Stefan Wohlfeil

Sicherheit im Internet

Kurseinheit 4:
Anbietersicherheit im Internet

mathematik
und
informatik

FernUniversität in Hagen

Inhaltsverzeichnis

Diese Seite bleibt aus technischen Gründen frei!

Kapitel 4

Anbietersicherheit im Internet

In dieser Kurseinheit wird das Thema *Sicherheit* aus dem Blickwinkel desjenigen betrachtet, der für einen Rechner im Internet verantwortlich ist. Diese Person ist dafür zuständig, dass der Rechner im Internet verfügbar ist, d. h. dass seine Dienste für potentielle Benutzer zur Verfügung stehen. Weiterhin muss dafür gesorgt werden, dass an den Daten auf diesem Rechner keine unbefugten Manipulationen vorgenommen werden. Beispielsweise sollte kein Hacker auf dem Rechner neue Benutzer anlegen können, so dass er sich später immer wieder auf dem Rechner anmelden kann.

4.1 Einführung

Rechner im Internet bieten typischerweise verschiedene Dienste an, wie z. B. Informationsdienste oder auch Transaktionen im E-Commerce. Da sich jedermann mit diesen Rechnern verbinden kann, muss man sicherstellen, dass Angreifer keinen Schaden anrichten können. Abschnitt 4.2 befasst sich mit den Grundlagen der sicheren Konfiguration von Webservern.

Dienste

Neben den im Internet verfügbaren Rechnern haben insbesondere Firmen oder Organisationen ein eigenes internes Rechnernetz. Häufig setzt man in diesen internen Netzen dieselben Techniken (Protokolle, Dienste etc.) ein, die auch im Internet benutzt werden. Ein solches internes Netz nennt man auch **Intranet**. Die Rechner im Intranet enthalten vertrauliche Daten, die nicht für jedermann einsehbar sein sollen. Zugriffe von außen auf diese Daten sind unerwünscht. Andererseits sollen von diesen Rechnern aber auch Verbindungen ins Internet erfolgen, z. B. zum Versand von E-Mails, zur Informationsbeschaffung im WWW etc. Also muss eine gesicherte Verbindung zwischen Intranet und Internet existieren. Abschnitt 4.3 behandelt diese Schnittstelle, die man auch als „Brandschutzwand“ (engl. **firewall**) bezeichnet.

Intranet

Firewall

Zum Abschluss des Kurses werden organisatorische Aspekte von Sicherheit besprochen. Die Erfahrung hat gezeigt, dass technische Schutzmaßnahmen immer nur ein erster Schritt sind. Diese Maßnahmen müssen von Menschen angewendet und beachtet werden und hierbei entstehen häufig Risiken (z. B. Menschen kennen die Problematik nicht und schreiben Passwörter auf). In Abschnitt 4.4 wird darauf eingegangen, wie man diese Risiken mit geeigneten organisatorischen Maßnahmen vermindern kann.

4.2 Sichere Webserver

4.2.1 Betriebssystemunabhängige Aspekte

Egal welchen Rechner mit welchem Betriebssystem Sie verwenden, bestimmte Hinweise sollten Sie immer beachten. Dazu gehört, dass auf dem Rechner nur die unbedingt erforderlichen Programme installiert sind. Programme, die nicht benötigt werden, belegen unnötig Platz auf der Festplatte. Viel schlimmer ist aber, dass Angreifer diese Programme benutzen können und bei Sicherheitslücken in diesen Programmen auf ihrem Rechner Schäden anrichten können. Zu den nicht unbedingt erforderlichen Programmen auf einem Webserver gehören beispielsweise:

Minimales System

1. Grafische Benutzeroberflächen, wie beispielsweise X11.
2. Programme zur Software-Entwicklung, wie Compiler, Linker etc.
3. Treiber für Geräte (z. B. Modem), die gar nicht angeschlossen sind.

Bei der Standardinstallation eines Betriebssystems ist solch eine minimale Installation oft nicht gegeben. Das Betriebssystem ist gerne so konfiguriert, dass es *alle* Funktionen auf *jeder* potentiellen Hardware-Plattform erlaubt. Als Administrator eines Webservers sollten Sie bei der Installation also *nicht* die „Standard"-Installation benutzen, sondern das Betriebssystem und die Programme ganz konkret für ihr geplantes System einrichten.

Korrekte Konfiguration

In einem Betriebssystem existieren verschiedene Mechanismen, die die Verarbeitung eingehender TCP/IP-Pakete steuern. Anhand der Portnummer (siehe auch Abschnitt 1.3.3) muss das Betriebssystem entscheiden, welcher Prozess das Paket übergeben bekommt und es bearbeiten soll. Während des Starts des Rechners (engl. **to boot**) werden einige Dienstprozesse automatisch gestartet. Diese laufen ständig im Hintergrund und bearbeiten dann die TCP/IP-Pakete für „ihren" Port. Wenn der Rechner nur einen bestimmten Dienst anbieten soll, d. h. wenn er nur auf bestimmten Ports Pakete entgegen nehmen soll, dann sollten Sie alle anderen Ports sperren. Konkret bedeutet dies, dass die für diese anderen Ports zuständigen Prozesse auf dem Rechner *nicht* laufen sollten.

Port

Nachdem auf dem Rechner nun nur die unbedingt erforderliche Software installiert ist, sowie nur die Ports „offen" sind, die man braucht, sind trotzdem weitere Konfigurationshinweise zu beachten. Sie sollten nur die Benutzerkennungen (engl. **user id**) einrichten, die tatsächlich benötigt werden. Insbesondere „Testbenutzer", die evtl. automatisch erzeugt wurden und die vielleicht ein allgemein bekanntes Standardpasswort haben, sollten entfernt werden. Weiterhin können Sie für die einzelnen Dienste (WWW, ftp etc.) eigene Benutzerkennungen einrichten. Diesen Benutzern werden nur die Rechte eingeräumt, die für die Erledigung der Aufgabe erforderlich sind. Der WWW-Benutzer darf also nur die HTML-Seiten lesen, der ftp-Benutzer nur die ftp-Dateien usw. Die Serverprozesse laufen dann unter der Kennung des jeweiligen Benutzers und *nicht* unter der Administratorkennung.

Benutzer

Rechte

Sollte es einem Angreifer nun gelingen, einen Serverprozess zu beschädigen und dadurch auf eine Kommandozeile (engl. **shell**) zuzugreifen, dann hat er in der Kommandozeile keine Administratorrechte, sondern nur die eingeschränkten Rechte des „Serverbenutzers". So ist der Umfang des potentiellen Schadens

begrenzt. Solche Angriffe sind in der Vergangenheit häufig vorgekommen. Ein typisches Beispiel sind die sogenannten *Buffer-Overflow*-Angriffe. Sie nutzen aus, dass ein Prozess beim Einlesen einer Zeichenkette nicht prüft, ob diese in den vorgesehenen Speicherbereich passt. Ist der Bereich zu klein bzw. die Zeichenkette zu lang, so werden Teile des Programms überschrieben. Wird der überschriebene Teil dann ausgeführt, so kann der Angreifer damit einen Kommandointerpreter starten und beliebige Kommandos ausführen. Weitere Details zu möglichen Angriffstechniken werden in Kurs *(01867) Sicherheit im Internet 2* vorgestellt.

Buffer-Overflow

Sie sollten also darauf achten, dass sie keine veraltete Version eines Serverprogramms benutzen, das eine solche Schwachstelle vielleicht noch enthält. Über neue Sicherheitslücken in Programmen informieren die Hersteller i. d. R. umgehend. Außerdem gibt es Mailinglisten, die über gefundene Schwachstellen berichten. Häufig werden dort auch direkt die erforderlichen Änderungen (engl. **patches**) angeboten. Das Computer-Emergency-Response-Team (CERT) der Carnegie Mellon University bietet eine derartige Mailingliste. Weitere Informationen hierzu finden sie unter der URL

aktuelle Versionen

Patches

```
http://www.cert.org/
```

Unter dem Namen *Deutsches Forschungsnetz (DFN)* gibt es einen DFN-Verein und wird ein Kommunikationsnetz zwischen Hochschulen und Forschungseinrichtungen betrieben. Weiterhin wird dort ein CERT betrieben und die Details hierzu kann man unter dieser URL nachlesen:

```
http://www.cert.dfn.de/
```

Auch Hacker machen sich das Leben heute einfach. Statt von Hand alle potentiellen Schnittstellen zu prüfen, setzen sie Hilfsprogramme, sogenannte *Scanner* ein. Diese prüfen systematisch alle Schnittstellen eines Rechners auf potentielle Schwachstellen. Finden die Programme eine Lücke, so melden sie dies dem Hacker. Natürlich finden diese Programme nicht *jede* Schwachstelle. Selbst wenn das Programm nichts meldet, können trotzdem Probleme auftreten. Als Administrator sollten Sie sich diese Programme trotzdem besorgen (es gibt freie Versionen im Internet, z. B. *nmap* oder *OpenVAS*) und auf Ihr eigenes System „loslassen“. Da immer wieder neue Fehler in Serverprogrammen gefunden werden, sollten Sie diese Tests regelmäßig wiederholen und dabei die aktuellsten Hackerwerkzeuge einsetzen. In Kurs *(01867) Sicherheit im Internet 2* werden solche Programme genauer vorgestellt.

automatisierte Angriffe

Trotz dieser Sicherheitsmaßnahmen wird früher oder später bestimmt jemand einen Angriff auf Ihr System starten. Falls derjenige einen neu bekannt gewordenen Fehler ausnutzt, bevor Sie Ihr System aktualisiert hatten, kann der Angreifer bereits Veränderungen vorgenommen haben. Die Veränderungen an einem großen System zu finden ist nicht einfach. Sie können sich dabei von *Integritätstest-Programmen* unterstützen lassen. Diese arbeiten nach folgendem Schema:

Integrität prüfen

1. Zuerst wird ein „Schnappschuss“ des Systemzustands erstellt. Das heißt, dass zu jeder Datei eine kryptografische Prüfsumme (siehe auch Abschnitt 2.5) berechnet wird. Die Prüfsummen für alle Dateien werden separat gespeichert, z. B. auf einer Diskette, einer CD-ROM oder einem USB-Stick.

2. Später kann man dann die Berechnung wiederholen und die Prüfsummen vergleichen. Bei unterschiedlichen Prüfsummen wurde die betroffene Datei nach der Erstellung des Schnappschusses verändert.

Intrusion Detection System (IDS)

Solche Integritätstestprogramme sind oft ein Bestandteil eines **Intrusion Detection System (IDS)**. Dieses Thema wird in Kurs *(01867) Sicherheit im Internet 2* genauer behandelt.

Übungsaufgabe 4.1 *Warum sollten die Prüfsummen auf einer Diskette, einer CD-ROM oder einem USB-Stick gespeichert werden und nicht auf der Festplatte?*

Log-Dateien kontrollieren

Nicht alle Angriffe auf ein System sind erfolgreich. Der Angreifer probiert vielleicht erst einmal nur herum, ohne Änderungen vorzunehmen. Auch solche Angriffe sollten erkannt werden. Dazu können Sie die Protokoll- und Log-Dateien benutzen. Dort speichern Serverprozesse alle erfolgten Zugriffe, egal ob sie fehlerfrei ausgeführt wurden oder zu einem Fehler führten (z. B. falsches Passwort). Somit können Sie erkennen, ob beispielsweise von einem bestimmten Rechner `x` immer wieder erfolglos versucht wurde, eine SSH-Verbindung aufzubauen. Als Administrator gewinnen Sie so zumindest Zeit, um geeignete Abwehrmaßnahmen einzuleiten. Außerdem bekommen Sie ein Gefühl dafür, wie das System normalerweise benutzt wird. Sie können dann in Zukunft verändertes Benutzerverhalten erkennen und Ihr Angebot entsprechend anpassen.

Backup erstellen

Falls ein Angreifer einmal Erfolg haben sollte, dann möchten Sie die unbefugten Änderungen wieder rückgängig machen. Wie schon in Abschnitt 1.4.1 (Viren) beschrieben, benötigen sie hierzu Kopien der Programme und der Daten auf dem System. Zu den Grundpflichten eines Administrators gehört daher auch die regelmäßige Erstellung von Sicherungskopien (engl. **backup**). Außerdem sollte der Administrator auch testen, ob die Sicherungskopien im Ernstfall auch tatsächlich wieder zurück kopiert werden können. Nichts ist ärgerlicher als ein Sicherungsband, das man vorliegen hat, das man dringend braucht, das aber nicht lesbar ist.

Neben der Konfiguration des Betriebssystems und der Serverprogramme müssen Sie sich auch Gedanken über die Organisation der Daten auf dem Webserver machen. Konkret ist zu planen, in welche Verzeichnisse welche Dateien gelegt werden. Weiterhin gehört zur Organisation der Daten auch die Frage, wie die Daten später aktualisiert werden.

In modernen Dateisystemen werden Daten hierarchisch organisiert. Konkret bedeutet dies, dass Verzeichnisse angelegt werden, die dann Dateien oder andere Verzeichnisse enthalten können. Es entsteht eine Baumstruktur wie in Abbildung 4.1. In der Konfiguration des Webservers trägt man das Verzeichnis ein, das für den Webserver die Wurzel sein soll. In Abbildung 4.1 ist die „Webwurzel“ grau hervorgehoben. Der Webserver bildet nun die URL `http://servername/index.html` auf den Dateinamen `/web/index.html` ab. Somit sind *alle* Dateien und Verzeichnisse unterhalb der Webwurzel grundsätzlich über das Web (also durch einen HTTP-Request) abrufbar. Das ist unabhängig davon, ob tatsächlich ein Link in einer beliebigen HTML-Seite auf die Datei oder das Verzeichnis zeigt. Ein Angreifer kann beispielsweise verschiedene Dateinamen raten und ausprobieren.

Vertrauliche Daten nicht im Web-Bereich ablegen.

Vertrauliche Daten sollten besser nicht unterhalb der Webwurzel gespeichert werden. Auf einem E-Commerce-Server sollten also keine Kundendaten oder Bestelldaten zwischen den HTML-Seiten mit den Angeboten gespeichert sein.

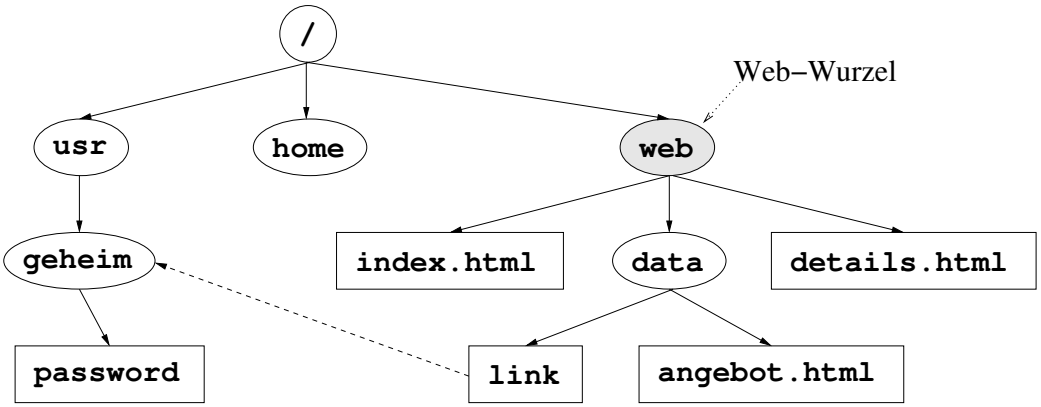

Abbildung 4.1: Beispiel für einen Dateibaum

Falls Sie es doch tun müssen (einige Webservice-Provider sehen das vielleicht vor), dann müssen Sie sich explizit um die Sicherung der Daten kümmern. Der *Apache*-Webserver bietet beispielsweise einen Mechanismus, der den Zugriff auf ein komplettes Verzeichnis unterhalb der Webwurzel verhindern kann (siehe auch Abschnitt 4.2.3).

In modernen Dateisystemen kann man auch Verweise (engl. **links**) anlegen. Diese Verweise können verschiedene Stellen des Baumes miteinander verbinden. Abbildung 4.1 zeigt ein Beispiel für einen Verweis. Der Verweis zeigt aus dem Web-Bereich heraus und ermöglicht somit den potentiellen Zugriff auf alle Daten des Systems. Die vertraulichen Daten wären so, trotz Speicherung außerhalb des Web-Bereichs, im Internet abrufbar. Die Webserver-Software sollte also so konfiguriert werden, dass der Webserver keine symbolischen Links verfolgt. Damit sind Zugriffe auf andere Bereiche des Dateibaums nicht mehr möglich.

Keine symbolischen Links verfolgen.

Der große Vorteil des Internets liegt auch darin, dass man die Daten schnell und einfach ändern und aktualisieren kann. Vergleichen Sie hierzu die Zeit, die ein Versandhaus benötigt, um eine Preisänderung mit einem papierbasierten Katalog zu kommunizieren, mit der Zeit, die es braucht einen Katalog im Internet zu ändern. Wenn man den Webserver nicht bei sich zu Hause stehen hat, ist man darauf angewiesen, Änderungen und Aktualisierungen über ein Netz durchzuführen. Dazu gibt es im Prinzip zwei Möglichkeiten:

1. Man schließt ein Modem an den Webserver an und kann sich dann später direkt dort einwählen.

2. Man benutzt die ohnehin vorhandene Internetanbindung zur Administration aus der Ferne.

Alternative 1 hat den Nachteil, dass zusätzlicher Aufwand anfällt. Man braucht ein Modem und einen Telefonanschluss für das Modem. Beides ist mit zusätzlichen Kosten verbunden. Außerdem sind Modemverbindungen nicht besonders schnell, so dass größere Aktualisierungen viel Zeit benötigen. Die Modemverbindung hat den Vorteil, dass die übertragenen Daten nur schwer abgehört oder manipuliert werden können.

Nimmt man die Administration über das Internet vor, dann muss man auf eine sichere Verbindung achten. *telnet* und *ftp* sind zwei Werkzeuge, die man zwar gut einsetzen kann, die aber beide unsicher sind. Sie sollten stattdessen *SSH* benutzen. Es enthält ein Programm *scp* (secure copy), das Daten sicher verschlüsselt von einem Rechner zu einem anderen Rechner übertragen kann.

SSH

Außerdem sollten Sie die Benutzerauthentisierung in SSH *nicht* mit Hilfe von Benutzerkennung und Passwort durchführen. Erstellen Sie stattdessen ein Schlüsselpaar für den Benutzer und benutzen dann die RSA-Authentisierung für diesen Benutzer. In Abschnitt 3.4.1 wurde *SSH* ausführlich vorgestellt.

Zusammenfassung

Zusammenfassung: Auf einem Server im Internet sollte der Administrator die folgenden allgemeinen Sicherheitsregeln einhalten:

Minimales System: Es werden nur die unbedingt erforderlichen Programme installiert.

Richtige Konfiguration: Programme werden so konfiguriert, dass nur die benötigten Funktionen (Stichwort: offene Ports) angeboten werden. Im Betriebssystem werden spezielle Benutzer angelegt, mit deren Kennung die Serverprozesse laufen. Diese Benutzer bekommen nur die erforderlichen Rechte zugewiesen.

Aktuelle Software: Werden, beispielsweise in Mailinglisten, neue Sicherheitslöcher bekannt, so sollten die Programme umgehend aktualisiert werden. Neue Patches werden regelmäßig installiert.

Selbst „hacken“: Mit den „Waffen des Gegners“ kann man das eigene System prüfen. Gefundene Fehler werden umgehend behoben, die Tests in regelmäßigen Abständen durchgeführt.

Integritätstest: Das System wird regelmäßig auf Änderungen untersucht. Gefundene Änderungen, die man sich nicht erklären kann, deuten auf einen erfolgreichen Angriff.

Log-Dateien kontrollieren: Durch regelmäßige Kontrolle der Log-Dateien erkennt man das „normale“ Verhalten der Benutzer. Außerdem können so erfolglose Angriffe erkannt werden. Präventive Gegenmaßnahmen werden dann umgehend getroffen.

Regelmäßiges Backup: In regelmäßigen Abständen wird das System gesichert. Außerdem wird geprüft, ob die Sicherung im Ernstfall auch dazu taugt, das System zu rekonstruieren.

Keine vertraulichen Daten im Web-Bereich: Alle vertraulichen Daten werden an anderen Stellen im Dateisystem gespeichert. Der Webserver wird so konfiguriert, dass er keine symbolischen Links verfolgt, die diese anderen Stellen doch über das Web zugänglich machen könnten.

Sicherer Administrationszugang mit SSH: Damit man den Server aus der Ferne administrieren kann sollte man einen sicheren SSH-Zugang einrichten.

4.2.2 Betriebssystemabhängige Aspekte

Microsoft Windows: Die Firma *Microsoft* hat mit dem *Internet Information Server (IIS)* einen eigenen HTTP-Server im Programm. Bei Windows Server 2003 wurde er bei der Betriebssysteminstallation automatisch mit installiert. Seit Windows Server 2008 muss man als Administrator auswählen,

IIS-Installation

welche **Rolle** ein Server spielen soll. Insgesamt gibt es 17 Rollen wie *Datei- und Speicherdienste*, *Druck- und Dokumentdienste*, *Remotedesktopdienste*, usw. Nur wenn man dem Server die Rolle *Webdienste (IIS)* zuweist, wird der IIS auch installiert. Die Rollenzuweisung und Installation der Software macht man mit einem speziellen grafischen Programm, dem *Server Manager*. Rolle

Ist die Rolle eingerichtet, so kann man mit dem Programm *Server Manager* weitere Features für den IIS nachinstallieren und auch die Konfiguration des IIS selbst vornehmen. Man kann hier einstellen, dass mehrere sog. *web sites* auf diesem Server angeboten (auf Englisch sagt man auch *hosted*) werden. Für die web site muss man nun prüfen, ob dort nur statische Seiten angezeigt werden oder ob auch dynamisch erzeugte Inhalte enthalten sein sollen. Letztere werden oft über spezielle Microsoft-Technologien wie dem *.NET framework* oder *ASP.NET* generiert. Weiterhin kann man spezifizieren, ob Seiten unverschlüsselt oder mit SSL verschlüsselt abgerufen werden können. Falls SSL eingesetzt werden soll, braucht man ein Server-Zertifikat und muss den IIS passend konfigurieren. Insgesamt sind also sehr viele Parameter richtig zu wählen. Ziehen Sie daher immer die Dokumentation zu Rate. IIS-Konfiguration

Es ist außerdem sehr sinnvoll, regelmäßig die sog. Security-Updates des Systems zu installieren. Das kann auch automatisiert werden. Falls man auf diese Automatisierung verzichtet, dann sollte man regelmäßig manuell nach neuen Updates suchen und diese installieren. Von *Microsoft* werden jeden zweiten Dienstag im Monat am sogenannten *patch day* Sicherheitsaktualisierungen veröffentlicht.

Alle hier genannten Hinweise sind nur als Startpunkte zu verstehen. Aufgrund der schnellen Weiterentwicklung der Programme sind sie fast schon zwangsläufig veraltet und unvollständig, sobald sie gedruckt sind. Die Firma *Microsoft* bietet im Internet aktuelle Sicherheitshinweise und speziellere Informationen zum IIS unter den folgenden URLs an: aktuelle Hinweise

`http://www.microsoft.com/security/` `http://www.iis.net/`

UNIX: Bereits bei der Installation des UNIX-Systems sollte man sich Gedanken über die Aufteilung (Partitionierung) der Festplatte machen. Systemverzeichnisse und Benutzerverzeichnisse sollten genauso getrennt werden wie variable und temporäre Verzeichnisse. Auch sollten schreibbare Verzeichnisse (z. B. `/var`, `/tmp` oder `/home`) in anderen Partitionen liegen als nicht-schreibbare Verzeichnisse wie `/` oder `/usr`. Bei einem Webserver bietet es sich auch an, das Wurzelverzeichnis des Serverprozesses in eine eigene Partition zu legen. Statt der festen Partitionierung der Festplatte kann man auch einen sog. **Logical Volume Manager (LVM)** benutzen. Damit kann man mehr „logische Partitionen“ anlegen, als die vier im BIOS eines typischen PCs vorgesehenen Partitionen. Außerdem kann man später zur Laufzeit des Systems Partitionen (logical volumes) einfach vergrößern, verkleinern, sichern, usw. Logical Volume Manager (LVM)

Beim Systemstart eines UNIX-Rechners werden viele verschiedene Dienstprozesse automatisch gestartet. Am Beispiel von SuSE Linux soll das Konzept kurz vorgestellt werden. Für andere UNIX-Systeme gilt im Prinzip dasselbe, allerdings können die Dateien oder Verzeichnisse anders heißen. Ziehen Sie für Details bitte die Dokumentation Ihres Systems zu Rate.

Zuerst wird der init-Prozess gestartet. Er liest seine Konfigurationsdatei `/etc/inittab` und startet eine Reihe von weiteren Prozessen, die im Verzeichnis `/etc/init.d/` stehen. Anschließend wird ein Shell-Skript gestartet, init

nämlich `/etc/init.d/rc`. Es bekommt die Nummer des Runlevels übergeben, in dem das System starten soll. Nun werden alle weiteren Dienste gestartet. Für jeden Runlevel $X \in \{0, 1, \ldots, 6, S\}$ gibt es ein Verzeichnis mit dem Namen `/etc/init.d/rcX.d`. Alle Dienste aus diesem Verzeichnis werden vom Shellskript nach einander gestartet. Technisch sieht das so aus, dass in dem Verzeichnis symbolische Links auf die tatsächlichen Start-Skripte für jeden Dienst stehen. Das Skript `/etc/init.d/rc` ist dafür verantwortlich, allen vorhandenen symbolischen Links zu folgen und das zugehörige Skript zu starten. Außerdem muss `/etc/inid.d/rc` darauf achten, dass die weiteren Skripte in der richtigen Reihenfolge gestartet werden.

Das Administrationsprogramm *YaST* erlaubt die bequeme Konfiguration der Dienste. In einer grafischen Oberfläche kann man die Dienste der einzelnen Runlevel aktivieren oder deaktivieren. Kontrollieren Sie also mit Hilfe von *YaST* und durch Anschauen der genannten Konfigurationsverzeichnisse, welche Prozesse/Dienste gestartet werden.

inetd

Besonders wichtig ist auch der inetd-Prozess. Er wartet auf ankommende IP-Pakete, die an einen Port aus einem bestimmten Intervall adressiert sind. Der inetd-Prozess entscheidet nun, welcher Dienst für diese Portnummer zuständig ist und startet den zugehörigen Serverprozess. Dieser kann dann das Paket bearbeiten. Somit braucht nur ein Serverprozess ständig zu laufen. Erst wenn weitere Serverprozesse benötigt werden (weil gerade ein IP-Paket ankommt), werden sie gestartet. Anhand der Konfigurationsdatei `/etc/inetd.conf` entscheidet der inetd-Prozess, welchen weiteren Serverprozess er starten soll. Um ein minimales System einzurichten, müssen Sie also diese Konfigurationsdatei anschauen und die Dienste, die nicht benötigt werden, auskommentieren oder besser noch ganz löschen. Inzwischen wird dieser Prozess oft durch einen verbesserten Nachfolger ersetzt. Er heißt xinetd und die Konfiguration befindet sich dann im Verzeichnis `/etc/xinetd.d/`.

xinetd

Immer mehr Linux-Distributionen stellen vom o. g. init-Konzept auf ein funktional äquivalentes System um. Dieses System heisst **systemd**. Eine wesentliche Verbesserung besteht darin, dass die Dienste eines Runlevels nun nicht mehr sequentiell gestartet werden, wie es beim klassischen Konzept der Fall war, sondern parallel. Das sorgt dafür, dass der Rechner beim Systemstart schneller bereit ist. Dazu bietet systemd eine Verwaltung der **sockets** an. Sockets werden i. d. R. dazu benutzt, dass verschiedene Prozesse eines Systems mit einander kommunizieren können. Dieser Mechanismus funktioniert auch zwischen Prozessen auf unterschiedlichen Rechnern, also über ein Netz.

systemd

sockets

Bei beiden Mechnismen passiert bei einem eintreffenden HTTP-Request aber letztlich dasselbe. Das System (inetd, xinetd oder systemd) stellt fest, dass der Request am definierten Socket (i. d. R. Portnummer 80 des Rechners) angekommen ist, startet ggf. den Webserverprozess und übergibt diesem das Socket und somit den Request. Der Webserverprozess bearbeitet nun den Request und antwortet dem Client.

4.2.3 Konfiguration des Webserverprozesses

Apache

Am Beispiel des weit verbreiteten Webservers *Apache* wird im Folgenden dargestellt, worauf man bei einer sicheren Installation achten muss. Die Konfiguration des *Apache* erfolgt durch Einträge in Konfigurationsdateien [Ris05]. Diese werden beim Start des Servers gelesen und entsprechend interpretiert. Nach einer

Änderung an einer der Konfigurationsdateien muss man den Server also neu starten, damit die Änderungen wirksam werden. Unter *Linux* gibt es dazu drei Möglichkeiten:

1. Man schickt dem Prozess ein entsprechendes Signal, wie in

   ```
   kill -HUP <Prozessnummer>
   ```

   Die Prozessnummer steht entweder in der Datei `/var/run/httpd.pid` oder man kann sie mit dem Kommando `ps` erfahren.

2. Man startet den Prozess mit dem Kommando

   ```
   /etc/init.d/apache restart
   ```

   neu. Einige Linux-Systeme bieten ein weiteres Shellskript mit dem Namen `service` an. Als Administrator kann man mit dem Kommando

   ```
   service apache restart
   ```

   dann auch das o. g. Shellskript starten.

3. Falls das System den systemd benutzt, startet man den Dienst (engl. **service**) mit dem Kommando

   ```
   systemctl restart apache.service
   ```

   neu. In diesem Beispiel wird angenommen, dass der Dienst den Namen apache.service trägt.

In der Dokumentation „Ihres“ Servers können Sie nachlesen, welche Kommandos Sie dort genau benutzen sollten. Die Konfigurationsdateien des *Apache* waren früher:

`httpd.conf`	„Haupt“-Konfigurationsdatei
`access.conf`	Konfiguration der Zugriffsrechte
`srm.conf`	Ressourcenkonfiguration
`mime.types`	Zuordnung von Namensendungen zu MIME-Typen

Diese Dateien sollten dem Administrator des Webservers gehören und nur von ihm verändert oder gelöscht werden können! Inzwischen geht man dazu über, die Dateien `access.conf` und `srm.conf` leer zu lassen, bzw. gar nicht erst anzulegen, und alle Konfigurationen in der Datei `httpd.conf` vorzunehmen.

Der Serverprozess wird zunächst unter der Root-Benutzerkennung gestartet. In der Datei `httpd.conf` kann man jedoch einstellen, dass der Prozess sich selbst einem anderen Benutzer zuordnet. Dazu dienen die *User*- und *Group*-Zeilen:

```
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
#  . On SCO (ODT 3) use "User nouser" and "Group nogroup".
#  . On HPUX you may not be able to use shared memory as nobody, and the
# suggested workaround is to create a user www and use that user.
#  NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
#  when the value of (unsigned)Group is above 60000;
#  don't use Group nogroup on these systems!
#
User wwwrun
Group nogroup
```

Der Serverprozess gehört nun dem Benutzer *wwwrun* und zur Gruppe *nogroup*. Dieser Benutzer und diese Gruppe sollten keine Zugriffsrechte auf Dateien haben, die außerhalb der Web-Seiten (bzw. außerhalb der vom Webserverprozess nachzuladenden Programme oder Module) haben. Sollte ein Angreifer den Webserverprozess zum Absturz bringen und evtl. eine Shell starten können, dann kann er keinen weiteren Schaden anrichten. Das verhindern die restriktiven Zugriffsrechte des Benutzers.

Übungsaufgabe 4.2 *Warum gehören die o. g. Konfigurationsdateien wie* ***`httpd.conf`*** *nicht dem Benutzer* wwwrun *sondern dem Administrator des Webservers?*

Die Einträge *ServerRoot* und *DocumentRoot* spezifizieren, an welcher Stelle im Dateibaum die Wurzel des Webservers und die Wurzel der Dokumente (HTML-Seiten) liegen.

```
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE!  If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available at <URL:http://www.apache.org/docs/mod/core.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/usr/local/httpd"
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/usr/local/httpd/htdocs"
```

Ein weiterer wichtiger Eintrag ist die Zeile *DirectoryIndex*. In ihr spezifiziert man, welche Datei der Webserver auf eine Anfrage liefert, die nur aus einem Verzeichnisnamen in der URL besteht. Ein Beispiel für so eine URL ist

`http://servername/data/`

In der Datei `httpd.conf` sieht der Eintrag standardmäßig wie folgt aus:

```
# DirectoryIndex: Name of the file or files to use as a pre-written HTML
# directory index.  Separate multiple entries with spaces.
#
DirectoryIndex index.html
```

Der Webserver liefert bei einer „Directory“-Anfrage dann die Datei `index.html` zurück. Existiert diese Datei im angegebenen Verzeichnis nicht, so liefert der Webserver eine Liste aller Dateien aus dem Verzeichnis. Diese Liste ist vergleichbar mit der Ausgabe des Kommandos `ls -l` unter *UNIX*. Wie die folgende Aufgabe zeigt, ist diese Option nicht ganz ungefährlich. Sie sollte daher mit Bedacht eingestellt werden.

Übungsaufgabe 4.3 *Welches Sicherheitsrisiko entsteht, wenn man die DirectoryIndex-Option nicht benutzt bzw. wenn man keine Datei* ***`index.html`*** *in einem Verzeichnis hat?*

Zugriffsrechte auf einzelne Verzeichnisse werden mit Hilfe des *Directory*-Parameters gesetzt. Der Parameter hat folgenden prinzipiellen Aufbau:

```
<Directory DirectoryName>
...
</Directory>
```

Innerhalb des Parameters kann man mit Hilfe der Parameter *allow* und *deny* steuern, von wo auf dieses Verzeichnis (engl. **directory**) zugegriffen werden darf. Dazu kann man DNS-Namen oder IP-Adressen zu *allow* oder *deny* hinzufügen. Da ein Angreifer einfacher einen DNS-Server manipulieren kann und somit einen falschen DNS-Namen vorspiegeln kann, ist es besser IP-Adressen zu benutzen. Ein Beispiel für die Zugriffsrechte auf ein Verzeichnis ist:

```
<Directory /usr/local/httpd/htdocs/internes>
allow from 10.71.144.4 10.71.144.168
deny from all
Options -FollowSymLinks
</Directory>
```

Diese Einstellung bewirkt, dass zunächst einmal *alle* Zugriffsversuche abgewiesen werden und davon dann zwei IP-Adressen ausgenommen sind. Mit diesem Mechanismus kann man zumindest zwischen internen Rechnern, die eine „private“ IP-Adresse haben, und Internetrechnern, die eine „offizielle“ IP-Adresse haben unterscheiden. Die *Options*-Zeile erlaubt es, weitere Eigenschaften zu spezifizieren. Mit einem + oder − kann man die Eigenschaft ein- oder ausschalten. Im Beispiel wird eingestellt, dass keine symbolischen Verweise (engl. **symbolic link**) verfolgt werden. Neben dieser Zugriffsbeschränkung auf Basis der IP-Adressen kann man den Zugriff auch für einzelne Benutzer freigeben oder sperren. Die Benutzer müssen dafür zusammen mit einem Passwort in einer speziellen Datei stehen. Diese Datei ist im Prinzip mit der Passwortdatei des Betriebssystems vergleichbar.

Nachteil

Ein Nachteil dieses Mechanismus ist allerdings, dass *jede* Änderung an den Zugriffsrechten erst dann wirksam wird, wenn der Webserver angehalten und neu gestartet wird. Außerdem muss jeder, der Zugriffsrechte für ein Verzeichnis vergeben möchte, die Konfigurationsdatei ändern dürfen. Daher bietet *Apache* auch an, die Zugriffsrechte auf ein Verzeichnis in einer Datei in diesem Verzeichnis zu speichern. Diese Datei heißt normalerweise `.htaccess` und kann in der Konfigurationsdatei durch den Parameter *AccessFileName* auch einen anderen Namen bekommen. In dieser Datei stehen dann die *allow*- und *deny*-Parameter für das Verzeichnis. Erfolgt ein Zugriff auf das Verzeichnis, so liest *Apache* zunächst die `.htaccess`-Datei und gewährt anhand der gelesenen Rechte dann den Zugriff oder nicht. Dadurch sinkt allerdings der Durchsatz, denn bei jedem HTTP-Request muss der Server nun mehrere Dateizugriffe machen: Zuerst muss er testen, ob die `.htaccess`-Datei existiert, und sie dann lesen. Anschließend, wenn der Zugriff erlaubt ist, muss er die eigentlich angeforderte HTML-Seite lesen.

htaccess

Man muss nun natürlich auch steuern können, was passieren soll, wenn ein *Directory*-Parameter *und* eine `.htaccess`-Datei für ein Verzeichnis existieren. Diese könnten beispielsweise widersprüchliche Vorgaben enthalten. Details hierzu finden Sie bei Roßbach [Roß99] oder bei Wolfgarten [Wol07]. Es ist übersichtlicher und sicherer, wenn Sie im *Directory*-Parameter nur wenige

allgemeine Vorgaben definieren und diese bei Bedarf durch `.htaccess`-Dateien ergänzen.

In *Apache* ist standardmäßig keine Unterstützung von SSL eingebaut. Es gibt jedoch eine Erweiterung in Form eines Moduls (*mod_ssl*), das man in *Apache* einbauen kann. Außerdem gibt es das Projekt *Apache-SSL*, das eine Version von *Apache* erstellt, die SSL-Unterstützung enthält. Näheres hierzu finden Sie im Internet unter der URL `http://www.apache-ssl.org/`. In der Konfigurationsdatei `httpd.conf` gibt es dann weitere Parameter, deren Namen mit *SSL* beginnen (z. B. *SSLCertificateFile*).

Der sinnvolle Einsatz von SSL erfordert ein Zertifikat (siehe auch Abschnitt 2.7.1). Das kann man bei Zertifizierungsstellen bekommen oder auch selbst erstellen. Möchte man sich ein Zertifikat selber erstellen, dann kann man hierfür das Programm *OpenSSL* benutzen. Es ist frei verfügbar und unter
URL folgender URL finden Sie weitere Informationen:

`http://www.openssl.org/`

Ein selbst erstelltes Zertifikat wird von den Webbrowsern i. d. R. nicht akzeptiert, so dass man es nur für erste Tests verwenden sollte. Die meisten Zertifizierungsstellen bieten allerdings auch kostenlose Testzertifikate an, die eine kurze Gültigkeitsdauer haben.

Möchte man nun, dass SSL-verschlüsselte Requests auf Portnummer 443 von Apache entgegen genommen werden, so kann man das mit diesen Einträgen in der Konfigurationsdatei erreichen:

```
LoadModule ssl_module modules/mod_ssl.so

Listen 443
<VirtualHost *:443>
    ServerName www.example.com
    SSLEngine on
    SSLCertificateFile /path/to/www.example.com.cert
    SSLCertificateKeyFile /path/to/www.example.com.key
</VirtualHost>
```

In Abschnitt 3.3.1 wurde der SSL-Handshake vorgestellt. In ihm handeln Client und Server die Parameter der zu benutzenden Verschlüsselungsverfahren aus. In Apache ist das unsichere Protokoll SSL 2 standardmäßig deaktiviert, so dass nur SSL 3 oder TLS akzeptiert werden. Möchte man, dass Apache nur starke Verschlüsselungsalgorithmen benutzt, muss man folgende Zeile ergänzen:

```
SSLCipherSuite HIGH:!aNULL:!MD5
```

Hier kann man auch explizit die Algorithmen eintragen, die für Verschlüsselung, Authentisierung und Integritätssicherung benutzt werden sollen. Beachten Sie dabei aber, dass Sie auch Algorithmen einstellen, die von den aktuellen Browsern (*Firefox*, *Safari*, *Chrome*, *Internet Explorer*, usw.) tatsächlich unterstützt werden.

Von Apache gibt es mehrere Programmversionen. Setzt man die aktuellste Version ein (im Januar 2014 ist Apache 2.4 aktuell) so wird statt des Namens `apache` oft der Name `apache2` in den Skripten oder Dateinamen benutzt. Im Kurstext wurde immer auf die `2` verzichtet.

Da Apache ein sehr umfangreiches Programm ist kann es hier nur in Ausschnitten behandelt werden. Weitere Details zu Apache finden Sie bei Roßbach [Roß99], bei Wolfgarten [Wol07] oder auf der Seite des Apache-Projekts im Internet unter der URL:

`http://httpd.apache.org/`

4.3 Firewalls

Die Schnittstelle zwischen einem privaten (internen) Netz und einem öffentlichen Netz wie dem Internet muss Sicherheitsmaßnahmen realisieren. Die Sicherheitsmaßnahmen sollen verhindern, dass nicht autorisierte Benutzer aus dem Internet auf Ressourcen des privaten Netzes zugreifen. Außerdem sollen sie verhindern, dass vertrauliche Daten aus dem privaten Netz nach außen dringen. Auch soll ein, evtl. mit einem Virus infizierter, interner Rechner nicht als SPAM-Verteiler auftreten und massenhaft E-Mail ins Internet versenden. In einer Firewall wird dazu definiert, welche Außenstehenden auf welche Ressourcen oder Dienste des privaten Netzes zugreifen dürfen. Genauso wird festgelegt, welche internen Benutzer oder Systeme auf welche Dienste des Internets zugreifen dürfen. Damit eine Firewall diese Aufgaben erfüllen kann, muss sie die *einzige* Verbindung zwischen internem Netz und Internet sein (siehe Abbildung 4.2).

Aufgaben

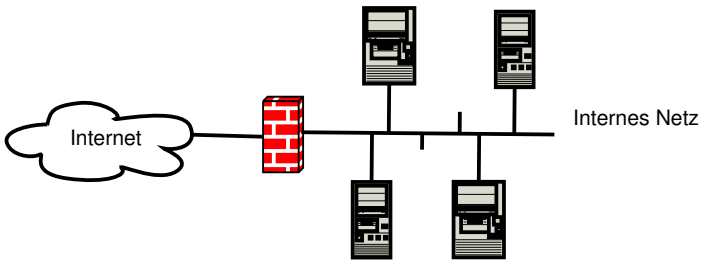

Abbildung 4.2: Prinzip einer Firewall

Ohne eine Firewall wäre jeder Rechner des internen Netzes Angriffen aus dem Internet ausgesetzt. Die Sicherheit des internen Netzes wäre nur so groß wie die Sicherheit des schwächsten Rechners. Jede Kette ist schließlich nur so stark wie ihr schwächstes Glied. Eine Firewall kann dagegen besonders sicher installiert und konfiguriert werden. Ein potentieller Angreifer kann nicht mehr eine schlecht konfigurierte Maschine suchen, sondern muss sich mit einem speziell geschützten System auseinander setzen.

Vorteile

Der oder die Administratoren des internen Netzes haben mit der Firewall einen zentralen Punkt, der gepflegt und überwacht werden muss. Das erfordert deutlich weniger Aufwand, als jeden Rechner eines internen Netzes komplett abzusichern. In der Firewall können auch alle Verbindungen protokolliert werden. Diese Protokolle muss der Administrator regelmäßig kontrollieren. So können Angriffe erkannt und Gegenmaßnahmen eingeleitet werden.

In einer Firewall können auch Adressen umgesetzt werden. Man nennt das auch **Network-Address-Translation (NAT)**. Dies ist beispielsweise dann erforderlich, wenn man in seinem privaten Netz keine „offiziell“ registrierten IPv4-Adressen benutzt. Auch einige Router bieten diese Funktion an. Im RFC 1597 sind drei IPv4-Adressbereiche für private Netze vorgesehen. Es sind die Bereiche

Network-Address-Translation (NAT)

von	bis
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

IPv4-Pakete an diese Adressen werden im Internet nicht weitergeleitet. Erst wenn man sein privates Netz an das Internet anschließt, braucht man eine offiziell registrierte Adresse. Mit Hilfe einer Firewall können dann Rechner mit den oben genannten privaten Adressen auf Rechner im Internet zugreifen. Die Firewall ersetzt bei einer Anfrage die private Absenderadresse durch ihre eigene offizielle Adresse. Der angesprochene Rechner im Internet schickt seine Antwort an die Firewall, die dann die Antwort an den internen Rechner weiterleitet.

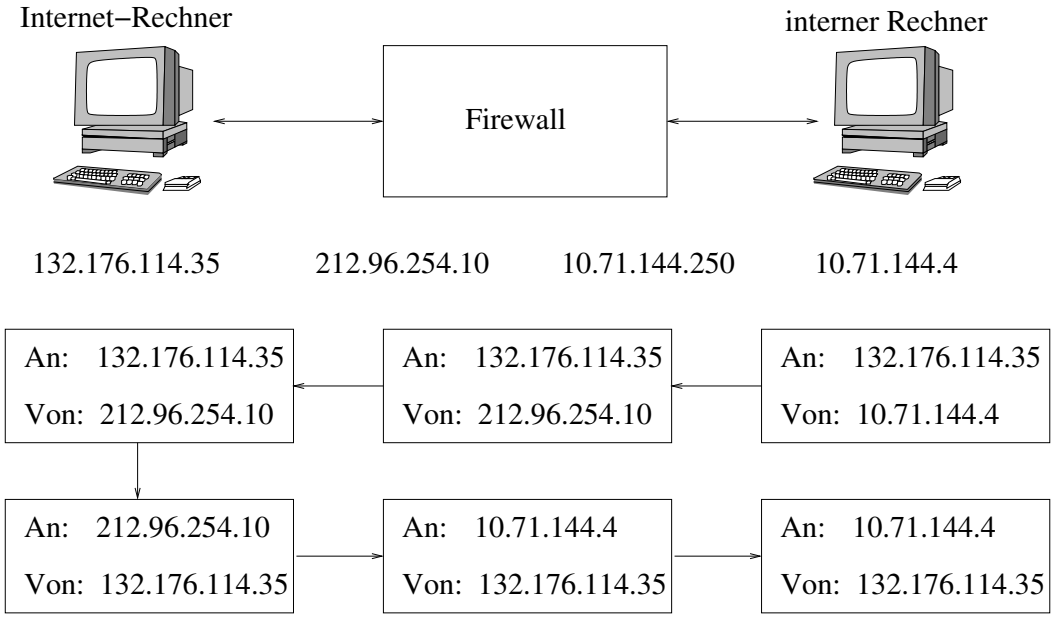

Abbildung 4.3: Beispiel für eine Adressumsetzung in einer Firewall

Abbildung 4.3 zeigt ein Beispiel für die Adressumsetzung. Der interne Rechner hat eine IPv4-Adresse (10.71.144.4) aus dem privaten Bereich. Die Firewall besitzt mehrere Netzanschlüsse und somit auch mehrere IPv4-Adressen. Der Anschluss an das interne Netz hat auch eine Adresse (10.71.144.250) aus dem internen Bereich. Der Anschluss an das Internet hat eine registrierte IPv4-Adresse (212.96.254.10). Wenn der interne Rechner ein IPv4-Paket an einen Rechner im Internet schickt, so wird in der Firewall die Absenderadresse geändert. An die richtige, interne IP-Adresse 10.71.144.4 kann kein Rechner im Internet eine Antwort schicken. Also muss die Antwort an die Firewall gehen und von dort an den ursprünglichen Rechner weitergeleitet werden. Damit das klappt merkt sich die Firewall woher das Paket kam, trägt seine eigene IPv4-Adresse als Absender ein und leitet das Paket dann an den geplanten Empfänger. Der schickt seine Antwort an die eingetragene Absenderadresse, also an die Firewall. Die erkennt das Paket und trägt die oben gemerkte, interne IPv4-Adresse wieder als Empfänger ein.

Einschränkungen

Eine Firewall alleine kann keine Sicherheit garantieren. Beispielsweise kann keine Firewall verhindern, dass von einem Modem im internen Netz vertrauliche Daten nach außen verschickt werden. Diese Übertragung würde an der Firewall komplett vorbeilaufen. Eine Firewall kann auch nicht verhindern, dass vertrauliche Daten per E-Mail nach außen verschickt werden, wenn es grundsätzlich

erlaubt ist, E-Mails nach außen zu verschicken. Ebenso können vertrauliche Daten auf einen kleinen USB-Stick kopiert werden und dann an der Firewall vorbei nach außen transportiert werden.

Auch der Einsatz von Verschlüsselungsverfahren führt dazu, dass eine Firewall den Inhalt der Kommunikation nicht mehr prüfen kann. In einer einmal aufgebauten SSH-Verbindung kann die Firewall nicht mehr prüfen, welche Kommandos ausgeführt werden. Auch bei HTTP-Verbindungen, die durch SSL gesichert sind (vergleiche Abschnitt 3.3.1 aus Kurseinheit 3), kann eine Firewall weder den Request, noch die Response mitlesen.

Trotz dieser Einschränkungen stellt eine Firewall einen wichtigen Baustein für die Sicherheit eines lokalen Netzes dar, der zusammen mit anderen Maßnahmen den Schutz vor Angriffen deutlich verbessert.

4.3.1 Firewall-Architekturen

Packet-Filtering-Router (Paketfilter) bestehen i. d. R. aus einem speziell eingerichteten Router (siehe auch Abschnitt 1.3.2). Für jedes IP-Paket entscheidet der Router, ob es passieren darf oder nicht. Eine interne Tabelle mit Regeln steuert diese Entscheidung. Die Bedingung formuliert Kriterien, die ein IP-Paket erfüllen kann. Die Informationen aus dem Kopf des IP-Pakets (sowie auch der Informationen aus dem Kopf des im IP-Paket enthaltenen TCP- oder UDP-Pakets) werden bei der Formulierung der Bedingung benutzt. Zu den Informationen des IP-Pakets gehören:

Packet-Filtering-Router

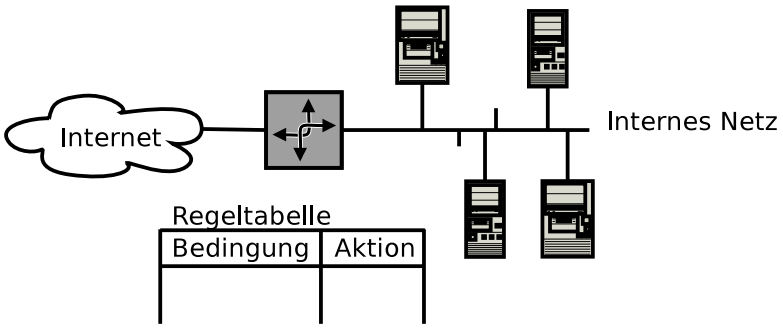

Abbildung 4.4: Paketfilter als Firewall

Regelsteuerung

- IP-Adresse des Absenders,
- Portnummer des Absenders,
- IP-Adresse des Empfängers,
- Portnummer des Empfängers,
- und weitere Informationen wie z. B. die TCP flags.

Findet der Router eine Regel, die zum aktuellen IP-Paket passt, so wendet er sie an. Für die Aktion gibt es im wesentlichen zwei Möglichkeiten:

1. *Durchlassen:* Die Bedingung hatte formuliert, wie ein erlaubtes IP-Paket aussieht. Der Packet-Filtering-Router leitet das Paket also weiter.

2. *Verwerfen:* Diese Aktion besagt, dass der Packet-Filtering-Router das Paket einfach vergisst und nicht weiterleitet. An dieser Stelle hat der Paketfilter nun zwei weitere Möglichkeiten: (a) Der Absender des Pakets bekommt eine ICMP-Fehlermeldung. Sie informiert den Absender, dass das Paket nicht zugestellt werden konnte. (b) Niemand wird darüber informiert, dass das Paket verworfen wurde.

Konzept

Damit kann die Firewall bestimmte Dienste komplett sperren. Möchte man beispielsweise keine Telnet-Verbindungen von außen nach innen zulassen, so werden alle IP-Pakete zu Port 23 verworfen. Möchte man, dass ein Rechner keine E-Mails zugestellt bekommt, so verwirft man alle Pakete an die SMTP-Portnummer 25, sollen keine HTTP-Anfragen erlaubt sein, so verwirft man Pakete an die HTTP-Standardportnummer 80, usw. Dieses Konzept stößt natürlich dann an seine Grenzen, wenn der Administrator eines internen Rechners dort Dienste an eine andere Portnummer bindet. Hat der Firewalladministrator SSH-Verbindungen erlaubt, d. h. Pakete an die SSH-Standardportnummer 22 dürfen passieren, und ein Rechneradministrator bindet auf seinem Rechner einen Webserver an Portnummer 22, so können HTTP-Anfragen von außen an diesen Webserver geschickt werden, auch wenn keine Pakete an Portnummer 80 erlaubt sind. Daher spricht man in diesem Zusammenhang auch häufig davon, „dass bestimmte Ports in der Firewall freigeschaltet sind."

Beispiel

Sollen nun SSH-Verbindungen zu Rechnern des internen Netzes zugelassen werden, dann kann man auch dafür eine Regel angeben. Sollen interne Rechner Verbindungen ins Internet aufbauen können, so braucht man auch hierfür Regeln. Die folgende Tabelle zeigt, wie solche Regeln im Prinzip aussehen.

Bedingung	Aktion
Empfänger Portnummer = 22	erlauben
Empfänger IP-Adresse = 10.71.144.4 UND Empfänger Portnummer = 80	erlauben

Die Tabellen, in denen die Regeln einer Firewall tatsächlich gespeichert sind, sehen in der Praxis allerdings komplexer aus. Sie enthalten häufig eigene Spalten für Absender-IP-Adresse und Absender-Portnummer sowie Empfänger-IP-Adresse oder Empfänger-Portnummer. Die Spaltenbedingungen werden dann logisch mit dem Operator *und* verknüpft. Als Werte dürfen dann auch „Wildcards" benutzt werden, die eine Menge von Werten darstellen. Oder man benutzt eine Netzadresse wie 141.71.1.0/24 und meint damit alle Adressen aus dem angegebenen Subnetz (141.71.1.1 bis 141.71.1.255).

Abs. IP	Abs. Port	Empf. IP	Empf. Port	Aktion
*	*	10.71.144.4	23	erlauben
*	*	*	23	verwerfen
*	*	10.71.144.1	25	erlauben
141.71.1.1	*	10.71.144.5	110	erlauben

Die erste Zeile der Tabelle sagt, dass jeder Rechner eine Telnet-Verbindung zur Adresse 10.71.144.4 aufbauen darf. Zu anderen Rechnern dürfen keine Telnet-Verbindungen aufgebaut werden. Die letzte Zeile sagt, dass von der IP-Adresse 141.71.1.1 eine POP3-Verbindung zum Rechner 10.71.144.5 aufgebaut werden darf.

Damit nun aber der POP3-Server auch Antwortpakete an 141.71.1.1 schicken kann, braucht es eine weitere Regel im Paketfilter, die genau das erlaubt. Für jede TCP-Verbindung, die man erlauben möchte, muss man *zwei* Filterregeln schreiben. Eine Regel erlaubt den Transfer von Paketen vom einen Ende der TCP-Verbindung an das andere Ende. Die zweite Regel erlaubt die Rückrichtung. In ihr sind Absender und Empfänger vertauscht. Man muss die Tabelle oben also um diese Zeile erweitern:

Abs. IP	Abs. Port	Empf. IP	Empf. Port	Aktion
10.71.144.5	110	141.71.1.1	*	erlauben

Damit der Paketfilter schnell arbeiten kann, geht er die Liste der Regeln normalerweise sequentiell durch und wendet die erste passende Regel an. Man muss also genau planen, welche Regeln in welcher Reihenfolge eintragen werden. Die „allgemeineren“ Regeln (wie alle HTTP-Pakete verwerfen) müssen hinter den „spezielleren“ Regeln (wie HTTP zum Rechner *www* erlauben) stehen. Es ist für einen Administrator also keine triviale Aufgabe, den Überblick über die Regeln als Ganzes zu behalten. Weiterhin erstellt ein Paketfilter i. d. R. *keine* Protokolle. Ein Administrator kann also nur schwer erkennen, ob gerade ein Angreifer versucht, den Paketfilter zu überlisten.

Nachteile

Normalerweise sinkt der Durchsatz eines Routers, wenn er viele Filterregeln beachten muss. Damit kann er zum Engpass bei der Anbindung an das Internet werden und man muss Zeit und Geld in leistungsstärkere Router investieren. Mit einem Paketfilter kann man auch keine benutzerbezogenen Regeln erstellen, da im IP-Paket keine Information über den absendenden oder empfangenden Benutzer enthalten ist. Im IP-Paket steht nur die IP-Adresse eines Rechners, der evtl. von vielen Benutzern genutzt wird.

Ein weiterer Nachteil von Paketfiltern ist, dass sie ihre Entscheidung auf der Basis einzelner IP-Pakete treffen müssen. Das bedeutet, dass man beispielsweise *telnet* oder *ftp* für einzelne Rechner aus dem internen Netz frei schalten kann, man aber keinen weiteren Einfluss auf den Inhalt der Telnet- oder ftp-Sitzung hat. Man kann also nicht einstellen, dass ftp nur aus bestimmten Verzeichnissen erlaubt ist, oder dass in der Telnet-Sitzung nur bestimmte Kommandos erlaubt sind.

Auch kann ein Paketfilter einem eingehenden Bestätigungspaket (SYN-ACK-Paket oder ACK-Paket) nicht ansehen, ob es tatsächlich die Antwort auf einen Verbindungsaufbauwunsch (SYN-Paket) ist oder ob es von einem Angreifer in böser Absicht geschickt wurde.

Diese Einschränkungen lassen sich nur umgehen, wenn die Firewall auf einer höheren Ebene im Protokoll-Stack ansetzt. Dies können die in den folgenden Abschnitten vorgestellten *Stateful-Inspection-Filter* oder die *Application-Level-Gateways*.

Übungsaufgabe 4.4 *Welche weitere Funktion einer Firewall kann ein Packet-Filtering-Router nicht erfüllen?*

Paketfilter haben den Vorteil, dass sie für die Anwender transparent sind. Wenn ein Dienst frei geschaltet ist, dann sieht es für den Benutzer so aus, als existierte gar keine Firewall. Es muss auch keine zusätzliche Software auf den Rechnern im internen Netz installiert werden. Moderne Router besitzen heute eingebaute Paketfilter, so dass man den Router, den man ohnehin benötigt, auch gleich als Firewall mitbenutzen kann.

Vorteile

Diese Art von Firewall werden Sie vermutlich von zu Hause kennen. Haben Sie dort einen Internetzugang, so haben Sie von Ihrem Internet-Service-Provider bestimmt ein entsprechendes Gerät bekommen. Je nach Zugangstechnik enthält es ein DSL- oder ein Kabel-Modem für den Datentransport über die Anschlussleitung. Dazu kommt ein Router, der als DHCP-Server für das interne Netz arbeitet, Routingfunktionen bereitstellt und Paketfilterregeln enthält. Normalerweise ist es voreingestellt, dass Pakete von außen nach innen nicht erlaubt sind, es sei denn sie sind Antworten auf Pakete, die vor kurzem von innen nach außen geflossen sind. Das führt zur nächsten Kategorie von Firewallgeräten.

Stateful-Inspection-Filter

Stateful-Inspection-Filter sind erweiterte Paketfilter, die ein lokales Gedächtnis haben. Darin werden aktuelle Status- und Kontextinformationen einer Kommunikationsbeziehung festgehalten. Der Router kann nun also die Regel implementieren, ACK-Pakete nur dann durchzulassen, wenn zuvor eine Verbindung aufgebaut wurde, also wenn SYN-Pakete verschickt wurden.

Für den Administrator sind diese Geräte auch deutlich einfacher zu konfigurieren. Statt für jede Flussrichtung einer TCP-Verbindung eine eigene Regel schreiben zu müssen, braucht der Administrator hier nur für jede Verbindung eine Regel zu schreiben, die genau diesen Verbindungsaufbau erlaubt. Dann reicht eine einzige zusätzliche Regel, in der dann geschrieben steht, dass alle Pakete zu einer bereits existierenden Verbindung auch durchgelassen werden sollen. Die erforderliche Zahl an Regeln für n Verbindungstypen sinkt also von $2n$ auf $n + 1$. Das in Abschnitt 4.3.2 vorgestellte *iptables* ist eine Implementierung eines Stateful-Inspection-Filters. Sie wird im Linux-Umfeld sehr häufig eingesetzt.

Application-Level-Gateway
Proxy

Application-Level-Gateway: Ein Application-Level-Gateway arbeitet auf der Ebene der Anwendungsprotokolle. Ein spezielles Programm, genannt **Proxy**, läuft auf dem Gateway und fungiert als Stellvertreter zwischen dem Client und dem Server. Ein Benutzer, der beispielsweise von innen einen HTTP-Request zu einem Rechner ins Internet senden will, wendet sich nicht direkt an den Rechner im Internet, sondern an das Gateway. Das Gateway kann nun eine Identifikation vornehmen und nur bei erfolgreicher Authentisierung den HTTP-Request weiterleiten. Wichtig ist hierbei, dass der Proxy die Verbindung nach außen aufbaut und dass der Benutzer von innen immer nur mit dem Proxy kommuniziert. Somit kann der Proxy bestimmten internen Benutzern das surfen im Internet verbieten.

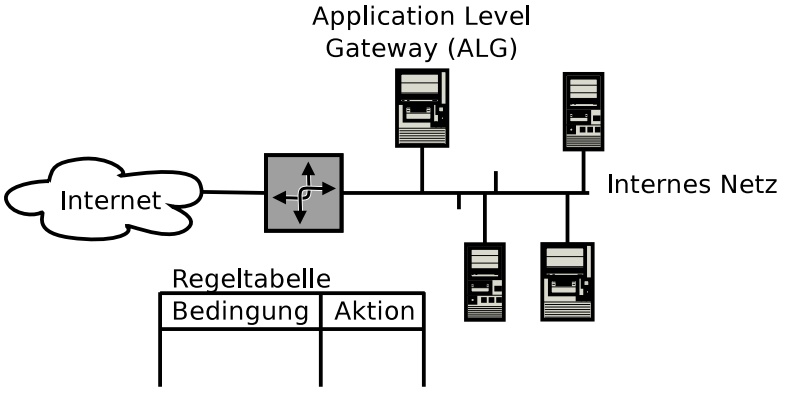

Abbildung 4.5: Single-Homed Application-Level-Gateway

Durch die passenden Regeln im Paketfilter steuert den Administrator nun, dass Pakete mit HTTP-Requests von innen nur vom Application-Level-Gateway erlaubt sind. Möchte ein Benutzer des internen Netzes im Internet surfen, also HTTP-Verbindungen in das Internet aufbauen, so muss das über den Proxy geschehen. Dazu muss der Benutzer seinen Browser so konfigurieren, dass *alle* HTTP-Requests vom Browser an den Proxy geschickt werden. Normalerweise würde der Browser den Request direkt an den in der URL genannten Rechner schicken.

Der Proxy kann nun eine Benutzeridentifikation verlangen und erst danach die Kommunikation erlauben. Außerdem könnten im Proxy eine Reihe von Servern angegeben sein, die während der Arbeitszeit nicht benutzt werden dürfen. Requests an diese Server beantwortet der Proxy direkt mit einer speziellen „Fehlerseite“.

HTTP-Proxies können neben der Absicherung auch der Effizienzsteigerung dienen. Hat der erste interne Benutzer eine bestimmte Seite geladen, so können nachfolgende Anforderungen für dieselbe Seite aus dem Cache des Proxy bedient werden. Dies senkt die Netzlast und führt zu kürzeren Antwortzeiten. Ein HTTP-Proxy anonymisiert auch die Benutzer des internen Netzes. Jeder HTTP-Request von innen wird vom Proxy nach außen weitergegeben. Für einen Webserver im Internet kommt der HTTP-Request also vom Proxy und nicht mehr von einem einzelnen Benutzer.

Cache

Für jeden Dienst, der über das Gateway laufen soll, wird ein eigener Proxy eingerichtet. Neben HTTP können auch Protokolle wie ftp oder SMTP nur über spezielle Proxies erlaubt werden. Ein Proxy kann nun die Kommunikation nicht nur filtern, sondern auch protokollieren.

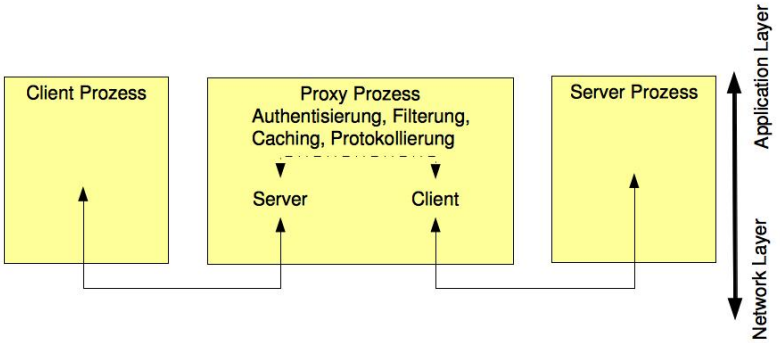

Abbildung 4.6: Funktionsweise eines Proxy

In Abbildung 4.6 ist die Funktionsweise eines Proxy dargestellt. Der Proxy trennt die ursprüngliche Verbindung zwischen dem Client und dem Server auf und macht daraus zwei Verbindungen. Eine zwischen dem Client und dem Proxy und die zweite zwischen dem Proxy und dem Server. Die Informationen der ursprünglichen Verbindung werden auf dem Proxy verarbeitet, beispielsweise gefiltert.

Funktionsweise

Eine weitere wichtige Funktion von Application-Level-Gateways besteht in der Filterung von HTTP-Nachrichten, die zwischen Clients und Webanwendungen ausgetauscht werden. Man spricht in diesem Fall auch von **Web Application Firewalls (WAF)**. Sie sollen die Sicherheit von gut und sicher implementierten Webanwendungen kontrollieren sowie die Sicherheit bei

Web Application Firewalls (WAF)

schlecht und unsicher implementierten Webanwendungen erhöhen. Charakteristisch bei Webanwendungen ist ja, dass HTTP ein zustandsloses Protokoll ist und Webanwendungen daher durch Cookies einzelne Benutzer und/oder Benutzersitzungen wiedererkennen. Speichert ein Webshop nun zwei Cookies bei jedem Benutzer, eines mit Namen `USERID` und eines mit Namen `ALLOWEDDISCOUNT`, so kann die Anwendung einzelne Benutzer erkennen und für jeden Benutzer einen Prozentwert speichern, der den Rabatt bezeichnen soll, den dieser Benutzer bei Einkäufen bekommen soll. Ein Angreifer kann sich das nun zu Nutze machen, indem er einen Browser benutzt, mit dem er die Cookiewerte selbst verändern kann. Dann kann man aus den vom Webshop vorgegebenen 5% Rabatt schnell mal 95% Rabatt machen. Man setzt einfach den Cookiewert von 5 auf 95.

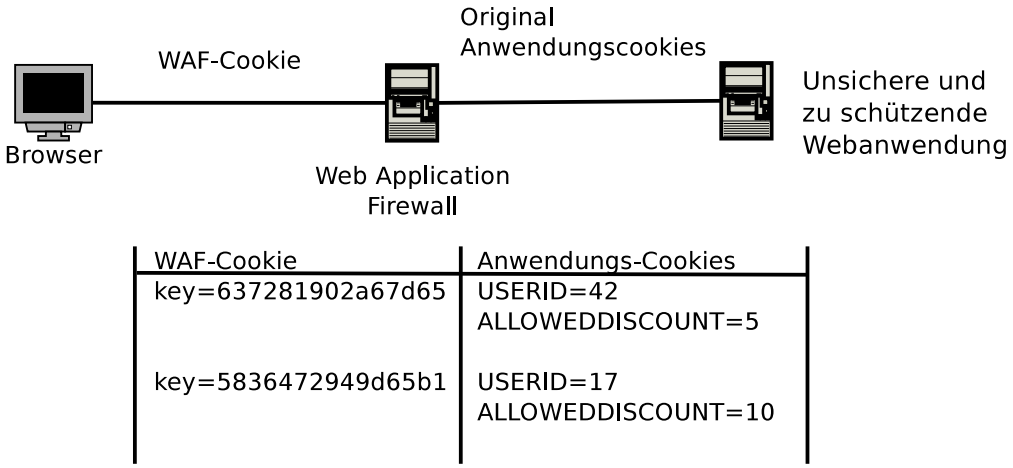

Abbildung 4.7: Web-Application-Firewall

Auch solche Webanwendungen kann man schützen, indem eine Web-Application-Firewall als Application-Level-Gateway *vor* die eigentliche Webanwendung platziert wird. Abbildung 4.7 zeigt das Prinzip. Will die Webanwendung nun das Cookie `ALLOWEDDISCOUNT` setzen, so macht die WAF folgendes:

1. Die WAF erzeugt eine große Zufallszahl für diesen Benutzer. Sie ist nicht zu erraten.

2. Die WAF speichert alle Cookies, die die Webanwendung im Browser setzen will unter der oben erzeugten Zufallszahl als Schlüssel.

3. Die WAF ersetzt die SetCookie-Zeile und sendet dem Browser nur noch den Schlüssel als Cookie.

Hat der Benutzer nun einige Artikel im Webshop ausgewählt und möchte bezahlen, so sendet er den nächsten HTTP-Request. In diesem steht nur der Schlüssel der WAF als Cookie. Die WAF macht nun das:

1. Sie sucht unter dem Schlüssel nach den Original-Cookies der Webanwendung.

2. Das Schlüssel-Cookie wird durch die Original-Cookies ersetzt.

3. Der so entstandene HTTP-Request wird dann an die Webanwendung geschickt.

Ein böswilliger Benutzer kann nun nur noch das Schlüssel-Cookie manipulieren. Die Existenz des Cookies `ALLOWEDDISCOUT` bleibt ihm verborgen und es ist auch nicht mehr manipulierbar. Das ist nur ein Beispiel, was eine WAF erreichen kann. Allgemein soll eine Web-Application-Firewall vor allen Angriffen auf Webanwendungen schützen, also auch gegen Buffer Overflow Angriffe, Injection Angriffe aller Art, unerwünschte Informationsflüsse, usw. Dazu muss die WAF natürlich die Anwendung kennen und beispielsweise wissen, welche Formularfelder wie heißen und welchen Werte-Typ die Anwendung dort erwartet. Nur dann kann die WAF unerlaubte Benutzereingaben in diesen Feldern erkennen und verhindern. Natürlich kann man die hier dargestellten Methoden auch direkt in die Webanwendung selbst einbauen.

Ein Nachteil der in Abbildung 4.5 dargestellten Architektur ist die Tatsache, dass ein Angreifer das ALG angreifen kann (es ist ja von außen erreichbar). Nach einem erfolgreichen Angriff auf das ALG kann der Angreifer nun vom ALG aus andere interne Rechner angreifen, *ohne* dass die Firewall davon etwas mitbekommt. Das lässt sich einfach verhindern, indem man zwischen die von außen erreichbaren Rechner und die internen Rechner einen weiteren Paketfilter schaltet. Das führt zur nächste Kategorie von Firewalls.

Screened Subnet bezeichnet ein eigenes Teilnetz, das zwischen dem internen Netz und dem Internet liegt. Dieses Teilnetz wird an beiden Enden durch Paketfilter geschützt. In Abbildung 4.8 ist das Screened Subnet grau hinterlegt. Das so geschützte Teilnetz bezeichnet man auch als **Demilitarized Zone (DMZ)**. Der Router zum Internet wird nun so konfiguriert, dass er nur Pakete nach innen durchlässt, die an einen der Rechner in der DMZ gerichtet sind. Der Router zum internen Netz lässt nur Pakete von Rechnern aus der DMZ nach innen durch. Der Weg nach außen ist analog aufgebaut. Der innere Router leitet nur Pakete an einen Rechner in der DMZ weiter, der äußere Router nur Pakete von einem Rechner der DMZ. Ein Angreifer muss nun also durch mehrere Systeme, bevor er Zugriff auf einen Rechner des internen Netzes hat.

Demilitarized Zone (DMZ)

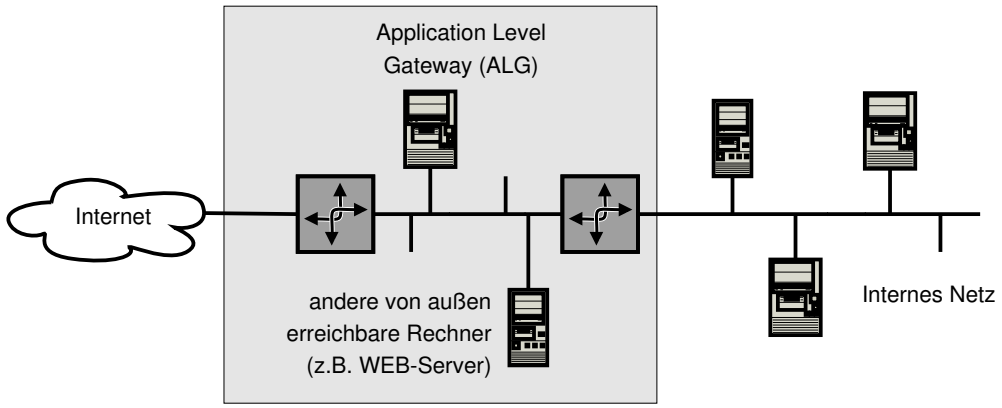

Abbildung 4.8: Screened Subnet bzw. Demilitarized Zone (DMZ)

Neben der Filterung auf Paketebene wird zusätzliche Sicherheit durch das Application-Level-Gateway erzielt. Wie bereits besprochen, filtert es die Kommunikation auf der Anwendungsebene. So können die Informationen aus allen Ebenen ausgenutzt werden.

Vorteile

Diese Architektur ist auch gut skalierbar. Sollte das Application-Level-Gateway zuviel Zeit brauchen, um die Proxy-Funktion für die verschiedenen

Dienste auszuüben, so kann man diese Funktionen auf verschiedene Rechner verteilen. Ein Application-Level-Gateway ist dann für das Surfen im Web (also HTTP) zuständig, ein weiteres für Dateiübertragungen (also ftp) und noch eines für *telnet*. Man kann diese Architektur auch recht einfach für neue Dienste erweitern, selbst wenn für diese Dienste noch kein Proxy zur Verfügung stehen sollte. In diesem Fall kann man nämlich die Router so konfigurieren, dass die IP-Pakete zu diesem Dienst direkt weiter geleitet werden und nicht erst über ein Gateway laufen müssen. Dadurch wird die Firewall insgesamt natürlich etwas unsicherer.

In der DMZ kann man neben dem Application-Level-Gateway auch weitere Systeme aufstellen, beispielsweise einen öffentlichen Webserver. Dort liegen dann die Internetseiten des Betreibers des internen Netzes. Diese Seiten sollen natürlich aus dem Internet abgerufen werden können. Hierbei besteht allerdings die Gefahr, dass ein Hacker den Webserver angreift und dann andere Rechner in der DMZ (z. B. ein Application-Level-Gateway) vom Webserver aus angreifen kann.

Da in Abbildung 4.8 insgesamt drei Geräte (innerer Paketfilter, äußerer Paketfilter und Application-Level-Gateway) vorkommen, die jedes bestimmte Firewall-Aufgaben übernehmen, werden solche Architekturen gerne auch „dreistufige Firewall“ genannt. In Werbe-Prospekten wird dann auch mal behauptet, man hätte nicht nur eine Firewall, sondern sichere das Netz sogar durch *drei Firewalls* hintereinander. Besser wäre es allerdings, wenn man auch in diesem Fall von *einer Firewall* spricht, die einfach aus mehreren Geräten bzw. Stufen besteht.

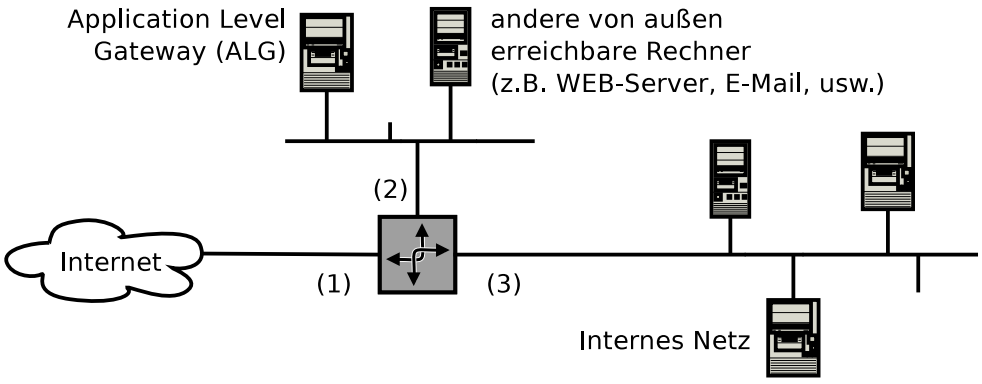

Abbildung 4.9: Screened Subnet bzw. Demilitarized Zone (DMZ) mit einem einzigen Paketfilter

Abbildung 4.9 zeigt eine in der Praxis oft eingesetzte Variation. Statt zwei Paketfiltern mit jeweils zwei Netzanschlüssen benutzt man einen einzigen Paketfilter mit drei Netzanschlüssen. Anschluss (1) verbindet mit dem Internet, Anschluss (2) führt in die DMZ und Anschluss (3) bindet das interne Netz an. Der Paketfilter enthält nun Regeln, die Verbindungen von außen nur kontrolliert zu Rechnern in der DMZ zulassen, aus der DMZ nur kontrolliert nach innen, von innen nur kontrolliert in die DMZ und von innen ebenfalls kontrolliert nach außen. Nun muss sich der Administrator nur noch um einen Regelsatz auf einem Gerät kümmern. Es gibt weniger Chancen, unabsichtlich Fehler bei den Regeln zu machen. Der Nachteil dieses Ansatzes ist, dass ein erfolgreicher Angriff auf den Paketfilter dem Angreifer danach beliebigen Zugriff auf interne Rechner und in die DMZ ermöglicht.

4.3.2 Firewall-Implementierungen

Es gibt verschiedene Implementierungsmöglichkeiten für die o. g. Firewall-Architekturen. Einige dieser Varianten werden in diesem Abschnitt kurz vorgestellt.

iptables

iptables ist ein *Linux*-Kommandozeilenprogramm, mit dem man Paketfilterregeln erstellen kann. Es benutzt die Netfilter-Architektur, die im *Linux*-Kernel dann die eigentliche Paketfilterung vornimmt. Zusätzlich kann iptables auch Network-Address-Translation (NAT) vornehmen [Spe11]. In iptables gibt es drei Gruppen mit Regeln:

Filter: Diese Regeln werden für die Paketfilterung benötigt. Sie beschreiben, welche Pakete die Firewall passieren dürfen und welche nicht.

NAT: Diese Regeln beschreiben wie Network-Address-Translation genau vorgenommen werden soll.

Mangle: Diese Regeln erlauben es, beliebige Veränderungen an Paketen vorzunehmen, z. B. den TTL-Zähler zu ändern.

Ketten

In jeder Regelgruppe gibt es nun Regel-**Ketten**. In einer Regelkette stehen die einzelnen Regeln, die das System für jedes Paket prüft. Die erste auf das Paket passende Regel entscheidet, was mit dem Paket passieren soll (weiterleiten, verwerfen). In iptables gibt es mehrere Regelketten:

INPUT: Die in der INPUT-Kette stehenden Regeln werden nur auf Pakete angewendet, die von außen kommend an einen Prozess auf dem Paketfilter selbst gerichtet sind. Hier könnte beispielsweise eine Regel stehen, die SSH-Zugang zum Paketfilter erlaubt, damit der Administrator den Paketfilter von seinem Schreibtisch aus administrieren kann.

OUTPUT: Diese Regeln werden nur auf Pakete angewendet, die von einem Prozess auf dem Paketfilter kommend an einen anderen Rechner gerichtet sind. Hier könnte beispielsweise eine Regel stehen, die es dem Paketfilter erlaubt, Betriebssystem-Updates vom Server des Betriebssystemherstellers anzufordern.

FORWARD: In der FORWARD-Kette stehen nun die eigentlichen Paketfilterregeln. Sie werden auf alle Pakete angewendet, bei denen der Paketfilter selbst weder die Quelle noch das Ziel ist. Von diesen Regeln werden also alle Pakete erfasst, die der Paketfilter nur durchreicht (engl. **to route**). Hier könnte beispielsweise stehen, dass Rechner aus dem Internet Pakete an den Webserver schicken dürfen. Vergleichen Sie hierzu Abbildung 4.8, wo der Webserver hinter dem ersten Paketfilter aufgestellt ist.

User defined: Solche Ketten können vom Benutzer des Paketfilters definiert werden und aus anderen Ketten „angesprungen“ werden.

Immer wenn also ein Paket beim Paketfilter eintrifft, wird zuerst geprüft, von wo nach wo das Paket gerichtet ist. In Abhängigkeit davon wird dann entweder die INPUT-Kette oder die OUTPUT-Kette oder die FORWARD-Kette durchlaufen. Die Regeln der Kette werden sequentiell geprüft und bei der ersten passenden Regel wird die angegebene Aktion ausgeführt. Eine Regel sieht beispielsweise wie folgt aus:

```
iptables -A FORWARD -p tcp -s 10.0.0.1 -d 10.2.0.1 --dport 22 -j ACCEPT
```

Die Option `-A` beschreibt, an welche Kette diese Regel angehängt werden soll. Im Beispiel also an die FORWARD-Kette. Mit der Option `-j` am Ende der Zeile wird die auszuführende Aktion beschrieben und die anderen Optionen (in der Mitte der Zeile) beschreiben wie ein Paket aussehen muss, damit die Regel angewendet werden kann. Im Beispiel muss das Paket ein TCP-Paket sein, von der Quell-IP-Adresse `10.0.0.1` kommen und an die Ziel-IP-Adresse `10.2.0.1` gerichtet sein; konkret an Port-Nr. 22 des Ziels. Damit kann iptables als Stateless-Paket-Filter arbeiten. In jeder Kette kann man natürlich auch eine *Default Policy* angeben. Die dort angegebene Aktion wird immer dann ausgeführt, wenn keine der anderen Regeln zum aktuellen Paket passt.

In iptables gibt es auch eine Zustandsmaschine (engl. **state machine**). Mit ihr kann man iptables auch als *Stateful-Inspection-Filter* betreiben. Man muss dann „nur noch“ die Regeln explizit hinschreiben, die den jeweiligen Verbindungsaufbau betreffen, also z. B. drei Regeln, die (1) SSH-Verbindungen zum SSH-Server, (2) Mailverbindungen zum Mailserver und (3) HTTP-Verbindungen zum Webserver erlauben. In *einer* einzigen weiteren Regel kann man dann festlegen, dass Antwortpakete im Rahmen von aufgebauten Verbindungen auch erlaubt sein sollen.

```
iptables -A FORWARD -d SSH-Server --dport 22 -m state --state NEW -j ACCEPT
iptables -A FORWARD -d Mail-Server --dport 25 -m state --state NEW -j ACCEPT
iptables -A FORWARD -d Web-Server --dport 80 -m state --state NEW -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Im Internet finden Sie weitere Informationen zu Netfilter/iptables unter der URL:

`http://www.netfilter.org/`

Es gibt eine Reihe von grafischen Administrationsprogrammen mit denen Administratoren die iptables-Regeln „zusammenklicken“ können. Man kann dort interaktiv einen Rechner als Webserver markieren und das Administrationsprogramm erzeugt dann die iptables-Regeln, damit man von außen auf diesen Webserver zugreifen kann.

PF **PF:** Das Packet-Filter-Paket (PF-Paket) ist unter BSD-UNIX das Analogon zu iptables. Es verwaltet grundsätzlich auch den Zustand einer Verbindung. Dadurch werden die Regelsätze einfacher und effizienter auszuführen. In PF wird beispielsweise zuerst geprüft, ob ein Paket zu einer bereits aufgebauten Verbindung gehört. Wenn ja, dann wird das Paket durchgelassen, ohne dass die Filterregeln überhaupt geprüft werden. Wenn nein, dann werden die Regeln geprüft.

In PF werden immer alle Regeln geprüft und die Aktion der letzten passenden Regeln wird ausgeführt. Somit kann man also am Anfang der Regelkette sämtlichen Verkehr blockieren und anschließend in spezielleren Regeln bestimmten Verkehr erlauben.

Möchte man beispielsweise keine eingehenden Verbindungen erlauben, aber ausgehende Verbindungen schon, dann reichen zwei Zeilen

```
block in all
pass out all
```

in der Konfigurationsdatei `/etc/pf.conf` hierfür aus. Da ausgehende Verbindungen erlaubt sind und die eingehenden Antwortpakete zunächst auf die Zugehörigkeit zu einer Verbindung geprüft werden, wird die `block in all`-Regel auf das Antwortpaket gar nicht angewendet.

Squid ist eine Application-Level-Gateway-Software (Proxy-Software) für das Anwendungsprotokoll HTTP. Es wird überwiegend eingesetzt, um Bandbreite zu sparen, indem häufig besuchte Webseiten im Proxy zwischengespeichert werden. Darüber hinaus kann Squid aber auch die Zugriffe auf Webseiten kontrollieren, also nur bestimmten Benutzern den Zugriff auf bestimmte Webseiten zu bestimmten Uhrzeiten erlauben.

Um den Proxy zu benutzen, muss das Anwendungsprogramm wissen, dass es den Proxy benutzen soll. Im Firefox-Browser kann man das in der Konfiguration einstellen. Danach sendet Firefox *jeden* HTTP-Request an den Proxy, egal welcher Rechnername in der URL stand. Auf dem Proxy terminiert also die TCP-Verbindung (siehe auch Abbildung 4.6). Der Proxy-Prozess prüft nun, ob die in der URL angeforderte Seite im Cache steht und der HTTP-Request somit direkt beantwortet werden kann. Steht die Seite nicht im Cache, so baut der Proxy nun seinerseits eine HTTP-Verbindung zum Server aus der URL auf und ruft die Seite ab. Anschließend leitet der Proxy die Seite an den anfragenden Browser weiter.

In Squid gibt es auch die Möglichkeit, Benutzer zu authentisieren. Dann kann man wie bei Betriebssystemen **Zugriffskontrolllisten** (engl. **access control lists (ACL)**) definieren und speziellen Benutzern den Zugriff beispielsweise nur an bestimmten Tagen, zu bestimmten Uhrzeiten oder auf bestimmte Webseiten erlauben.

Zugriffskontrolllisten

Socks: Während *Squid* ein spezieller Proxy (im Wesentlichen für HTTP) ist kann man mit Socks verschiedene Protokolle benutzen. Daher nennt man Socks auch einen *generischen Proxy*. Das Problem mit Proxies ist, dass die Anwendungsprogramme bzw. die Anwendungsprotokolle wissen müssen, dass sie über einen Proxy kommunizieren. In der HTTP-Spezifikation werden Proxies explizit erwähnt und das Verhalten definiert. In einem Browser wie *Firefox* muss man als Benutzer einen Proxy auch explizit einstellen.

Socks

Mit Hilfe der Socks-Bibliothek kann man nun Anwendungsprogramme und Serverprogramme schreiben, die nicht direkt über **Sockets** mit einander kommunizieren, sondern die Socks-API benutzen. Somit ist es möglich, einen Proxy zwischen Anwendungsprogramm und Serverprogramm zu setzen.

Sockets

ModSecurity [Ris10] ist eine Open Source Implementierung einer Web Application Firewall. Es kann die HTTP-Requests und -Responses protokollieren, inspizieren und somit Angriffsversuche erkennen. Darüber hinaus können die übertragenen Daten auch verändert werden. Die Webanwendung sieht also bestimmte Angriffe gar nicht mehr.

ModSecurity ist ein Modul für den *Apache* Webserver. Hat man eine Webanwendung auf einem anderen Webserver, so kann man einen Apache-Server mit ModSecurity als sogenannten **Reverse Proxy** vor die Webanwendung schalten. Alle HTTP-Requests gehen also nicht an die Webanwendung, sondern erst an den Reverse Proxy. Er prüft auf Angriffe und leitet nur „ungefährliche“ HTTP-Requests weiter. Auch die HTTP-Responses werden zunächst gefiltert

Reverse Proxy

und erst danach an den Anfrager weitergeleitet. Auf der Startseite des Projekts im Internet

`http://www.modsecurity.org`

finden Sie weitere Informationen.

4.3.3 Firewall-Konfiguration

Bei der Konfiguration einer Firewall kann man nicht für jede mögliche Situation eine eigene Regel aufstellen. Stattdessen wird man für einige häufige Situationen Regeln definieren. Was soll nun in den anderen Situationen geschehen? Hier gibt es zwei Möglichkeiten:

Alles, was nicht explizit verboten ist, ist erlaubt: In dieser Konfiguration wird im Wesentlichen eine Liste der verbotenen Verbindungen definiert. Alles, was nicht auf dieser Liste steht, ist erlaubt und darf die Firewall passieren.

Diese Konfiguration hat den Nachteil, dass sie Freiheiten gewährt, die vielleicht doch von einem Angreifer ausgenützt werden können. Außerdem fällt ein Fehler in der Konfiguration im normalen Betrieb nicht auf, sondern erst wenn er ausgenutzt wurde und ein Angreifer bereits Schaden angerichtet hat.

Alles, was nicht explizit erlaubt ist, ist verboten: In dieser Konfiguration wird im Wesentlichen eine Liste der erlaubten Verbindungen definiert.

Der Nachteil dieser Konfiguration besteht darin, dass sie ständig angepasst werden muss, wenn die Benutzer neue Verbindungswünsche (beispielsweise Telnet) haben. Dafür hält sie neue Angriffsformen wirksam ab. Ein weiterer Vorteil ist, dass Fehler in der Konfiguration schnell erkannt werden. Wenn Benutzer etwas nicht machen können, was sie können sollten, dann wenden sie sich i. d. R. sehr schnell an den Administrator. Der kann den Fehler dann korrigieren.

Diese beiden Alternativen sind der typische *Trade-Off* zwischen Bequemlichkeit und Sicherheit. Es ist einfacher, erst einmal nur die bekannten Sicherheitslöcher zu stopfen und ansonsten den Benutzern das Leben so einfach wie möglich zu machen. Diese Strategie ist allerdings gefährlich. Im Gegensatz dazu steht die Strategie, erst einmal alles zu verbieten. Sie ist sicher, denn sie erlaubt erst einmal keine Verbindungen. Der Administrator muss für jeden Dienst einzeln eintragen, ob er erlaubt sein soll. Dies führt sicherlich dazu, dass der Administrator mehr Arbeit hat als bei der bequemeren Strategie.

Bestimmte Dienste im Internet sind einfach zu unsicher. Man sollte sie in einer Firewall komplett sperren und keine diesbezüglichen Verbindungen zulassen. Zu diesen Diensten gehören:

Sperren

rlogin, rsh und rexec: Diese Dienste wurden ursprünglich für die Berkeley-UNIX-Variante entwickelt. Sie übertragen Passwörter im Klartext und können auch so konfiguriert werden, dass die Authentisierung umgangen wird. Die Ports 513, 514 und 512 sollten also gesperrt werden.

NIS, NFS und RPC: Das Network-Information-System (NIS) und das Network-File-System (NFS) erlauben eine einfachere Konfiguration eines lokalen Netzes (LAN). Sie basieren auf Remote-Procedure-Calls (RPCs) und benutzen standardmäßig den Port 111. Erfolgreiche Angriffe auf diesem Port können dazu führen, dass Passwörter ausspioniert werden und Dateien gelesen oder verändert werden können. Diese Dienste sollten daher nur innerhalb eines lokalen Netzes benutzt werden.

X Windows: Das X11-Protokoll erlaubt es, ein Programm und die Ein-/Ausgabe desselben auf verschiedenen Rechnern laufen zu lassen. Der Serverprozess, der für die Ein-/Ausgaben zuständig ist, kann missbraucht werden, so dass z. B. Tastatureingaben, die für ein bestimmtes Programm gedacht sind, auch an ein anderes Programm übermittelt werden. Passwörter können damit ausspioniert werden. Man kann X11 auf einem entfernten Rechner jedoch sicher benutzen, wenn man es durch *SSH* tunnelt.

Telnet: Da bei telnet Daten im Klartext übertragen werden sollte es nicht mehr benutzt werden. Mit SSH (siehe Abschnitt 3.4.1) gibt es einen sehr guten und auch frei verfügbaren Ersatz.

Die folgenden Dienste sind mit geringeren Risiken behaftet bzw. sie werden von Benutzern stärker nachgefragt. Sie sollten von der Firewall im Prinzip durchgelassen werden. Allerdings sollten sie in der Firewall entsprechend kontrolliert werden:

Filtern, aber durchlassen

HTTP: Dieses Protokoll ist das mit Abstand am häufigsten benutzte Protokoll im Internet. Es komplett zu sperren würde bedeuten, die meist genutzte Anwendung zu verbieten. Stattdessen sollte man die Firewall so konfigurieren, dass HTTP-Verbindungen kontrolliert zugelassen sind. Man kann den Webserver z. B. in die DMZ stellen und den Paketfilter so konfigurieren, dass nur von den internen Rechnern der Administratoren des Web-Auftritts Verbindungen von innen zum Webserver erlaubt sind.

SMTP: Das Protokoll zum Austausch von E-Mails sollte von der Firewall nur zu den speziell konfigurierten E-Mail-Servern durchgelassen werden. Häufig stellt man dieses SMTP-Gateway in die DMZ und führt auf dem Gateway dann eine Virenprüfung und eine SPAM-Filterung durch. Die danach noch verbliebenen E-Mails leitet das SMTP-Gateway aus der DMZ zu einem internen E-Mail-Server weiter. Dieser wird dann von den Benutzern des internen Netzes benutzt.

SSH: Sollen Benutzer von außen und über das Internet auf interne Systeme zugreifen können, so sollte SSH benutzt werden. Allerdings sollte man sich auf *einen* internen SSH-Server beschränken, zu dem die Firewall Verbindungen zulässt. Außerdem ist es sinnvoll, dass der SSH-Server die Benutzer-Authentisierung *nur durch SSH-Schlüssel* und nicht durch Passwörter erlaubt. Damit verhindert man brute force Angriffe auf schwach gewählte Benutzerpasswörter. Alternativ kann man auch ein **Virtual Private Network** einsetzen und ein VPN-Gateway in die DMZ stellen. VPNs werden in Kurs *(01867) Sicherheit im Internet 2* genauer behandelt.

Virtual Private Network

NNTP: Internetnews werden mit Hilfe dieses Protokolls übertragen. Es sollte von der Firewall nur zu den speziell konfigurierten Newsservern durchgelassen werden.

DNS: Die Zuordnung von Rechnernamen zu IP-Adressen ist ein wichtiger Dienst. Die Firewall sollte eigene DNS-Server bereitstellen (z. B. in der DMZ), die dafür sorgen, dass die internen Rechnernamen und IP-Adressen nicht von außen einsehbar sind. Außerdem sollen die internen Rechner natürlich alle öffentlichen DNS-Namen auflösen können.

4.3.4 Firewall-Betrieb

Nachdem man sich eine Sicherheitspolitik überlegt, eine Firewall-Architektur ausgewählt und die Firewall installiert hat, muss die Firewall betrieben werden. *Sicherheit ist keine einmalige Aktion, sondern ein Prozess, der ständig weiter läuft.* Das bedeutet, dass die Sicherheitsmaßnahmen (die in der Firewall realisiert werden sollen) regelmäßig auf ihre Einhaltung überprüft werden müssen. Dazu gehört beispielsweise, dass man immer wieder kontrollieren muss, ob die Firewall tatsächlich die einzige Verbindung zwischen dem Intranet und dem Internet ist. Dieser IT-Sicherheits-Prozess wird in Abschnitt 4.4.2 genauer beschrieben.

Sicherer Admin-Zugang

Der oder die Administratoren müssen über einen sicheren Zugang zur Firewall verfügen, d. h. entweder einen physischen Zugang zu den Geräten selbst oder ein eigenes sicheres Administrationsnetz. Lässt sich keine dieser Alternativen realisieren, sollte man einen SSH-Zugang für die Administratoren einrichten, bei dem die Benutzerauthentisierung durch RSA-Schlüssel erfolgt.

In regelmäßigen Abständen sollten die Administratoren prüfen, ob keine Veränderungen an der Firewall vorgenommen wurden. Veränderungen könnten an der Hardware, der Software oder den Konfigurationsdaten vorkommen. Gegen die unbefugte Änderung der Konfigurationsdaten, z. B. der Filterregeln, kann man sich schützen, indem diese Daten auf einem Medium gespeichert sind, das keinen Schreibzugriff gestattet.

Wichtig ist auch, dass man die Reaktion der Firewall auf einen Systemabsturz untersucht. Die Firewall sollte in diesem Fall nicht automatisch neu starten. Ferner sollte es beim Ausfall der Firewall nicht mehr möglich sein, Netzverbindungen zwischen Internet und Intranet aufbauen zu können.

Logfiles

Im laufenden Betrieb erzeugt die Firewall Protokolldateien, die auch **Logfiles** genannt werden. Solche Logfiles werden teilweise direkt vom Betriebssystem der Firewallhardware erzeugt, von der Firewallsoftware (d. h. den Paketfiltern oder den Proxies) oder von weiterer, zusätzlich angeschaffter Protokollierungshardware bzw. -software. Die Protokolle sollten sicher gespeichert werden, z. B. auf einer eigenen Partition, einer eigenen Festplatte oder einem eigenen Log-Server. Es ist für einen Menschen unmöglich, den dabei anfallenden Datenmengen Herr zu werden. Mit Hilfe von Filterprogrammen sucht man stattdessen die Meldungen aus den Protokollen, die wichtig sind. Wichtig sind die Meldungen, die keine „normalen“ Meldungen sind. Zweck der Protokollierung ist es,

Zweck der Protokollierung

1. Angriffe oder Angriffsversuche zu erkennen und
2. im Falle eines Angriffs Beweise zu sichern und eine Analyse der Vorgänge zu ermöglichen.

Im System werden viele verschiedene Logfiles vorhanden sein. Da sie teilweise dieselben Ereignisse protokollieren, kann man sie auf Konsistenz untersuchen. Steht beispielsweise im Proxy-Logfile, dass der Benutzer *xy* von Rechner *z* eine Telnet-Verbindung aufgebaut hat, und im Logfile von Rechner *z* ist kein Eintrag zu finden, dass Benutzer *xy* dort angemeldet war, hat man einen Widerspruch gefunden. Dieser Widerspruch könnte ein Hinweise darauf sein, dass jemand die Systeme erfolgreich angegriffen und dann versucht hat, durch Manipulation der Logfiles seine Spuren zu verwischen.

Häufig genug gibt es aber auch andere Ursachen für vermeintliche Inkonsistenzen. Beispielsweise kann die Software Fehler enthalten oder dem Administrator fehlte einfach nur die richtige Erklärung für das beobachtete Verhalten. Es ist deshalb besonders wichtig, nach der Inbetriebnahme einer Firewall den Datenverkehr und die Logfiles genau zu analysieren. Die so gewonnenen Erfahrungen sind für einen Administrator sehr wichtig.

Intrusion-Detection-Systeme (IDS)

Intrusion-Detection-Systeme (IDS) haben die Aufgabe, den Administrator bei der Erkennung von Angriffen zu unterstützen. Genauer gesagt sollen sie

- zeitnah, also möglichst genau dann, wenn es passiert, bzw. nur sehr kurze Zeit später, und
- zuverlässig, also möglichst nur dann, wenn es tatsächlich der Fall ist,

erkennen, ob ein Angriff vorliegt. Der Administrator muss ein Intrusion-Detection-System daher so konfigurieren, dass die Ereignisse, die einen Angriff kennzeichnen, im System bekannt sind. Außerdem muss man Schwellwerte definieren, deren Über- oder Unterschreiten einen Angriff signalisiert. Es gibt zwei Verfahren, die von Intrusion-Detection-Systemen benutzt werden:

Anomalie-Erkennung: Nachdem dem IDS bekannt gemacht wurde, wie „normales“ Benutzerverhalten aussieht, vergleicht das IDS die Protokolle mit dem normalen Benutzerverhalten. Abweichungen werden gemeldet.

Ein Beispiel hierfür ist die Uhrzeit, zu der sich ein Benutzer normalerweise an- oder abmeldet. Meldet sich ein Benutzer immer nur montags bis freitags zwischen 7:00 und 9:00 Uhr am System an, so ist eine Anmeldung sonntags um 15:00 Uhr zumindest merkwürdig.

Signaturanalyse: Bei diesem Verfahren vergleicht das IDS die Protokolldaten mit der Signatur typischer Angriffe. Häufig beginnen Angreifer damit, erst einmal alle Ports eines Systems zu prüfen, um offene Schnittstellen zu erkennen.

Werden beispielsweise nacheinander die Ports $1 \ldots 10\,000$ eines Systems von demselben externen Rechner angesprochen, so könnte ein Angriff vorbereitet werden. Anschließend könnte der Angreifer über einen der offenen Ports versuchen, das System zu manipulieren.

In der Praxis ist es natürlich nicht ganz so einfach, denn ein Angreifer würde die Ports nicht in aufsteigender oder absteigender Reihenfolge prüfen. Es ist also schwierig, eine passende Signatur zu diesem Angriffsversuch zu definieren.

Intrusion-Detection-Systeme lassen sich natürlich auch dazu einsetzen, Angriffe von innen zu erkennen. In Kurs *(01867) Sicherheit im Internet 2* wird das Thema IDS genauer behandelt.

Reaktion auf Zwischenfälle

Reaktion auf Zwischenfälle: Hat das Intrusion-Detection-System oder der Administrator erkannt, dass ein Angriff vorliegen könnte, so sind die folgenden Schritte durchzuführen:

1. **Keine Panik!** Schnelle unüberlegte Reaktionen können den Schaden vergrößern und die Ursachenforschung erschweren. Man sollte also Ruhe bewahren und erst nachdenken und dann handeln.

2. Überlegen, ob Sofortmaßnahmen erforderlich sind. Das bedeutet, dass man zuerst klären muss, ob der Angriff im Moment noch stattfindet oder ob man nur erkannt hat, dass z. B. letzte Nacht ein Unbefugter im System war. Mögliche Sofortmaßnahmen sind:

   - Man lässt alle Systeme weiterlaufen wie bisher.
   - Man nimmt die vom Angriff betroffenen Systeme vom Netz.
   - Man trennt das gesamte Intranet vom Internet.

   Welche der Sofortmaßnahmen man wählt, hängt auch davon ab, wie gravierend der Angriff ist. Lässt man die Systeme weiterlaufen, so hat man die Chance, den Angreifer zurück zu verfolgen. Nimmt man Systeme vom Netz, begrenzt man den weiteren Schaden, der angerichtet werden kann.

3. Beweise sichern. Hierzu erstellt man ein Backup der betroffenen Systeme. Am besten benutzt man hierzu ein Medium, das nur ein einziges Mal beschreibbar ist. Damit kann der potentielle Vorwurf entkräftet werden, dass man nachträglich das Backup verändert hat.

4. Problem analysieren. Hierzu sucht man nach den Ursachen, wie es zu dem Angriff kommen konnte. Mögliche Ursachen können sein:

   - Falsche Konfiguration der Firewall
   - Fehler und/oder Sicherheitslücken der Firewall-Komponenten

5. Maßnahmen zur Problemlösung ergreifen. Nachdem man die Ursache für den Angriff erkannt hat, schließt man die gefundene Lücke. Dazu ändert man evtl. die Konfiguration der Firewall oder installiert die neusten Patches.

Falls der Angreifer erfolgreich war, sollte man immer davon ausgehen, dass er das System soweit kompromittiert hat, dass er weitere Möglichkeiten geschaffen hat sich wieder in das System einzuwählen. Also auch wenn man die ursprüngliche Sicherheitslücke gestopft hat, muss man damit rechnen, dass der Angreifer weitere „Hintertüren“ für sich geöffnet hat. Daher ist es nach einem erfolgreichen Angriff i. d. R. erforderlich, das *ganze System* neu aufzusetzen.

4.3.5 Zusammenfassung: Firewall

Eine Firewall hat die Aufgabe, ein internes Netz vom Internet abzugrenzen und auf der Schnittstelle gewisse Sicherheitsmaßnahmen zu realisieren. Die wichtigsten Ziele sind:

- Verhindern, dass vertrauliche interne Daten nach außen gelangen.
- Erlauben, dass interne Benutzer die für sie freigegebenen Dienste des Internets benutzen können.

Es gibt verschiedene Firewall-Architekturen, die diese Ziele auf verschiedenen Wegen und mit unterschiedlichem Erfolg erreichen. Auf den unteren Ebenen des Protokoll-Stack können Router als Paketfilter arbeiten. Sie können auch den Zustand von TCP-Verbindungen speichern und somit den Aufwand beim erstellen der Regelsätze deutlich reduzieren.

Auf der Anwendungsebene bieten Application-Level-Gateways die Möglichkeit, Firewall-Funktionen zu übernehmen. Besonders für HTTP und Webanwendungen werden solche Systeme benutzt. Die Kombination von Paketfiltern und Application-Level-Gateways führt zu einer Architektur, die unter dem Oberbegriff Demilitarized Zone (DMZ) bekannt ist.

Es gibt allerdings keine optimale Firewall für alle Anforderungen. Stattdessen muss im Einzelfall entschieden werden, welchen Firewall-Typ man einsetzen möchte. Entscheidungskriterien sind die Größe des zu schützenden Netzes und der Schutzbedarf dieses Netzes.

Paketfilter und Application-Level-Gateways müssen konfiguriert werden. Konkret muss man entscheiden, welche Pakete oder Dienste die Firewall passieren dürfen und welche nicht. Für nicht explizit definierte Pakete oder Dienste muss man festlegen, ob sie erlaubt sind oder nicht. Höhere Sicherheit erhält man, wenn man die Strategie „alles, was nicht explizit erlaubt ist, ist verboten“ bei der Konfiguration benutzt.

Eine Firewall wird im Betrieb ständig überwacht. Bei erkannten Angriffen sind evtl. Schutzmaßnahmen erforderlich. Aber auch ohne erfolgte Angriffe sollte eine Firewall immer auf dem aktuellsten Stand bleiben. Dazu muss der Administrator die Firewall-Software regelmäßig aktualisieren und die Konfiguration anpassen.

Firewalls sind komplexe Gebilde, die im Rahmen dieses Kurses nicht vollständig abgehandelt werden können. Weitergehende Informationen zum Thema Firewall findet sich in der Literatur bei Chapman und Zwicky [CZ02], bei Goncalves [Gon99] oder bei Spenneberg [Spe11]. Bevor Sie eine Firewall installieren, sollten Sie sich mit Hilfe der Literatur tiefer in das Thema einarbeiten.

4.4 Organisatorische Sicherheitsmaßnahmen

Aufgrund der allgemeinen Aufmerksamkeit im Hinblick auf IT-Sicherheit ist eigentlich jedem klar, dass man sich um die Sicherheit seiner IT-Systeme aktiv kümmern muss. Einige Tipps für Privatanwender mit wenigen PCs wurden bereits in Kapitel 3 gegeben. In diesem Abschnitt soll es darum gehen, was Administratoren größerer Netze mit vielen Systemen machen können, um die Sicherheit ihres Netzes und ihrer Systeme zu erhöhen. Größere Netze werden i. d. R. von Firmen oder Behörden betrieben. Im Folgenden soll der

Begriff *Organisation* eine Firma oder Behörde oder andere ähnliche Einrichtung bezeichnen. Neben der konkreten Administration eines einzelnen Rechners durch einen einzelnen Administrator geht es also auch um weitere Aspekte, wie:

- Wer ist eigentlich der Administrator für welche Rechner? Typischerweise gibt es viele Mitarbeiter und viele Rechner.
- Wer legt fest, was Administratoren machen dürfen, sollen oder müssen? Sicherheit und Bequemlichkeit sind manchmal entgegengesetzte Ziele. Welches ist wichtiger?
- Was sollen Administratoren in bestimmten Situationen bzw. bei bestimmten Ereignissen (z. B. im Notfall) genau machen?
- „Wie viel Sicherheit braucht die Organisation überhaupt?“ In der Regel wird diese Entscheidung nicht vom Administrator eines einzelnen Systems getroffen.
- Wie gefährdet sind die einzelnen Systeme eigentlich? Nicht jedes System ist gleich verwundbar und ein Ausfall kann nahezu irrelevant oder existenzbedrohend sein.

Informationssicherheits-Managementsystem (ISMS)

Eine Organisation als Ganzes braucht hierfür ein **Informationssicherheits-Managementsystem (ISMS)**. Es „umfasst alle Regelungen, die für die Steuerung und Lenkung zur Zielerreichung der Institution sorgen.“ [BSI08b] Dabei bezieht sich der Begriff „Zielerreichung“ natürlich auf die Ziele im Hinblick auf IT-Sicherheit bzw. Informationssicherheit.

In Abschnitt 4.4.1 werden einige Standarddokumente vorgestellt. Sie befassen sich mit denselben Inhalten wie dieser Abschnitt des Kurses und schlagen „Standards“ vor, so dass man nach diesen Standards vorgehen kann. Tut man das, so hat man gute Chancen ein vernünftiges Maß an Sicherheit in der eigenen Organisation zu erreichen; einfach weil man bekannte und bewährte Vorgehensweisen wiederverwendet und nicht „das Rad (möglicherweise fehlerhaft) neu erfindet“.

Sicherheitskonzept

Bevor man aber tatsächlich handelt, sollte man sich überlegen, wie die Situation eigentlich genau ist, welche Risiken existieren und mit welchen Maßnahmen man diesen Risiken begegnen will. Was zu einem **Sicherheitskonzept** gehört, wird in Abschnitt 4.4.3 vorgestellt. Für die Umsetzung des Konzeptes braucht man dann eine passende Vorgehensweise (siehe Abschnitt 4.4.2) und die passende Aufbauorganisation (siehe Abschnitt 4.4.4)

Diese Inhalte werden auch im Kurs *(41760) Informationsmanagement* behandelt. Neben den allgemeinen Aspekten des Informationsmanagements widmet sich die Kurseinheit 5 dieses Kurses explizit dem Thema IT-Sicherheitsmanagement.

4.4.1 IT-Sicherheitsstandards

Standards

Eine Reihe von Organisationen haben sich über das Thema „IT-Sicherheit in der eigenen Organisation“ Gedanken gemacht und die Ergebnisse aufgeschrieben. Damit diese Erkenntnisse wiederverwendet werden können, wurden einige dieser Dokumente als sog. **Standards** veröffentlicht. Die ersten dieser Dokumente beschäftigten sich mit der Sicherheit eines einzelnen IT-Systems (PC, Workstation oder Mainframe), später auch mit dem Thema Sicherheit in einem *Local*

Area Network (LAN) oder auch mit dem weltweiten Internet. Heute werden die Inhalte weiter gefasst. Das Bundesamt für Sicherheit in der Informationstechnik benutzt in diesem Zusammenhang den Begriff **Informationsverbünde** und meint damit „das Zusammenspiel von infrastrukturellen, organisatorischen, personellen und technischen Komponenten, die zur Umsetzung von Geschäftsprozessen und Fachaufgaben dienen." [BSI08b]

Informationsverbünde

Es gibt eine ganze Reihe von Standards im Bereich IT-Sicherheit. In diesem Abschnitt sollen einige der wichtigsten Standards kurz vorgestellt werden.

Trusted Computer System Evaluation Criteria (TCSEC) und Information Security Evaluation Criteria (ITSEC): Unter dem englischen Namen *Orange Book* wurden von 1980 bis 1985 am US National Computer Security Center die *Trusted Computer System Evaluation Criteria* entwickelt. Darin wurden die Schutzstufen A (entspricht formal nachgewiesener Sicherheit), B (systembedingter Schutz), C (benutzerbestimmbarer Schutz) und D (minimaler Schutz) definiert. Dazu kamen die nummerischen Unterstufen C1, C2 und B1, B2 sowie B3.

Einige Betriebssysteme (*UNIX*, *MS Windows*) wurden in die Schutzstufe C eingruppiert, u. a. weil Benutzer Zugriffsrechte vergeben können.

In Europa wurden die Kritikpunkte am *Orange Book* aufgegriffen und ein neues Kriterienwerk, genannt ITSEC, erstellt. Darin werden Funktionsklassen F1 bis F10 definiert. Die Klasse F6 stellt dabei besondere Anforderungen an die Integrität von Daten, F7 an die Verfügbarkeit, F9 an die Vertraulichkeit. Daneben wurden auch Qualitätsstufen (oder auch Evaluationsstufen) festgelegt. E1 erfordert keinerlei besondere Maßnahmen, in Stufe E2 müssen Sicherheitsanforderungen dokumentiert sein, in E3 wird eine informelle Beschreibung gefordert. In den folgenden Stufen werden die Anforderungen immer größer, so dass die oberste Stufe E6 nur durch den Einsatz formaler, mathematischer Methoden erreichbar ist. ITSEC wurde 1991 veröffentlicht.

Common Criteria: Die bis dahin unterschiedlichen Kriterienwerke sollten dann in den *Common Criteria (CC)* vereint werden. 1996 begann die Entwicklung und 1999 wurde die Version 2.1 der Common Criteria veröffentlicht. Sie besteht aus drei Teilen:

1. Introduction and General Model
2. Security Functional Requirements
3. Security Assurance Requirements

Dazu kommen sieben Evaluierungsstufen EAL1 bis EAL7, die an ITSEC angelehnt sind.

BS 7799 / ISO 17799: Der *British Standard (BS)* Nummer 7799 wurde im Jahr 1995 in England vom *Department of Trade and Industry (DTI)* unter dem Titel *Information Technology — Code of practise for information security management* entwickelt. Im Jahr 2000 wurde er von der *International Standards Organisation (ISO)* als ISO-Standard Nummer 17799 akzeptiert. Als die verschiedenen ISO-Standards als Serie mit einheitlichen Nummern (ab 27000 aufwärts) zusammengefasst wurden, bekam dieser Standard die neue Nummer ISO/IEC 27002.

Im Jahr 1999 hat das *British Standards Institute* (BSI[1]) einen zweiten Teil mit dem Titel *Information Security Management System — Specification with guidance for use* veröffentlicht.

Die ISO/IEC 27000 Serie wird gemeinsam von der ISO und der *International Electrotechnical Commission (IEC)* veröffentlicht. Auf den Webseiten der ISO (`http://www.iso.org/`) kann man sich den „Abstract" der Standarddokumente ansehen. Das komplette Dokument kann man dann käuflich erwerben. Bisher sind die folgenden Standards veröffentlicht:

ISO/IEC 27000 enthält eine Einführung in das Thema und definiert die wichtigen Fachbegriffe, die in den folgenden Standards benutzt werden. Der englische Titel lautet *Information security management system — Overview and vocabulary.*

ISO/IEC 27001 , mit dem Titel *Information security management systems — Requirements*, beschreibt die Anforderungen um ein ISMS aufzubauen, zu betreiben, zu überwachen, zu überarbeiten, zu warten, zu verbessern und zu etablieren. Anschließend soll man geeignete Sicherheitsmechanismen kennen, so dass die eigenen Daten und Systeme hinreichend geschützt sind.

ISO/IEC 27002 heißt *Code of practise for Information security management* und enthält allgemeine Richtlinien und konkrete Vorgaben zum Thema ISMS. Man kann es sich als „Best Practice Recommendation" vorstellen.

ISO/IEC 27005 beschäftigt sich mit dem Thema Risikomanagement. Ohne eine konkrete Managementmethode vorzugeben werden Rahmenempfehlungen gegeben. Sie sollen helfen, die Anforderungen, die in ISO/IEC 27001 beschrieben werden, umzusetzen.

ISO/IEC 27006 trägt den langen Namen *Information technology – Security techniques – Requirements for the accreditation of bodies providing certification of information security management systems.* Dahinter verbergen sich Anforderungen für die Überprüfung (Akkreditierung) von „Bodies", die ihrerseits ISMS prüfen und zertifizieren wollen.

IT-Grundschutzhandbuch

Die BSI-100-Reihe: In Deutschland hat das *Bundesamt für Sicherheit in der Informationstechnik (BSI)* seit 1994 das **IT-Grundschutzhandbuch** veröffentlicht. Darin wird neben den Bereichen Organisation, Personal und Infrastruktur auch sehr ausführlich auf die technischen Aspekte eingegangen. Ein Katalog von *Gefährdungen* und ein Katalog von *Maßnahmen* werden in sogenannten *Bausteinen* zusammengeführt. Bei jedem Baustein (z. B. allgemeiner Server, UNIX-Server, Netzkomponente, Betriebssystem, Mainframe usw.) stehen sich die potentiellen Gefährdungen (z. B. Personalausfall, fehlerhafte Datenübertragung, Softwareschwachstellen, Diebstahl usw.) sowie die möglichen Maßnahmen (z. B. Vertretungsregelungen, Integritätsschutzmaßnahmen, Einsatz von Verschlüsselung usw.) gegenüber. Da Gefährdungen und Maßnahmen auf viele Bausteine zutreffen, kann man sich die Beziehungen jeweils als n:m Beziehungen wie in Abbildung 4.10 vorstellen.

[1]Eine Abkürzung (BSI), zwei Institutionen. (1) British Standards Institute und (2) Bundesamt für Sicherheit in der Informationstechnik: unglücklich gelaufen.

Abbildung 4.10: Beziehung zwischen Gefährdungen, Bausteinen und Maßnahmen

BSI 100-1 ist das Pendant zu ISO/IEC 27001. In diesem Dokument beschreibt sich der Standard selbst wie folgt:

> Der vorliegende Standard definiert allgemeine Anforderungen an ein ISMS. Er ist vollständig kompatibel zum ISO-Standard 27001 und berücksichtigt weiterhin die Empfehlungen der ISO-Standards 27000 und 27002. Er bietet Lesern eine leicht verständliche und systematische Anleitung, unabhängig davon, mit welcher Methode sie die Anforderungen umsetzen möchten. Das BSI stellt den Inhalt dieser ISO-Standards in einem eigenen BSI-Standard dar, um einige Themen ausführlicher beschreiben zu können und so eine didaktischere Darstellung der Inhalte zu ermöglichen. Zudem wurde die Gliederung so gestaltet, dass sie mit der IT-Grundschutz-Vorgehensweise kompatibel ist. Durch die einheitlichen Überschriften in den zuvor genannten Dokumenten ist eine Orientierung für die Leser sehr einfach möglich. [BSI08b]

BSI 100-2 beschreibt die IT-Grundschutz-Vorgehensweise. In BSI 100-1 beschreibt das BSI dieses Dokument wie folgt:

> Die IT-Grundschutz-Vorgehensweise beschreibt Schritt für Schritt, wie ein Managementsystem für Informationssicherheit in der Praxis aufgebaut und betrieben werden kann. Die Aufgaben des Informationssicherheitsmanagements und der Aufbau einer Organisationsstruktur für Informationssicherheit sind dabei wichtige Themen. Die IT-Grundschutz-Vorgehensweise geht sehr ausführlich darauf ein, wie ein Sicherheitskonzept in der Praxis erstellt werden kann, wie angemessene Sicherheitsmaßnahmen ausgewählt werden können und was bei der Umsetzung des Sicherheitskonzeptes zu beachten ist. Auch die Frage, wie die Informationssicherheit im laufenden Betrieb aufrechterhalten und verbessert werden kann, wird ausführlich beantwortet.
>
> IT-Grundschutz in Verbindung mit dem BSI-Standard 100-2 interpretiert damit die sehr allgemein gehaltenen Anforderungen der zuvor genannten ISO-Standards 27000, 27001 und 27002 und hilft den Anwendern in der Praxis bei der Umsetzung mit vielen Hinweisen, Hintergrund-Informationen und Beispielen. Die IT-Grundschutz-Kataloge erklären nicht nur, was gemacht werden sollte, sondern geben sehr konkrete Hinweise, wie eine Umsetzung (auch auf technischer Ebene) aussehen kann. Ein Vorgehen nach IT-Grundschutz ist somit eine erprobte und

> effiziente Möglichkeit, allen Anforderungen der oben genannten ISO-Standards nachzukommen. [BSI08b]

Eine prinzipielle Idee des IT-Grundschutzes ist es, dass gemäß der klassischen 80:20-Regel (80% des Ergebnisses erreicht man mit 20% des Aufwands) „vernünftiges Verhalten in den meisten Fällen ausreicht“. Nur in wenigen Fällen (besonders hoher Schutzbedarf) sind aufwendigere Sicherheitsmaßnahmen (angefangen bei einer genaueren Risikoanalyse bis hin zu zusätzlichen Maßnahmen) nötig.

BSI 100-3 hat den Titel Risikoanalyse auf der Basis von IT-Grundschutz.

> Das BSI hat eine Methodik zur Risikoanalyse auf der Basis des IT-Grundschutzes erarbeitet. Diese Vorgehensweise bietet sich an, wenn Unternehmen oder Behörden bereits erfolgreich mit dem IT-Grundschutz arbeiten und möglichst nahtlos eine ergänzende Sicherheitsanalyse an die IT-Grundschutz-Analyse anschließen möchten. [BSI08b]

BSI 100-4 befasst sich mit Notfallmanagement. Auch hier erklärt das BSI den Inhalt dieses Dokumentes in [BSI08b] sehr schön selbst:

> Im BSI-Standard 100-4 wird eine Methodik zur Etablierung und Aufrechterhaltung eines behörden- bzw. unternehmensweiten Notfallmanagements erläutert. Die hier beschriebene Methodik baut dabei auf der in BSI-Standard 100-2 beschriebenen IT-Grundschutz-Vorgehensweise auf und ergänzt diese sinnvoll.

ITIL: In der Betriebswirtschaft hat man erkannt, dass die Informationsverarbeitung in einem Betrieb inzwischen genauso bedeutsam geworden ist wie die originären Geschäfts- oder Produktionsprozesse. Die IT unterstützt die Geschäftsprozesse, wobei heute viele Geschäftsprozesse auf IT angewiesen sind und ohne IT gar nicht mehr ausgeführt werden können. Damit diese kritischen IT-Dienste auch zuverlässig funktionieren, muss ein Betrieb also auch in die IT investieren und die IT sowie die *IT-Services* managen.

IT-Infrastructure-Library (ITIL)

In den 80er Jahren wurde in Großbritannien eine Sammlung von Büchern zum Thema *IT-Service-Management* als **IT-Infrastructure-Library (ITIL)** zusammengefasst. Darin werden *Best Practises*, also bewährte Vorgehensweisen, vorgestellt. Sie bestehen zum einen darin, wie man IT-Dienste vernünftig (zuverlässig, kosteneffizient usw.) anbietet und zum anderen auch darin, wie die Prozesse zur Änderung der IT-Dienste aussehen sollten. Da sich Geschäftsprozesse immer schneller verändern, muss auch die IT sich immer schneller anpassen und evtl. veränderte oder neue Dienste anbieten.

Das Thema IT-Sicherheit wird insofern berührt, als Sicherheitsprobleme immer zu Beeinträchtigungen führen und damit zu Geschäftsproblemen.

4.4.2 Der IT-Sicherheitsprozess

Sicherheit ist keine einmalige Aktion. Es reicht nicht aus, einmal z. B. eine Firewall zu installieren und dann zu hoffen, dass das interne Netz hinreichend geschützt ist. Sicherheit ist eher als Prozess, also als nicht endende Folge

von Aktionen zu verstehen. In diesem Abschnitt soll die Organisation dieses Prozesses besprochen werden.

In der Wirtschaft ist der sog. „kontinuierliche Verbesserungsprozess“ die Grundlage des Qualitätsmanagements. In einem zyklischen Prozess werden immer wieder dieselben Schritte durchlaufen, nämlich:

Plan: Erkennen des Handlungsbedarfes und „geistiges Vorwegnehmen von zukünftigem Handeln“ (planen).

Do: Durchführung der geplanten Maßnahmen, manchmal auch erst in kleinerem Rahmen als Test.

Check: Analysieren und prüfen, ob die Maßnahmen die angestrebten Ziele tatsächlich erreichen (Erfolgskontrolle).

Act: Die erfolgreichen Maßnahmen werden zum „Standard“ erklärt und komplett umgesetzt (falls bei *Do* nur ein kleinerer Test erfolgte).

Dieses PDCA-Modell genannte Vorgehen wurde von Deming [Dem00] eingeführt. Nachdem der Act-Schritt durchlaufen wurde, geht es wieder am Anfang im Plan-Schritt weiter. Die veränderte Situation kann zu weiterem Handlungsbedarf führen und eine erneute Planung erfordern.

Das Bundesamt für Sicherheit in der Informationstechnik BSI (siehe auch `http://www.bsi.de`) schlägt in seinem Grundschutzhandbuch die in Abbildung 4.11 dargestellten Schritte für den Sicherheitsprozess vor. BSI

Der Prozess besteht demnach aus vier Phasen, der Initiierung, der Planung, der Umsetzung der geplanten Maßnahmen und der anschließenden Aufrechterhaltung des Sicherheitsniveaus. Während der Initiierung muss die Leitung im Wesentlichen das grobe Ziel vorgeben, die Organisation aufbauen und Ressourcen bereitstellen.

Die Planung besteht aus der Erstellung eines detaillierten Sicherheitskonzepts. Im ihm wird festgehalten, was zu schützen ist und wie es geschützt werden soll. Die Umsetzung besteht auch aus zwei Schritten: der Realisierung der Maßnahmen aus dem Konzept und der Sensibilisierung und Schulung der Benutzer. Der letzte Schritt besteht in der Aufrechterhaltung und Verbesserung, also der Kontrolle des laufenden Betriebs. Aufgrund neuer Bedrohungen kann es dann erforderlich sein, das Sicherheitskonzept zu überarbeiten. Die Pfeile in Abbildung 4.11 deuten den zeitlichen Ablauf der Schritte an. Im Folgenden werden die einzelnen Schritte des Prozesses beschrieben.

Initiierung des IT-Sicherheitsprozesses: Die Leitung der Organisation muss größere Veränderungen „von oben“ anstoßen. Und die Einführung eines IT-Sicherheitsprozesses ist eine größere Veränderung in einer Organisation.

Die IT-Sicherheitsleitlinie enthält das grundsätzliche Bekenntnis einer Organisation zum Thema IT-Sicherheit. Im Englischen wird die Sicherheitsleitlinie auch *Security Policy* genannt.

In diesem Dokument wird allgemein verständlich beschrieben „für welche Zwecke, mit welchen Mitteln und mit welchen Strukturen Informationssicherheit innerhalb der Institution hergestellt werden soll.“ [BSI08a] Die Leitung der Organisation/Institution ist für die Sicherheitsleitlinie verantwortlich, denn es handelt sich um eine strategische Vorgabe für alle Mitglieder/Beschäftigte.

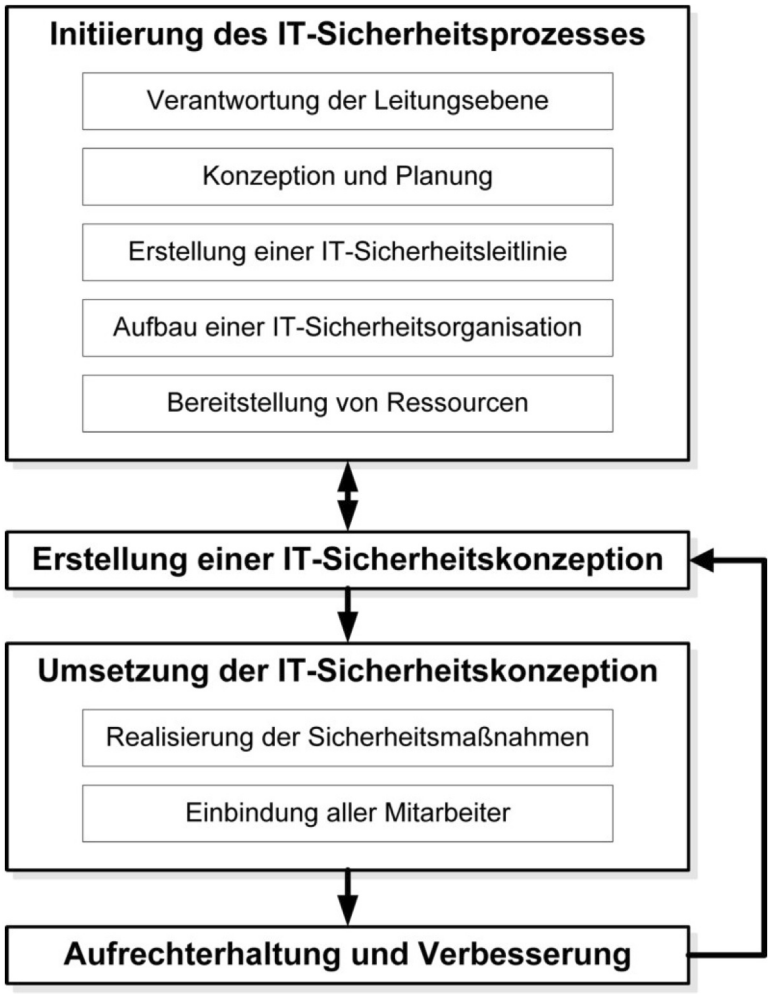

Abbildung 4.11: IT-Sicherheitsprozess des BSI

Neben dem Geltungsbereich enthält die Sicherheitsleitlinie auch eine Beschreibung der Sicherheitsziele und der Organisationsstruktur (siehe Abschnitt 4.4.4). Diese Organisationsstruktur muss natürlich auch geschaffen werden. All das ist mit Aufwand verbunden und dazu sind entsprechende Personal- und Finanzressourcen erforderlich.

Erstellung einer IT-Sicherheitskonzeption: Dieser Schritt ist etwas umfangreicher und wird deshalb in Abschnitt 4.4.3 genauer erklärt.

Umsetzung der IT-Sicherheitskonzeption: Nachdem im Sicherheitskonzept festgelegt wurde, mit welchen Maßnahmen man die identifizierten Risiken minimieren will, wird ein Plan zur Realisierung der Maßnahmen erstellt. Er enthält die konkreten Aktionen, die erforderlich sind. Die folgenden Punkte gehören hierzu:

Realisierung der Maßnahmen

- Arbeitspläne
- Verantwortlichkeiten
- Kosten von Investitionen, Arbeitsaufwand etc.
- Termine

Somit ist definiert, *wer* für die Realisierung einer Maßnahme zuständig ist, *was* genau zu tun ist, *wieviel* es kosten darf und bis *wann* es fertig sein soll. Ein Beispiel:

Maßnahme	Wer	Wieviel	Bis wann
Zugangsschutz zum Serverraum durch neue Schlösser realisieren	Herr Meier	EUR 90	30.4.2015
Mit AES verschlüsselte Kommunikation zwischen A und B realisieren	Frau Müller	EUR 30 000	1.11.2017

In der Praxis können die Pläne zur Realisierung der Maßnahmen auch schon detaillierter vorbereitet sein. Das hängt i. d. R. von der Art der Maßnahme und der Person ab, die damit beauftragt wird.

Einbindung der Mitarbeiter

Die realisierten Sicherheitsmaßnahmen alleine reichen noch nicht. Sie müssen von den Benutzern auch umgesetzt werden. Das neue Schloss, das den Zugang zum Serverraum beschränken soll, nützt nichts, wenn der Schlüssel immer steckt. Ein Schulungs- und Sensibilisierungsprogramm soll die Mitarbeiter

- über die Sicherheitsleitlinie und die Sicherheitsmaßnahmen *informieren*,
- durch weitergehende Erläuterungen *motivieren*, d. h. es soll klar machen, warum Sicherheit erforderlich ist,
- mit den Maßnahmen *vertraut machen* und die korrekte Umsetzung erläutern,
- auf die Konsequenzen aufmerksam machen, wenn gegen die Sicherheitsleitlinie verstoßen wird.

Diese Schulungsmaßnahmen sollten zeitnah erfolgen, d. h. kurz bevor oder kurz nachdem die Sicherheitsmaßnahmen realisiert wurden. Neue oder versetzte Mitarbeiter sollten möglichst umgehend an ihnen teilnehmen.

Kontrolle

Aufrechterhaltung und Verbesserung: Im laufenden Betrieb ist es laut IT-Grundschutzhandbuch des BSI notwendig zu prüfen,

- ob die ausgewählten Sicherheitsmaßnahmen eingesetzt werden,
- ob sie korrekt eingehalten werden,
- ob sie den Anforderungen im laufenden Betrieb genügen, und
- ob sie bei Änderungen nach wie vor relevant sind oder ebenfalls verändert werden müssen.

Da sich die Risiken und Bedrohungen in der IT immer weiter entwickeln und neue Bedrohungen (z. B. neue Viren, verteilte *Denial-of-Service*-Angriffe usw.) ergeben, muss darauf angemessen reagiert werden. Sie führen dazu, dass man erneut über das Sicherheitskonzept nachdenkt. Die Schleife in Abbildung 4.11 zeigt diese Rückkopplung, die nach der Anpassung des Sicherheitskonzepts auch die Überarbeitung der nachfolgenden Schritte erforderlich macht.

4.4.3 Die IT-Sicherheitskonzeption

Das IT-Sicherheitskonzept ist die Dokumentation des Informationsverbundes, der Gefährdungen/Risiken, des Schutzbedarfes und der Maßnahmen, die man umsetzen möchte. Seine Bestandteile werden in diesem Abschnitt erläutert.

Strukturanalyse

Strukturanalyse: Die Erstellung des Sicherheitskonzepts beginnt mit einer Strukturanalyse. Darin beschafft man sich einen Überblick über die zu schützenden IT-Ressourcen. Zu diesen Ressourcen gehören Hardwaresysteme aller Art (Computer, Netzwerke etc.), aber auch die Anwendungen und Daten (Dateien, Datenbanken etc.). Ein Plan der Netztopologie ist ein guter Ausgangspunkt für die Strukturanalyse.

Anwendungen

Auf den Systemen laufen Anwendungen, die Geschäftsprozesse der Organisation/Institution unterstützen. Dazu verarbeiten die Anwendungen Daten, die zu schützen sind. Daher sind in der Strukturanalyse auch die Anwendungen, die Daten der Anwendungen, die Benutzer der Anwendung und die Geschäftsprozesse, die von der Anwendung unterstützt werden, festzuhalten. Diese Informationen sind für den nächsten Schritt (die Schutzbedarffeststellung) wichtig. Meistens ergibt sich der Schutzbedarf einer Anwendung aus dem Schutzbedarf der Daten, die diese Anwendung verarbeitet. Und den Schutzbedarf der Daten kennen die Benutzer der Anwendung i. d. R. am besten.

Da in größeren Organisationen sehr viele Systeme, Anwendungen, Daten und Benutzer existieren, ist es unumgänglich die Komplexität der Strukturbeschreibung zu reduzieren. Dazu gruppiert man in geeigneter Art und Weise. Beispielsweise muss man bei Arbeitsplatzrechnern nicht jeden einzelnen Rechner aufnehmen, sondern kann eine Gruppe *Buchhaltungs-Clients* o. ä. definieren. Der Schutzbedarf aller Systeme einer Gruppe ist dann identisch.

Schutzbedarf-feststellung

Schutzbedarffeststellung: Nachdem nun geklärt ist, was geschützt werden soll, überlegt man wie gut es zu schützen ist. Man legt dabei fest, welches Sicherheitsniveau angestrebt werden soll. Das Niveau kann von „kaum schützenswert“, bis zu „unter allen Umständen geschützt halten“ reichen. Man muss also entscheiden, was notwendig und angemessen ist. Die Verfügbarkeit von Zahlungsverkehrssystemen ist für eine Bank überlebenswichtig, während ein „Marketing“-Webserver auch einmal ausfallen darf, ohne dass daran eine Firma zu Grunde geht.

Der Schutzbedarf ergibt sich aus den möglichen Schäden, die bei der Verletzung eines der Schutzziele (Vertraulichkeit, Integrität, Authentizität und Verfügbarkeit) eintreten können. Die Schäden können bestimmten Kategorien zugeordnet werden:

Schaden-Kategorien

Finanzschäden: Die Organisation verliert Geld, weil Umsätze verloren gehen, Strafen bezahlt werden müssen, Geld für Schadensbehebung ausgegeben werden muss, ...

Juristische Verstöße: Im Schadensfalle würde die Organisation gegen Gesetze verstoßen oder Verträge nicht erfüllen.

Operationelle Schäden: Die Organisation kann ihre Aufgaben nicht mehr erfüllen.

Persönliche Schäden: Die Organisation könnte dafür verantwortlich sein, dass anderen Personen Schäden an Leib und Leben entstehen oder dass das Grundrecht auf informationelle Selbstbestimmung nicht beachtet wird.

Imageschäden: Die Organisation muss mit Ansehensschäden und Vertrauensbeeinträchtigungen rechnen.

Je nachdem wie groß die Schäden in den Szenarien werden können, ist dann auch der Schutzbedarf. Für die „Höhe“ des Schutzbedarfes benutzt man gerne Kategorien. Das BSI schlägt drei Kategorien vor [BSI08a]: Schutz-Kategorien

1. **Normal**. Dieser Schutzbedarf liegt vor, wenn die Schadensauswirkungen begrenzt und überschaubar sind. Normal
2. **Hoch**. Dieser Schutzbedarf liegt vor, wenn die Schadensauswirkungen beträchtlich sein können. Hoch
3. **Sehr hoch**. Dieser Schutzbedarf wird gewählt, wenn die Schadensauswirkungen ein existentiell bedrohliches oder katastrophales Ausmaß erreichen können. Sehr hoch

Umsetzung eines Grundschutzes: Damit nun nicht für jeden Punkt aus der Strukturanalyse eine Risikoanalyse durchgeführt werden muss, sollten für alle „normalen“ Punkte die „normalen“ Sicherheitsmaßnahmen ausreichen. Im Grundschutzhandbuch hat das BSI die typischen Risiken in sehr vielen Einsatzszenarien analysiert und entsprechende Sicherheitsmaßnahmen vorgeschlagen. Zur Umsetzung des Grundschutzes braucht eine Organisation also nur zu prüfen, ob die Maßnahmen aus dem Grundschutzhandbuch umgesetzt werden.

Nur bei signifikant höherem Schutzbedarf, beispielsweise wenn streng geheime Daten verarbeitet werden sollen, sollte man die im Folgenden beschriebene Risikoanalyse durchführen und bei Bedarf zusätzliche Sicherheitsmaßnahmen umsetzen.

Ergänzende Risikoanalyse und Sicherheitsmaßnahmen: Für die Risikoanalyse gibt es zwei Vorgehensweisen: (1) die allgemeine Risikoanalyse und (2) die Risikoanalyse auf der Basis von IT-Grundschutz [BSI08c]. Variante (2) setzt voraus, dass eine Sicherheitskonzeption nach IT-Grundschutz erstellt wurde.

Betrachten wir zunächst die allgemeine Risikoanalyse. Dabei geht es nicht um allgemeine Risiken, sondern um die Risiken, die das Erreichen der oben definierten Sicherheitsziele gefährden können. Verschiedene Risiken haben natürlich verschiedene Auswirkungen auf die Schutzziele. Deshalb bewertet man die Risiken nach folgenden Kriterien:

1. Eintrittswahrscheinlichkeit
2. Ausmaß des potentiellen Schadens

In einer *Probability-Impact*-Matrix visualisiert man das Ergebnis der Bewertung. Dazu wird auf der X-Achse die Eintrittswahrscheinlichkeit und auf der Y-Achse das Ausmaß der potentiellen Schäden eingetragen.

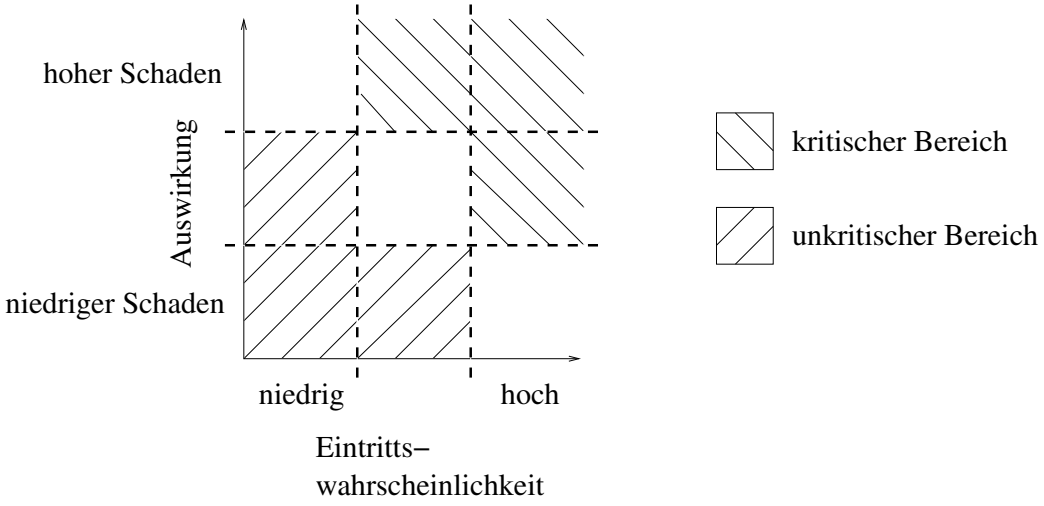

Abbildung 4.12: Probability-Impact-Matrix

Abbildung 4.12 zeigt das Prinzip. Dabei reicht es zunächst, wenn man pro Achse drei Klassen (niedrig, mittel, hoch) bildet und alle identifizierten Risiken in das Diagramm als Punkte einträgt. Die Risiken in der linken unteren Ecke, d. h. aus den drei Bereichen unterhalb der Diagonale von links oben nach rechts unten, können für die folgenden Betrachtungen zunächst ignoriert werden. In Abbildung 4.12 sind sie als unkritische Bereiche markiert. Die Risiken aus der rechten oberen Ecke, also wieder den drei Bereichen oberhalb der Diagonalen, müssen durch geeignete Schutzmaßnahmen vermindert werden. Sie sind die kritischen Risiken. Die Risiken in den weißen Feldern sollten mindestens beobachtet werden, auch wenn sie zunächst tolerierbar erscheinen. Sie könnten zukünftig in der Eintrittswahrscheinlichkeit steigen oder zu größerem Schadenausmaß führen. Dann würden sie in die rechte obere Ecke „wandern“ und müssten aktiv angegangen werden.

Beispiel

Das Risiko „Passwort wird bei der Übertragung durch das Internet ausspioniert“ wird man i. d. R. in den rechten oberen Bereich einordnen. Potentielle Gegenmaßnahmen wären die Auswahl eines anderen Übertragungsweges (z. B. Briefpost) oder die Verschlüsselung des Passworts vor der Übertragung.

Übungsaufgabe 4.5 *Ordnen Sie die folgenden Risiken in eine Probability-Impact-Matrix ein:*

1. *Ein Hacker manipuliert die Homepage derart, dass dort dumme Witze und Links zu zwielichtigen Webservern installiert werden.*
2. *Der Administratorzugang zum Webserver ist eine Telnet-Verbindung über das Internet.*
3. *Durch Probleme in der internen Ablauforganisation werden die HTML-Seiten auf dem Server nicht rechtzeitig aktualisiert, sondern mit einer Stunde Verzögerung.*

Weiterhin bestimmen gesetzliche Regelungen, welches Sicherheitsniveau beispielsweise von Aktiengesellschaften mindestens eingehalten werden muss.

KonTraG

Im Gesetz zur Kontrolle und Transparenz im Unternehmensbereich (KonTraG) werden Aktiengesellschaften zur Risikovorsorge verpflichtet. Weiterhin sind natürlich immer auch die Datenschutzgesetze und -richtlinien zu beachten.

Die Realisierung eines Sicherheitsniveaus ist nicht umsonst zu haben. Es ist immer auch mit Aufwand verbunden, der umso größer ist, je höher das angestrebte Sicherheitsniveau ist. Bei der Erstellung eines Sicherheitskonzepts muss man also auch abwägen, mit wie viel Aufwand die Realisierung verbunden ist. Ein Sicherheitskonzept, das mit einem finanziellen Aufwand von EUR 1 Million ein Risiko minimiert, das im Schadensfall einen Schaden von EUR 10 verursacht, ist nicht wirtschaftlich. In diesem Fall lebt man besser mit dem Risiko und legt EUR 10 für den Schadensfall[2] zurück.

Risikoanalyse auf der Basis von IT-Grundschutz

Bei der Risikoanalyse auf der Basis von IT-Grundschutz streicht man alle Systeme/Anwendungen/Daten, die durch die Grundschutzmaßnahmen bereits hinreichend geschützt sind. Für die verbleibenden Punkte (mit besonders großem Schutzbedarf) betrachtet man nun die im Grundschutzhandbuch vorgestellten Gefährdungen und erstellt eine Risikoanalyse für diese Gefährdungen. Außerdem überlegt man, ob es weitere Gefährdungen gibt. Wenn ja, dann wird auch für sie eine Risikoanalyse durchgeführt. Weitere Details hierzu stehen in BSI-Standard 100-3 [BSI08c].

Ausgehend von den Sicherheitsmaßnahmen im Grundschutzhandbuch prüft man dann weitere Sicherheitsmaßnahmen. Zur Reduzierung des Risikos kann man auch das Risiko abwälzen, beispielsweise indem man eine Versicherung darüber abschließt.

4.4.4 Eine IT-Sicherheitsorganisation

Jede größere Behörde, Firma oder Organisation hat eine interne Struktur. In dieser Struktur ist festgelegt, wie die Aufgaben und Verantwortlichkeiten verteilt sind. Man bezeichnet diese Struktur auch als Organisation. In Form von Organigrammen wird grafisch dargestellt, aus welchen Teilen die Organisation besteht. In diesem Abschnitt geht es darum, wie man diese Struktur definieren kann, so dass sie den Anforderungen an das Thema Sicherheit besser entspricht.

Bei der Erstellung einer IT-Sicherheitsleitlinie und eines IT-Sicherheitskonzepts müssen Personen aus dem Geschäftsbereich einer Organisation und IT-Experten zusammenarbeiten. Die Geschäftsexperten kennen die verarbeiteten Daten, deren Wert und die Geschäftsprozesse. Nur sie können die Auswirkungen möglicher Bedrohungen für das Geschäft der Organisation abschätzen. Allerdings kennen sie die verschiedenen technischen Bedrohungen nicht. Hierzu ist das Know-how von IT-Security-Experten erforderlich. Sie kennen die Technik und die Missbrauchsmöglichkeiten.

In einer Organisation müssen die Art dieser Zusammenarbeit, die jeweiligen Aufgaben, Zuständigkeiten und Verantwortlichkeiten möglichst genau definiert werden. Hierfür gibt es allerdings keine allgemeingültige Aufbauorganisation. Vielmehr hängt sie davon ab, welche Geschäfte die Organisation macht, wie sie bisher aufgebaut ist und wie groß sie ist. Im Folgenden sollen zumindest einige Rollen definiert werden. Abbildung 4.13 zeigt eine Möglichkeit für die Organisation.

Topmanagement

Topmanagement: Das Topmanagement der Organisation muss sich zum Thema IT-Security bekennen. Konkret muss es die Bedeutung des Themas

[2] Vorausgesetzt, dass der Schadensfall nun nicht alle 10 Sekunden eintritt.

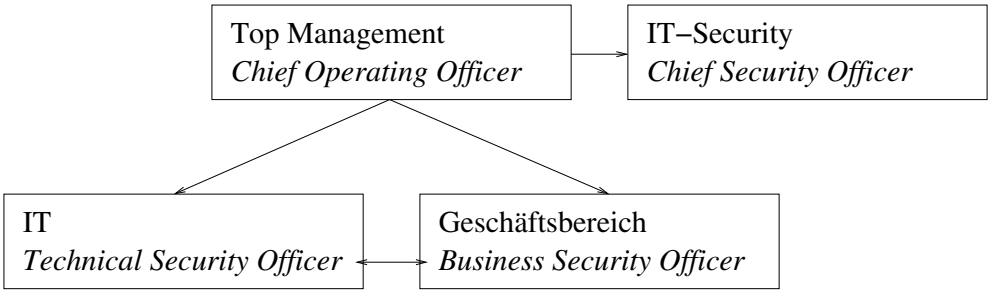

Abbildung 4.13: Mögliche Organisationsstruktur für IT-Security

akzeptieren und alle Teile der Organisation für das Thema sensibilisieren. Weiterhin kann nur das Topmanagement Veränderungen in der Organisationsstuktur und den internen Verfahren und Arbeitsabläufen anstoßen. Außerdem bestimmt das Topmanagement die Verantwortlichen, die für die Aufrechterhaltung der Sicherheit zuständig sind.

IT-Security **IT-Security:** Fachlich wird das Management vom Bereich IT-Security unterstützt. Dort wird die organisationsweite Sicherheitsleitlinie ausgearbeitet. In ihr ist festgelegt, welches die zu schützenden Ressourcen der Organisation sind und welches Sicherheitsniveau angestrebt werden soll. Weiterhin kümmert sich dieser Bereich auch um neue Entwicklungen, die Einfluss auf die Sicherheitsleitlinie haben können. Zu diesen Entwicklungen gehören beispielsweise:

- gesetzliche Vorgaben (Gesetz zur digitalen Signatur, KonTraG, Datenschutzgesetze usw.),
- Standardisierungsbestrebungen (DES, AES usw.) und
- aktuelle Forschungsergebnisse (z. B. Analyse von Schwachstellen in Verschlüsselungsverfahren).

Daher hat dieser Bereich auch die Aufgabe, die anderen Bereiche, insbesondere die IT und die Geschäftsbereiche zu unterstützen.

In diesem Bereich wird auch über die erforderlichen Schulungs- und Sensibilisierungsmaßnahmen entschieden. Hier können Informationskampagnen zum Thema organisiert werden oder es werden spezielle Kurse zur Schulung aller Mitarbeiter konzipiert.

Geschäftsbereich **Geschäftsbereiche:** Ein Geschäftsbereich kennt die dort anfallenden Daten, deren Wert und den Schutzbedarf. Er ist also verantwortlich dafür, dass die Daten (bzw. allgemeiner alle Ressourcen) hinreichend geschützt sind. Es bietet sich an, hierfür eine Person zu benennen, die diese Verantwortung trägt. Dieser *Business-Security-Officer* arbeitet zur Erreichung der angestrebten Schutzziele mit dem Bereich IT zusammen.

IT **IT:** In der IT übernimmt ein *Technical-Security-Officer* die Verantwortung dafür, dass die vom Geschäftsbereich vorgegebenen Schutzziele durch passende technische Schutzmaßnahmen umgesetzt werden. Allgemein hat die IT die Aufgabe, die IT-Systeme für den Geschäftsbereich zu betreiben, so dass der Geschäftsbereich mit Hilfe der Systeme seine Geschäfte machen kann. Außerdem

entwickelt die IT die Systeme so weiter, dass sie den neuen und kommenden Anforderungen der Geschäftsbereiche entsprechen. Zusammen mit dem Bereich IT-Security sorgt der Technical-Security-Officer dafür, dass die konkreten Systeme für den Geschäftsbereich in Übereinstimmung mit den organisationsweiten IT-Security-Vorgaben realisiert werden.

Das folgende kurze und unvollständige Beispiel soll diese Zusammenarbeit verdeutlichen: Beispiel

Topmanagement: Das Topmanagement beschließt, dass die Organisation besonders hohe Sicherheitsanforderungen hat, da in ihr vertrauliche Daten bearbeitet werden. Man legt daher fest, dass vertrauliche Daten grundsätzlich verschlüsselt werden sollen.

Weitergehende Vorgaben werden an dieser Stelle nicht getroffen.

IT-Security: Dieser Bereich legt nun fest, welche Verschlüsselungsverfahren und welche Schlüssellängen in der Organisation eingesetzt werden sollen. Man gibt beispielsweise vor, dass nur gut untersuchte und standardisierte Verfahren zum Einsatz kommen sollen.

Diese Festlegung könnte wie folgt aussehen:

Algorithmus	Schlüssellänge
Triple-DES	168 Bit
AES	128 Bit
RC5	128 Bit

Geschäftsbereich: Hier wird für einen konkreten Anwendungsfall festgelegt, welche der Daten vertraulich sind und welche nicht. In einem Prüfungsamt einer Universität könnte die Analyse ergeben, dass Namen und Matrikelnummern nicht vertraulich sind, die Prüfungsergebnisse allerdings schon.[3] Man legt also fest, dass die Datei `xyz` mit den Prüfungsergebnissen verschlüsselt werden muss.

IT: Unter Berücksichtigung dieser Informationen und der vorhandenen IT-Infrastruktur wird überlegt, welche Maßnahmen erforderlich sind. Dazu könnte gehören, dass man *GnuPG* in der Version 2.0.13 einsetzen wird. Es wird so implementiert, dass AES mit 128 Bit Schlüssellänge eingesetzt wird.

Dieses Beispiel ist stark verkürzt und verdeutlicht nur das prinzipielle Zusammenspiel der Rollen. In der Praxis sind selbst bei diesem kleinen Beispiel viele weitere Aspekte zu beachten. Dazu gehört u. a. das Schlüsselmanagement (wer erzeugt die Schlüssel, wer kennt sie, wie sind die Vertretungsverfahren geregelt usw.).

Zum Abschluss dieses Abschnitts soll noch einmal auf das Bundesamt für Sicherheit in der Informationstechnik hingewiesen werden. Dort werden insbesondere auch die organisatorischen Aspekte von IT-Security behandelt und beispielsweise im IT-Grundschutzhandbuch und den weiteren Standards dokumentiert. Im Internet findet man das BSI unter der URL:

`http://www.bsi.de/`

[3] Ohne es genau zu wissen, vermute ich, dass die Prüfungsämter der FernUniversität alle Daten als vertraulich betrachten. Das Beispiel ist also hypothetisch.

4.5 Zusammenfassung

Nach dem Durcharbeiten dieser Kurseinheit sollten Sie die folgenden Fragen beantworten können:

- Welche vom Betriebssystem unabhängigen Maßnahmen sind für den sicheren Betrieb eines Webservers erforderlich?
- Welche zusätzlichen Maßnahmen sind unter den Betriebssystemen *Windows XP* (bzw. den Nachfolgeversionen) und *UNIX* erforderlich?
- Wo finden Sie aktuelle Informationen zu Sicherheitsproblemen?
- Was ist bei der Konfiguration eines Webservers wie *Apache* zu beachten?
- Welche Funktionen bietet eine Firewall?
- Wie funktionieren Paketfilter, Stateful Paketfilter und Application-Level-Gateways?
- Welche Eigenschaften habe diese Systeme und welche Schutzziele kann man mit ihnen erreichen?
- Wie kann man diesen Systemen komplexere Firewall-Architekturen realisieren?
- Was ist beim Betrieb einer Firewall zu beachten?
- Wie sollte man bei erkannten Angriffen auf die Sicherheit vorgehen?
- Wie sieht der IT-Sicherheitsprozess aus?
- Was enthält eine IT-Sicherheitskonzeption?
- Welche organisatorischen Regelungen kann man in einer Firma oder Behörde treffen, damit man IT-Sicherheit einfach realisieren kann?

Lösungen der Übungsaufgaben

Übungsaufgabe 4.1 Wenn die Datei mit den kryptografischen Prüfsummen auf der Festplatte gespeichert ist, dann kann ein Angreifer seine Manipulationen an anderen Dateien verschleiern, indem er einfach auch die betreffenden Prüfsummen anpasst. Dazu berechnet er die kryptografische Prüfsumme der veränderten Datei und schreibt sie zu den anderen Prüfsummen.

Deshalb sollte man die Prüfsummen auf einem Medium speichern, auf das der Angreifer keinen Zugriff hat. Ein USB-Stick müsste also abgezogen und getrennt vom Rechner gelagert werden.

Übungsaufgabe 4.2 Sollte ein Angreifer eine Schwachstelle im Webserver ausnutzen und dann eine Shell zur Kommandoeingabe bekommen, könnte er als Benutzer *wwwrun* Kommandos ausführen. Er könnte sich dann Schreibrechte für `httpd.conf` geben und alle Sicherheitsmechanismen, die darin beschrieben sind abschalten. Nach dem nächsten Start des Webservers würden sich somit viele weitere Angriffsmöglichkeiten ergeben.

Übungsaufgabe 4.3 Eine Liste der in einem Verzeichnis auf dem Webserver gespeicherten Dateien erlaubt einem Angreifer Einblick in Informationen, die man eigentlich nicht veröffentlichen wollte. Kennt man den Dateinamen, so kann man einfach die zugehörige URL in den Browser eintippen und bekommt vom Webserver die Seite geliefert. Dies funktioniert auch dann, wenn *überhaupt kein* Link auf diese Seite zeigt.

Übungsaufgabe 4.4 Der Paketfilter kann keine Adressumsetzung zwischen privaten IP-Adressen und den „offiziellen“ Internet-IP-Adressen durchführen. Ein Router leitet i. d. R. Pakete nur weiter. Wollte der Router auch die Adressen umsetzen, so müsste er Informationen über den Zustand der Verbindungen speichern. Externe Rechner kennen ja keine internen Adressen und schicken alle Pakete an den Router. Der muss daher nachschauen können, wohin das Paket weitergeleitet werden soll. Also muss er auch wissen, auf welche ausgehende Anfrage das eingehende Paket eigentlich antwortet.

Übungsaufgabe 4.5 Abbildung 4.14 zeigt, wie man die drei Risiken klassifizieren kann. Diese Klassifizierung ist natürlich subjektiv und hängt von vielen Randbedingungen ab, die in der Aufgabenstellung nicht genannt wurden.

Durch die grafische Anordnung erkennt man allerdings gut, welchen Risiken man sich zuerst widmen muss. Auch wird bei der Auswahl der Maßnahmen klar, dass diese Maßnahmen mehrere Risiken auf einmal verändern können. Der Einsatz von *SSH* beispielsweise vermindert häufig mehrere Risiken auf einmal.

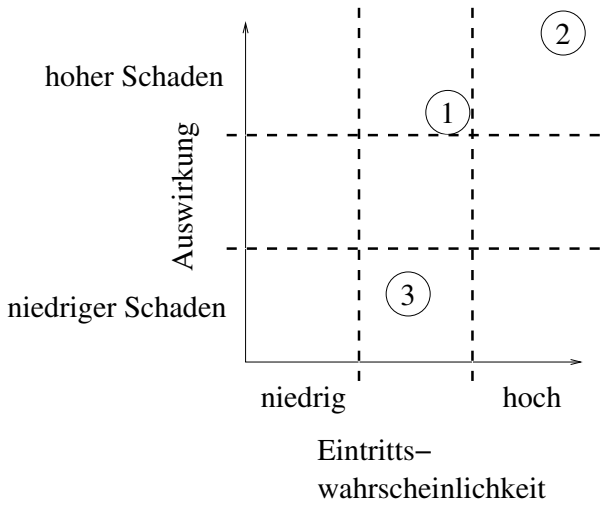

Abbildung 4.14: Risikoklassifizierung

Literatur

[AKS02] Manindra Agrawal, Neeraj Kayal und Nitin Saxena. „PRIMES is in P“. In: *Ann. of Math* 2 (2002), S. 781–793.

[And08] Ross J. Anderson. *Security Engineering*. 2. Aufl. Wiley und Sons, 2008.

[Ano03] Anonymous. *Hacker's Guide*. Übersetzung von Maximum Security, 4th ed. München, Germany: Markt+Technik Verlag, 2003.

[Ano98] Anonymous. *Maximum Security*. 2. Aufl. Indianapolis, Indiana: SAMS, 1998.

[Bau97] Friedrich L. Bauer. *Entzifferte Geheimnisse. Methoden und Maximen der Kryptologie*. Heidelberg, Germany: Springer-Verlag, 1997.

[Ben+08] Jens Bender, Dennis Kügler, Marian Margraf und Ingo Naumann. „Sicherheitsmechanismen für kontaktlose Chips im deutschen elektronischen Personalausweis - Ein Überblick über Sicherheitsmerkmale, Risiken und Gegenmaßnahmen“. In: *Datenschutz und Datensicherheit* 32.3 (2008), S. 173–177.

[Ber+07] G. Bertoni, J. Daemen, M. Peeters und G. Van Assche. *Sponge functions*. Ecrypt Hash Workshop 2007. Mai 2007.

[Ber+11a] G. Bertoni, J. Daemen, M. Peeters und G. Van Assche. *The* Keccak *reference*. `http://keccak.noekeon.org/`. Jan. 2011.

[Ber+11b] G. Bertoni, J. Daemen, M. Peeters und G. Van Assche. *The* Keccak *SHA-3 submission*. `http://keccak.noekeon.org/`. Jan. 2011.

[BF01] Dan Boneh und Matthew K. Franklin. „Identity-Based Encryption from the Weil Pairing“. In: *CRYPTO*. Hrsg. von Joe Kilian. Bd. 2139. Lecture Notes in Computer Science. Springer, 2001, S. 213–229. ISBN: 3-540-42456-3.

[BPH02] L. Bassham, W. Polk und R. Housley. „Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation Lists (CRL) Profile“. `ftp://ftp.rfc-editor.org/in-notes/rfc3280.txt`. Apr. 2002.

[Bra99] John R. T. Brazier. „Possible NSA Decryption Capabilities“. In: *DuD Datenschutz und Datensicherheit* 23.10 (Okt. 1999), S. 576–581.

[BSB05] Daniel J. Barrett, Richard Silverman und Robert G. Byrnes. *SSH The Secure Shell — The Definitive Guide*. 2. Aufl. O'Reilly, 2005.

[BSI08a] BSI. *IT-Grundschutz-Vorgehensweise*. BSI-Standard 100-2, Version 2.0. Mai 2008.

[BSI08b] BSI. *Managementsysteme für Informationssicherheit (ISMS)*. BSI-Standard 100-1, Version 1.5. Mai 2008.

[BSI08c] BSI. *Risikoanalyse auf der Basis von IT-Grundschutz*. BSI-Standard 100-3, Version 2.5. Mai 2008.

[BSI12a] BSI. *Advanced Security Mechanisms for Machine Readable Travel Documents – Part 1 - eMRTDs with BAC/PACEv2 and EACv1*. Techn. Ber. TR-03110-1. Bundesamt für Sicherheit in der Informationstechnik, 2012.

[BSI12b] BSI. *Advanced Security Mechanisms for Machine Readable Travel Documents – Part 2 - Extended Access Conrol Version 2 (EACv2), Password Authenticated Connection Establishment (PACE) and Restricted Indetification (RI)*. Techn. Ber. TR-03110-2. Bundesamt für Sicherheit in der Informationstechnik, 2012.

[BSI12c] BSI. *Advanced Security Mechanisms for Machine Readable Travel Documents – Part 3 - Common Specifications*. Techn. Ber. TR-03110-3. Bundesamt für Sicherheit in der Informationstechnik, 2012.

[Coc01] Clifford Cocks. „An Identity Based Encryption Scheme Based on Quadratic Residues“. In: *IMA Int. Conf.* Hrsg. von Bahram Honary. Bd. 2260. Lecture Notes in Computer Science. Springer, 2001, S. 360–363. ISBN: 3-540-43026-1.

[CZ02] D. Brent Chapman und Elizabeth D. Zwicky. *Einrichten von Internet Firewalls*. Sebastopol, CA: O'Reilly, 2002.

[Dem00] W. Edwards Deming. *Out of the Crisis*. 2. Aufl. MIT Press, Oktober 2000.

[DH76] W. Diffie und M. Hellman. „New directions in cryptography“. In: *IEEE Transactions on Information Theory* 22.6 (Sep. 1976), S. 644–654. ISSN: 0018-9448. DOI: 10.1109/TIT.1976.1055638. URL: http://dx.doi.org/10.1109/TIT.1976.1055638.

[DR02] Joan Daemen und Vincent Rijmen. *The Design of Rijndael*. Berlin Heidelberg: Springer Verlag, 2002.

[DR99] Joan Daemen und Vincent Rijmen. *AES Proposal: Rijndael*. http://www.esat.kuleuven.ac.be/~rijmen/rijndael/rijndaeldocV2.zip. 1999.

[Eck13] Claudia Eckert. *IT-Sicherheit*. 8. Aufl. München: Oldenbourg Wissenschaftsverlag GmbH, 2013.

[El 85] Taher El Gamal. „A public key cryptosystem and a signature scheme based on discrete logarithms“. In: *Proceedings of CRYPTO 84 on Advances in Cryptology*. Santa Barbara, California, USA: Springer-Verlag New York, Inc., 1985, S. 10–18. ISBN: 0-387-15658-5. URL: http://dl.acm.org/citation.cfm?id=19478.19480.

[Ele98] Electronic Frontier Foundation. *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design*. Sebastopol, CA: O'Reilly & Associates Inc., 1998.

[Ert03] Wolfgang Ertel. *Angewandte Kryptographie*. München: Fachbuchverlag Leipzig im Carl Hanser Verlag, 2003.

[Ger+04] Helmar Gerloni, Barbara Oberhaitzinger, Helmut Reiser und Jürgen Plate. *Praxisbuch Sicherheit für Linux-Server und -Netze*. München: Hanser Verlag, 2004.

[Ges04] Alexander Geschonneck. *Computer Forensik*. Heidelberg: dpunkt.verlag, 2004.

[Gon99] Marcus Goncalves. *Firewalls: A Complete Guide*. McGraw-Hill, Okt. 1999.

[Gut98] Peter Gutmann. „Software Generation of Practically Strong Random Numbers“. In: *Proc. Usenix Security Symposium*. Eine wesentlich erweiterte Version des Artikels ist verfügbar unter `http://www.cypherpunks.to/~peter/06_random.pdf`. 1998.

[Hel80] Martin E. Hellman. „A cryptanalytic time-memory trade-off“. In: *IEEE Transactions on Information Theory* 26.4 (1980), S. 401–406.

[Hou+02] R. Housley, W. Polk, W. Ford und D. Solo. „Internet X.509 Public Key Infrastructure“. `ftp://ftp.rfc-editor.org/in-notes/rfc3280.txt`. Apr. 2002.

[KN07] Dennis Kügler und Ingo Naumann. „Sicherheitsmechanismen für kontaktlose Chips im deutschen Reisepass - Ein Überblick über Sicherheitsmerkmale, Risiken und Gegenmaßnahmen“. In: *Datenschutz und Datensicherheit* 31.3 (2007), S. 176–180.

[Knu97] Donald E. Knuth. *The Art of Computer Programming Vol. 2: Seminumerical Algorithms*. Addison-Wesley, 1997.

[LS07] P. L'Ecuyer und R. Simard. „TestU01: A C Library for Empirical Testing of Random Number Generators“. In: *ACM Transactions on Mathematical Software* 33.4, Article 22 (Aug. 2007). `http://www.iro.umontreal.ca/~simardr/testu01/tu01.html`.

[Mar] G. Marsaglia. *The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness*. `http://www.stat.fsu.edu/pub/diehard/`.

[MOV96] A. Menezes, P. van Oorschot und S. Vanstone. *Handbook of Applied Cryptography*. `http://www.cacr.math.uwaterloo.ca/hac`. CRC Press, 1996.

[MR99] Günter Müller und Kai Rannenberg, Hrsg. *Multilateral Security in Communications*. München: Addison Wesley Longman GmbH, 1999.

[Oec03] Philippe Oechslin. „Making a Faster Cryptanalytic Time-Memory Trade-Off“. In: *CRYPTO*. Hrsg. von Dan Boneh. Bd. 2729. Lecture Notes in Computer Science. Springer, 2003, S. 617–630. ISBN: 3-540-40674-3.

[Ris05] Ivan Ristic. *Apache Security — The Complete Guide to Securing Your Apache Web Server*. O'Reilly, 2005.

[Ris10] Ivan Ristic. *ModSecurity Handbook: The Complete Guide to the Popular Open Source Web Application Firewall*. FeistyDuck, 2010.

[Roß99] Stephan Roßbach. *Der Apache Webserver*. Bonn, Germany: Addison-Wesley, 1999.

[Ruk01] A. Rukhin et al. *NIST Special Publication 800-22: A Statistical Test Suite for Random And Pseudorandom Number Generators for Cryptographic Applications*. `http://csrc.nist.gov/rng/SP800-22b.pdf`. Mai 2001.

[Sch+99] Bruce Schneier, John Kelsey, David Wagner, Chris Hall, Niels Ferguson und Doug Whiting. *Twofish Encryption Algorithm : A 128-Bit Block Cipher*. John Wiley and Sons, 1999.

[Sch00] Bruce Schneier. *Secrets & Lies. IT-Sicherheit in einer vernetzten Welt*. Heidelberg, Weinheim, Germany: dpunkt.Verlag / Wiley-VCH, 2000.

[Sch96] Bruce Schneier. *Angewandte Kryptographie. Protokolle, Algorithmen und Sourcecode in C*. Bonn, Germany: Addison-Wesley, 1996.

[SGG08] Abraham Silberschatz, Peter Galvin und Greg Gagne. *Applied Operating System Concepts*. 8. Aufl. New York, NY, USA: John Wiley, Inc., 2008.

[Sha84] Adi Shamir. „Identity-Based Cryptosystems and Signature Schemes“. In: *CRYPTO*. Hrsg. von G. R. Blakley und David Chaum. Bd. 196. Lecture Notes in Computer Science. Springer, 1984, S. 47–53. ISBN: 3-540-15658-5.

[Sot+08] Alexander Sotirov, Marc Stevens, Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik und Benne de Weger. *MD5 considered harmful today*. `http://www.win.tue.nl/hashclash/rogue-ca/`. Dez. 2008.

[Spe11] Ralf Spenneberg. *Linux Firewalls – Sicherheit für Linux-Server und -Netzwerke mit IPv4 und IPv6*. 2. Aufl. München, Deutschland: Addison-Wesley, 2011.

[Sta00] William Stallings. *Network Security Essentials*. Uppser Saddle River, New Jersey: Prentice Hall, 2000.

[Sta06] William Stallings. *Cryptography and Network Security*. 4. Aufl. Upper Saddle River, New Jersey: Prentice Hall, 2006.

[Tan02] Andrew S. Tanenbaum. *Moderne Betriebssysteme*. Pearson Studium, 2002.

[Vie09] John Viega. *the myths of security*. Sebastopol, CA: O'Reilly, 2009.

[Wol07] Sebastian Wolfgarten. *Apache Webserver 2. Installation, Konfiguration, Programmierung*. Addison-Wesley, 2007.

[WWS02] Tobias Weltner, Kai Wilke und Björn Schneider. *Windows-Sicherheit, Das Praxisbuch*. Konrad-Zuse-Str. 1, D-85716 Unterschleißheim: Microsoft Press, 2002.

[WY05] Xiaoyun Wang und Hongbo Yu. „How to Break MD5 and Other Hash Functions“. In: *Advances in Cryptology - EUROCRYPT 2005*. Hrsg. von Ronald Cramer. Lecture Notes in Computer Science 3494. Berlin: Springer-Verlag, 2005, S. 19–35.

Inhaltsverzeichnis

Diese Seite bleibt aus technischen Gründen frei!

002456672
(04/17)

01866-5-04-S 1