durch Einbruch in diese Systeme unterbinden. Des Weiteren ist das Ausspionieren bei der Übertragung zwischen Systemen (Mitlesen), sowie die Manipulation von Verbindungen (Übernehmen einer Verbindung mit Hilfe gefälschter Adressen) zu verhindern. Als letztes sei hier noch das Verhindern der normalen Funktion eines Rechners (Denial of Service Attack) genannt, zum Beispiel durch Syn Flooding.

Um den Schutz vor den eben genannten Gefahren zu gewährleisten, werden folgende Schutzmaßnahmen bereitgestellt:

- Blockieren unerwünschten Verkehrs
- Weiterleitung eingehenden Verkehrs an vertrauenswürdigeren, interne Systeme
- Verbergen verwundbarer Systeme, welche nicht auf einfache Art gesichert werden können, vor dem Internet
- Protokollierung des Verkehrs von und zum privaten Netzwerk
- Verstecken von Informationen wie Systemnamen, Netzwerktopologie, Netzwerk-Gerätetypen und interne Usernamen vor dem Internet
- Anbieten von robusterer Authentifizierung als durch die meisten Standardprogramme

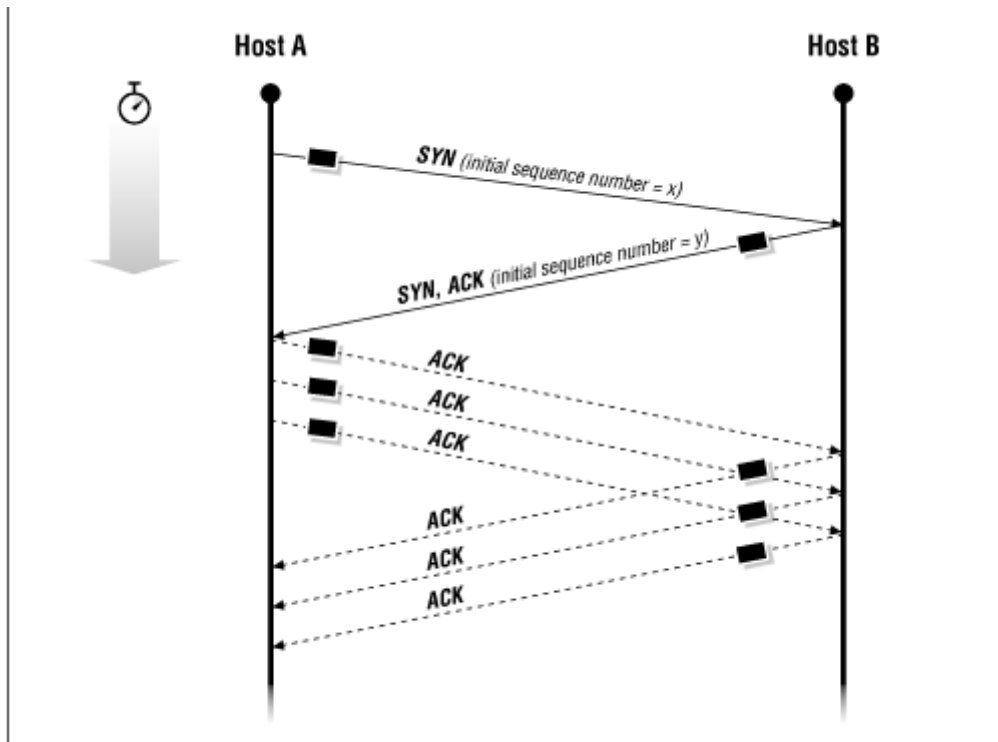
### 1.1.1 Angriffsarten auf das zu schützende Netz

Wie bereits erwähnt, sind die Gefahren vielfältig, vor denen es zu schützen gilt. Im Folgenden sollen beispielhaft zwei sehr verbreitete Angriffsarten kurz vorgestellt werden. Zum einen ist das die DOS-Attacke (Denial-of-Service-Attacke), welche dazu dient, einen Rechner lahm zu legen. Zum anderen soll das Spoofing näher erläutert werden, welches zum Zweck hat, Zugriff auf einen Rechner zu erhalten.

### 1.1.2 SYN-Flooding (Denial-of-service-Attacke)

Bei dieser Form des Angriffs wird das Drei-Wege-Handshaking von TCP benutzt, um so genannte halboffene Verbindungen herzustellen. Da TCP ein verbindungsorientiertes Übertragungsprotokoll ist, gibt es Mechanismen, um eine Verbindung zu synchronisieren. Dies wird über das Drei-Wege-Handshaking von TCP erledigt. Dabei werden drei Schritte durchgeführt:

- 1) Der Client sendet eine Synchronisationsnachricht (SYN) an den Server
- 2) Der Server antwortet mit einem entsprechenden Acknowledgement (ACK/SYN)
- 3) Darauf sendet der Client sein Acknowledgement (ACK) an den Server



**Drei-Wege-Handshaking, Bild aus [3]**

Mit diesen drei Schritten ist das Handshaking abgeschlossen. Nach Schritt 2 befindet sich auf dem Server ein Eintrag für die Verbindung, der bestehen bleiben muss, bis der Client seine Antwort gesendet hat. Eine Verbindung in diesem Stadium nennt man halboffen.

Eine SYN-Attacke nutzt nun die Tatsache aus, dass der Server die halboffenen Verbindungen speichern muss, bis er eine Antwort darauf erhält. Wird diese Antwort allerdings nie gesendet, so muss der Server die halboffene Verbindung trotzdem im Speicher behalten. Tatsächlich hört das Opfer erst nach einiger Zeit auf, auf die Bestätigung zu warten, wodurch natürlich recht schnell die gesamte Bandbreite des Rechners "aufgebraucht" ist.

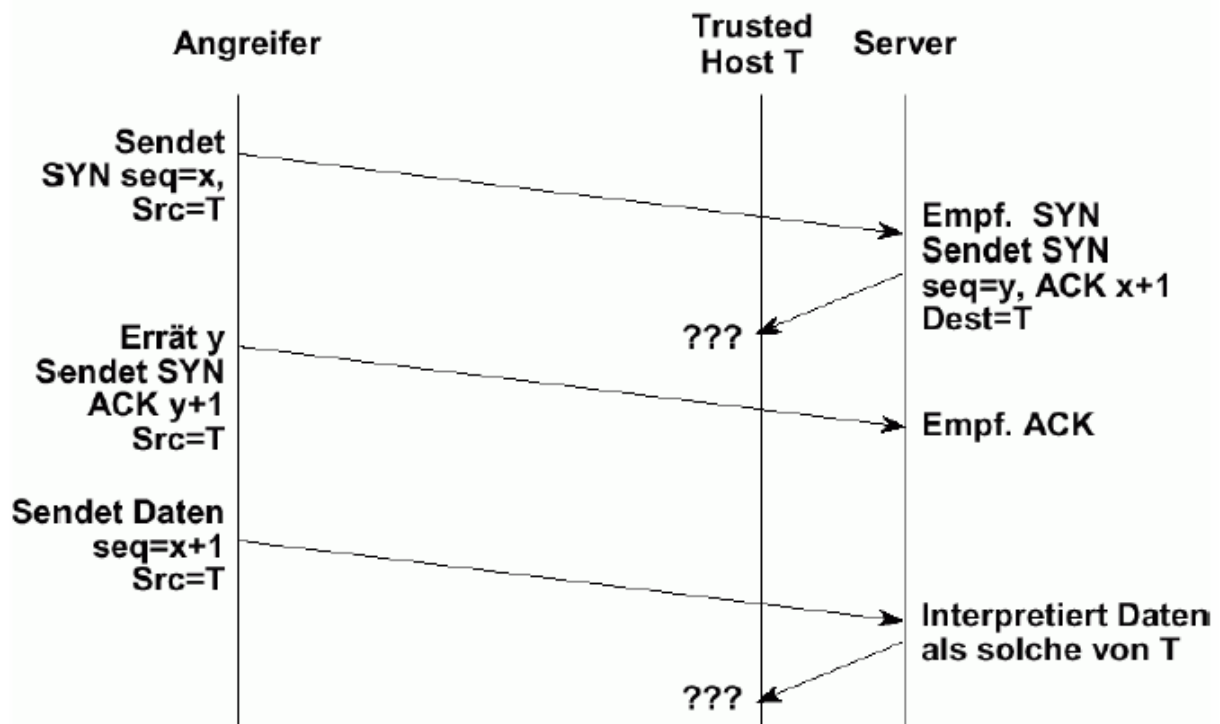
In den Implementierungen dieses Protokolls wird in der Regel eine Query benutzt, die einen gewissen endlichen Speicher für die halboffenen Verbindungen zur Verfügung stellt. Erstellt nun ein Angreifer eine größere Menge dieser halboffenen Verbindungen, so ist abzusehen, dass der Speicher der Query irgendwann zu Ende ist. An dieser Stelle ist es dem Server nun nicht mehr möglich, eine weitere TCP-Verbindung aufzubauen. Er kann somit nicht mehr auf Anfragen seiner Clients reagieren. Im schlimmsten Falle kann es sogar dazu führen, dass der Serverrechner durch den Überlauf der Query abstürzt, wodurch er dann völlig lahm gelegt ist.

### **1.1.3 IP Spoofing**

Spoofing ist eine Technik, mit der man vortäuscht, jemand anderes zu sein (Absenderadresse eines anderen angeben) mit der Absicht, authentifiziert zu werden und Zugang zum System zu erhalten.

Der Cracker weiß, dass der Host X eine Vertrauensbeziehung mit Y hat. X hat der Angreifer lahm gelegt. Jetzt muss der Cracker sich als X ausgeben und mit Y eine Verbindung aufbauen. Um das zu erreichen, muss er die korrekte Sequenznummer erraten.

## TCP: Voraussage von TCP-Folgennummern



Spoofing, Bild aus [6]

Nachdem er dann als X authentifiziert wurde, muss er sich ein geeigneteres Loch suchen, um ins System Y einzudringen.

Der Hacker schreibt nun bestimmte Dateien so um, dass Y Verbindungen von jeder Quelle akzeptiert, ohne sich zusätzlich authentifizieren zu müssen.

### 1.2 Schwächen des IP-Protokolls

Die TCP/IP-Protokollsuite ist Basis der im Internet verwendeten Protokolle. Gegenwärtig wird vorwiegend die vierte Version des Internet Protokolls eingesetzt. Allerdings mangelt es dieser an grundlegenden Funktionen zur Authentifizierung und zur Vertraulichkeit. Folglich sind die auf den TCP/IP-Protokollen aufsetzenden Schichten zunächst ebenfalls nicht gesichert. Jede Applikation und jedes Protokoll, das auf TCP/IP basiert, muss gegebenenfalls eigene Sicherungsmechanismen bereitstellen.

Durch das Wissen um Sicherheitsmängel im TCP/IP können Benutzer bereits bestehender TCP- bzw. UDP-basierter Anwendungen und Protokolle kritische Situationen einschätzen und im Bedarfsfall ihre sensitiven Informationen zusätzlich sichern.

Angriffe auf TCP/IP-Protokolle betreffen heute zahlreiche Anwendungen. Einige Angriffstechniken erfordern direkte Manipulationen der Protokollheader, welche nur mit Systemadministrator-Privilegien durchführbar sind. Heutzutage können Privatleute mit Hilfe von Internet-Diensteanbietern mit ihren eigenen Maschinen am Internet teilnehmen. Folglich haben sie die nötigen Privilegien inne, um ihre eigenen Rechner als Ausgangsbasis für Netzwerk-Angriffe einzusetzen. Überdies dürfte ein versierter Hacker ohne große Schwierigkeiten in ein fremdes System eindringen und dort die nötigen Privilegien erlangen können.

### 1.3 Firewalltypen

Es existieren verschiedene Firewalltypen, welche im Folgenden näher beschrieben werden sollen:

- Packet Filtering Firewalls
- Circuit Level Gateways
- Application Level Gateways
- Statefull Inspection Firewalls

Die wenigsten professionellen Firewalls gehören nur einer einzigen Kategorie an. Meistens werden verschiedene Typen miteinander kombiniert.

#### 1.3.1 Packet Filtering Firewalls

Diese arbeiten auf dem Netzwerk-Layer des OSI-Modells oder auf dem IP-Layer des TCP/IP Modells. Sie sind üblicherweise Teil eines Routers. Der Router erhält Pakete von einem Netzwerk und leitet sie an ein anderes Netzwerk weiter. Dabei wird jedes Paket mit einer Anzahl Regeln, sogenannten Screening Regeln, verglichen, bevor es weitergeleitet wird. Abhängig vom TCP- und IP-Header der Pakete und von den Regeln kann die Firewall das Paket ablehnen, es weiterleiten oder eine Nachricht zum Ursprung zurücksenden.

Screening Regeln bestehen aus zwei Bestandteilen, den Selektionskriterien und einer sogenannten Politik.

Die Selektionskriterien bestehen aus:

- IP-Adressen im IP Header
- Portnummern im TCP/UDP Header
- Protokollnummern je nach benutztem Service (Datagrammtyp) (TCP/UDP/...) unterschiedlich
- SYN-/ACK-Flag zur Richtungsfeststellung (TCP)

nach diesen Kriterien wird das Paket dann jeweils gefiltert.

Die Politik bezeichnet zwei Arten von Regeln, die Gebotsregeln (Allow) und die Verbotsregeln (Deny). Bei den Gebotsregeln ist alles, was nicht explizit erlaubt ist, verboten. Bei den Verbotsregeln ist alles, was nicht explizit verboten ist, erlaubt. Dabei ist die erste Strategie der Gebotsregeln der zweiten vorzuziehen.

Hierzu ein Beispiel:

Nr	Typ	Quell-adr.	Ziel-adr.	Quell-port	Ziel-port	Aktion
1	TCP	*	123.4.5.6	>1023	23 (telnet)	Allow
2	TCP	*	123.4.5.7	>1023	80 (WWW)	Allow
3	TCP	129.6.48.254	123.4.5.8	>1023	22 (SSH)	Allow
6	*	*	* 6	*	*	Deny

Stärken von Packet Filtern:

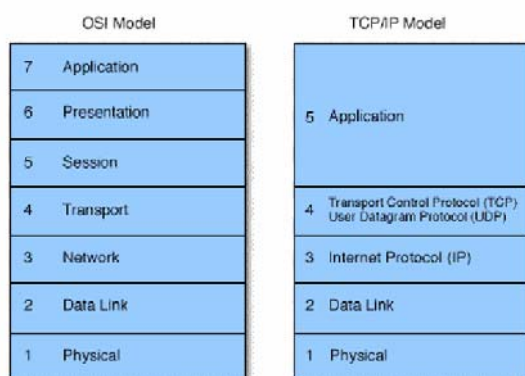
Paketfilterung ist eine kostengünstige Technologie mit recht guter Performance. Ein Paketfilter ist heute auf fast allen Router-Produkten standardmäßig implementiert. Oft ist kein zusätzlicher Administrations- und Konfigurationsaufwand notwendig. Des Weiteren unterliegt Paketfiltertechnologie keinen US-Exportbeschränkungen wie z.B. Kryptographie-Software. Packet Filter sind leicht erweiterbar, wenn neue Dienste oder Protokolle transportiert werden müssen (hinzufügen neuer Regeln reicht im Normalfall). Eine Implementation einer Packet Screen auf einem Rechner ist auch möglich und wird praktiziert. Der Vorteil hierbei liegt in weitaus verbesserten Protokollierungsmöglichkeiten.

Schwächen von Packet Filtern:

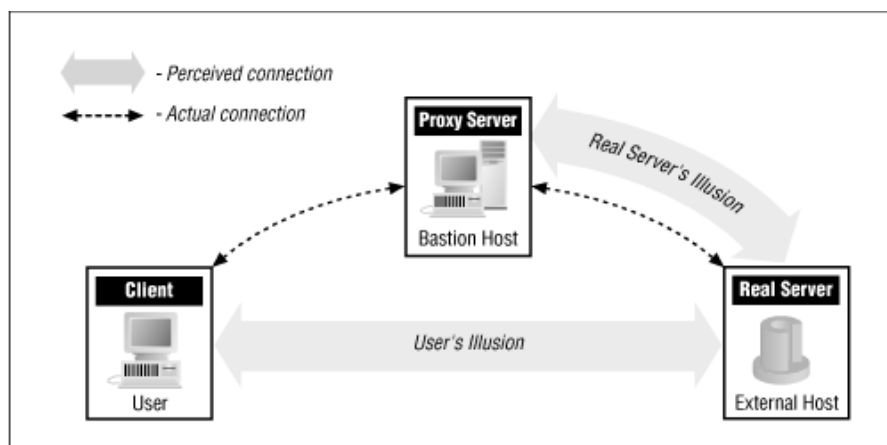
Paketfilterregeln sind für den Durchschnittsbenutzer oft recht verwirrend. Bei großen Netzen können Filterregeln sehr umfangreich und schwer nachvollziehbar werden. Protokollmeldungen enthalten oft keine Informationen über den Inhalt der übertragenen und verworfenen Pakete. Einige Protokolle sind für Packet Filter ungeeignet, da variable Portnummern verwendet werden. Des Weiteren besitzen Portnummern und IP-Adressen eine unzureichende Integrität der, da diese leicht gefälscht werden können (IP-Spoofing). Ein weiterer Nachteil ist die fehlende Benutzerauthentifizierung und die fehlende Kontrolle der Inhalte der Datagramme (keine Content-Filterung).

### 1.3.2 Proxies / Application Gateways

Packet Filter werten die Informationen der ISO/OSI-Schichten 3 und 4 aus, wohingegen Proxies die Informationen der Anwendungsschicht (5-7) heranziehen. In der Regel bilden mehrere Proxy-Prozesse einen Application Gateway. Proxies/Application Gateways bestehen aus Circuit-Level-Proxies und/oder Application-Level-Proxies.



Proxies / Application Gateways, Bild aus [3]



### 1.3.2.1 Circuit Level Gateways

Circuit Level Gateways arbeiten auf dem Session-Layer des OSI-Modells oder auf dem TCP-Layer des TCP/IP-Modells. Sie überwachen das TCP-Handshaking zwischen Paketen von vertrauenswürdigen Servern oder Clients und nicht vertrauenswürdigen Hosts und umgekehrt, um herauszufinden, ob eine Session legitim ist oder nicht. Um Pakete auf diesem Weg zu filtern, benutzen Circuit Level Gateways die Daten, welche im Header des TCP Session-Layer Protokolls vorhanden sind. Wurde das Handshaking als legitim erkannt, baut das C.-L.-Gateway die Verbindung auf und die Pakete werden nur noch hin und her transportiert ohne weiteres Filtern. Wurde die Session vollendet, wird sie aus der Tabelle gelöscht.

Der Client wird vollständig hinter dem Gateway verborgen (Verbergen der Netztopologie). C.-L.-Proxies sind unabhängig vom Protokoll einsetzbar und i.d.R. für den Client transparent. Die notwendigen Informationen werden vom Client Prozess erzeugt und dem Proxy zur Verfügung gestellt.

### 1.3.2.2 Application Level Gateways

Application Level Gateways arbeiten auf dem Application-Layer. Für jeden Dienst ist ein spezifisches Proxyprogramm auf dem Proxy-Server erforderlich (telnet, FTP, HTTP). Im Gegensatz zu Packet Filtern ist eine Nutzdatenanalyse möglich, das heißt, Daten können analysiert und z.B. nach bestimmten Schlüsselwörtern durchsucht werden (z.B. E-Mail, HTML-Seiten). Einige HTTP-Proxies bieten sogar die Möglichkeit, alle Zeilen innerhalb einer Seite, die zu Java-Applets gehören, zu löschen. Des Weiteren können bestimmte Dienstmerkmale eingeschränkt werden. Meist ist eine Cache Funktionalität für Webseiten verfügbar. Im Gegensatz zu Paket-Filtern wird mit Application-Level-Gateways eine verlässliche Trennung zwischen unterschiedlich vertrauenswürdigen Netzwerksegmenten erreicht. Der Proxy fungiert dabei als Stellvertreter zwischen Client und Server. Es gibt keinen direkten Aufbau einer Kommunikationsbeziehung, vielmehr kontaktiert der Client zuerst den Proxy, welches dann das eigentliche Zielsystem – den Server – kontaktiert. Manipulierte Pakete werden zuverlässig abgefangen und entsprechend protokolliert.

Stärken von Proxies / Application Level Gateways:

- Bieten ein hohes Maß an Sicherheit
- Sehr umfangreiche Protokollierung ist möglich
- Authentisierung des Benutzers kann vorgenommen werden (im Gegensatz zu Packet Filtering)
- Granularität auf Dienstebene
- Dienste können benutzerabhängig erlaubt werden
- Verbindung zwischen dem zu schützenden Netz und dem Internet wird durch Application Gateway völlig entkoppelt

Schwächen von Proxies / Application Level Gateways:

- Höherer Rechenaufwand nötig
- Wenig skalierbar
- Angriffe auf unteren Protokollebenen werden nicht erkannt und protokolliert, daher ist es oftmals angeraten einen zusätzlichen Packet Filter einzusetzen

### 1.3.3 Statefull Inspection Firewalls

Statefull Inspection Firewalls kombinieren die Aspekte der ersten drei Firewallgruppen. Sie filtern Pakete auf dem Netzwerk-Layer und erkennen, ob Session Pakete legitim sind oder nicht. Es handelt sich bei Statefull Inspection Firewalls um eine zustandsabhängige Paketfilterung, welche ein hohes Maß an Sicherheit, gute Leistung und Transparenz bietet.

Während ein konventioneller Packet Filter i.d.R. von außen ankommende ACK-Pakete unabhängig vom Status eines eventuellen Verbindungsaufbaus passieren lässt, wird ein Statefull Inspection Filter ACK-Pakete nur nach dem Auftreten eines von innen nach außen gesandten SYN-Paketes nach innen weiterleiten. Die potentielle Gefahr, dass von außen manipulierte ACK-Pakete nach innen gesandt werden, ohne dass von innen ein Verbindungsaufbau nach außen stattfand, wird durch diese Maßnahme stark reduziert.

Ähnliche Filtermöglichkeiten bestehen für UDP-basierte Dienste: z.B. "erlaube DNS-Antwort nur falls eine DNS-Anfrage gestellt wurde". Voraussetzung für eine qualifizierte Umsetzung dieses Beispiels ist jedoch, dass nicht nur Quell- und Zieladresse sowie Quell- und Zielport, sondern auch der DNS-Header im Anfrage-Paket in die Speicherung der Status- und Kontextinformation einbezogen wird. Hintergrund ist hier die verhältnismäßig leichte Fälschbarkeit von UDP-Paketen.

Ein vorteilhafter Aspekt von Stateful Inspection Filtern ist die Fähigkeit, die Daten auf allen Protokollebenen (d.h. von Netzwerk- bis Anwendungsebene) zu prüfen. So kann z.B. ein FTP-GET erlaubt, ein FTP-PUT jedoch verboten werden.

Ein positiver Effekt der im Vergleich zu konventionellen Paketfiltern erhöhten Eigenintelligenz ist die Option, einzelne Pakete während einer Kommunikationsbeziehung zu assemblieren und damit erweiterte Möglichkeiten zur Benutzer-Authentisierung zur Anwendung zu bringen.

Statefull Inspection Firewalls sind teuer und sehr komplex und damit oft schwer zu bedienen und zu warten. Des Weiteren sind sie anfällig für Sicherheitslöcher (falsche Konfiguration). Als Folge der nicht verlässlichen Trennung der Netzwerksegmente sind Stateful Inspection Filter nicht immun gegen bestimmte auf unteren Protokollebenen stattfindende Angriffe. So z.B. werden fragmentierte Pakete i.d.R. von außen nach innen ohne weitere Prüfung durchgelassen.

Eine weitere konzeptionelle Eigenschaft der meisten Stateful Inspection Filter besteht darin, dass nicht für jeden Dienst bzw. jede Kommunikationsbeziehung ein separater Proxy-Prozess gestartet wird. Die Filterung findet im gleichen Prozessraum statt. Ein einziger potentieller Schwachpunkt kann somit zum Absturz der gesamten Filterkomponente führen.

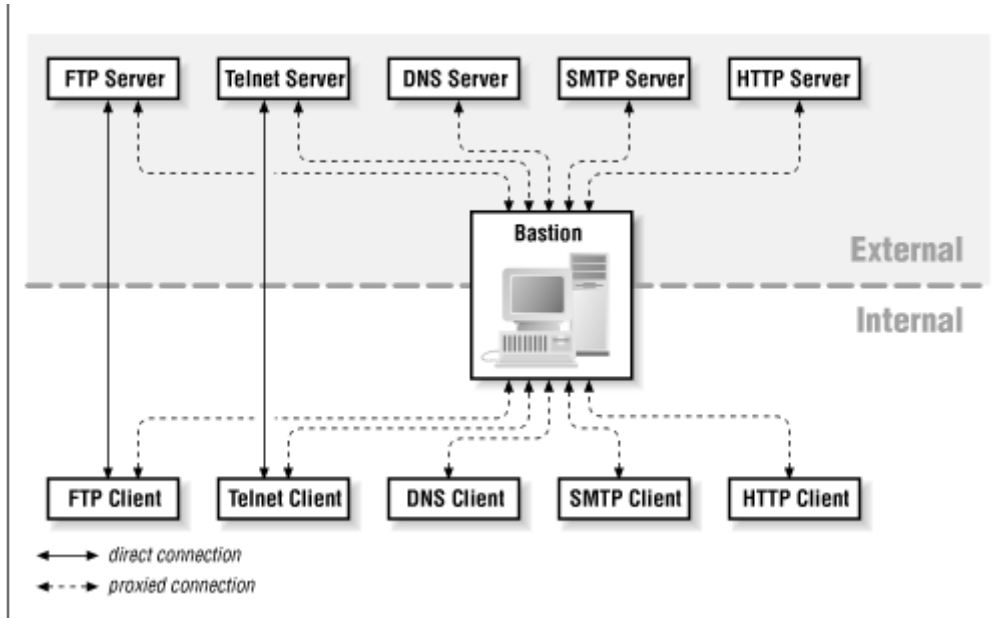
## 1.4 Firewallarchitekturen

Als allgemeine Grundsätze bei der Verwirklichung/Implementation einer Firewall sind folgende zu nennen:

- Position der eingesetzten Geräte sollte möglichst weit außen sein
- Bei stark zu schützenden Netzen sollten Geräte redundant eingesetzt werden
  - Unterschiedliche Hersteller
  - Unterschiedliche Filterformate
- Anzahl und Art der Firewallkomponenten sollte dem Sicherheitskonzept angepasst sein
  - Es gilt nicht unbedingt der Grundsatz  
⇒ je mehr Komponenten desto höher die Sicherheit

Im Folgenden sollen nun beispielhaft einige Firewallarchitekturen kurz vorgestellt werden.

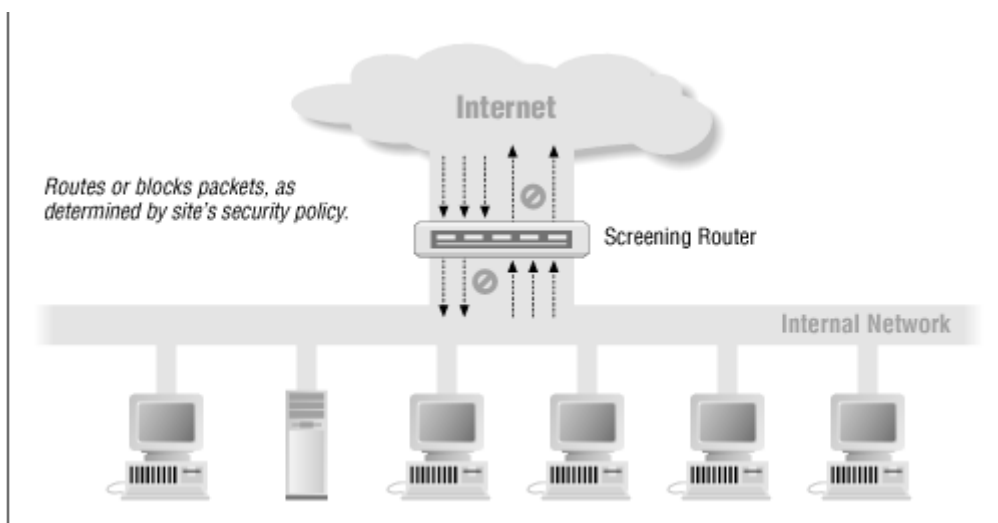
### 1.4.1 Bastion Host Architektur



**Bastion Host Architektur, Bild aus [3]**

Der Bastion Host ist der erste oder einzige Rechner, der aus dem Internet erreichbar ist. Für diesen ist daher höchste Hostsicherheit erforderlich. Das heißt, dass die Softwareausstattung so einfach wie möglich gehalten werden sollte (Minimalsystem/Least Privilege). Dabei erbringt er Internetdienste und leitet diese weiter. Es sollten keine Benutzeraccounts auf dem Bastion Host eingerichtet werden und er darf nicht die Funktionalität eines Routers erfüllen. Er übernimmt oftmals die Protokollfunktionen (Logging/Auditing).

### 1.4.2 Screening Router Architektur



**Screening Router Architektur, Bild aus [3]**

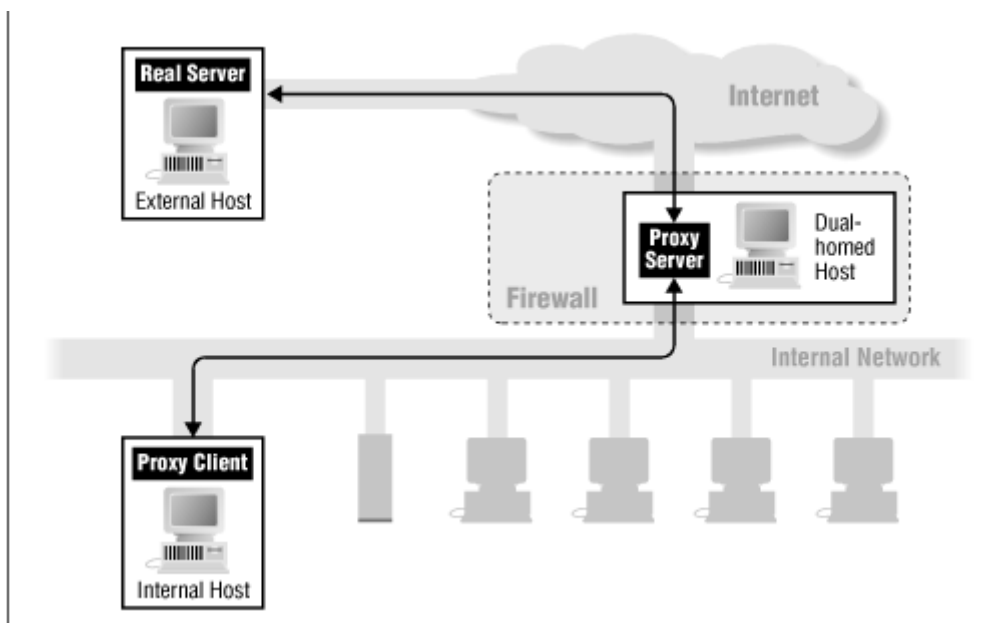
Vorteile einer Screening Router Architektur:

- Einfache Installation
- Geringer Administrationsaufwand
- Kostengünstig

Nachteile einer Screening Router Architektur:

- Nur geringer Schutz
- Gelingt es, dem Angreifer die Packet-Screen zu überwinden, liegt das gesamte Netz offen
- Begrenzte Protokollmöglichkeiten
- Gefahr durch IP-Spoofing

### 1.4.3 Dual Homed Host / Gateway Firewall



**Dual Homed Host / Gateway Firewall, Bild aus [3]**

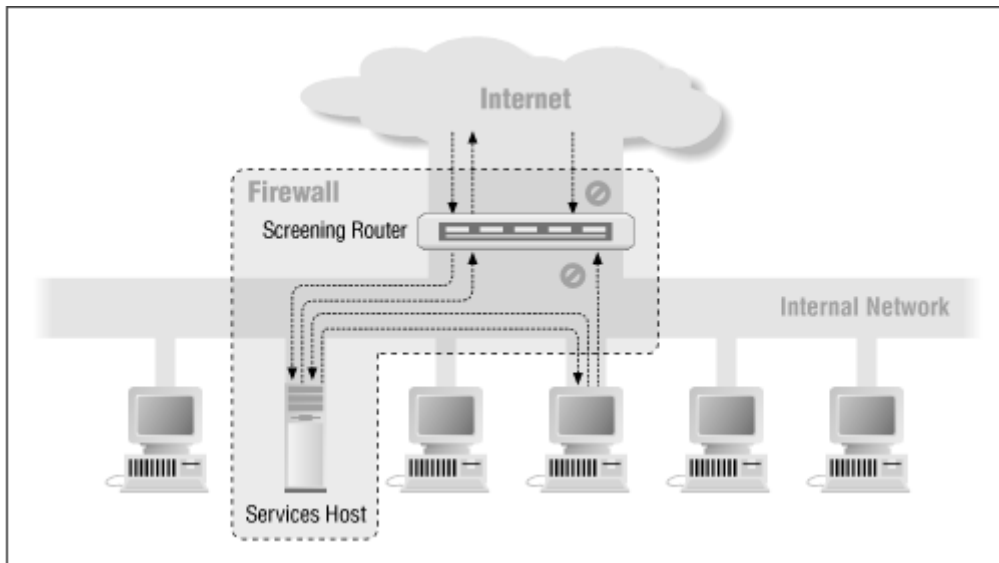
Vorteile einer Dual Homed Host / Gateway Firewall Architektur:

- Umfangreiche Zugriffskontrolle
- Angriffe können gut nachvollzogen werden
- Nur erlaubte Dienste können in Anspruch genommen werden

Nachteile einer Dual Homed Host / Gateway Firewall Architektur:

- Hoher Installationsaufwand
- begrenzte Erweiterungsmöglichkeiten
- Bastions müssen sehr gut gegen Angriffe geschützt werden

### 1.4.4 Screened Host



Screened Host, Bild aus [3]

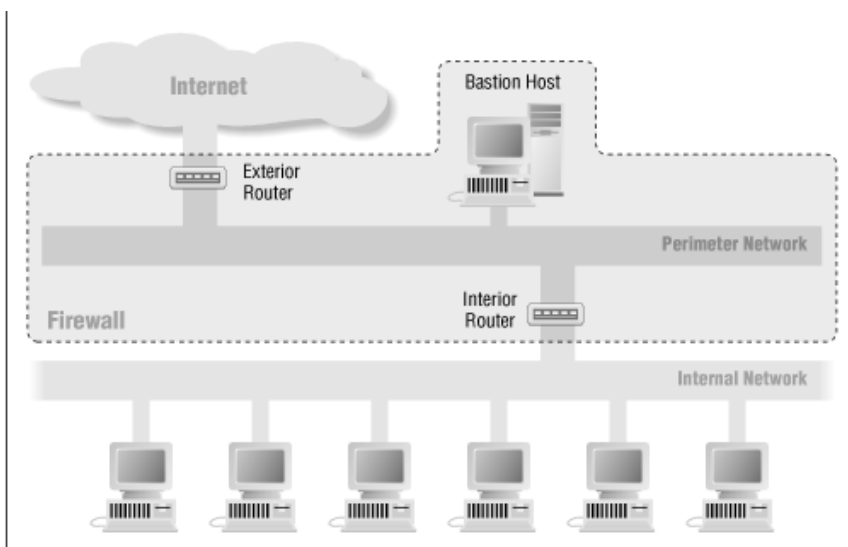
Vorteile der Screened Host Architektur:

- Verbindet Vorteile der ersten beiden Architekturen
- Bastion durch Screening Router geschützt
- mehrere Bastion Rechner sind möglich

Nachteile der Screened Host Architektur:

- Höhere Kosten
- innerer Netzwerkverkehr kann nach erfolgreichem Angriff der Bastion mitgehört werden (sniffing)
- Router als Single Point of Failure

### 1.4.5 Screened Subnet



Screened Subnet, Bild aus [3]

Die Screened Subnet Architektur wird auch als Perimeter Netzwerk, Grenznetz oder demilitarisierte Zone (DMZ) bezeichnet.

Vorteile der Screened Subnet Architektur:

- Gute Skalierbarkeit
- Es müssen zwei Packet Filter überwunden werden (am besten unterschiedlicher Bauart)

Nachteile der Screened Subnet Architektur:

- Höhere Kosten
- Hoher Administrationsaufwand

### **1.5 Die modernen Firewalls von heute**

Die heute auf dem Markt erhältlichen Firewalls sind technologisch ausgereift. Daher konzentrieren sich die Hersteller auf das Anbieten von Zusatzfunktionalitäten, wie z.B. Remote-Management-Tools in verteilten Unternehmen, VPN-Funktionalitäten, Virenschutz oder URL-Filter.

Von technologischer Seite her beruhen die heute erhältlichen professionellen Firewalls auf der Statefull-Inspection oder der Application-Proxy Technologie.

## 2 Hochgeschwindigkeits- Firewalls

### 2.1 Überblick

Hochgeschwindigkeits-Firewalls sind nötig, da moderne Datenverbindungen mit Geschwindigkeiten arbeiten, die jenseits des Leistungsvermögens normaler Paketscreens und Firewalls liegen. ATM, Glasfaser und Gigabit – Ethernet sind in der Lage, 155 mbit/s und mehr zu übertragen. Zusätzliche Probleme ergeben sich durch Besonderheiten der einzelnen Übertragungsarten. Meist werden die IP Pakete zerteilt und in neue Pakete aufgeteilt. Das IP Format, an dem sich Herkunft und Zieladresse schnell und leicht ablesen lassen, liegt nicht mehr vor. Der Paketscreen wird zum Flaschenhals, da alle Daten ihn passieren müssen, er gibt die Kapazitätsbegrenzung vor. Außerdem liegt bei z.B. ATM nicht unbedingt eine physische Trennung der Netze vor. Durch Manipulation der Router oder Switches kann ein Umgehen der Firewall möglich sein.

### 2.2 Lösungsmöglichkeiten

Die grundsätzliche Lösung, wie diese Kapazitätsengpässe beseitigt werden, ist die Parallelisierung von Vorgängen. Dabei kann man grob unterscheiden, ob man entweder für jede IP Verbindung einen neuen Prozess benutzt (Verbindungsparallel) oder die einzelnen Pakete auf mehrere Prozessoren verteilt und parallel verarbeitet (Paketparallel). Der Ausdruck Prozess/Prozessor ist hier nicht im „Hardware technischem“ Sinne (CPU, etc) zu verstehen, sondern meint eine verarbeitende Einheit (Paketscreen, Filter, etc).

Bei der verbindungsparallelen Verarbeitung gibt es den offensichtlichen Nachteil, dass das Verhalten bei nur wenigen offenen Verbindungen unter Umständen sehr schlecht sein kann. Das ist z.B. dann der Fall, wenn eine Verbindung die komplette Leitungskapazität benutzt. Sie kann dann maximal die Kapazität eines Prozessors als Leistung nutzen.

Die paketparallele Verarbeitung bietet sich an, weil bei ihr die Eigenschaften des IP Protokolls elegant ausgenutzt werden. Die einzelnen Pakete sind von einander relativ unabhängig und müssen nicht unbedingt in der originalen Reihenfolge beim Empfänger ankommen. Außerdem lässt sich, im Gegensatz zur verbindungsparallelen Methode, eine sehr feine Granulierung erreichen.

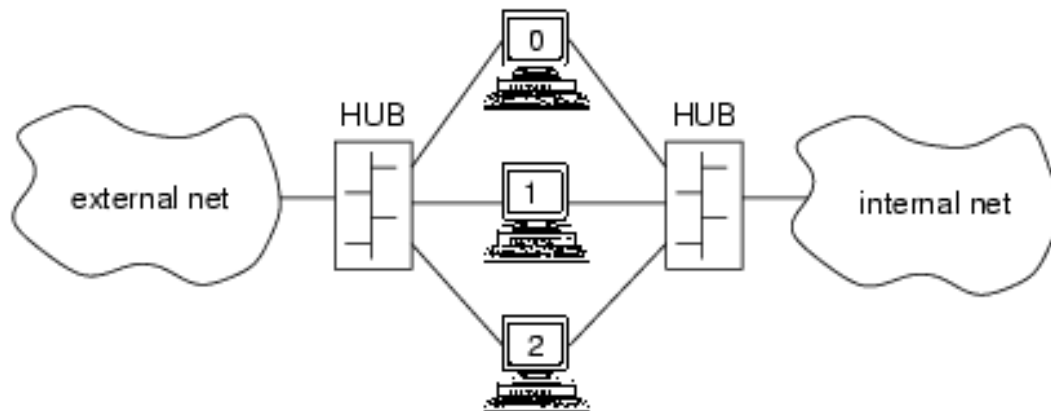
### 2.3 Anforderungen

Wichtig für alle Verteilungsverfahren sind einige wenige Kriterien, die möglichst optimal erfüllt sein sollten:

- Skalierbarkeit : Lässt sich das System einfach erweitern oder können neue verarbeitende Einheiten nur aufwendig hinzugefügt werden
- Overhead : Wie groß ist der zusätzliche Aufwand, der durch das Verteilen der Pakete / Verbindungen entsteht? Macht er eventuell den Nutzen zunichte?
- Aufwand bei der Implementation und Konfiguration: Der Aufwand sollte natürlich möglichst gering gehalten werden.
- Zuverlässigkeit : Entstehen durch das Verteilen der Pakete / Verbindungen Probleme mit der Zuverlässigkeit ? „Single-point-of-failures“ sollten vermieden werden.
- Einschränkungen: Einschränkungen sollten vermieden werden, damit bestehende Lösungen durch neue parallelisierte Systeme einfach ersetzt werden können.

Wenn man alle diese Kriterien abwägt, kommt man recht schnell zu dem Entschluss, dass für kapazitätsstarke Datenleitungen nur die paketparallele Verarbeitung in Betracht gezogen werden sollte. Bis auf die letztgenannte Einschränkung bietet sie in allen Punkten die fast optimalen Voraussetzungen. Die verbindungsparallele Verarbeitung hat Probleme mit der Zuverlässigkeit und der Skalierbarkeit und ist deshalb nicht unbedingt erste Wahl.

## 2.4 Paketparallele Verarbeitung



Beispiel für eine parallele Paketverarbeitung, Bild aus [2]

Die zu filternden Pakete kommen entweder aus dem internen oder aus dem externen Netz und erreichen den Verteiler (im Bild ein Hub, mehr zum Verteiler später). Hier werden sie auf alle Prozessoren verteilt. Jeder der Prozessoren sucht sich mit einem Algorithmus die Pakete raus, die er (und nur er) bearbeitet. Pakete, die er nicht bearbeitet, werden fallen gelassen (DROP). Danach werden die Pakete wieder zusammen geführt und in das Zielnetz geschickt.

Diese Lösung hat einige Vorteile, auf die wir etwas genauer eingehend sollten. Die Eigenheiten des IP Protokolls werden geschickt genutzt. Da die Reihenfolge der IP Pakete weitgehend egal ist, kann es ruhig zu Verschiebungen kommen. Da als kleinste Einheit die einzelnen IP Pakete dienen (und nicht ganze Verbindungen), kann die Last sehr gleichmäßig verteilt werden.

Des Weiteren benötigen die einzelnen Prozessoren (erst mal) keine weiteren allgemeinen Informationen, um die Pakete zu bearbeiten, eine zentrale Verwaltungsinstanz entfällt. Die Lösung ist sehr gut skalierbar und ausfallsicher. Sollte die Leistung nicht mehr ausreichen, können einfach neue Prozessoren hinzugefügt werden, der Datenstrom verteilt sich auch dann auf diese. Außerdem ist nur ein Minimum an Konfigurationsaufwand nötig, da alle Prozessoren die gleichen Regeln benutzen. Es muss also nur einmal ein Satz an Regeln erstellt werden, der dann einfach auf alle anderen Prozessoren dupliziert werden kann.

Für die Verteilung der Daten auf die Prozessoren bieten sich zwei grundsätzliche Lösungen an. Einmal die aktive Verteilung, bei der ein zentraler Verteiler die Aufgabe übernimmt, die Daten gleichmäßig auf die Prozessoren verteilt und gleichzeitig deren Status überwacht. Fällt einer aus, kann der Verteiler sofort reagieren und die Daten dementsprechend auf einen der anderen Prozessoren umleiten. Diese Lösung hat aber auch Nachteile. Der Verteiler in seiner zentralen Position wird zum Risiko („single-point-of-failure“). Fällt er aus, kommt es zum Totalausfall des ganzen Systems.

Die Alternative ist eine passive Verteilung durch einen Hub. Dieser „broadcastet“ alle Pakete an alle Prozessoren, die müssen sich dann „ihre“ Pakete aussuchen, die sie bearbeiten. Dies ist bei weitem sicherer und der aktiven Verteilung vorzuziehen. Daraus ergibt sich jetzt die

Aufgabe, dass den Prozessoren die Pakete eindeutig zugeordnet werden müssen, so dass keines doppelt und keines gar nicht bearbeitet wird. Diese Zuordnung sollte möglichst schnell erfolgen, um die Belastung der Prozessoren niedrig zu halten und keine neuen Flaschenhälse einzubauen. Die Verteilung sollte außerdem gleichmäßig über die Prozessoren sein und unabhängig von deren Anzahl, so dass bei Ausfall oder Erweiterung keine aufwendigen Anpassungen zu machen sind.

## **2.5 Lösungsansatz für einen paketparallelen Packetscreen**

Ein möglicher Lösungsansatz soll hier kurz beschrieben werden. Zuerst bildet der Prozessor einen Hashwert über ein oder mehrere ausgewählte Felder des zu bearbeitenden IP Paketes. Dazu sollten u.a. die IP Adresse von Sender und Empfänger, die IP Sequenz Nummer und die Header Checksumme gehören. Am besten ist eine Kombination aus mehreren Feldern, da das eine möglichst große Varianz des Hashwertes zur Folge hat. Dieser Hashwert wird dann bei  $n$  Prozessoren modulo  $n$  genommen und man erhält einen Wert zwischen 0 und  $(n - 1)$ . Der Prozessor (Nummer 0) bearbeitet also alle Pakete, die den Wert 0 haben, der zweite (Nummer 1) alle mit Wert 1 und so weiter. Alle Pakete die nicht zum jeweiligen Prozessor gehören, werden fallen gelassen (DROP).

Damit ist die Bearbeitung aller Pakete gewährleistet und keines wird von zwei Prozessoren verarbeitet. Sollte ein Paket wiederholt geschickt werden, so sollte sich auch der Hashwert ändern, da sich ja auch z.B. die IP Sequenz Nummer oder die Header Checksumme ändert. Und damit ändert sich wahrscheinlich auch der zugehörige Prozessor. Damit sollte zumindest kurzfristig ein Ausfall eines Prozessors überbrückbar sein.

Nachteile bei der parallelen Verteilung gibt es allerdings auch. Hubs lösen das Problem sehr unelegant, da sie einen einfachen Broadcast auf alle Prozessoren machen. Was ein Vorteil bei der Zuverlässigkeit ist, stellt sich als Nachteil heraus, wenn es um die Belastung der Netzwerkverbindungen geht. Denn immerhin muss noch jedes Paket vom Prozessor zumindest kurz betrachtet werden, ob es für diesen in Frage kommt. Außerdem unterstützen Hubs keine „Full Duplex“ Übertragung.

Man könnte die Hubs durch Switches ersetzen und in Verbindung mit Multicast Adressen eine ähnliche Funktionalität bereitstellen. Dies führt aber zu zusätzlichem Aufwand bei der Konfiguration bei den Switches und den Packetscreens.

## **2.6 Performance**

Die Performance der Packetscreens ist stark abhängig von der Anzahl der Filterregeln, die benutzt wird. Eine OC-3 Leitung ohne jeden Filter kommt auf ca. 155mbit/s. Bei nur 10 Filterregeln sinkt der Durchsatz auf ca. 60 Prozent, also ungefähr 95 mbit/s. Bei Hundert Filterregeln sind es nur noch 30 Prozent, also ca. 50 mbit/s.

## **2.7 Parallele Proxyserver**

Weitere Möglichkeiten für parallele Firewalls ist der Einsatz von parallelen Proxyservern. Diese arbeiten aber nach Protokoll getrennt. Es werden auf jedem Prozessor mehrere Instanzen des Proxy gestartet, wobei sich jede Instanz um eine Verbindung kümmert. Zusätzlich sollte noch eine Verteilung über mehrere Rechner stattfinden. Das kann aber zu Problemen bei z.B. FTP Verbindungen führen, weil der FTP Server unaufgefordert

Datenverbindungen zum Client öffnet, die dann durch die Firewall abgelehnt werden, weil keine Initiierung der Verbindung aus dem internen Netz vorhanden war.

Ein weiteres Problem ist der verhältnismäßig langsame Verbindungsaufbau, der vor allem bei vielen kurzen Verbindungen negativ auffällt.

Die Verteilung der Verbindungen kann wieder auf zwei Arten erfolgen. Einmal statisch, dass jeder Client auf eine fest zugewiesene Instanz des Proxy zugreift. Oder es werden dynamische Verteilungsverfahren benutzt, wie z.B. „Round-Robin DNS“ oder eine „Meta-Proxy“, der die Verbindungen verteilt. Hier entsteht aber wieder das bekannte Problem, dass zentrale Verteilungseinheiten bei Ausfall einen Totalausfall des Systems verursachen.

## **3 Desktop Firewalls**

### **3.1 Überblick**

„Desktop Firewalls“ sind ein Oberbegriff für Produkte verschiedener Hersteller, die alle dem Einzelanwender die Möglichkeit geben sollen, auf einem einzelnen Desktop PC den Netzwerkverkehr zu beschränken und zu kontrollieren. Dies wird meist über einfache Filterregeln gelöst, die zum Teil auch auf einer „pro Programm“ Basis eine Kontrolle über den Zugang zum Netzwerk regulieren. Bekanntere Produkte sind z.B.:

- McAfee Desktop Firewall
- Norton / Symantec Personal Firewall
- Tiny Personal Firewall
- ZoneAlarm
- Microsoft Windows XP (eingebaut im Betriebssystem)

### **3.2 Nachteile, Gefahren und Probleme**

Desktop Firewalls verleihen ein Gefühl von Sicherheit, das sie aber nicht immer gewährleisten können. Der Benutzer ist im guten Glauben, dass die Firewall ihn vor Angriffen und ungewollten Datenverkehr schützt, was aber nicht der Fall sein muss. Programme können sich „tarnen“ und sich für eine Applikation ausgeben, der schon der Zugang zum Netz gewährt wurde, in dem sie diese z.B. überschreiben. Ein anderes Problem ist wenn Programmen der Zugang gewährt wurde, die regelmäßig benutzt werden, wie z.B. dem Internet Explorer und dieser dann von einem zweiten Programm benutzt wird, um Seiten aus dem Netz abzurufen oder Daten zu senden.

Eine weitere Möglichkeit ist auch, dass das ganze lokale (Firmen-) Subnetz als vertrauenswürdig eingestuft wird, aber z.B. ein Proxy in diesem Subnetz die Verbindung in externe Netze ermöglicht. Dieser kann dann von Programmen ungehindert benutzt werden, da er sich ja auch im Subnetz und damit in der vertrauenswürdigen Zone befindet.

Auch ist es meist möglich, die Firewall einfach auszuschalten. Das kann Sinn machen, wenn z.B. eine Fehlfunktion vorliegt oder aber zu Testzwecken. Das sollte eigentlich nur dem Anwender erlaubt und zumindest eine Art Authentifizierung erfordern. Bei nachlässiger Implementation ist es aber auch für Dritte möglich, die Firewall zu deaktivieren. Ein bösesartiges Programm kann dann die Firewall deaktivieren und danach ungehindert Kontakt über das Netz aufnehmen.

Außerdem lassen die Reporting - und Logging – Fähigkeiten der meisten Firewalls deutlich zu wünschen übrig. Man kann selten erkennen, welchen Programmen der Zugriff auf Netzwerkressourcen gewährt wurde und welche daran gehindert wurden. Auch eine Übersicht über bestehende Verbindungen ist meist nicht selbstverständlich.

### **3.3 Beispiel : Tiny Personal Firewall**

Die Tiny Personal Firewall (TPF) [1] ist eine (für Privatanwender) kostenlose Desktop Firewall, die einige interessante Ideen umsetzt.

Einmal als Service auf dem PC installiert, fragt sie bei jedem Zugriff eines Programms auf das Netzwerk, ob dieser gestattet werden soll. Zusätzlich bietet die TPF sofort die Erstellung einer Regel an, wobei man dabei festlegen kann, ob der Zugriff nie oder immer, nur für bestimmte Quell- und Zieladressen und Ports erfolgen darf. Liegt für das entsprechende Programm schon eine Regel vor, wird nicht noch mal nachgefragt.

Es ist außerdem möglich, ganze IP und Portbereiche zuzulassen oder zu sperren. Des Weiteren wird für jede Applikation, die erfasst wurde, ein MD5 Hash gebildet, der bei jedem Zugriff geprüft wird. Ändert sich diese Checksumme für eine ausführbare Datei wird sofort nachgefragt, ob die Datei absichtlich (z.B. durch ein Update) ersetzt wurde und ob man den Netzwerkzugriff wieder gestatten möchte. Das verhindert effektiv das Ersetzen von Dateien.

## **4 Grenzen von Firewalls**

Dieser Abschnitt soll erläutern, vor welchen Gefahren auch Firewalls nicht schützen können. Da die Firewall nur den Verkehr zwischen zwei Rechner überwachen kann, ist die Schutzfunktion auch nur auf Strom an Daten zwischen zwei Rechnern beschränkt.

Kommt der Feind oder der Angreifer aus dem internen Netz oder bei mehrbenutzerfähigen Rechnern sogar vom eigenen Rechner, kann die Firewall nichts ausrichten. Denn wenn die Daten nicht über die Firewall geroutet werden, ist diese natürlich machtlos. Ein mögliches Szenario für diesen Fall wäre z.B. das ein Angreifer schon Zugang zu einem Rechner im internen Netz hat. Jetzt kann er neue Angriffe auf andere Rechner im internen Netz starten, die alle nicht die Firewall passieren müssen, und deshalb erst mal unerkannt und unverhindert bleiben.

Des Weiteren ist die Firewall hilflos, wenn sie bestimmte Formen des Angriffes nicht kennt und deshalb auch nicht verhindern kann. Der „Ping of Death“ ist ein solches Beispiel. Hier wird ein besonders formuliertes Ping-Paket an einen Rechner geschickt, der auf Grund einer fehlerhaften TCP/IP Implementation abstürzt. Für die Firewall ist das Ping Paket zuerst einfach nur ein Paket, das einen gültigen Empfänger und Sender besitzt. Außerdem sind Ping Pakete grundsätzlich sinnvoll und sollten, wenn es keine besonderen Vorbehalte gibt, durchgelassen werden. Die Firewall kann aber auf diese „Ping of Death“ Pakete „trainiert“ werden, da diese besondere Merkmale aufweisen, nach denen gefiltert werden kann. Dazu gehören z.B. bestimmte Flags und vor allem die Größe des Paketes.

Eine weitere Bedrohung, vor der Firewalls nicht automatisch schützen können sind Viren. Diese werden meist in ausführbaren Dateien oder eMails von Rechner zu Rechner übertragen und sind somit für die Firewall nicht automatisch zu erkennen. Aber auch hier gilt wieder einschränkend zu sagen, dass manche Virenangriffe (besonders von Würmern via http) zu verhindern sind, wenn der angreifende Virus oder Wurm eine eindeutige Signatur hat, also von der Firewall erkannt werden kann. Hierzu gehören Merkmale wie z.B. die URL, die aufgerufen wird.

Eine weitere Angriffstaktik, gegen die Firewalls nichts ausrichten können, ist das so genannte „Social Engineering“. Dabei werden Angestellte vom Angreifer auf geschickte Weise ausgehorcht, in dem der Angreifer zum Beispiel anruft und vorgibt aus der Firma zu sein (bei großen Firmen sehr einfach) und nach Einstellungen, Passwörtern, Geburtstagen oder ähnlichen Daten fragt, die bei einem Einbruch hilfreich sein könnten. Da der Angreifer sich nachher eventuell bei der Firewall mit den gewonnenen Daten authentifizieren kann, wird er nicht als Eindringling erkannt.

Und zu guter Letzt sind schlecht konfigurierte Firewalls die größte Bedrohung. Der Systemadministrator muss zwar immer abwägen zwischen Sicherheit für das interne Netz und dem Komfort der Benutzer (die möglichst nicht unnötig eingeschränkt werden sollen). Allerdings sollte die Firewall immer aktuell gehalten werden, um neue Würmer und sonstige Angriffe erkennen und verhindern zu können.

## 5 Referenzen

- [1] Tiny Personal Firewall – <http://www.tinysoftware.com>
- [2] Parallele Firewalls: Skalierbare Lösungen für Hochgeschwindigkeitsnetze von Uwe Ellermann und Carsten Benecke in DFN-Bericht Nr. 85
- [3] Building Internet Firewalls; D. Brent Chapman & Elizabeth D. Zwicky; O'REILLY 1995
- [4] Firewallsysteme – Konzeption-Implementation-Audit; Thomas Veit; Cebit 2000
- [5] Diplomarbeit: Sicherheitsaspekte in TCP/IP-Rechnernetzen von cand. inform. Ulrich Flegel; TU Braunschweig Juni 1997
- [6] „Firewalls“ – Seminararbeit Kommunikation und verteilte Systeme von Besson, Mettler, Simitovic, Steinmann; Institut für Informatik der Universität Zürich 2001