Firewalls

Begleitscript zum Workshop

Am 11. April 2002

Für die RWE Systems AG

Von Malte von dem Hagen

root@DocValde.net

Inhalt

1. Definition „Firewall“

Ein Firewall-System weist eine unverkennbare Analogie zu einem elektronischen Pförtner und einer elektronischen Brandschutzmauer auf. Sie sichert und kontrolliert den Übergang von einem zu schützenden Netz zu einem unsicheren öffentlichen Netz. Dabei muss ein Firewall-Element grundsätzlich zwei Aspekte erfüllen, um der besagten Analogie gerecht zu werden:

Brandschutzmauern

Als erstes ist ein Firewall-Element dafür zuständig einen Bereich in einem Netzwerk abzusichern, um eine Schadensbegrenzung in einem Notfall einzuräumen. Die eskalierende Seite wird so abgeschottet, dass Schäden nicht auf andere Teile des Netzes überschwappen können. Entsprechend wird das Gebäude einer Organisation in bestimmte Abschnitte unterteilt, damit beim Ausbruch von Feuer in einem Segment nicht andere Teile der Lokation ohne weiteres griffig für Schäden werden können. Auf Kommunikationsnetze bezogen bedeutet dies, dass das Firewall-Element das zu schützende Netz gegen Gefahren aus dem unsicheren Netz abkapselt. Es wird nur ein einziger, sicherer und bewachter Durchgang zwischen den beiden Teilnetzen gewährleistet: Der so genannte "Common Point of Trust".

Pförtner

Ein Firewall-System hat zudem die Aufgabe als Analogie zum Pförtner den Transfer zu kontrollieren. Möchte ein Besucher das Gebäude der Organisation betreten, so wird er identifiziert und authentisiert. Mitarbeiter werden als Mitarbeiter vermerkt, und Gäste werden als Gäste notiert. Außerdem wird kontrolliert, welche Gegenstände in das Gebäude eingeführt und ausgeführt werden. All diese Ereignisse werden sorgfältig beim Pförtner protokolliert, zum Beispiel, wann welcher Besucher gekommen und gegangen ist. Ebenso, wen er besucht hat und welche Gegenstände er beim jeweiligen Übertritt des Kontrollpunktes bei sich trug. Eventuell auftretende Unregelmäßigkeiten oder verdächtige Aktionen können anhand der Protokollierung im Nachhinein analysiert werden. Das elektronische Äquivalent zum Pförtner ist ein Firewall-Element, das überprüft, wer aus dem unsicheren Netz auf das zu schützende Netz zugreifen darf. Es kontrolliert, über welche Protokolle und Dienste zugegriffen wird und mit welchen Hosts kommuniziert werden darf. In diesem Sinne ist ein Firewall-System also gleichzeitig eine Brandschutzmauer und ein elektronischer Pförtner. Eine Firewall-Lösung und -Implementierung fällt jeweils sehr individuell aus und muss den jeweiligen Ansprüchen angepasst werden. Außerdem darf nicht außer Acht gelassen werden, dass ein solcher Knotenpunkt technische, personelle, organisatorische und infrastrukturelle Sicherheitsmechanismen erfordert.

Zielsetzung

Ein Firewall-System wird quasi als Schranke zwischen das zu schützende und das unsichere Netz geschaltet, so dass der ganze Datenverkehr zwischen den beiden Netzen nur über das Firewall-Element möglich ist. Ziel dieser Maßnahme ist es im Standardfall, das interne Netz (normalerweise das Netz des Betreibers, der auch die Firewall installiert) vor Angriffen aus dem externen Netz zu schützen sowie unerwünschten Datenabfluss vom internen in das externe Netz zu verhindern.

Extern steht dabei im Allgemeinen für die Kommunikationszugänge in den WAN-Bereich. Es kann aber auch bei Intranets für die weniger geschützten Bereiche innerhalb eines Unternehmens- oder Behördennetzes stehen.

Es stellt also im wahrsten Sinne des Wortes den "Common Point of Trust" für den Übergang zwischen unterschiedlichen Netzen dar. Auf der Firewall werden Mechanismen implementiert, die die ganzen Transaktionen sicher und beherrschbar machen sollen. Dazu analysiert das Firewall-System die Kommunikationsdaten, kontrolliert die Kommunikationsbeziehungen und Kommunikationspartner, reglementiert die Kommunikation nach einer Sicherheitspolitik, protokolliert Ereignisse und alarmiert gegebenenfalls bei bestimmten Verstößen den Security-Administrator. Firewalls werden in erster Linie genutzt, um die Anbindung ans Internet in vielerlei Hinsicht sicherer zu machen. Doch auch das aufteilen in Segmente oder Subnetze macht Sinn; besonders bei großen Netzwerken. Die Vorteile des "Common Point of Trust" lässt sich auf die geringen Kosten, Umsetzung der Sicherheitspolitik, Möglichkeiten, erhöhte Sicherheit und Überprüfbarkeit aufsummieren.

Allgemeine Ziele

Die allgemeinen Ziele von Firewall-Systemen sind folgende:

- Zugangskontrolle auf Netzwerk-, Benutzer- und Datenebene
- Rechteverwaltung
- Kontrolle auf der Anwendungsebene
- Entkoppelung von Diensten
- Beweissicherung und Protokollauswertung
- Alarmierung
- Verbergen der internen Netzstruktur
- Vertraulichkeit von Nachrichten

Um diese Ziele in greifbare Nähe rücken zu lassen, muss das Firewall-System selber gewisse Anforderungen erfüllen:

- Das Firewall-System muss selbst resistent gegen Angriffe sein.
- Accounting (IP- und benutzerorientiert)
- NAT - Network Address Translation (auch IP-Masquerading genannt)
- Jeglicher Datenverkehr von innen nach außen (und umgekehrt) läuft über die Firewall.
- Nur autorisierter Verkehr darf die Firewall passieren. Welcher Verkehr autorisiert ist, wird in einer Sicherheitspolitik definiert.
- Alles was nicht ausdrücklich erlaubt ist, wird von der Firewall abgewiesen.

2. Bedarf

Früher:

– **öffentliche Netze**: abgeschlossen, zentral verwaltet

– **Internet**: reines Forschungsnetz, kein lohnendes Angriffsziel, Benutzer vertrauen einander

Heute:

- zunehmende Dezentralisierung öffentlicher Netze durch Deregulierung der Telekommunikationsmärkte
- zunehmende kommerzielle Nutzung des offenen, dezentralen, „anarchischen" Internets

Folge:

– Sicherheitsmechanismen werden zum unverzichtbaren Bestandteil moderner Kommunikationssysteme

- In den letzten 10 Jahren haben sich Bedrohungsszenarien entwickelt, die vorher die absolute Ausnahme waren, was früher nur im Zusammenhang mit Spionage auftrat, ist heute Alltag! Pressemeldungen, die das verdeutlichen, erreichen uns regelmäßig, die Dunkelziffer von Sicherheitsvorfällen dürfte ungleich höher sein.

Jahr	1997	1998	1999	2000
Sicherheitsvorfälle	3285	4942	9859	17672

Quelle: http://www.cert.org/

Bedrohungen:

- **Script-Kiddies**

Durch die weite Verbreitung und den leichten und kostengünstigen Zugang zum Internet ist die Personenmenge, die das Internet nutzt, unüberschaubar groß. Die riesige Anzahl von Webseiten (über 1 Mio. Suchergebnisse für das Wort „Hacker") , die dem Internetnutzer Zugriff auf einfach zu bedienende Programme für schädliche Aktionen erlauben, lässt die so genannten Script-Kiddies zu einer ernstzu-nehmenden Bedrohung für Computernetze werden. Script-Kiddies sind meist Teenager, die zwar Programme benutzen können, aber nicht über tiefgehendes Wissen über Computer(-netze) verfügen. Sie benutzen und verbreiten diese schädlichen Programme (Trojaner, Viren, Scanner etc.) „aus Spaß" und ohne großes Unrechtsbewusstsein.

- **Cracker**

  Als Cracker werden Computerfreaks bezeichnet, die sich meist ein solides Fachwissen angeeignet haben, dieses jedoch aus Spaß und Geltungsdrang, aber auch politischen oder finanziellen Gründen, zu schädlichen Zwecken einsetzen. Sie schreiben die oben erwähnten Programme und brechen in Computernetze ein, um dort Schaden zu verursachen. Am beliebtesten ist das so genannte „Defacement“ von Webseiten großer Unternehmen, also das manipulieren oder austauschen der Internetseiten. Je größer das Unternehmen, desto größer der Ruhm, den sie in ihren Kreisen ernten. Es existieren Webseiten, die solche Defacements sammeln, archivieren und veröffentlichen.

- **Hacker**

  Hacker sind gemäß ihrer sich selbst auferlegten Ethik „gutmütig“, hochgradig technisch interessiert ist es nicht ihre Absicht, Schaden anzurichten. Falls sie in Computernetze einbrechen, richten sie dort kein Unheil an, sondern teilen dem Betreiber des Netzes die von ihnen ausgenutzten Lücken mit, damit er auf diese reagieren kann.

  Leider gibt es überall schwarze Schafe, so dass auch aus diesem Personenkreis die Angreifer für **Industrie- und Wirtschaftsspionage** rekrutiert werden.

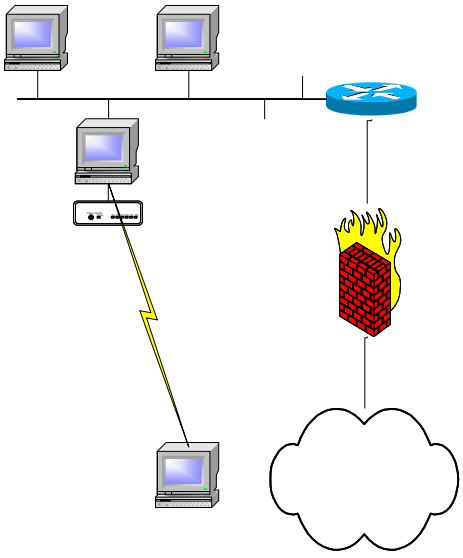
- **Mitarbeiter**

  Nicht zu vernachlässigen ist das Risikopotential der eigenen Mitarbeiter eines Unternehmens. Statistiken zufolge kommen etwa 50% der Angriffe auf ein Netzwerk von innen, von Mitarbeitern, die vom Unternehmen enttäuscht sind oder die so etwas als Spiel sehen.

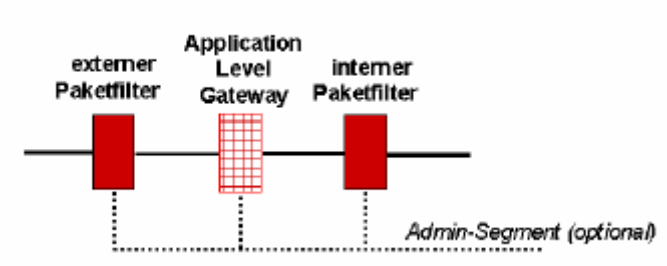
3. Netzwerkdesign

Einige grundsätzliche Überlegungen zu Firewalls: Je simpler eine Firewallarchitektur ist, desto weniger ist sie fehleranfällig, denn wenn mehrere Komponenten betreut und konfiguriert werden müssen, steigt auch die Anzahl der Fehlerquellen. Eine Firewall mit einem hohen Sicherheitsniveau ist teuer gegenüber einer mit einem niedrigen Sicherheitsniveau. Bei der Auswahl sollte darauf geachtet werden, dass die zu treffenden Schutzmaßnahmen im Preis nicht höher als der zu schützende Gegenwert liegen.
Ausnahme: Für persönliche Daten von Kunden, Klienten oder Patienten ist das höchste Sicherheitsniveau gerade gut genug, denn im Fall eines Systemeinbruchs ist nicht mehr der Netz- oder Systembetreiber der einzig Betroffene (hier hat der Betreiber eine besondere Sorgfaltspflicht). Die Sicherheit eines Netzes wird durch die Serienschaltung diverser Sicherheitsstufen, mit jeweils anderem Betriebssystem (bei gleichem BS. wäre eine mögliche Sicherheitslücke direkt auf allen Systemen), immens erhöht. Bei einer solchen Architektur sind der Aufwand der Administration und die Anforderungen an den Administrator sehr hoch.

Voraussetzung für das effiziente Ausnutzen (im positiven Sinne!) der Möglichkeiten eines Firewall-Systems ist ein durchdachtes Sicherheitskonzept. Ähnlich wie beim Pförtner gilt, dass nicht die Firewall etwas sicher macht, sondern mit ihr kann man etwas sicher machen, wenn sie richtig betrieben wird. Das bedeutet, dass ein Unternehmen sich vor dem Angehen an die Implementierung einer proprietären Zwischenlösung an sich an die Ausarbeitung eines Sicherheitskonzeptes machen sollte. In diesem analytischen Zusammentragen muss definiert werden, was vor wem und wer von was geschützt werden soll. Es macht wenig Sinn, sich Hals über Kopf für eine vermeintliche ultimative Lösung zu entscheiden, wenn man sich eigentlich gar nicht über das Problem im Klaren ist. Die Reduzierung des Zugriffs erhöht die Sicherheit und erleichtert die Kontrolle und Administration des Firewall-Systems. Das Fehlen von Überschaubarkeit kann bei einem solchen Projekt zur Achilles-Ferse werden.

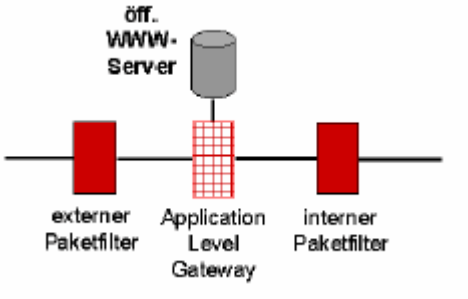


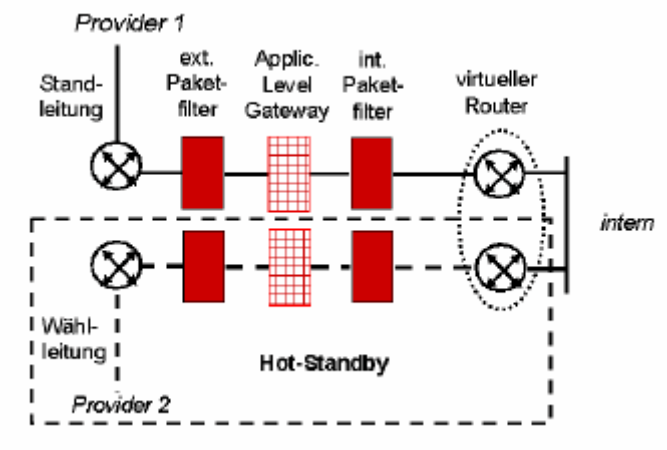
Wie schon gezeigt wurde, muss eine Firewall an allen Übergängen eines sicheren zu einem unsicheren Netz eingesetzt werden, es darf keinen Bypass geben.
Der häufigste Fehler, der hier gemacht wird, ist das undokumentierte Installieren von Modems oder ISDN-Karten, häufig von Mitarbeitern, denen der Internetzugang sonst verwehrt bliebe oder die den Restriktionen der Security Policy des Unternehmens entgehen wollen. Wie gefährlich dies ist, kann man sich vorstellen, wenn man bedenkt, dass es in Internet unzählige Wardialer-Programme gibt, die einen großen Telefonnummernbereich sehr schnell abtelefonieren können, um eben solche Modems oder ISDN-Karten zu finden.



Des Weiteren macht die Kombination verschiedener Firewall-Systeme Sinn, z.B. eines Paket Filters und eines Application Level Gateways. Selbst wenn ein System davon angreifbar ist, bleibt das Netz durch das andere noch einigermaßen geschützt.

Was aber ist mit Bereichen, die sowohl aus dem sicheren als auch dem unsicheren Netz erreichbar sein sollen? Das können bspw. Webserver, FTP-Server etc. sein. Für diesen Fall gibt es das Konzept der DMZ (De-Militarisierte Zone).
Eine DMZ ist Zwischenbereich, auf den von beiden Seiten aus zugegriffen werden kann, jedoch ist auch hierüber keine direkte Kommunikation von unsicher zu sicher erlaubt.
Moderne Firewalls besitzen (optional) mehrere Interfaces, die beliebig konfiguriert werden können, für jedes Interface kann ein eigenes Regelset erstellt werden, so dass eine DMZ leicht eingerichtet werden kann.





Am Rande erwähnt sei, dass im Zuge der Ausfallsicherheit ein redundantes Design erarbeitet werden sollte, da für die meisten Bereiche mit erhöhtem Schutzbedarf gleichzeitig eine hohe Verfügbarkeit gefordert ist.

Jede Firewall muss administriert und gepflegt werden. Das höchste Maß an Sicherheit bietet hier ein so genanntes „Out-of-Band-Management“. Das bedeutet, dass zur Administration Wege (und nur diese Wege) genutzt werden, die keine Verbindung zum Nutz-Netzwerk haben. Dies kann ein eigenes Netzwerk sein oder (im einfachsten Fall) über eine lokale Konsole erfolgen.

Wenn ein separates Netz nicht in Frage kommt, muss zur Administration zwingend eine sichere Kommunikation erfolgen, in Verbindung mit Verschlüsselung und Authentifizierung (Als Schlagworte seien hier genannt: SSH, TACACS+)

4. Klassifizierung

4.1 Personal Firewall

Die Gefahr, beim Durchforsten des Internets auf Viren oder korrupten Programmcode zu stoßen, egal ob nun in Form eines ActiveX-Elements oder eines Java-Applets, wächst mit der Bedeutung des Internets. Für eine professionelle Lösung muss relativ tief in die Tasche gegriffen werden, um das eigene Netzwerk vor solchen Gefahren hermetisch abzuriegeln. Daher werden auf Software-Ebene so genannte Personal-Firewalls realisiert, welche vorzugsweise Windows-Systeme von Normalanwendern vor Gefahren aus dem Internet bewahren sollen.

Leider ist es so, dass viele auf dem Markt erhältliche Systeme nicht die Anforderungen erfüllen können, die eigentlich an das Objekt in Extremsituationen gestellt werden. Auch die Performance auf dem System nimmt rapide ab, obwohl dies heutzutage bei der eingesetzten Hardware, auch im privaten Bereich, nicht mehr so als negativ ausschlaggebend eingestuft werden muss. Die größte Angriffsfläche bietet jedoch in den wenigsten Fällen die Personal-Firewall selbst, sondern das Betriebssystem, auf dem sie aufsetzt.

Trotzdem ist es durchaus sinnvoll für den Surfer von heute, stets mit aktueller Anti-Viren-Software und Personal-Firewall in die Weiten des Internets vorzudringen, da hierdurch wenigstens sämtliche TCP/IP-Pakete kontrolliert werden können, die den Rechner erreichen. Auch fungieren einige Personal-Firewalls automatisch als Viren-Scanner, indem sie automatisch das Verhalten von ActiveX- und Java-Elementen überprüfen, die lokale Daten löschen oder auf dem eigenen Rechner unbemerkt im Hintergrund Daten per FTP an einen entfernten Cracker schicken könnten. Die Personal-Firewall schafft, um ihren Zweck zu erfüllen, einen Schutzbereich, der so eine Art Quarantäne bildet. Dieser Schutzbereich wird Sandbox genannt, wobei ein Programm, welches innerhalb dieser virtuellen Umgebung ausgeführt wird, nur sehr begrenzten Zugriff auf Ressourcen des Systems gewährt bekommt. Zwar ist der Ansatz dieses Sandbox-Systems sehr gut, doch sind die verschiedenen Umsetzungen noch nicht genug ausgereift, um einen umfassenden Schutz in dieser Hinsicht zu geben.

Vorteile von Personal-Firewalls

- Niedrige Kosten.
- Meist für Normal-Anwender zugeschnitten und daher leicht verständlich.

Nachteile von Personal-Firewalls

- Nicht besonders zuverlässig, da schon alleine das Betriebssystem zu viel Angriffsfläche bietet.
- Viele Anwender können aufgrund fehlender TCP/IP-Kentnisse keine korrekten Filter-Regeln setzen und die Protokollierung auswerten.
- Performance-Einbussen auf der Workstation.

4.2 Paket Filter

Das aktive Firewall-Element Paket Filter analysiert und kontrolliert die ein- und ausgehenden Pakete auf der Netzzugangs-, der Netzwerk- und der Transportebene. Dazu werden die Pakete, nicht nur TCP/IP-Protokolle, aufgenommen und analysiert. Diese Analyse wird aufwendiger, sobald auch der Inhalt der einzelnen Pakete durchforstet werden soll; daher wird normalerweise nur ein rascher und hastiger Blick auf den Header geworfen. Die beiden Netze werden bei einer solchen Implementierung physikalisch entkoppelt. Ein Paket-Filter verhält sich wie eine normale Bridge und wird transparent in eine Leitung eingefügt.
Nach der Analyse des Pakets wird verifiziert, ob es unter eine bestimmte Regel fällt, und dementsprechende Reaktionen ausgeführt. In den Regeln wird vorzugsweise definiert, dass nur die notwendigste Kommunikation erlaubt ist; alle Verstöße werden abgewiesen (engl. reject) oder verworfen (engl. deny). Dadurch können sicherheitskritische Aktionen, wie zum Beispiel IP-Fragmentierung, von vornherein ausgeschlossen werden.
Diese Aufgabe kann z.B. ein einfacher Linux-PC (ipfwadm) übernehmen oder auch ein Cisco Router mit geeignetem Cisco IOS (FW-Option).

Allgemeine Arbeitsweise

- Es wird überprüft, von welcher Seite das Paket empfangen wird (Informationen aus dem Einbindungsmodul).
- Auf der Netzzugangsebene werden die Quell- und Ziel-Adresse und der verwendete Protokoll-Typ kontrolliert.
- Auf Netzwerkebene wird je nach Protokoll-Typ das Paket anders kontrolliert.
    - IP: die Ziel-, Quell-Adresse, verwendetes Schicht-4-Protokoll, Optionsfeld und Flags
    - ICMP: die ICMP-Kommandos/-Typen
    - IPX: Network und Node
- Auf Transportebene findet
    - bei UDP und TCP eine Überprüfung der Portnummern (Quelle und Ziel) statt.
    - bei TCP eine Überprüfung der Richtung des Verbindungsaufbaus statt.
- Zusätzlich kann überprüft werden, ob der Zugriff in einem erlaubten Zeitrahmen geschieht.

Die entsprechenden Prüfinformationen werden aus dem Regelwerk (Accessliste und Rechteliste) entnommen und mit den Ergebnissen der jeweiligen Analysen verglichen.
Paketfilter sind geeignet für paketorientierte Dienste (ftp, login, ...), jdeoch NICHT für kontextabhängige Dienste (X11, http, mail, ...)

Vorteile des statischen Paketfilters sind die relativ einfache Handhabung sowie der geringe finanzielle und zeitliche Aufwand bei der Installation. Statische Paketfilter schützen vor IP-Spoofing, einer der Hauptangriffsmethoden zur Umgehung von Abweisungsregeln für gesperrte Dienste. Hinzu kommt der Vorteil, daß in der Regel für die Anbindung an das externe Netz ohnehin ein Router benötigt wird, der heute zumeist die Funktionalität eines statischen Paketfilters bereits integriert hat.

Nachteile des statischen Paketfilters sind hauptsächlich die Einschränkungen bei den Filterregeln: es können nur die Dienste ohne Sicherheitsrisiko durchgelassen werden, die feste Portnummern verwenden und verbindungsorientiert arbeiten (dann lässt sich der Initiator der Verbindung ermitteln), also TCP-Pakete. Bei UDP-Paketen ist es auf UDP-Ebene nicht möglich, festzustellen, ob es sich um eine Anfrage oder eine Antwort handelt. Daher sind UDP-Pakete durch statische Paketfilter nur sicher handhabbar, wenn es sich um Dienste handelt, bei denen sichergestellt ist, dass sie keinen Schaden anrichten (etwa DNS bei entsprechender Konfiguration des internen DNS-Servers sowie der internen DNS-Clients). Daher ist aus Sicherheitssicht ein statischer Paketfilter allein in der Regel nur dann tragbar, wenn man sich auf die Nutzung weniger Dienste wie z.B. FTP, TELNET, SMTP und DNS beschränkt. Auch ist es nicht möglich, Dienste nur für bestimmte Benutzer zuzulassen. Ein weiterer entscheidender Nachteil von (statischen und dynamischen) Paketfiltern ist schließlich, dass Angriffe nicht oder erst zu spät erkannt werden können, da die Protokollierung auf den Inhalt der *sockets* beschränkt ist.

Dynamische Paketfilter

Bei verbindungslosen Kommunikationsverbindungen, wie dies zum Beispiel bei UDP der Fall ist, kann nicht grundsätzlich festgelegt werden, von wem ein Verbindungsaufbau durchgeführt wird. Dynamische Paketfilter besitzen in einem solchen Fall die zusätzliche Eigenschaft, sich die Informationen (IP-Adresse und Port) der nach Außen geschickten UDP-Pakete zu merken und nur die entsprechend passenden Antworten der virtuellen Verbindung zurückzulassen. Das bedeutet genauer, dass nur die Antwortpakete durchgelassen werden, die vom selben Host und gleichen Port kommen, an den das ursprüngliche UDP-Paket gesendet worden ist und entsprechend zum gleichen System retourniert wird. Der Name rührt also daher, dass die Filter-Regeln intern dynamisch gehandhabt werden. Dienste wie SNMP können über dynamische Paketfilter also viel sicherer angeboten werden.

Die **Konfiguration von Paketfiltern** erfolgt in drei Schritten:

1. Festlegen einer Sicherheitspolitik (was ist erlaubt, was ist verboten).
Die Sicherheitspolitik sollte bereits explizite Aussagen enthalten, welche Dienste in welche Initiierungsrichtung zugelassen werden.

2. Mittels obiger Aussagen werden die zugelassenen Pakettypen formal spezifiziert.

3. Abschließend werden diese formalen Spezifikationen in die Syntax des Paketfilters übersetzt. Die dabei entstehenden formalen Regeln wendet der Paketfilter üblicherweise streng sequentiell an. Sobald eine Regel auf ein Paket zutrifft, bricht die Regelprüfung ab und die betreffende Regel wird auf dieses Paket angewendet. Danach wird die Regelprüfung für das nächste Paket ausgeführt. Pakete, die nicht explizit durch eine Filterregel zugelassen sind, müssen somit durch eine abschließende generelle Sperr-Regel abgewiesen werden (Realisierung des allgemeinen Sicherheitsprinzips: alles, was nicht ausdrücklich erlaubt ist, ist verboten). Durch die sequentielle Abarbeiten kann es unter Umständen zu unbeabsichtigten Effekten, insbesondere zu Sicherheitslücken kommen, wenn die Reihenfolge der Filterregeln falsch gewählt wurde.

4.3 Stateful Inspection

Als auf Paketfiltern basierende Technologie sind Stateful Inspection Filter in der Lage, sich die aktuelle Status- und Kontextinformation einer Kommunikationsbeziehung in internen Zustandsautomaten zu merken.
Diese grundsätzliche Arbeitsweise lässt sich am 3-stufigen Aufbau einer TCP-Verbindung veranschaulichen:

1. Verbindungsanfrage des Quellrechners : (SYN)
2. Bestätigung der Anfrage vom Zielrechner : (SYN-ACK)
3. Bestätigung vom Quellrechner : (ACK)
4. Weiterer Datentransfer : (ACK)

Während ein konventioneller Paketfilter i.d.R. von außen Pakete mit gesetztem ACK-Flag unabhängig vom Status eines eventuellen Verbindungsaufbaus passieren lässt, wird ein Stateful Inspection Filter diese 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.

Als „Stateful Inspection Packet Filter“ können beispielsweise ein OpenBSD-PC mit der ipfilter Software, eine Cisco PIX oder die weit verbreitete Checkpoint FW-1 Software zum Einsatz kommen.
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. 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 (vgl. Paketfilter).

Eine Konfigurationsoption einiger Stateful Inspection Filter besteht in der Aktivierung sog. Fast-Modi für einzelne oder alle TCP-Verbindungen, was die Sicherheitseigenschaften jedoch auf die eines konventionellen Paketfilters reduziert. 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. Zwingend erforderlich sollte deshalb ein spezieller Prozess (evtl. doppelte Auslegung) sein, der Aktivität, Funktionsfähigkeit und insbesondere die Integrität der Filterkomponente überwacht.
Ein aus Anwendungssicht positiver Aspekt ist die sehr gute Transparenz eines Stateful Inspection Filters. Modifikationen auf Clients sind in der Regel nicht nötig. Zu den vorteilhaften betrieblichen Eigenschaften zählen weiterhin die guten Performancewerte.

Vorteile von Paketfiltern

- Transparent für Benutzer und die Rechensysteme. Ausnahmen sind natürlich explizite Authentifizierungen.
- Einfach erweiterungsfähig für neue Protokolle.
- Flexibel für neue Dienste.
- Für andere Protokollfamilien verwendbar (IPX, OSI, DECNET, SNA, ...).
- Hohe Performance durch optimale Mechanismen (Hardware, Betriebssystem, Treiber)
- Leicht realisierbar, da geringe Komplexität.

Nachteile von Paket-Filtern

- Daten, die oberhalb der Transportebene sind, werden in der Regel nicht analysiert.
- Für die Anwendungen (FTP, HTTP, SMTP, ...) besteht keine Sicherheit.
- Falsch konfigurierte oder kompromittierte Hosts im internen Netz können für normalerweise nicht erlaubte Kommunikationen zwischen den beiden Netzen missbraucht werden.
- Protokolldaten werden nur bis zur Transportebene zur Verfügung gestellt.

4.4 Application Gateway

Application-Level-Gateways leiten Anwendungen weiter, wobei für jede Anwendung ein eigener Code verwendet wird, der so genannte Proxy-Server. Dieser Proxy-Server untersucht den Verkehr und vermittelt zwischen interner und externer Seite. Bei dieser Aufgabe hat er die vollständige Kontrolle über den Verkehr, und somit die Möglichkeit einer vollständigen Protokollierung des Datenaustausches. Standardmäßig ist die TCP/IP-Weiterleitung unterbunden, so dass, wenn kein für diesen Dienst entsprechender Proxy-Server installiert ist, keine Verbindung möglich ist.

Ein Benutzer, der über ein Application-Gateway kommunizieren möchte, muss sich zuerst beim Firewall-System identifizieren und authentisieren. Es gibt verschiedene Möglichkeiten, wie diese Authentifizierung von Statten gehen kann. Danach wird die Kommunikation für den Anwender transparent weitergeführt: Für ihn sieht es so aus, als würde er einen direkten Datenaustausch mit seinem Ziel abhalten. Damit diese Lösung realisiert werden kann, muss auf dem Application-Gateway ein Dienst laufen, der über einen definierten Port ansprechbar ist. Die Pakete an diesen Port werden analysiert und gegebenenfalls weitergeleitet. Eine solche Software ist in der Regel nur für einen Dienst (HTTP, FTP, SMTP, ...) konzipiert worden und wird als Proxy bezeichnet. Für jeden Dienst, der über das Application-Gateway ansprechbar und weiterleitbar sein soll, muss ein eigener Proxy vorhanden sein, aber auch keine weitere Software, die diesen Dienst ermöglichen könnte.
Jeder Proxy auf dem Application-Gateway kann speziell für seinen Dienst, für den er zuständig ist, weitere Sicherheitsdienste anbieten. Auch die Analyse ist auf dieser Kommunikationsebene sehr intensiv möglich, da der Kontext der Anwendungsdaten für den jeweiligen Dienst klar definiert ist. Der Vorteil ist, dass kleine überschaubare Module genutzt werden, und so die Fehleranfälligkeit erfreulich niedrig gehalten werde kann.
Das Security-Management darf aus Sicherheitsgründen nicht auf demselben Rechnersystem laufen oder zumindest nicht zur gleichen Zeit, wie das Application-Gateway. Zudem sollen keine Routing-Funktionen auf dem Host aktiv sein, damit nichts an den Proxies vorbeigeschmuggelt werden kann.
Da das Application-Gateway bei der Kommunikation jeweils zum Rechnersystem des unsicheren Netzes und zu dem zu schützenden Netzes eine Kommunikationsverbindung hat, bietet es eine "Network Address Translation". Dabei hat das Application-Gateway für die jeweiligen Interfaces zwei verschiedene IP-Adressen, die auch jeweils nur für die jeweilige Richtung genutzt werden.

Application-Level-Proxies

Application-Level-Proxies sind für bestimmte Dienste/Anwendungen implementiert. Aus diesem Grund kennen sie die Kommandos der Anwendungsprotokolle und können diese detailliert analysieren und ebenso haargenau kontrollieren. Application-Level-Proxies arbeiten mit der gängigen, unveränderten Clientsoftware für ihre individuellen Dienste. Oft wurde aber zudem eine veränderte Vorgehensweise im Falle einer gewünschten Authentifikation beim Application-Level-Proxy implementiert.

Ein **SMTP-Proxy** beispielsweise arbeitet nach dem Store-and-Forward-Prinzip, welches in gewissem Masse eine Analogie zum Sammelbriefkasten aufweist. Dabei wird als erstes die komplette Mail vom SMTP-Proxy abgespeichert. Ein Weitersenden wird erst eingeleitet, wenn die erste Phase des Annehmens erfolgreich verlief. Für die Mail-Kommunikation ist also keine end-to-end Beziehung zwischen eigentlichem Sender und seinem nächst direktem Empfänger notwendig.

Der SMTP-Proxy arbeitet nicht benutzerorientiert und erfordert daher keine Authentifikation. Eine ankommende E-Mail wird standartmässig auf dem TCP-Port25 (SMTP) entgegen-genommen und nach der Überprüfung des Absenders auf dem Application-Gateway in einem speziellen Verzeichnis abgelegt. Der SMTP-Daemon prüft periodisch, ob neue Nachrichten eingegangen sind. Der Mail Transfer Agent (MTA) stellt dem Adressaten die elektronische Post direkt oder über einen oder mehrere MTAs zu. Der SMTP-Proxy verhindert damit, dass der MTA direkt mit dem unsicheren Netz kommunizieren kann.

Der wohl beliebteste MTA ist unbestritten Sendmail. Er wird aufgrund seiner hohen Skalierbarkeit vielerorts eingesetzt. Sendmail ist jedoch auch für seine Vielzahl von Sicherheitslücken und Implementierungsfehlern bekannt.
Ein SMTP-Proxy verarbeitet daher nur die folgenden Befehle nach RFC 821 (SMTP - Simple Mail Transfer Protocol), die nicht sicherheitskritisch sind:

- HELO
- MAIL
- RCPT
- DATA
- QUIT
- RSET
- NOOP.

Einige weitere Befehle werden mit Standartantworten quittiert:

- HELP
- VRFY
- EXPN.

Bei Nutzen eindeutig sicherheitsrelevanten Befehlen wie

- DEBUG

wird eventuell direkt der Security-Manager mit einer Nachricht informiert. Da die Befehle zuerst den SMTP-Proxy durchlaufen müssen, kann der DEBUG-Befehl einfach ignoriert werden, um den potentiellen Angriff durch eine Suche nach Implementierungsfehlern zu unterbinden. Durch die Verwendung des Store-and-Forward-Prinzips wird eine Entkoppelung des komplexen und fehlerbehafteten MTAs, in diesem und vielen Beispielen Sendmail, erreicht. Sendmail wird nicht direkt mit Befehlen angesprochen, sondern nur die stellvertretende Software des SMTP-Proxies. Der SMTP-Proxy ist überschaubar und damit eine gut testbare Software.

Im Logbuch des Application-Gateways können durch das SMTP-Proxy die folgenden Protokolldaten für eine spätere Auswertung festgehalten werden:

- IP-Adresse und Rechnername des Quell-Rechnersystems
- Uhrzeit und Datum des Verbindungsaufbaus
- Absender der Nachricht (wie im Mail-Header angegeben)
- Empfänger der Nachricht (wie im Mail-Header angegeben)
- Anzahl der übertragenen Bytes
- Uhrzeit und Datum des Verbindungsabbaus

Benutzerorientierte Application-Level-Proxies

Der im Folgenden als Beispiel angeführte **HTTP-Proxy** steht hier stellvertretend für benutzerorientierte Proxies (weitere wären telnet oder ftp). Diese führen selbst eine Authentifikation mit dem entsprechenden Benutzer durch. Im Falle einer erfolgreichen Identifikation und Authentifikation des Anwenders beim Proxy gilt diese nur für jenen speziellen Proxy. Für das Nutzen eines anderen Dienstes/Proxies, muss sich der User erneut authentifizieren. Benutzerorientierte Proxies haben den Vorteil, dass die Zuordnung zwischen Benutzer und IP-Adresse und dem gewünschten Dienst eindeutig und lückenlos ist. Der HTTP-Proxy ist für die kontrollierte Kommunikation über das Hypertext Transfer Protocol verantwortlich und stellt entsprechende Sicherheitsfunktionen für diesen Dienst zur Verfügung.

Der Verbindungsaufbau erfolgt vom Quell-Rechnersystem auf TCP-Port80 (HTTP) des Application-Gateways. An diesem well-known Port für HTTP übernimmt der HTTP-Proxy automatisch die Verbindung. Der Benutzer auf dem Client identifiziert und authentisiert sich unter der Angabe des Verbindungsziels beim HTTP-Dienst. Wurde diese zweite Phase erfolgreich abgewickelt, wird ein individuelles Benutzerprofil aktiviert, welches sich aus folgenden Punkten zusammensetzt:

- IP-Adresse des Quell-Rechnersystems, das die Verbindung aufbauen möchte
- Benutzername, mit dem die Authentifikation erfolgte
- IP-Adresse des Ziel-Rechnersystems.
- Nun baut der HTTP-Proxy vom Application-Gateway eine zweite Verbindung zum TCP-Port80 (HTTP) des eigentlichen Ziel-Rechnersystems auf. Jetzt kann der Benutzer mit seinem Browser den Dienst des Ziel-Hosts unter transparenter Einwirkung des Application-Gateways nutzen.

Da das HTTP-Protokoll nicht session-orientiert arbeitet, ist der HTTP-Proxy dementsprechend auch nicht in der Lage, das Ende einer Sitzung zu erkennen. Bei jeder Anforderung einer WWW-Seite wird eine Verbindung über das Firewall-System aufgebaut, die Dokumente übertragen und die Kommunikationsverbindung wieder abgebaut. Beim ersten Mal wird vor der Übertragung die Authentifikation durchgeführt. Aus diesem Grund wird ein Timer gesetzt, der den Beginn der Session festhält. Nach Ablauf dieses Timers wird der HTTP-Proxy automatisch abgeschaltet. Bei jeder weiteren Kommunikation über den HTTP-Proxy wird der Timer nach erfolgreicher Authentifikation erneut gesetzt.

Der Kommando-Filter analysiert und überprüft die verwendete Methoden (FTP, HTTP, NNTP, SMTP, ...) und die dafür verwendeten Befehle (z.B. put, get, post). Jeder Versuch, eine nicht zulässige Anforderung abzusetzen, wird ihm angezeigt und es erfolgt der entsprechende Eintrag in die Protokolldateien. Es können auch spontane Benachrichtigungen vom Security-Management durchgeführt werden, falls grobe Regelverstöße registriert worden sind.

Mit der Hilfe eines Daten-Filters im HTTP-Proxy können definierte URLs zugelassen oder geblockt werden. So können zum Beispiel nur bestimmte Top-Level-Domains (z.B. ch und de) ansprechbar sein. Durch den Daten-Filter können jedoch auch bekannte, nicht gewünschte Dateien oder HTTP-Seiten durch den Proxy herausgefiltert werden.

Aktive Inhalte innerhalb von HTML-Dokumenten können eine Gefährdung für einen Host darstellen, der das Dokumentiert interpretieren soll. Hier kommt Content-Security ins Spiel, die vor solchen schädlichen Webapplikationen schützen soll. Ein Applet-Filter kann Java, JavaScripts und ActiveX kontrollieren oder ausschließen. Durch den so genannten Malware-Filter können korrupter Programmcode (z.B. Viren, trojanische Pferde, Würmer, ...) aufgespürt und ihnen durch externe Lösungen (z.B. Antiviren-Software) entgegengewirkt werden.

In der Logdatei des Application-Proxies für HTTP können die folgenden Protokolleinträge standartmässig vorgenommen werden:

- IP-Adresse und Rechnername des Quell-Rechnersystems
- IP-Adresse und Rechnername des Ziel-Rechnersystems
- Uhrzeit und Datum des Verbindungsaufbaus
- Name des Benutzers
- Anzahl der übertragenen Bytes
- Name der übertragenen Dateien oder übertragenen HTML-Seite
- Uhrzeit und Datum des Verbindungsabbaus

5. Schwachstellen und Angriffspunkte

Grundsätzlich gilt für alle Firewall-Systeme, dass es zweierlei Arten von Angriffen gibt:
Zum einen Angriffe der Form, dass die Firewall übergangen wird, dass also Verkehr hindurchgeschleust wird, der eigentlich geblockt werden sollte.
Zum anderen Angriffe auf das Firewall-System selbst, also die Plattform, auf der die Firewall-Software realisiert wurde. Deshalb sollte hier eine bedachte Auswahl erfolgen. Beispiele aus der Praxis für sichere Betriebssysteme:

- **OpenBSD**

  OpenBSD ist ein Un*x-Derivat, welches sich rühmt, seit bald fünf Jahren keine aus der Ferne ausnutzbare Sicherheitslücke aufzuweisen, was auf eine sehr zurückhaltende Auswahl bei der Aufnahme von neu entwickelten Systemkomponenten in den Quellcode zurückzuführen ist sowie auf eine vor einiger Zeit durchgeführte akribische Kontrolle des gesamten Quellcodes auf Sicherheitslücken hin.

- **Debian GNU Linux**

  Debian GNU Linux gilt als eine sehr sichere Linux Distribution, da sie einem versierten Benutzer erlaubt, jeden Bestandteil detailliert zu konfigurieren und nur minimale automatische Einstellungen durchführt. Es werden zwar sehr häufig so genannte „Vulnerabilities“ bekannt, diese werden jedoch sehr schnell gefixed. Mit entsprechendem Aufwand ist also eine sehr sichere Plattform realisierbar.

- **proprietäre Systeme**

  Manche Hersteller haben eigene Betriebssysteme entwickelt, allen voran Cisco bei seiner PIX Firewall. Proprietäre Betriebssysteme haben den Vorteil, dass sie nur aufs nötigste reduziert sind und somit von vorn herein weniger Angriffspunkte bieten. Jedoch ist man hier allein auf die sorgfältige Arbeit der Entwickler angewiesen, die stetige Überprüfung durch die Community wie bei OpenBSD oder Debian GNU fehlt hier.

- **Windows**

  Windows-Betriebssysteme, egal in welcher Version, können **nicht als sicher** bezeichnet werden. Zwar hat sich auch hier in den letzten Jahren einiges getan, jedoch hat der Benutzer nur völlig unzureichende Kontrolle über das Betriebssystem selbst. Des Weiteren hat Microsoft nicht gerade Schnelligkeit und Zuverlässigkeit im Umgang mit Sicherheitslücken gezeigt. Hinzu kommt, dass die am weitesten verbreitete Betriebssystemfamilie natürlich entsprechend viele Anwender hat, die sich mit der Ausnutzung der Lücken beschäftigen.

5.1 Personal Firewalls

Das Werbebanner-Einblendungsprogramm der Firma Aureate wird vom Setup-Programm der Wirtsanwendung kurzerhand als Netscape Navigator bzw. Internet Explorer **Plugin** installiert. Als Plugin kommuniziert es nicht *direkt* mit dem Internet, sondern nutzt die Plugin-Schnittstelle des Browser dafür.
Dadurch umgeht es auf einfache Weise die Probleme, die sonst bei einem Internetzugang über ein Netzwerk auftreten würden. Aureate ist nicht nur um einiges älter als ZoneAlarm, sondern wird von sehr vielen Freeware/Shareware-Programmen wie zum Beispiel von GoZilla und WebCopier verwendet.
Anscheinend greifen neuere Versionen des Werbeflächen-Tools über die Wirtsanwendung auf das Internet zu, sofern es sich bei diesen Programmen um "Internet-Software" handelt.
Suchstichwörter: advert.dll, Radiate

Weiterhin gibt es **HTML-Dateien** im TEMP-Verzeichnis, die ein paar Bilder aus dem Internet laden. Die Bild-URLs sind dabei ganz besonders aufgebaut: Einige bestehen unter anderem aus den Dateinamen, die man unter {Start | Dokumente} findet.
Der Realplayer kommuniziert übrigens auf diese Weise unter Umgehung der Firewall mit dem Internet (auch wenn die Personal Firewall die "normale" Kommunikation erkannt und unterbunden hat):

C:\WINDOWS\TEMP\RN7080.htm
|<HEAD>
|<META HTTP-EQUIV="refresh" CONTENT="0;URL=http://presets6.real
|.com/sitesmenu/rphurl.html?xx00xx00x0X0xxx00xXxxxx0x0xxxxxxxXx
|xxxxxxxx0xxxxxx0xx0xxxxxxXxxxx0x00xx0x0xx00xxx0xxxxx0X0X0x0X0x
|xx0X0xxx0X0000xx0xxx0X0xxx0X0xxx0X0X0x0X0Xxx00xxxxXxxx0xx0x0x
|xx0xx0x00xxxxX00xxXx0xXxxxx0xxXx0X0X0x0x00x00x0Xxxx">
|</HEAD>
(“x” und “0” repräsentieren Zahlen und Buchstaben)

Namen sind Schall und Rauch:
Do you want Microsoft Internet Explorer to access the internet?
Do you want Netscape Navigator to access the internet?
Do you want Microsoft Windows 95 to access the internet?
Do you want DFÜ-Netzwerk to access the internet?
Do you want Zone Alarm to access the internet?

Vernünftig programmierte Spyware wird sich ja kaum als "The ultimative hacking tool" in Windows anmelden.

Protokoll-Tunnel: zum Beispiel IP over E-Mail oder http. Für einen HTTP-Proxy sieht das wie eine ganz normale Web-Seiten-Anforderung aus. Theoretisch kann man jedes Protokoll über jedes andere tunneln, solange man Einfluss auf eine entsprechende Gegenstelle hat. Bei DNS-Abfragen zum Beispiel geht das auch über "viele Ecken".

Jedes Programm hat unter Windows 9x die **Zugriffsrechte** auf derselben Ebene wie die PF mit dem Netzwerk zu kommunizieren (also nebenher). Unter Windows NT (2000, XP) gilt das gleiche, wenn man sich als "Administrator" angemeldet hat; z.B. um Software im guten Glauben zu installieren. Happy99 und Hybris kommen dem recht nah, in dem sie die WSock32.dll ersetzen. Mittlerweile gibt es einen Proof-Of-Concept:
http://www.securityfocus.com/archive/1/244026 (Englisch)

Inzwischen gibt es auch die ersten bösen Programme [tm], die einige Personal Firewalls (bzw. bestimmte Versionen) **einfach beenden**:
http://groups.google.com/groups?selm=3B3B7A4E.C2EF27C4%40hrz.tu-chemnitz.de
http://de.geocities.com/pseueq/y3k.htm
Theoretisch dürften im Worst Case (also wenn das böse Programm [tm] Administrator- bzw. Root-Rechte auf dem Rechner hat) alle PFs ziemlich machtlos sein.

Außerdem kann man viele Personal Firewalls durch ähnlich aussehende Programme **ersetzen**, indem man im simpelsten Fall den Treiber-Aufruf in der Registry löscht und den Aufruf des User-Frontends mit dem Dateinamen eines entsprechend präparierten Programmes überschreibt.

Normale Benutzerrechte reichen bei den meisten PFs, um **neue Regeln** einzufügen. Natürlich ist das auch automatisch möglich:
http://www.heise.de/newsticker/data/pab-18.05.01-001
http://my-forum.netfirms.com/zone/zcode.htm (bestätigt "Yes-Button")

Und zum Schluss sind da noch die bösen Programme, die überhaupt nicht mit dem Internet kommunizieren. z.B.: Ein Trojanisches Pferd, das angeblich ein Virenscanner ist (und auch wirklich andere Viren findet) allerdings zusätzlich Zifferndreher in Excel-Tabellen verursacht. Oder das böse Programm ersetzt die Telefonnummer, die an das Modem bzw. die ISDN-Karte gesendet wird, durch eine 0190-Nummer.

Von außen gibt es grob gesagt drei Möglichkeiten, Zugriff auf ein System zu erlangen:

Eine **Fehlkonfiguration**, bei der zum Beispiel die Datei- und Druckerfreigabe nicht nur an die lokale Netzwerkkarte, sondern auch an das Internet-Interface gebunden ist. (Bei diesem Beispiel sind Personal-Firewalls recht erfolgreich).
Viele Fehlkonfigurationen spielen sich auch auf Anwendungsebene ab; zum Beispiel im Browser oder Mailprogramm. Leider sind solche Geschichten (z.B. automatisches Starten von Programmen) häufig die Standard-Einstellung, die man erst mehr oder weniger mühsam ändern muss.
Hier sieht es mit dem Schutz durch PFs sehr schlecht aus, da die PF nicht mitbekommt, wenn ein Programm ungewollt Daten verändert oder löscht. Und wenn das Mailprogramm plötzlich Mails senden will, dann wird die PF es nicht davon abhalten.

Die zweite große Möglichkeit besteht in der Ausnutzung von **Bugs**.
Ein häufiges Angriffsszenario entsteht, wenn ein Programm die Länge eines Speicherbereiches beim Kopieren/Einlesen nicht prüft und über das Ende seines Puffers hinausschreibt. Wenn in einer der dahinterliegenden Speicherzellen ein Verweis auf eine Speicheradresse mit Programmcode liegt (Rücksprungadresse bei Funktionen), dann kann diese überschrieben werden. Im simpelsten Fall steht dann dort Müll und es gibt einen "Fehler in Anwendungsprogramm". Mit etwas Mühe ist es in dieser Situation häufig möglich, in die eigenen Daten zu springen, die in Wirklichkeit Programm-Anweisungen in Maschinensprache sind.

Technische Hintergrundinformationen dazu:
"Smashing The Stack For Fun And Profit" (Englisch)
http://www.phrack.org/phrack/49/P49-14

Eine Personal Firewall könnte theoretisch solche "zu langen" Daten erkennen und abfangen. Das geht allerdings nur, wenn sie weiß, wonach sie suchen muss. Bevor die Personal Firewall-Hersteller ihre Programme angepasst haben, hat MS (bzw. die Hersteller des fehlerhaften Programms) idR. ihre Sicherheitspatches schon längst veröffentlicht.

Zu guter Letzt könnte noch eine **Fernwartungssoftware** oder ein anderes böses Programm [tm] gestartet sein; wahrscheinlich im Glauben ein nützliches Programm zu installieren.
Allerdings machte es neben den bunten Effekten auf der Webseite (ActiveX) im Hintergrund noch andere Sachen. Oder die Mail von einem Bekannten ist in Wirklichkeit von dem Wurm verschickt worden, den er im gleichen Irrglauben gestartet hat.
Dagegen sind PF ebenfalls machtlos:
Es befindet sich in diesem Fall bereits ein Programm auf dem Rechner, das die PF so verändern kann, dass sie Verbindungsaufbauten von außen ohne Rückfrage annehmen. Oder noch leichter: Es baut selbst eine Verbindung nach außen auf und holt sich seine Befehle ab. (Damit fällt es in den ersten Abschnitt).

5.2 Kommerzielle Produkte

Im folgenden werden einige ausgewählte Attacken beschrieben, die von richtig konfigurierten Firewalls teilweise aufgehalten werden können. Auf technische Details zur Durchführung wurde bewusst verzichtet.

IP-Spoofing

Bei dieser Art der Attacke wird mit Hilfe einer falschen IP-Adresse dem angegriffenen System eine falsche Identität vorgetäuscht. Die gegenseitige Identifikation zweier kommunizierender Netze (Systeme) erfolgt bei den meisten TCP/IP-Protokollen ausschließlich über die IP-Adresse. Im Internet sind jedoch sehr viele "Hackertools" als Freeware erhältlich, die es ermöglichen, eine falsche IP-Adresse vorzutäuschen.

Um IP-Spoofing abzuwehren, sollten Paketfilter von außen kommende Pakete, die interne Adressen als Absender angegeben haben, abweisen. Ferner sollten Dienste, deren Authentisierungskriterium IP-Adressen sind, gesperrt oder restriktiv behandelt werden. Mit der derzeitigen Implementierung des Internet-Protokolls ist es nicht möglich, Pakete vollständig zu eliminieren, die durch IP-Spoofing entstanden sind. Die Bedrohung durch IP-Spoofing kann lediglich reduziert werden. Ausgehende Pakete, die als Quelladresse keine interne Adresse besitzen, sollten ebenfalls abgewiesen werden. Somit wäre z.B. eine Zurückverfolgung von TCP-SYN-Attacken möglich, falls alle Netzprovider dies befolgten.

Datenpakete mit extravaganten TCP-Headern

Diese Angriffsmethode ist sehr simpel. Der Angreifer schickt einem der Netzserver des zu attackierenden Netzes einen unbekannten Paketheader. Der Server interpretiert diesen Header falsch, und wird so zu unvorhergesehenen Reaktionen verleitet. In Folge dieser Reaktionen ist es dem Angreifer dann möglich in das System einzudringen.

Missbrauch des Source-Routing

Einem IP-Paket lässt sich die Route, die es nehmen soll, um ans Ziel zu gelangen, vorschreiben, genauso wie die Route, den das Antwortpaket zu nehmen hat. Während der Übertragung besteht die Möglichkeit, die Wegbeschreibung zu manipulieren, so dass nicht der vorgeschriebene, sichere Weg (z.B. über die Firewall) genommen wird, sondern ein oder mehrere unkontrollierte Wege.

Missbrauch des ICMP-Protokolls

ICMP steht für das Internet-Controll-Message-Protokoll. Es hat die Aufgabe Fehler- und Diagnosefunktionen zu übermitteln. Leider lässt es sich zum Ändern der Routingtabellen missbrauchen, so dass z.B. nicht geschützte Routen benutzt werden. Oder der Angreifer schleust über diesen Weg gefälschte destination-unreachable-Pakete in eine bestehende Verbindung, um diese zu unterbrechen.

Missbrauch der Routing-Protokolle

Routing-Protokolle haben die Aufgabe, zwei vernetzten Systemen evtl. Routenänderungen mitzuteilen. So ist es möglich, mit einer dynamischen Routingtabelle zu arbeiten. Für einen Angreifer ist es aber möglich falsche RIP-Pakete (Route-Information-Protokoll)zu erzeugen, und so die Systeme zu veranlassen, ungewünschte Routen zu nehmen.

TCP-SYN-Flooding

TCP-SYN-Flooding kann auf zwei Arten für denial-of-service-Angriffe auf Server verwendet werden. Zum einen kann ein Angreifer auf Basis von IP-Spoofing halboffene Verbindungen etablieren. Der Angreifer-Client sendet SYN, der Opfer-Server antwortet mit SYN-ACK, aber der Client bestätigt nicht mit ACK. Solange diese Bestätigung fehlt, ist die Verbindung also halboffen. Bei einer gewissen Anzahl von halboffenen Verbindungen ist der Server nicht mehr in der Lage, neue Verbindungen anzunehmen. Dieser Angriff kann verhindert werden, indem z.B. eine Firewall IP-Spoofing nicht zulässt (siehe Abschnitt 3.1). Zum anderen ist TCP-SYN-Flooding auch ohne IP-Spoofing möglich, indem der Angreifer als Quelle eine Adresse angibt, zu der das SYN-ACK nicht geroutet werden kann, weil sie nicht existiert, oder für den Server nicht erreichbar ist. Um dies abzuwenden, können zustandsabhängige Filter nur eine begrenzte Anzahl von SYN-Paketen zulassen.

Fragmentation Attack

Der Fragmentierungsangriff zielt darauf ab, die Implementierung der IP-Fragmentierung derart auszunutzen, dass Filterregeln von Paketfiltern nicht auf die fragmentierten IP-Pakete passen und somit diese Fragmente durchgelassen werden. Im [RFC 1858] wird der **Tiny Fragment Attack** beschrieben. Dabei zerlegt der Angreifer seine Nutzdaten (z.B. TCP-Pakete) in sehr kleine Fragmente. Dadurch wird z.B. ein TCP-Header auf mehrere Fragmente aufgeteilt und kann daher von statischen Paketfiltern nicht analysiert werden. Eine Filterung aufgrund von Regeln, die z.B. den TCP-Header betreffen, ist nicht mehr möglich. Als Gegenmaßnahme sollten Implementierungen von Paketfiltern verwendet werden, bei denen die Länge des Tiny Fragments nicht verändert werden kann.

Ein zweiter Angriff, der die IP-Fragmentierung ausnutzt, ist der so genannte **Overlapping Fragment Attack** [RFC 1858]. Die derzeitige Internet-Protokoll-Spezifikation [RFC 791] beschreibt einen Reassemblierungsalgorithmus, der neue Fragmente produziert und dabei jeden überlappenden Teil der zuvor erhaltenen Fragmente überschreibt. Wird ein solcher Algorithmus angewendet, so könnte ein Angreifer eine Folge von Paketen konstruieren, in denen das erste Fragment (mit einem Offset der Länge Null) harmlose Daten beinhaltet (und dadurch von einem Paketfilter weitergeleitet werden kann). Ein beliebiges nachfolgendes Paket mit einem Offset, der größer als Null ist, könnte TCP-Header-Informationen (z.B. destination port) überlappen.

Diese würden durch den Algorithmus modifiziert (überschrieben) werden. Dieses zweite Paket wird von vielen Paketfiltern nicht gefiltert. Gegenmaßnahme hierzu ist, Paketfilter zu verwenden, die ein Minimum an Fragment-Offset für Fragmente verlangen.

Tunneln

Durch die Fragmentierung von IP-Paketen existiert wenig Information, auf die man eine Filterentscheidung basieren kann, denn bis auf das erste Fragment besitzen die Fragmente keine Portnummern. Besitzt ein Angreifer einen Komplizen im zu schützenden Bereich, so können Fragmente ohne Portnummern zum Tunneln einer Firewall verwendet werden. Das erste Fragment beinhaltet zwar die Portnummer und kann entsprechend gefiltert werden. Wird dies nicht weitergeleitet, so ist das Paket im Inneren unvollständig, und die übrigen Fragmente werden schließlich vom Zielknoten verworfen. Ist der Zielknoten allerdings im Besitz eines Komplizen des Angreifers, so kann dieser die Fragmente zusammensetzen. Analog kann der Komplize im zu schützenden Bereich gefälschte Fragmente ohne Portnummern erstellen, die vom externen Komplizen auf einer äußeren Maschine zusammengesetzt werden können. Um diesen drohenden Infomationsverlust abzuwenden, sollten also Fragmente ohne Portnummern von innen nach außen und umgekehrt bereits an der Firewall verworfen oder dort zusammengesetzt und überprüft werden. Dies ist mit zustandsabhängigen Filtern bzw. Paketfiltern, die Kontext speichern können, realisierbar.

Grenzen einer Firewall

Keine Firewall kann einen absoluten Schutz gegen ein Eindringen in ein System gewährleisten. Sie kann lediglich einen Einbruch so schwer wie möglich, und damit auch so unwahrscheinlich wie möglich machen. Dies bedeutet, dass eine Firewall zwar großen Schutz, jedoch keine absolute Sicherheit bietet.

Sie kann Angriffe oder deren Versuche auf niedriger Protokollebene feststellen, meist abblocken, teilweise auch protokollieren und zurückverfolgen, gegen Angriffe auf höherer Ebene ist sie jedoch nutzlos (Sie kann z.B. Virenbefall nicht verhindern, denn dazu müsste jedes einzelne Datenpaket zeitaufwendig untersucht werden).

Gegen Angriffe, die aus dem internen Netz heraus verübt werden, kann eine Firewall ebenfalls nicht schützen, es sei denn, ein bestimmter Teil des internen Netzes ist durch eine Firewall geschützt. Schutz kann die Firewall auch nur dann bieten, wenn die gesamte Kommunikation mit externen Systemen über diese Firewall abgewickelt wird.

Ein einziger Kommunikationskanal (z.B. Modem) reicht aus, um das gesamte Netz zu gefährden, da so die Firewall umgangen werden kann. Aus dem o.a. Argument geht eindeutig hervor, dass das größte Risikopotential von den Menschen ausgeht, die mit und in dem Netz arbeiten, denn durch Unwissenheit und Leichtgläubigkeit können sie einem Angreifer viele Möglichkeiten des Eindringen öffnen. Leider hinkt die Entwicklung von Schutzmechanismen immer den Möglichkeiten des Angriffs hinterher, denn bevor die Entwickler einer Firewall einen Schutzmechanismus erstellen, müssen sie zuerst wissen, gegen welche Art des Angriffs geschützt werden muss.

6. Ruleset

Ein Ruleset besteht aus mehreren einzelnen Regeln, die eine Firewall bei jeder Anfrage in oder aus dem Internet/Netzwerk von oben angefangen nach unten abgearbeitet bzw. überprüft, und beim Zutreffen einer Regel anhält und diese ausführt, die darauf folgenden Regeln werden dann ignoriert.

Das Ruleset verarbeitet jede Kommunikation und filtert alle Datenpäckchen, die zwischen Computern z.B. im Internet / Netzwerk hin- und herlaufen.

Je mehr Regeln (bei den vorgegebenen Rulesets von den Herstellern meist ziemlich viele), desto mehr Speicher und Ressourcen braucht die Firewall. - Also Rulesets immer möglichst kompakt und übersichtlich halten!

Eine Regel ergibt sich aus folgenden Komponenten:

1.) Einer **Richtung** (=Direction), von wo aus eine Verbindung "aufgebaut" werden darf also von "drinnen" oder "draußen".

- OUTGOING - Aufbau einer Verbindung durch ein TCP_ACK Signal von innen
- INCOMING - Aufbau einer Verbindung durch ein TCP_ACK Signal von außen
- oder beides: BOTH

2.) Einem **Protokoll**: Das Internet basiert auf verschiedenen Protokollen, die für die Übertragung den Datenpäckchen zuständig sind.
Die wichtigsten von den meisten Firewalls unterstützten Protokolle sind TCP, UDP, ICMP, (...)

3.) Einer **Quelle**: Also woher eine Anfrage stammt:
- Host (=eine IP),
- Port (=von der Firewall zu öffnende & schließende "Tore“),
- Service/Application (=Programm)

4.) Einem **Ziel** (wohin das Packet soll):
- Host (=eine IP),
- Port (=von der Firewall zu öffnende & schließende "Tore“.),
- Service/Application (=Programm)

5.) Einer **Erlaubnis**: Blockieren (Block / Deny) oder Erlauben (Permit)

6.) Als letztes haben die meisten Firewalls noch eine Option, eine **Log-Datei** zu führen.

No	Type	Rule Description	Direction	Protocol	Local Port	Application	Remote Host	Remote Port	Erklärung der Regeln
1	Permit	ICMP	Outgoing	ICMP + Echo Request	Any	Any	Any	Any	ICMP ist in mehrere Unterbereiche aufgeteilt, siehe unten. Erlaubt ist hier der Echo Request, womit ihr andere Leute anpingen könnt!
2	Permit	ICMP	Incoming	ICMP + Echo Reply + Destination Unreachable + Time Exeeded	Any	Any	Any	Any	ICMP ist in mehrere Unterbereiche aufgeteilt, siehe unten. Erlaubt ist hier Echo (die Anwort auf euren Ping) sowie Ziel nicht erreichbar und Zeit abgelaufen! Der Rest wird unten in der letzten Regel geblockt!
3	Permit	Corporate DNS	Both	UDP	Any	Any	Corporate Name Servers IP's (DNS)	53	Diese Regel erlaubt deinem Computer sich mit dem Name-Server (die eine IP in einen Hostnamen auflösen und umgekehrt) deines Providers zu verbinden. Die IP's der Name Server findest du meist unter den Zugangsdaten für MacOs bei deinem Provider auf der Homepage. Trage diese unter Trustfull Addresses ein.
4	Permit	Loopback	Both	UDP/TCP	Any	möglichst explizit deine Internetprogs	127.0.0.1	Any	Da diese Loopback Regel eine sehr wichtige ist, gibt es mehr Informationen weiter unten.
5	Permit	NetBT Datagram	Both	UDP	137, 138	Any	Trustful addresses	Any	Diese Regel erlaubt NetBios (Windows Neighborhood protocol) datagrams (UDP) die bei jedem Computer im lokalen Netzwerk gesendet werden um sich zu identifizieren. Als vertrauenswürige Remote/Locale-Adresse die IP und Subnetzmaske der Rechner im Netwerk angeben; z.B. : Lokales Netz = Trustful addresses z.B. 172.23.0.0 / 255.255.0.0
6	Permit	NetBT Session	Outgoing	TCP	Any	Any	Trustful Addresses	139	Diese Regel erlaubt , das Windows Ressourcen im lokalen Netzwerk nur mit den "Custom Address Group", also vertrauenswürdigen IP's teilt, die du in deiner Firewall eintragen kannst.
7	Permit	NetBT Session	Incoming	TCP	139	Any	Trustful Addresses	Any	Diese Regel erlaubt "vertrauenswürdigen Computern" im Netzwerk auf deine Windows Ressourcen zuzugreifen.
8	Permit	Internet Browser	Outgoing	TCP	Any	Dein Browser	Any oder IP's deines Proxys	80, 443	Diese Regel erlaubt deinem Browser Zugang zum Internet (HTTP=Port80, HTTPS=Port443)
9	Permit	Mail/ Program	Outgoing	TCP	Any	Dein Mail Client	IP's deines Mailservers z.B. pop.gmx.net & mail.gmx.net	110, 995, 25, 143, 119	Diese Regel erlaubt deinem Mailclient Zugang zu deiner Mailbox Port 110 = POP3-Protokoll oder Port 995 = SPOP3-Protokoll (<-falls du deine Mails über eine sichere Verbindung prüftst) sowie Port 143 = IMAP falls du deine Emails über das IMAP Protikoll abholst. Port 25 ist für SMTP also das versenden von Mails zuständig. Port 119 = NNTP wird für Newsübertragung verwendet und somit nur benötigt, falls du Newsdienste abonnomiert hast.

10	Permit	Newsclient	Outgoing	TCP	Any	Dein Newsprog	IP's deines News-servers	119	Diese Rule ist nur nötig, falls du mit einem Newsprogramm oder deinem Emailprogramm in Newsgroups liest! Port 119 steht für NNTP
11	Permit	Download Butler	Outgoing	TCP	Any	Dein Filefetcher	Any	80, 21	Diese Regel erlaubt deinem Download Manager Zugang zum downloaden.
12	Permit	Download Butler	Incoming	TCP	Any	Dein Filefetcher	Any	20	Diese zweite Verbindung (Datenverbindung) ist für "FTP-Data" bzw. für die Übertragung der Daten zuständig. Ftp-Data=Port 20
13	Permit	Instant Messenger	Outgoing	TCP	Any	Dein Instant Messenger	Any oder IP's zum Terminal	5190	Diese Regel erlaubt deinem Instant Messenger z.B. AIM Zugang zum Internet; über den AOL-Port=5190
14	Permit	Telnet Client	Outgoing	TCP	Any	Dein Telnet Progi	Any	23	Diese Regel erlaubt deinem Telnet Programm kompletten Zugriff auf andere Server im Internet. Ebenfalls löschen, falls du Telnet nicht benutzt.
15	Permit	FTP Client	Outgoing	TCP	Any	Dein FTP Progi	IP's von deinem FTP Server / Website	21	Diese Regel erlaubt deinem FTP Client Zugriff auf zum Beispiel deine Website, wo du was uploaden möchtest. Über Port 21 wird die Steuer-verbindung auf-gebaut, auf der FTP-Befehle und Parameter zwischen Client und Server ausgetauscht werden! FTP=Port 21
16	Permit	FTP Client	Incoming	TCP	Any	Dein FTP Progi	IP's von deinem FTP Server / Website	20	Diese zweiteVerbindung (Datenverbindung) ist für "FTP-Data" bzw. für die Übertragung der Daten zuständig. Ftp-Data=Port 20
17	Permit	(...) <- Hier kannst du individuelle Regeln für hier nicht angeführte Programme z.B. nach der Anleitung unten einbauen. -> (...) Übrigens hier vorgeschlagende Regeln für Programme die du nicht benutzt, brauchst du natürlich nicht in deiner Firewall...							
18	Block	Block	Incoming	Any (UDP/TCP & ICMP)	Any	Any	Any	Any	Diese Regel gehört als letztes abgearbeitet, da sie alle Incoming Verbindungen die nicht zuvor erlaubt worden sind blockiert!

Kurzerklärung ICMP:

Neben dem Internetprotokoll (IP) sind auf der Internetschicht noch weitere Protokolle angesiedelt. Eins davon ist ICMP, es dient zum Austausch von Fehler- und Informationsmeldungen bei IP-, TCP- und UDP-Protokollen. ICMP-Pakete werden immer als IP-Datagramm verschickt. Es dient dazu, Hosts günstigere Routen zu einem Ziel bekanntzugeben, über Routing-Probleme zu informieren oder Verbindungen wegen Problemen im Datennetz abzubrechen. Auf ICMP basieren die Kommandos ping und traceroute. Die Meldungen des ICMP sind in zwei Klassen eingeteilt: die Fehlermeldungen und die Informationsmeldungen.

Bei ICMP muss genau abgewogen werden zwischen maximaler Sicherheit und Komfort bzw. Performance.

Kurzerklärung Loopback:

Die IP 127.0.0.1 existiert nur auf dem eigenen Computer, sprich diese Verbindungen spielen sich nur auf dem eigenen Computer ab! Problem dabei ist, das jedes Programm z.B. den Internetexplorer benutzen kann um sich mit dem Internet zu verbinden!

Wichtig also ist, dass man nur den Programmen, diese Verbindung erlaubt, die ins Internet dürfen und es anfragen! Zum Beispiel der Internet Explorer selbst braucht das Loopback nur auf UDP Ebene Outgoing; die meisten anderen Programme wie z.B. der Netscape Navigator oder dein Emailprogram kommen ohne ein Loopback aus!

7. Quellen

Besondere Erwähnung gebührt hier
Marc Ruef (marc.ruef@computec.ch), aus dessen Text „Einführung in Firewallsysteme“ wesentliche Teile der Kapitel 1 und 4 entnommen wurden,
Hendrik Brummermann (HendrikUsenet@nexgo.de), dessen Text nahezu unverändert den Abschnitt „5.1: Schwachstellen von Personal Firewalls“ darstellt sowie
Johan Lukas (fetzmail@gmx.de) von http://faq.at/firewalls/, dessen Beschreibung des Regelsets unter „6. Ruleset“ verwendet wurde.
Diese Texte sind so gut und anschaulich, dass man es kaum besser schreiben könnte.

Weitere Quellen:

Firewallsysteme – Konzeption, Implementation, Audit
Thomas Veit
Cebit 2000

Installation, Test und Bewertung von Internet-Sicherheitsmechanismen
Diplomarbeit von Jürgen Mayerhofer und Christian Rotter
Fachhochschule Regensburg 12. Mai 1998

Studie “Gesicherte Verbindung von Computernetzen mit Hilfe einer Firewall”
Andreas Bonnard, Christian Wolff
Siemens AG 1997

Informationssicherheit in der industriellen Kommunikation
Siemens AG 1999

Firewall Routers and Packet Filtering
Gary Kessler
February 1995

Sowie s. “8. Internet Ressourcen”

Bildquellen:

http://www.kes.info/_archiv/_heftarchiv/images/firewall.jpg
http://www.das-erste.de/krimi2/hansen.jpg
http://www.itso.iu.edu/staff/krulewit/ddos/Slide8.jpg
Firewallsysteme - Cebit 2000 - Thomas Veit
http://www.cafe17.de/images/tipps_tricks/filter.gif
http://www.netsol.com/en_US/promotions/offers/images/welcome-email-photo.jpg
http://www.ic-2000.com/images/activex.gif

8. Internet Ressourcen

Mit welchen Ports arbeitet die Anwendung ...
http://www.practicallynetworked.com/sharing/app_port_list.htm

de.comp.security.firewall FAQ
http://www.iks-jena.de/mitarb/lutz/usenet/Firewall.html

FAQ: Firewall Forensics (What am I seeing?)
http://www.robertgraham.com/pubs/firewall-seen.html

Firewall Handbuch für LINUX 2.0 und 2.2
http://www.little-idiot.de/firewall/zusammen.html

Firewall FAQ
http://faq.at/firewalls

Wie Personal Firewalls ausgetrickst werden können
http://home.arcor.de/nhb/pf-austricksen.html

Port numbers
http://www.iana.org/assignments/port-numbers

How to bypass your personal firewall outbound detection
http://www.keir.net/firehole.html

Firewall-Grundlagen
http://www.tecchannel.de/internet/682/index.html

Firewalls Complete
http://www.quanmongmo.net/computer/firewall/

Ruleset für iptables
http://it-secure-x.net/sites/iptablesruleset.shtml

Konzeption und Realisierung eines Firewallsystems zur Internet-Intranet Kopplung
http://www.klaus.camelot.de/dip/

Firewallsysteme - Konzeption - Implementation - Audit
http://www.bsi.de/literat/tagung/cebit00/vt_071.htm

Einrichtung eines Routers unter SuSE 7.2 mit Firewall und DHCP mit Konfiguration der Clients.
http://www.gcf.de/directdl.php?id=36&dl=papers/router.pdf

Informationssicherheit in der industriellen Kommunikation
http://www1.ad.siemens.de/net/html_00/ftp/fachartikel/security_whitep_de.pdf

Firewall Studie des BSI
http://www.bsi.bund.de/literat/studien/firewall/fwstud.htm

9. Über den Autor

Malte von dem Hagen ist 24 Jahre alt, Student der IT-Sicherheit an der Ruhr-Universität Bochum und Werkstudent im Competence Center IT-Netze der RWE Systems AG in Essen.
Dort arbeitet er projektbezogen z.B. an einer Implementierung des CiscoSecure ACS Server und im Bereich sicheres Netzwerkdesign (DMZ, sicheres Router- und Switch-Management, VPN, Intrusion Detection).
Erfahrung in Erwachsenenbildung konnte er im notfallmedizinischen Bereich als Ausbilder für Erste Hilfe und für Sanitätsdienstfortbildungen sammeln.

Vorträge und White Papers im IT-Bereich:

„Sicherheitslücken im wireless LAN nach IEEE 802.11“ (RWE 2001)
„Die Problematik von vLANs als Sicherheitsfeature“ (RWE 2001)
„Sicherheitsstrategien für Rechenzentren und Serverparks“ (RUB 2002)
„Workshop Firewall“ (RWE 2002)

Interessengebiete:

Kryptographie-Algorithmen
Intrusion Detection
IP-Routing
Netzwerkdesign

Kontakt:

Malte von dem Hagen
Martin-Luther-Str. 121
45144 Essen

mobil: 0160-99018074
web: http://www.DocValde.net/
email: root@DocValde.net

Das vorliegende Script wurde für einen vierstündigen Workshop geschrieben, dessen Zielgruppe Auszubildende der „neuen IT-Lehrberufe“ im ersten / zweiten Lehrjahr waren. Dementsprechend kann nur begrenzt auf technische Details eingegangen werden. Für die Richtigkeit wird keine Gewähr übernommen. Tests der Sicherheit einer Firewall, egal aus welcher Motivation heraus, können rechtliche Konsequenzen haben! Sie sollten deshalb ausschließlich mit der Zustimmung des jeweiligen Netzbetreibers durchgeführt werden!