Prof. Dr. Stefan Wohlfeil

Lehrveranstaltung

Sicherheit im Internet I

LESEPROBE

Fakultät für
Mathematik und Informatik

FernUniversität in Hagen

Inhaltsverzeichnis

Kapitel 1

Sicherheit in der Informationstechnik

Die Autoren: Dr. Stefan Wohlfeil, geb. 12.12.1964

- Studium der Informatik mit Nebenfach Elektrotechnik an der Universität Kaiserslautern (1984–1991)
- Wissenschaftlicher Mitarbeiter am Lehrgebiet Praktische Informatik VI der FernUniversität in Hagen (1991–1998)
- Promotion zum Dr. rer. nat. (1997)
- Mitarbeiter in der Deutsche Bank AG, Abteilung TEC — The Advanced Technology Group (1998–2002)
- Professor an der Hochschule Hannover, Fakultät IV, Abteilung Informatik; Arbeitsgebiet: Sichere Informationssysteme (seit 2002)

Einige Abschnitte in Kurseinheit 2 hat Prof. Dr. Jörg Keller (Lehrgebiet Parallelität und VLSI der FernUniversität in Hagen) geschrieben.

1.1 Einführung

Liebe Fernstudentin, lieber Fernstudent,
herzlich willkommen beim Kurs über Sicherheit im Internet!

Diese Einführung soll Ihnen einen Überblick darüber geben, worum es im vorliegenden Kurs geht. Dieser Kurs liefert eine Einführung in das Gebiet der *Sicherheit*. Dabei wird es um einzelne Computer, vernetzte Computer und das *Internet* gehen. Sie erfahren, welche Sicherheitsprobleme dort existieren und welche Möglichkeiten Sie haben, sich diesen Problemen entgegenzustellen.

Inhalt des Kurses und Vorkenntnisse: Dieser Kurs richtet sich an Informatik-Studierende[1] und setzt die Kenntnis einiger Inhalte aus einem Informatik-Grundstudium voraus. Konkret sollten Sie bereits wissen, wie ein Computer prinzipiell aufgebaut ist, was ein Betriebssystem typischerweise macht und welche Möglichkeiten sich durch die Vernetzung, wie beispielsweise

[1] Hierzu gehören alle Studierenden, deren Curriculum einen Informatikbestandteil enthält wie beispielsweise auch Studierende der Wirtschaftsinformatik.

im Internet, für Anwender bieten. Diese Themen werden im Kurs *(01801) Betriebssysteme und Rechnernetze* behandelt.

Die Kurseinheit 1 beschäftigt sich mit den Grundlagen des Themas *Sicherheit*. Die folgenden Fragen werden diskutiert:

- *Warum* ist das Thema Sicherheit überhaupt bedeutsam?
- Was *bedeutet* der Begriff „Sicherheit“ im Zusammenhang mit Computern eigentlich?
- Welche Probleme sind zu lösen?

In der Literatur werden eine Reihe von „klassischen Bedrohungen“ vorgestellt, auf die im Kurs auch eingegangen wird. Weiterhin wird der Aufbau und die Funktionsweise des Internets kurz vorgestellt. Konkret werden einige wichtige Protokolle und Dienste des Internets besprochen. Den Abschluss der ersten Kurseinheit bildet eine Beschreibung von ausgewählten konkreten Bedrohungen der Computersicherheit, wie beispielsweise Viren.

Die Kurseinheit 2 beschäftigt sich mit den Grundfunktionen, die die Hardware von Systemen und die Betriebssysteme selbst im Bereich Sicherheit anbieten. Außerdem werden dort die Grundlagen von Verschlüsselungsverfahren erklärt. Sie sind für die Lösung des *Vertraulichkeits-* und des *Integritätsproblems* ein wichtiges Hilfsmittel. Es wird weiterhin auf Authentifizierungsverfahren eingegangen. Dazu gehört auch das Konzept der *digitalen Signaturen*.

In Kurseinheit 3 wird das Thema Sicherheit aus der Perspektive eines World-Wide-Web-Benutzers beleuchtet. Die Fragen, wie versende/empfange ich sicher eine E-Mail, bzw. wie surfe ich sicher im Netz, werden beantwortet. Oft muss man auch an Rechnern arbeiten, die weit entfernt vom eigenen Schreibtisch stehen. Der Studentenrechner `baobab` (früher `bonsai`) der FernUniversität ist ein Beispiel hierfür. Wie man das sicher tun kann, wird auch in Kurseinheit 3 erklärt. Abschließend bekommen Sie einige Hinweise, wie Sie Ihren privaten PC zu Hause sicherer machen können.

Die Kurseinheit 4 beschäftigt sich mit dem Thema Sicherheit aus der Perspektive eines Systemadministrators und eventuell auch Web-Anbieters. Hier werden Verfahren und Systeme vorgestellt, mit denen ein internes Netz (auch Intranet genannt) so an das Internet angeschlossen wird, dass keine unbefugten Zugriffe und Modifikationen möglich sind. Neben dem Konzept der *Firewall* wird auch auf organisatorische und prozedurale Aspekte eingegangen.

Ergänzende Materialien: Das Thema Sicherheit im Internet ist derart umfangreich, dass es in diesem Kurs nur in Ausschnitten behandelt werden kann. Ziel des Kurses ist es, dass Sie die Grundlagen des Themengebietes kennenlernen und Sie sich dann darauf aufbauend tiefer in die Materie einarbeiten können. Dazu gibt es verschiedene weitere Informationsquellen.

Bücher

Die Menge an Büchern zum Thema Security wächst sehr schnell. Ausgehend vom Literaturverzeichnis dieses Kurses sollten Sie in der Universitätsbibliothek das eine oder andere Buch ausleihen und durchschauen. Aktuellste Bücher kann man bei den verschiedenen Buchhändlern im Internet suchen. Dort findet man u. U. auch Rezensionen der Bücher vor.

Internet

Überhaupt ist das Internet eine nahezu unerschöpfliche Quelle an Informationen zum Thema Security. Im Kurs werden eine Reihe von Verweisen auf

interessante Seiten im Internet genannt. Wenn Sie Zugang zum Internet haben, nehmen Sie sich doch die Zeit und schauen sich die eine oder andere Seite an. Ich hoffe, dass die Verweise noch stimmen, wenn Sie den Kurs lesen. Das Internet ändert sich ständig, so dass es gut sein kann, dass Sie einmal eine Seite nicht finden. In diesem Fall sollten Sie eine der vielen Suchmaschinen wie z. B. *bing* oder *Google* konsultieren. Vielleicht hat sich ja nur die Adresse der Seite leicht verändert. Informieren Sie dann auch bitte die Kursbetreuer, damit der Kurstext aktualisiert werden kann. Die Namen und Kontaktmöglichkeiten der Kursbetreuer wurden Ihnen im Anschreiben zusammen mit dieser Kurseinheit genannt.

Fortsetzungskurse im nächsten Semester

An der FernUniversität in Hagen ergänzen der Kurs *(01868) Sicherheit im Internet 1 – Ergänzungen* und der Kurs *(01867) Sicherheit im Internet 2* diesen Kurs. In Kurs *(01868) Sicherheit im Internet 1 – Ergänzungen* werden unter anderem diese Themen besprochen:

- Computer-Forensik
- Anonymität
- Biometrie
- Zugriffskontrollen, Benutzerauthentisierung
- Sicherheit in Telekommunikationsnetzen (WLAN, Voice over IP)
- Aktive Inhalte (ActiveX, Java, JavaScript)

In Kurs *(01867) Sicherheit im Internet 2* werden dann diese Themen behandelt:

- Konkrete Bedrohungen und Angriffe gegen Rechner
- Bezahlverfahren im Internet
- Überblick über wichtige Gesetze und Verordnungen, die im Internet von besonderer Bedeutung sind
- Virtual Private Networks (VPN)
- Intrusion Detection Systems (IDS)
- Entwurf und Implementierung sicherer Systeme

Informationen zu den Prüfungsmöglichkeiten dieses Kurses finden Sie in den Studien- und Prüfungsinformationen Ihres Studienganges an der FernUniversität oder bei der Studienberatung. Dort erfahren Sie, in welchen Studiengängen dieser Kurs eingesetzt wird, in welchen Modulen dieser Kurs ein Bestandteil ist, in welcher Form Sie Leistungsnachweise oder Prüfungen ablegen können bzw. müssen und so weiter.

1.2 Einleitung

1.2.1 Warum ist Sicherheit erforderlich?

Das Thema „Sicherheit in der Informationstechnik“ hat in den letzten Jahren mehr und mehr an Bedeutung gewonnen. Einer der Hauptgründe dafür ist die große Popularität des Internets. Für viele Menschen ist das Internet nicht nur das Informationsmedium, als das es ursprünglich entwickelt wurde, sondern immer öfter auch das Medium für private Geschäftstätigkeiten aller Art. Bücher, Flug- und Eisenbahntickets, Schuhe, Mode, Musik, usw. können nicht nur im Geschäft gekauft werden, sondern auch bequem von zu Hause aus. Über virtuelle Auktionshäuser wie beispielsweise *eBay* werden inzwischen täglich enorme Mengen von Gütern aller Art versteigert. Neben Privatleuten beteiligen sich auch Gewerbetreibende als Bieter, ebenso wie als Anbieter. Neben den materiellen Gütern können natürlich digitale Güter, also alles was sich digitalisieren lässt, sehr einfach über das Internet vertreiben lassen. Müssen klassische Bücher noch von einem Boten ausgeliefert werden, so können digitale Bücher (auch **eBooks** genannt) direkt über das Internet ausgeliefert werden. Das gilt ebenso für Musik, Filme oder Software.

private Geschäftstätigkeiten

eBooks

Auch Bankgeschäfte wie Überweisungen, Einrichten/Ändern von Daueraufträgen oder sogar An- und Verkäufe von Wertpapieren lassen sich über das Internet abwickeln. Die Vorteile für die Konsumenten sind vielfältig:

- Man ist nicht mehr an die Ladenöffnungszeiten gebunden, sondern kann rund um die Uhr tätig sein.
- Man muss nicht mehr persönlich im Geschäft vorbei schauen, sondern kann seine Geschäfte bequem von zu Hause erledigen.
- Verschiedene Angebote lassen sich einfacher vergleichen. Die Konkurrenz ist immer nur „einen Mausklick“[2] entfernt.
- Bei digitalen Gütern erfolgt die Lieferung sofort über das Internet und es fallen keine Wartezeiten an.

Aber nicht nur Privatleute, sondern auch viele Firmen benutzen das Internet für ihre Geschäftszwecke. War man zuerst nur durch eine „website“ präsent und hat sich und seine Produkte vorgestellt, so nutzt man das Internet bzw. die Internettechnologie heute auch für die Abwicklung von Geschäften. In der Automobilindustrie sind die Hersteller und ihre Zulieferer über das Internet miteinander verbunden und tauschen so beispielsweise Bestellungen aus. Der Vorteil für die Unternehmen besteht darin, dass viele solcher Tätigkeiten automatisierbar sind. Dadurch lassen sich Kosten einsparen. Den Umfang der wirtschaftlichen Bedeutung, die das Internet heute (Juli 2013) erreicht hat, verdeutlichen die folgenden Zahlen:

wirtschaftliche Bedeutung

- Alleine in Deutschland wurden 2006 ca. 35,4 Millionen Bankkonten online geführt.[3] Das sind etwa 80% der Girokonten bei privaten Banken. Etwa jeder zweite Internetbenutzer führt sein Konto online. Bis 2011 stieg

[2] Tatsächlich ist es nicht ganz so einfach. Die wenigsten Anbieter werden einen Verweis (engl. **link**) auf ihre Konkurrenz mit anbieten. Man muss diese Adressen also erst einmal finden. Suchmaschinen, wie *Google*, oder Verzeichnisdienste, wie *Yahoo!*, helfen bei der Suche.

[3] Quelle: Bundesverband Deutscher Banken (`http://www.bdb.de/`)

die Zahl der online geführten Girokonten auf 48 Millionen (von ca. 95 Millionen Konten insgesamt).

- Im Jahr 2006 wurden in Deutschland etwa 1,8 Milliarden Online-Überweisungen getätigt. Dabei wurden etwa 1 685 Milliarden Euro bewegt[4]. Im Jahr 2011 wurden bereits etwa 2,3 Milliarden Online-Überweisungen getätigt und dabei 11 849 Milliarden Euro bewegt.

- Das U.S. Census Bureau veröffentlicht Statistiken aus den USA. Im dritten Quartal des Jahres 2008 wurden etwa 34,4 Milliarden US-Dollar Umsatz im E-Commerce gemacht. Abbildung 1.1 zeigt die stetig steigenden Umsatzzahlen sowie die Spitzen zu Weihnachten. Dabei zählt ein Handel als E-Commerce, wenn entweder die Bestellung oder die Preisvereinbarung über das Internet oder ein anderes elektronisches Netz (z. B. Extranet oder EDI) erfolgt.

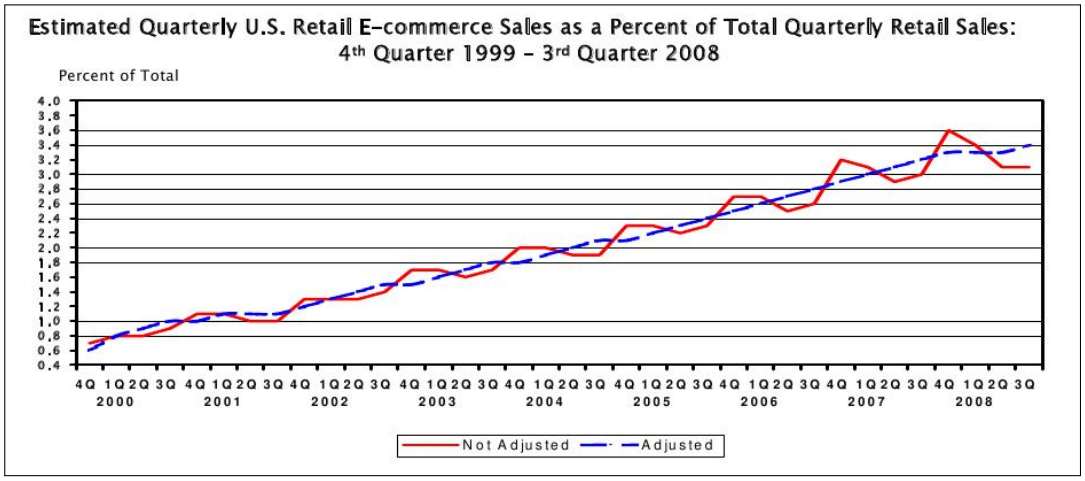

Abbildung 1.1: E-Commerce-Anteil an den Gesamtumsätzen in den USA

Im Jahr 2011 ist der Anteil des E-Commerce an den Umsätzen im Bereich „Produktion/Herstellung“ (engl. **manufacturing**) auf fast 50 Prozent gestiegen. Das heißt, dass in diesem Bereich fast jeder zweite Dollar durch E-Commerce umgesetzt wird und viele Firmen bereits sehr stark vom E-Commerce abhängen. Im Großhandelsbereich (engl. **wholesale**) liegt die Quote bei knapp 25 Prozent, im Einzelhandelsbereich (engl. **retail**) liegt die Quote bei knapp 5 Prozent.

- Der Buchhändler *Amazon* hat nach eigenen Angaben am 29. November 2010 weltweit mehr als 13,7 Millionen Bestellungen angenommen. Das entspricht etwa 158 Bestellungen pro Sekunde. Im Jahr 2007 hat *Amazon* knapp 14,9 Milliarden US-Dollar Umsatz (engl. **net sale**) gemacht. Im Jahr 2008 stieg der Umsatz trotz der Finanz- und Wirtschaftskrise auf ca. 19,2 Milliarden US-Dollar, um sich dann bis 2012 auf ca. 61 Milliarden US-Dollar zu verdreifachen.

Man sieht, dass das Internet eine große wirtschaftliche Bedeutung gewonnen hat und dass die wirtschaftliche Bedeutung kontinuierlich wächst. Ohne die

[4]Quelle: Deutsche Bundesbank (`http://www.bundesbank.de/`): *Statistiken zum Zahlungsverkehr 2002–2006 (Stand Januar 2008)* und *Statistiken zum Zahlungsverkehr 2007–2011 (Stand Januar 2013)*

entsprechende Sicherheit der beteiligten Informations-Technologie-Systeme (IT-Systeme) könnten also immense wirtschaftliche Probleme entstehen.

Aktuelle Sicherheitsprobleme: Sicherheit ist aber auch deshalb ein wichtiges Thema, weil heute schon viele Sicherheitsprobleme auftreten. Sie stören die normale Computernutzung, richten nicht unerhebliche Schäden an und finden daher auch mehr und mehr Beachtung in der Presse. Ein **Computer-Emergency-Response-Team (CERT)** ist eine Anlaufstelle, die Meldungen über Sicherheitsvorfälle entgegen nimmt und über mögliche Abwehrmaßnahmen informiert. Im Internet findet man unter der Adresse `http://www.cert.org/` eine Statistik der gemeldeten Vorfälle.

Computer-Emergency-Response-Team (CERT)

In Deutschland betreibt das Deutsche Forschungsnetz (DFN) auch solch ein Team. Zu seinen Aufgaben gehören neben der Betreuung der DFN-Mitglieder auch die Bereitstellung von Informationen über Sicherheitsvorfälle und Hilfsmitteln zur Bekämpfung von Sicherheitsproblemen. Außerdem betreibt es eine Zertifizierungsstelle (siehe auch Abschnitt 2.7). Im Internet finden Sie es unter der Adresse `http://www.cert.dfn.de/` Eine Liste von weiteren deutschsprachigen CERTs findet man unter der Adresse `http://www.cert-verbund.de`.

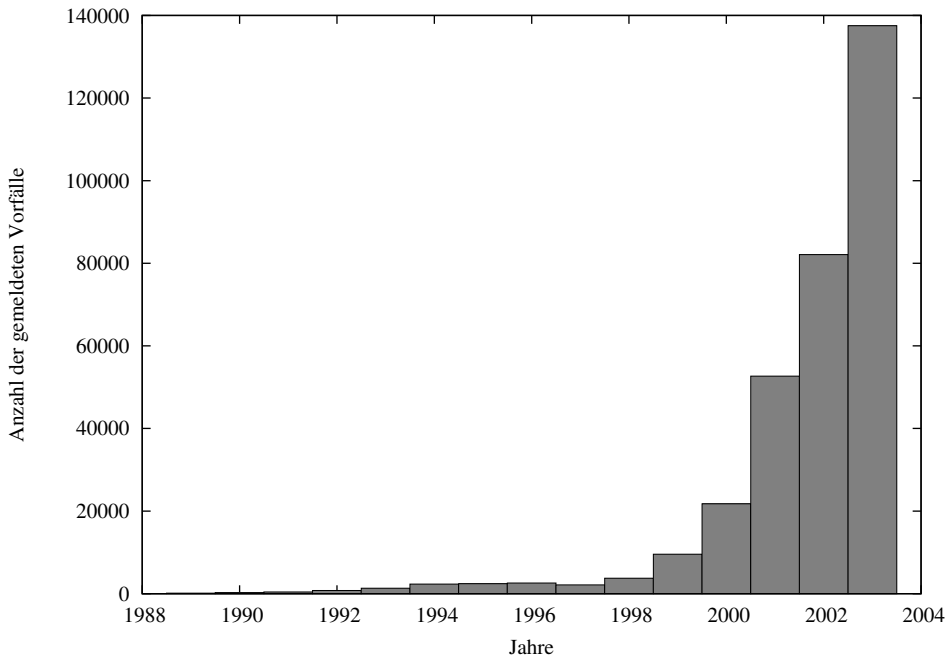

Abbildung 1.2: Dem CERT gemeldete Sicherheitsvorfälle

Abbildung 1.2 zeigt grafisch die Zahlen, die das CERT als Tabelle im Internet publiziert. Die steigende Zahl der Vorfälle kann nicht alleine auf ein gestiegenes Bewusstsein bzgl. Computersicherheit zurück geführt werden. Wenn mehr Personen auf Sicherheitsvorfälle achten, dann werden natürlich auch mehr Vorfälle erkannt. Eine leichte Steigung der gemeldeten Vorfälle wäre also auch bei einer eigentlich konstanten Zahl an Vorfällen zu erwarten. Dadurch alleine lässt sich der Anstieg der Zahlen in Abbildung 1.2 allerdings nicht erklären. Es ist tatsächlich eine steigende Zahl an Sicherheitsvorfällen vorhanden.

Hierfür gibt es zwei wesentliche Gründe: (1) Ein Grund ist die Verfügbarkeit von einfach zu bedienenden Angriffswerkzeugen, beispielsweise Virus-Konstruktions-Programmen oder konkreten „hacker tools“. Damit kann auch

ein weniger qualifizierter Angreifer Schwachstellen in Systemen ausnutzen. In diesem Fall spricht man oft von sog. **Script-Kiddies**, die verfügbare Programme zur Ausnutzung von Schwachstellen laufen lassen und beliebige Rechner angreifen. (2) Der zweite Grund ist wirtschaftlicher Natur. Der massenhafte Versand von unerwünschten Werbe-E-Mails (engl. **spam**) ist für die Versender ein potentiell lukratives Geschäft. Vergleichsweise niedrigen Kosten steht ein großer Nutzen gegenüber. Allerdings versuchen Anwender permanent, die steigende Flut von Werbe-E-Mails einzudämmen. Beispielsweise werden E-Mails von bekannten „Spammern“ grundsätzlich blockiert. Dies kann aber dadurch umgangen werden, indem die „Spammer“ nicht unter eigenem Namen agieren, sondern erfolgreich angegriffene Rechner und deren Benutzer hierzu missbrauchen. Die unerwünschten Werbe-E-Mails kommen dann nicht mehr von einem Rechner (den man als Absender evtl. nicht zulassen kann), sondern von beliebigen, vorher nicht bekannten Rechnern von Privatnutzern. Hacker vermieten solche **Bot-Netze** an zahlende Kunden um deren Werbung über die gehackten Rechner zu verbreiten.

Script-Kiddies

Bot-Netze

Das *SANS Institute* (siehe `http://www.sans.org/`) hat einige Jahre lang eine Liste mit den *Top 20 Security Risks* erstellt. Sie gruppierte die Liste der Probleme des Jahres 2007 in

- Client-side Vulnerabilities (Webbrowser, Office-Software, E-Mail-Clients, Mediaplayer etc.)
- Server-side Vulnerabilities (Webapplications, Windows-Services, UNIX- and Mac OS-Services, Backupsoftware, Antivirensoftware, Management-Servers, Database-Software etc.)
- Security-Policy and Personnel (Excessive User-Rights and Unauthorized Devices, Phishing, Unencrypted Laptops and Removable Media etc.)
- Application-Abuse (Instant Messaging, Peer-to-Peer-Programs etc.)
- Network-Devices (VoIP-Servers and Phones etc.)
- Zero-Day-Attacks

Man sieht, dass es eine Vielzahl von Schwachstellen und Problemen gibt. Da ständig neue Probleme auftauchen und glücklicherweise einige alte Probleme gelöst scheinen, ändert sich auch die Klassifikation der Liste mit der Zeit.

Seit 2009 werden andere Listen erstellt. Aktuell sind zwei Listen:

1. *Top 25 Most Dangerous Software Errors.* In dieser Liste werden die gefährlichsten Softwarefehler vorgestellt. Sie sind der Grund für entsprechende Sicherheitsprobleme. Software-Entwickler sollten diese Fehler kennen und in ihren Programmen unbedingt vermeiden. In der Liste des Jahres 2011 gehören die Fehler zu drei Kategorien: (1) Insecure Interaction between Components; hier geht es um unsichere Datenübertragung zwischen Komponenten eines Programms, zwischen Programmen, zwischen Prozessen, threads oder zwischen Systemen. (2) Risky Resource Management; hier geht es um schlechte Erstellung, Übertragung oder Löschung von wichtigen Ressourcen eines Systems und (3) Porous Defenses; also falsch realisierten oder schlicht vergessenen Sicherheitsmaßnahmen.

2. *Top 20 Critical Security Controls.* Hier werden die wichtigsten Sicherheitsmaßnahmen aufgezählt, mit denen man die eigene IT schützen sollte. Bei allen Maßnahmen wird erklärt, wie Angreifer das Fehlen dieser Maßnahme ausnützen können und wie man die Maßnahme effektiv umsetzt.

Conficker

Ein Beispiel für ein großes Sicherheitsproblem aus dem Jahr 2008 ist der Wurm *Conficker* alias *Downadup*. Es zeigt wie geschickt Angreifer vorhandene Sicherheitsprobleme ausnützen können. Der Wurm nutzt einige Software-Fehler in allen Betriebssystemen von Microsoft Windows 2000 bis Microsoft Windows Server 2008 aus. Der Remote-Procedure-Call-Dienst (RPC-Dienst) enthält einen Fehler, so dass ein Angreifer eine spezielle Nachricht an den Rechner schicken kann, um dann die komplette Kontrolle über diesen Rechner auszuüben. Der Angreifer braucht keine Benutzerkennung oder irgendein Passwort auf dem System zu kennen oder auszuspähen. Bereits im Jahr 2003 hatte der Wurm *MSBlaster* alias *Lovsan* eine vergleichbare Lücke in Microsoft Windows ausgenutzt und großen Schaden angerichtet. Der RPC-Dienst wird bei der Datei- und Druckfreigabe in Microsoft Windows benutzt.

Sich persistent machen

Als erstes kopiert sich der Wurm als DLL-Datei in das Betriebssystem des infizierten Rechners, trägt sich in die Registry ein und sorgt somit dafür, dass er beim Systemstart immer wieder neu gestartet wird.

In ihrem Malware-Protection-Center beschrieb *Microsoft* die weiteren Auswirkungen von *Conficker* wie folgt:

> This malware mostly spreads within corporations but also was reported by several hundred home users. It opens a random port between port 1024 and 10000 and acts like a web server. It propagates to random computers on the network by exploiting MS08-067. Once the remote computer is exploited, that computer will download a copy of the worm via HTTP using the random port opened by the worm. The worm often uses a .JPG extension when copied over and then it is saved to the local system folder as a random named dll.
>
> It is also interesting to note that the worm patches the vulnerable API in memory so the machine will not be vulnerable anymore. It is not that the malware authors care so much about the computer as they want to make sure that other malware will not take it over too...

Als Verbreitungsserver arbeiten

Andere aussperren

Jeder infizierte Rechner wird somit zum neuen Verteiler für den Wurm im Gegensatz zu anderer Schadsoftware, die ein neu infizierter Rechner immer von demselben Server laden würde.

Anschließend ruft der Wurm HTML-Seiten von einigen Webservern ab, um daraus das aktuelle Datum und die Uhrzeit zu entnehmen. Außerdem erfährt er so, welche im Internet sichtbare IP-Adresse der infizierte Rechner hat. (Der Rechner könnte in einem privaten Netz mit privaten IP-Adressen liegen und über einen Router, der Network-Address-Translation (NAT) durchführt, mit dem Internet verbunden sein.) Eine URL zum o. g. Webserver auf dem infizierten Rechner verteilt der Wurm dann an andere Rechner, die er zu infizieren versucht.

verbreiten

Letztlich versucht der Wurm sich (1) im lokalen Netz auf angeschlossene USB-Devices etc. zu verbreiten und (2) schickt der Wurm dann auch noch Anfragen an einige URLs, um von dort weitere (Schad-)Software nachzuladen. Diese URLs werden zufällig generiert, wobei das aktuelle Datum und die

Uhrzeit in die Erzeugung eingehen. Es werden also täglich andere URLs erzeugt, damit man diese Rechner nicht einfach sperren kann. Ein Hacker der den Erzeugungsalgorithmus kennt kann nun beispielsweise die URLs für einen Tag in etwa drei Monaten berechnen und dann eine dieser zufällig generierten DNS-Domains für sich registrieren, dort Schadsoftware aufspielen und abwarten bis infizierte Rechner versuchen, diese Schadsoftware zu laden und zu starten. Die Firma *F-Secure* hat einige solche DNS-Domains für sich registriert und dann gezählt wie viele infizierte Computer dorthin Anfragen schicken. Am 16. Januar 2009 schätzte man bei *F-Secure* die Zahl der mit dem *Conficker*-Wurm infizierten Rechner auf ca. 8,9 Millionen!

Das Nachladen von Funktionen haben die Autoren von *Conficker* noch zusätzlich geschützt. Die nachzuladenden (Schad-)Funktionen müssen mit einem speziellen RSA-Schlüssel (siehe Abschnitt 2.4.2) digital signiert sein. Die Autoren von Schutzsoftware können diesen Nachlademechanismus also nicht dazu benutzen dem Wurm eine „Selbstzerstörungsfunktion“ unter zu schieben.

Schutz

Der einzige Schutz vor dem Wurm besteht darin, den Betriebssystem-Patch von *Microsoft* rechtzeitig zu installieren.

PRISM: Im Sommer 2013 wurde bekannt, dass die *National Security Agency (NSA)* der USA ein weltweites Abhörnetz betreibt und so viele Kommunikationsdaten wie möglich mitliest. So wird u. a. erfasst wer mit wem per E-Mail kommuniziert und wer wann welche Internetseiten aufruft. Außerdem sind amerikanische Firmen wie *Google*, *Microsoft*, *Apple* und *facebook* verpflichtet, der NSA Daten auf Anfrage zur Verfügung zu stellen. Welche Daten das genau sind ist nicht öffentlich bekannt. Man muss also davon ausgehen, dass die eigene Privatsphäre doch nicht so privat ist wie man das gerne hätte. Möchte man vertrauliche Inhalte schützen, so muss man sie verschlüsseln (siehe Abschnitt 2.3). Möchte man nicht, dass jemand weiss, welche Internetseite man besucht, so muss man Anonymitätsnetze wie JAP oder Tor (siehe Kurs *(01868) Sicherheit im Internet 1 – Ergänzungen*) benutzen. Außerdem sollte man sich sehr genau überlegen, welche persönlichen Daten man in sozialen Netzen preisgeben will.

Gesetzliche Verpflichtungen: Am 1. Mai 1998 trat das Gesetz zur Kontrolle und Transparenz im Unternehmensbereich (KonTraG) in Kraft. Es zwingt börsennotierte Aktiengesellschaften zur Risikovorsorge durch die Etablierung eines Risikomanagement- und Kontrollsystems. Das heißt, dass diese Firmen gezwungen sind, sich mit den Risiken, die durch ihre IT-Systeme entstehen können, auseinander zu setzen. Natürlich reicht es nicht, sich nur mit den Risiken auseinander zu setzen. Die Unternehmen müssen die erkannten Gefahrenpotentiale auch verringern. Da 100-prozentige Sicherheit nicht erreichbar ist, müssen die Restrisiken auf ein vertretbares Maß reduziert werden. Verbleibende Gefahren sollen durch Überwachungsmaßnahmen kalkulierbar werden.

KonTraG

Im Geschäftsleben hat die Risikovorsorge aber auch handfeste finanzielle Gründe. Banken müssen für jeden Kredit, den sie vergeben, einen bestimmten Anteil an Eigenkapital vorhalten. Das dient dazu, dass beim Zahlungsausfall eines Gläubigers die Bank nicht in Liquiditätsprobleme kommt. Die Höhe dieses Eigenkapitalanteils richtet sich nach der Höhe des Risikos. Kreditnehmer mit hohem Risiko zwingen eine Bank zu höheren Eigenkapitalrücklagen. Daher werden Banken solchen Kreditnehmern ungünstigere Zinskonditionen anbieten. Das *Basel Committee on Banking Supervision* hat solche Vorschriften unter

Basel II
Basel III

dem Namen **Basel II** erarbeitet. Insbesondere die sog. operationellen Risiken werden zukünftig beurteilt. Ein Kreditnehmer, dessen Geschäfte stark von der IT abhängen und der wenig IT-Sicherheitsvorkehrungen trifft, muss also mit höheren Kreditkosten rechnen. Im Jahr 2010 wurde unter dem Namen **Basel III** ein weiteres Reformpaket veröffentlicht, das ab 2013 schrittweise in Kraft treten wird. Als Folge der weltweiten Finanzkrise von 2007 wurden u. a. weitere Vorschriften zum Risikomanagement erlassen.

Ein weiteres Problem stellen Computer im Börsenhandel dar. Sie können so programmiert werden, dass sie automatisch bei bestimmten Ereignissen Kauf- oder Verkaufs-Aufträge absetzen. Das kann zu Spiraleffekten führen. Sinkt der Kurs einer Aktie unter einen bestimmten Wert, dann will der erste Computer verkaufen. Dadurch sinkt der Wert weiter und weitere Computer wollen verkaufen usw. Dadurch können sehr starke Schwankungen entstehen. Im sogenannten Hochfrequenzhandel nutzen Computer Informationsvorsprünge im Bereich von *Millisekunden* aus, d. h. für einen Menschen bleibt eigentlich kaum Zeit, um gegebenenfalls einzugreifen.

Aber auch Sie persönlich müssen aufpassen, dass Angreifer Ihren Rechner nicht dazu missbrauchen, anderen einen Schaden zuzufügen. Möglicherweise tragen Sie daran eine Mitschuld, wenn Sie die erforderliche Sorgfalt missachten. Dann könnten Sie auf Schadenersatz verklagt werden.

Persönliche Arbeitserleichterung: Sobald Sie auf dem eigenen Rechner ein Sicherheitsproblem haben, müssen Sie sich darum kümmern. Falls Sie nichts tun, können Sie Probleme der vielfältigsten Art bekommen. Ihr Rechner arbeitet nicht mehr für Sie, er schädigt andere Benutzer oder Ähnliches. Um diese Probleme zu vermeiden, müssen Sie also etwas tun. Das kostet mindestens Zeit, die Sie möglicherweise mit Erfreulicherem verbringen wollen. Daher ist es einfacher, Probleme von Anfang an zu vermeiden, als sich später mit der Problembehebung auseinander zu setzen. In der Medizin heißt das: „Vorbeugen ist besser als heilen“

1.2.2 Was heißt eigentlich Sicherheit?

Umgangssprachlich versteht man unter Sicherheit in der Regel einen Zustand ohne Gefahren. Im dtv-Lexikon ist Sicherheit wie folgt definiert:

> **Sicherheit, 1)** Zivilrecht: Bürgschaft, Pfand oder jeder Vermögenswert, der zur Sicherheitsleistung gebracht wird. **2)** objektiv das Nichtvorhandensein von Gefahr, subjektiv die Gewissheit, vor möglichen Gefahren geschützt zu sein.

Für diesen Kurs ist die juristische Bedeutung uninteressant. Die zweite Interpretation unterscheidet zwischen *objektiver Sicherheit* und *subjektiver Sicherheit*. In beiden Fällen ist von Gefahren die Rede. Im Rahmen von IT-Sicherheit spricht man an Stelle von Gefahren häufiger von **Bedrohungen**. Neben den eher abstrakten Bedrohungen sind auch die potentiellen **Schäden** zu betrachten. Diese lassen sich in der Regel einfacher quantifizieren. In Abschnitt 1.2.3 wird kurz vorgestellt, *was* eigentlich bedroht ist. Es geht konkret um die Ziele von **Angriffen** (engl. **attacks**). In Abschnitt 1.2.4 werden einige Kriterien vorgestellt, anhand derer man Bedrohungen klassifizieren kann. Anschließend geht es in Abschnitt 1.2.5 um die „klassischen Bedrohungen“ der IT-Sicherheit. Hier

Bedrohungen
Schäden
Angriffe

werden die typischen Probleme, wie Integrität, Vertraulichkeit, Authentizität usw. diskutiert.

1.2.3 Angriffsziele

Die Sicherheit von IT-Systemen kann auf verschiedene Weise gefährdet sein. Abbildung 1.3 zeigt, an welchen Stellen die Sicherheit gefährdet werden kann. Der erste Angriffspunkt liegt an den Zugangswegen zu einem Computer. In

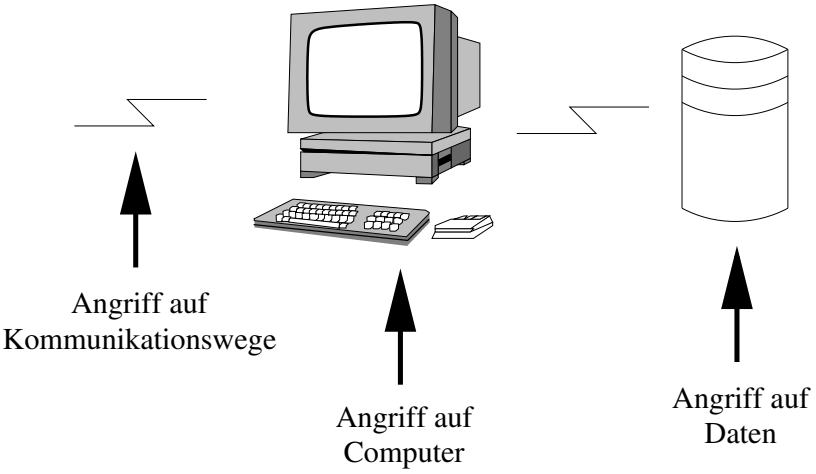

Abbildung 1.3: Ziele von potentiellen Angriffen auf die Sicherheit

einem vernetzten System wie dem Internet sind nicht alle Kommunikationswege unter der Kontrolle einer einheitlichen Instanz. Deshalb ist damit zu rechnen, dass einzelne Übertragungsleitungen abgehört werden. So kann jemand Ihre Kommunikation „mitlesen“.

Der zweite Angriffspunkt ist der Computer selbst. Bei Mehr-Benutzer-Computern können andere Benutzer Programme laufen lassen, die evtl. Ihre eigenen Programme beeinträchtigen oder Ihre eigenen Daten verfälschen oder ausspionieren. Außerdem könnte der Computer lahmgelegt werden und dadurch nicht mehr für Sie zur Verfügung stehen.

Das dritte Ziel von Angreifern sind die Daten selbst. Durch unbefugten Zugriff auf Datenbanken oder die Datenträger (Festplatten, Bänder, DVDs, USB-Sticks) können Daten ausspioniert oder manipuliert werden. Aber auch durch die Messung von elektromagnetischen Wellen, die von einem Bildschirm immer[5] abgestrahlt werden, kann man den Bildschirminhalt (und damit die Daten auf dem Bildschirm) rekonstruieren.

1.2.4 Systematik der Bedrohungen

nicht-technische und technische Bedrohungen

Bei den Bedrohungen kann man unterschiedliche Klassen differenzieren. Ein erstes Unterscheidungskriterium ist, ob es sich um *technische Bedrohungen* handelt oder ob eine *nicht-technische Bedrohung* vorliegt. Zu den einfachen technischen Bedrohungen gehört z. B. die kosmische Strahlung. Auch wenn es eher selten vorkommt, so kann diese Strahlung den Wert einzelner Bits verändern und so Daten verfälschen. Auch andere elektrische Probleme, z. B.

[5] Zumindest bei Bildschirmen auf Basis von Elektronenstrahlröhren.

auf Übertragungsleitungen, oder elektromagnetische Probleme bei Funkübertragungen gehören hierzu. Diesen Problemen kann man durch die Einführung von Redundanz, beispielsweise durch Prüfbits oder fehlerkorrigierende Codes, begegnen. Man begegnet technischen Bedrohungen also i. d. R. mit technischen Maßnahmen.

Ein nicht-technisches Problem wäre dagegen eine Person, die absichtlich Daten verfälscht und dann überträgt. Natürlich kann man hier auch argumentieren, dass eine Person die Daten ja nicht „von Hand“ ändern kann, sondern dazu einen Computer und ein Programm (also Technik) benutzen muss. Soll die Person diese Daten aber tatsächlich mit dem Computer verarbeiten (natürlich unverfälscht), dann arbeitet die Technik in diesem Fall tatsächlich so wie sie soll. Um das Problem der Datenfälschung zu vermeiden würden technische Maßnahmen weniger helfen als organisatorische Maßnahmen wie beispielsweise das Vier-Augen-Prinzip. Also wäre diese Bedrohung eher nicht-technisch.

beabsichtigte und unbeabsichtigte Bedrohungen

Das o. g. Problem führt zum zweiten Unterscheidungsmerkmal. Bedrohungen können *beabsichtigt* oder *unbeabsichtigt* entstehen. Eine unbeabsichtigte Bedrohung ist es, wenn ein Mitarbeiter aus Unwissenheit ein wichtiges Passwort auf einen Post-it-Zettel schreibt und diesen auf den Monitor des Computers klebt. Im Allgemeinen zählen die meisten Bedienungsfehler zu dieser Kategorie.

Im Gegensatz dazu ist jede Form von Spionage zu den beabsichtigen Bedrohungen zu zählen. Auch die sogenannten *Denial-of-Service-Attacks* sind beabsichtigte Bedrohungen. Hier versuchen Angreifer einen Dienst für die legitimen Benutzer nicht mehr verfügbar zu machen, beispielsweise durch Systemüberlastungen.

aktive und passive Bedrohungen

Ein drittes Merkmal unterscheidet *aktive* und *passive* Bedrohungen. Zu den passiven Bedrohungen zählt jede Form des Abhörens. Funkübertragungen lassen sich durch eine zusätzliche Antenne einfach mitschneiden und beeinträchtigen den regulären Empfang nicht. Selbst wenn der Angreifer aktiv seine Antenne in die Luft halten muss und das Abhörprogramm starten muss, so greift er doch nicht in die Übertragung ein. Auch das heute sehr oft benutzte *Ethernet* (siehe auch Abschnitt 1.3.1) überträgt in seiner Grundform jedes Datenpaket an alle angeschlossenen Computer. Normalerweise ignoriert ein Computer alle Datenpakete, die nicht an ihn selbst adressiert sind. Es ist jedoch sehr einfach, einen Computer so zu programmieren, dass er alle Pakete an einen bestimmten anderen Computer im Netz mitprotokolliert. Passive Bedrohungen sind schwer zu entdecken.[6]

Im Gegensatz dazu wird bei aktiven Bedrohungen direkt eingegriffen. Das Erzeugen neuer Nachrichten, das Unterdrücken oder Verzögern von Nachrichten und die Fälschung von Nachrichten durch einen Angreifer sind aktive Bedrohungen. Sie lassen sich i. d. R. einfacher entdecken als passive Bedrohungen.

Abbildung 1.4 visualisiert die Systematik der Bedrohungen. Im Rahmen dieses Kurses wird der Schwerpunkt der Betrachtungen auf den gemusterten Bereichen (vorne unten im Würfel) liegen.

1.2.5 Klassische Bedrohungen

Hauptzweck der Erstellung von Computernetzen ist es, den Austausch von Daten zwischen verschiedenen Computern so einfach wie möglich zu machen. In

[6] Edward Snowden hat 2013 über solche Aktivitäten durch die National Security Agency (NSA) der USA berichtet.

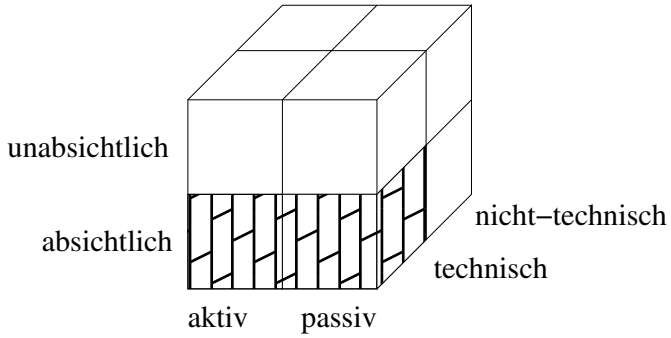

Abbildung 1.4: Klassifikation von Bedrohungen

diesem Abschnitt wird daher von einer normalen, ungestörten Kommunikation wie in Abbildung 1.5 ausgegangen. Ein Sender verschickt eine Nachricht an

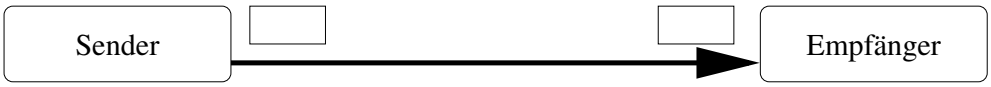

Abbildung 1.5: Ungestörter Nachrichtenaustausch

einen Empfänger und die Nachricht kommt dort unversehrt an. Die Nachricht könnte dabei eine E-Mail oder ein HTTP-Request oder eine beliebige andere elektronische Nachricht sein. In den folgenden Abschnitten wird dann aufgezeigt, welche Probleme beim Datenaustausch auftreten können.

Unbefugter Informationsgewinn: Der unbefugte Informationsgewinn ist ein Angriff auf die **Vertraulichkeit** (engl. **confidentiality**) der übertragenen Daten. Bei der elektronischen Abwicklung von Geschäften, z. B. dem Kauf von Aktien, möchten die beteiligten Parteien i. d. R. nicht, dass Dritte davon erfahren. Der unbefugte Informationsgewinn ist ein passiver Angriff, der absichtlich oder unabsichtlich erfolgen kann. Abbildung 1.6 zeigt das Prinzip des unbefugten Informationsgewinns.

Vertraulichkeit

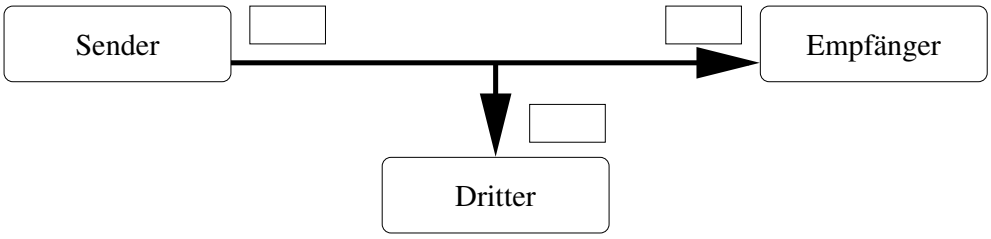

Abbildung 1.6: Unbefugter Informationsgewinn

Das „klassische“ Gegenmittel gegen diese Bedrohung besteht darin, die Nachricht für den Dritten unleserlich, unverständlich oder unerkennbar zu machen. Durch **Verschlüsselung** kann man Nachrichten unverständlich machen. Versteckt man eine Nachricht in einer anderen, unverfänglichen und i. d. R. größeren Nachricht, so spricht man von **Steganografie**.

Verschlüsselung

Steganografie

Der unbefugte Informationsgewinn ist nicht auf das Abhören von Datenübertragungen begrenzt. Wie bereits in Abschnitt 1.2.3 dargestellt, kann sich ein Unbefugter auch direkt Zugang zu einem Computer mit wichtigen Daten verschaffen. Diese kann ein Angreifer dann einsehen oder kopieren und mitnehmen.

In großen Rechenzentren sind die Computer daher in verschlossenen Räumen untergebracht. Der Zugang zu diesen Räumen ist genau reglementiert und wird u. U. durch Sicherheitsschleusen geregelt. Durch die steigende Verwendung von mobilen Computern, wie Notebooks, Smartphones oder Tablet-Computer, steigt jedoch die Gefahr von Diebstählen nicht nur der Geräte, sondern auch der darauf gespeicherten Daten.

Weiterhin ist unbefugter Informationsgewinn auch dadurch möglich, dass sich ein Angreifer Zugang zu den Datenträgern selbst verschafft. Die Festplatten sind normalerweise genauso sicher untergebracht wie die Computer selbst. Aber die Sicherungen, z. B. auf Magnetbändern oder dedizierten Backup-Festplatten, werden aus Sicherheitsgründen auch an anderen Orten als der Computer selbst gelagert. Eventuell sind diese Orte nicht so gut gesichert wie ein Rechenzentrum. USB-Sticks haben einen Kapazität von mehreren Gigabytes. Sie sind klein und können unbemerkt transportiert werden. Hat man Zugriff auf die USB-Schnittstelle eines Computers und steckt dort einen USB-Stick an, dann kann man Daten auf den Stick kopieren und dann mitnehmen.

Integrität

Unbefugte Modifikation: Die unbefugte Modifikation ist ein Angriff auf die **Integrität** (engl. **integrity**) der übertragenen Daten. Eine vom Sender als Kaufauftrag abgeschickte Nachricht könnte unterwegs von einem Dritten in einen Verkaufsauftrag umgewandelt werden. Die unbefugte Modifikation ist ein aktiver Angriff der i. d. R. absichtlich erfolgt. Doch auch durch technische Probleme ist eine unbeabsichtigte Modifikation möglich.

Abbildung 1.7: Unbefugte Modifikation

Abbildung 1.7 zeigt das Prinzip der unbefugten Modifikation einer Nachricht. Der Empfänger erhält eine andere Nachricht, als der Sender abgeschickt hat.

Gegen unbefugte Modifikationen kann man sich auf verschiedene Arten schützen. Durch die Einführung von Redundanz kann ein Empfänger erkennen, ob Daten auf dem Transportweg verändert wurden. Dazu werden an den Nutzinhalt zusätzliche Daten gehängt, die bestimmten Bedingungen genügen. Ein Beispiel hierfür sind die Paritätsbits von Hauptspeicherbausteinen. Bei gerader Parität wird das Paritätsbit so gesetzt, dass die Zahl der Einsen in einem Datenwort (einschließlich Paritätsbit) gerade ist. Folgende Gründe machen dieses Verfahren im Bereich der Sicherheit jedoch nicht einsetzbar:

- Einfache Verfahren erkennen nur bestimmte Veränderungen an den Daten, z. B. nur die Veränderung eines Bits. Andere Veränderungen bleiben dagegen unentdeckt.
- Bei einer beabsichtigten Modifikation der Daten kann ein Angreifer nicht nur die Nutzdaten, sondern auch die Redundanzdaten verändern. Für den Empfänger sieht die Nachricht dadurch korrekt aus. Voraussetzung hierfür ist, dass der Angreifer das Verfahren kennt, mit dem die Redundanzdaten berechnet werden.

Eine weitere Schutzmöglichkeit ist wiederum die Verschlüsselung der Daten mit einem geeigneten Verfahren. Ohne den Schlüssel zu kennen, kann ein

Angreifer dann keine sinnvolle Veränderung der Daten vornehmen. Verändert man wahllos einige Bytes einer gut verschlüsselten Nachricht, so entsteht bei der Entschlüsselung meistens etwas Unleserliches.

Angriffe auf die Integrität der Daten können nicht nur bei der Übertragung von Daten auftreten. Auch hier kann ein Unbefugter durch den direkten Zugang zu einem Computer, auf dem wichtige Daten gespeichert sind, diese Daten verändern. Eine weitere Gefährdung der Integrität von Daten sind ungeübte Benutzer. Durch die falsche Bedienung der Programme können Daten unbeabsichtigt modifiziert werden.

Unbefugte Erzeugung: Die unbefugte Erzeugung von Nachrichten ist ein Angriff auf die **Authentizität** (engl. **authenticity**). Dabei erzeugt jemand eine Nachricht und gibt darin vor, jemand anderes zu sein. Beispielsweise könnte Ihr Nachbar eine Bestellpostkarte auf Ihren Namen ausfüllen und damit bei irgendeinem Versandhändler eine Bestellung in Ihrem Namen aufgeben. Es kommt auch immer noch häufig vor, dass Personen einen Anruf von einem vorgeblichen Bankmitarbeiter erhalten. Dieser fragt den Angerufenen dann mit einer vorgeschobenen Begründung nach der Geheimzahl der EC-Karte. Geben Sie Ihre Geheimzahl oder ein Passwort *niemals* jemand anderem preis!

Authentizität

Abbildung 1.8 zeigt das Prinzip der unbefugten Erzeugung. Ohne dass der Sender aktiv wird, erhält der Empfänger eine Nachricht. Die unbefugte Erzeugung ist ein aktiver Angriff und erfolgt überwiegend absichtlich. Er kann bei Konfigurations- oder Bedienungsfehlern auch unabsichtlich erfolgen.

Abbildung 1.8: Unbefugte Erzeugung von Nachrichten

Normalerweise überzeugt man sich von der Identität einer Person durch einen Ausweis, wie z. B. einen Personalausweis. Um festzustellen, ob ein Brief tatsächlich vom vorgeblichen Absender kommt, bedient man sich der *eigenhändigen Unterschrift*. Diese Methoden haben den Vorteil, dass sie schwer zu fälschen oder zu kopieren sind. Man kann mit dem Auge recht gut feststellen, ob man ein Originaldokument oder eine Fotokopie vor sich hat. In der digitalen Welt der Computer kann man den Unterschied zwischen „Original-Bit“ und „Kopie-Bit“ jedoch nicht mehr feststellen. Man ist dort auf andere Verfahren angewiesen, auf die wir in Abschnitt 2.6 noch genauer eingehen werden.

Das Problem der Authentizität ist auch eng verwandt mit dem Problem der **Nicht-Zurückweisbarkeit** (engl. **non-repudiation**) von Nachrichten. Dabei geht es darum, dass weder der Sender noch der Empfänger die stattgefundene Kommunikation nachträglich abstreiten (zurückweisen) können. Konkret bedeutet dies, dass

Nicht-Zurückweisbarkeit

- der Empfänger beweisen kann, dass die Nachricht tatsächlich vom vorgegebenen Absender kommt (vorhandene eigenhändige Unterschrift) und
- der Sender beweisen kann, dass die Nachricht tatsächlich beim geplanten Empfänger und nicht bei jemand anderem angekommen ist (vgl. Einschreiben mit Rückschein bei der Post).

Verfügbarkeit

Unbefugte Unterbrechung: Die unbefugte Unterbrechung ist ein Angriff auf die **Verfügbarkeit** (engl. **availability**) von Daten, Computern und Kommunikationsmitteln. Wenn Ihr Nachbar Ihr Telefonkabel durchschneidet, so steht Ihnen dieses Kommunikationsmittel nicht mehr zur Verfügung. Insbesondere bei zeitkritischen Geschäften ist die Verfügbarkeit der Systeme sehr wichtig. Aktienkurse können sich beispielsweise sehr schnell ändern und eine Verzögerung Ihrer Kauf-/Verkaufsorder durch die Nicht-Verfügbarkeit der Kommunikationsstrecke kann enorme wirtschaftliche Konsequenzen haben. Angriffe auf die Verfügbarkeit sind aktive Angriffe, die i. d. R. absichtlich erfolgen.

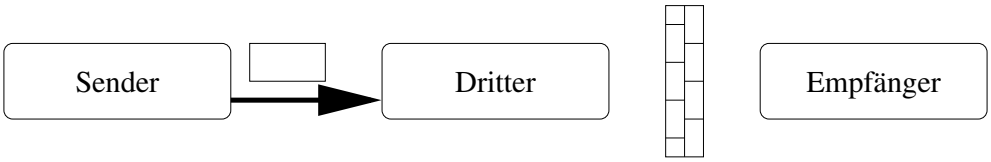

Abbildung 1.9: Unbefugte Unterbrechung

Angriffe auf die Verfügbarkeit können sich auch auf Computer selbst richten. Dabei werden speziell konstruierte Nachrichten an einen Computer geschickt, sodass bei der Bearbeitung der Nachricht das Betriebssystem abstürzt. Bis der Computer neu gestartet ist (evtl. ist dazu ein manueller Eingriff erforderlich!) stehen die Dienste dieses Computers nicht mehr zur Verfügung.

Das klassische Gegenmittel gegen diese Attacken besteht darin, sogenannte „hochverfügbare Systeme“ (engl. **high availability**) aufzustellen. Dem eigentlichen Rechnersystem stellt man ein zweites, redundantes System mit identischer Konfiguration zur Seite. Fällt ein System aus, übernimmt das andere den Betrieb. Das macht man nicht nur bei Rechnern, sondern auch anderen Komponenten, wie Stromversorgung, Festplatten, Internetanschlüssen usw. Bei Ausfällen aufgrund technischer Probleme (z. B. Ausfall der Festplatte, Blitzeinschlag, o. ä.) hat sich dieses Mittel bewährt. Gezielte Attacken, die Fehler im Betriebssystem ausnutzen und den Computer dadurch zum Absturz bringen, kann man durch ein identisches Backup-System nicht bekämpfen.

Allgemein gesagt entwirft man Architekturen, die sich nicht durch einen Fehler an einer einzelnen Stelle lahm legen lassen. Solche Stellen nennt man auf Englisch **single point of failure**. Sie sind zu vermeiden. Das Thema *sichere Architekturen* wird im Kurs *(01867) Sicherheit im Internet 2* ausführlicher besprochen.

single point of failure

Zusammenfassung der zu schützenden Eigenschaften:

Vertraulichkeit (engl. **confidentiality**)**:** Daten sind nur für befugte Personen zugänglich.

Integrität (engl. **integrity**)**:** Daten sind korrekt und unverändert.

Authentizität (engl. **authenticity**)**:** Daten stammen vom vorgeblichen Erzeuger.

Verfügbarkeit (engl. **availability**)**:** Daten können von befugten Personen gelesen/bearbeitet werden.

1.2.6 Nicht-technische Aspekte von Sicherheit

Im vorigen Abschnitt wurde der Begriff der Sicherheit in dem Sinne vorgestellt, dass ein sicheres System für den Benutzer *verlässlich* sein muss. Konkret bedeutet dies, dass die Funktionsweise des Systems den gestellten Anforderungen genügt. Der Benutzer kann sich auf die Korrektheit der Ergebnisse (z. B. der übertragenen Daten oder der Identität des Kommunikationspartners) ebenso verlassen wie auf die Verfügbarkeit des Systems. Neben dieser technischen Sicht gibt es auch die Sicht der Betroffenen. Aus dieser Sicht ist ein System sicher, wenn es für die Betroffenen *beherrschbar* ist. Dies bedeutet, dass der Einzelne und auch die gesamte Gesellschaft vor den unerwünschten Auswirkungen neuer Technologien und Systeme geschützt werden müssen [MR99].

Verlässlichkeit

Beherrschbarkeit

Neue Technologien und Systeme verändern das Verhalten der Menschen, die diese Systeme benutzen. Die neuen Möglichkeiten der Informationsverarbeitung beeinflussen auch die Struktur der Gesellschaft. Die Erfindung des Buchdrucks mit beweglichen Lettern durch Gutenberg hat schon einmal das Leben der Menschen und die Struktur der Gesellschaft grundlegend verändert. Ähnliches kann durch die Verbreitung von Personal Computern (PCs) und des Internets wieder vorkommen. Seit dem 11. September 2001 werden in den USA zum Zweck der Terrorabwehr vermehrt Daten gesammelt (Patriot Act).

Dazu kommen die Daten, die viele Menschen freiwillig von sich selbst in „sozialen Netzwerken“ wie *XING*, *studiVZ*, *Facebook*, *MySpace* usw. einstellen. Neben Namen, Anschrift und weiteren Kontaktdaten werden dort auch berufliche Informationen und persönliche Hobbys und Interessen eingetragen. Manchmal findet man dort auch einen Spitznamen einer Person, der in eigentlich anonymen Diskussionsforen als Benutzerkennung auftaucht. Die Beiträge unter dieser Benutzerkennung könnten dann der Person mit dem Spitznamen zugeordnet werden. Der Schutz der Privatsphäre (engl. **privacy**) eines Menschen ist durch IT-Technik möglicherweise gefährdet.

privacy

Wenn man einmal zusammenstellt in welchen Bereichen Daten über Personen erfasst werden, ergibt sich eine lange Liste:

Finanzdaten: Wo hat man Konten? Wie viel verdient jemand? Wohin wird Geld überwiesen? Wo hebt man Geld ab, d. h., wo hält man sich auf? Wo kauft man ein (und bezahlt mit Karte)?

Konsumdaten: Mit Hilfe von Rabattkartensystemen erfassen Geschäfte die Einkäufe ihrer Kunden. Was wird wann und wo gekauft? Daraus lassen sich beispielsweise Ernährungsgewohnheiten ableiten, an denen Krankenkassen Interesse haben könnten.

Kommunikationsdaten: Mit wem telefoniert jemand? Wer bekommt E-Mails von wem? Telefongespräche und E-Mails können abgehört werden (und werden es auch). Bei Mobiltelefonen kann man auch orten, wo sich der Teilnehmer aufhält. Wohin surfen Benutzer im Internet? In welchen Diskussionsgruppen beteiligen sich Personen?

Aufenthaltsdaten: Die Zahl von Überwachungskameras (Flughafen, Kaufhaus, öffentliche Plätze, Hotels, Banken usw.) steigt ständig und gibt Auskunft über den Aufenthaltsort von Personen. Bei Flugreisen muss man sich auch ausweisen und in den USA u. U. auch biometrische Daten (Fingerabdrücke) von sich selbst abgeben.

In modernen Digitalkameras sind immer öfter auch GPS-Sensoren eingebaut. Somit kann bei jedem Foto in den Metadaten gespeichert werden, wann und wo das Foto aufgenommen wurde. Speichert man seine Fotos im Internet (in einer Cloud, bei einem sozialen Netz, usw.) so kann jeder mit Zugriff auf diese Fotos wissen, wo sie (als Fotograf oder als Motiv auf dem Foto) zum Aufnahmezeitpunkt waren. Kombiniert mit automatischer Gesichtserkennung kann man auch Bewegungsprofile der abgelichteten Personen erstellen.

Persönliche Informationen: In sozialen Netzen findet man neben dem Namen einer Person oft auch Fotos, Informationen zur Ausbildung (den eigenen Kenntnissen und Fertigkeiten), zum Arbeitsplatz (in welcher Firma, welche Position, welche Tätigkeit) und zu den sonstigen Interessen, z. B. den Hobbys.

Diese Liste ist nicht vollständig. Trotzdem können diese Datenmengen zusammengefasst und mit Hilfe von **Data-Mining** durchsucht und korreliert werden. Daraus kann man

Data-Mining

> erschreckend komplette Dossiers über sie erhalten: über Lebensgewohnheiten, Interessen und Vorlieben, Lebensstile, persönliche Probleme und sexuelle Orientierungen, politische Neigungen, finanzielle Verhältnisse, Familienstand usw. [DIE ZEIT Nr. 34, 12. August 2004]

Der Aufbau der Computer- und Netztechnologie ist für den Einzelnen im Grundsatz verständlich. Trotzdem ist das Ausmaß der Veränderungen für die Gesellschaft nur zum Teil vorhersehbar. Die Auswirkungen des E-Commerce sind nur schwer vorhersehbar, insbesondere die Auswirkungen auf die bisher als Verkäufer oder allgemein im Vertrieb tätigen Menschen.

Daneben gibt es ein weiteres Problem. Die technische Zuverlässigkeit von Computern (sie können riesige Datenmengen in sehr kurzer Zeit fehlerfrei verarbeiten, z. B. Zahlenkolonnen addieren oder Gleichungssysteme lösen) verleitet die Benutzer zu einer unzulässigen Verallgemeinerung. Es resultiert ein falsches Vertrauen in die Objektivität der Datenverarbeitung nach dem Motto: „Was der Computer berechnet hat, wird schon stimmen."

Software-Fehler

Die Vernetzung der Systeme und die immer weiter steigende Komplexität der Systeme stellt die Beherrschbarkeit in Frage. Da man nur noch selten das gesamte System durchschaut und versteht, übersieht man möglicherweise Schwachstellen und Missbrauchsmöglichkeiten. Und welcher PC-Benutzer weiß heute schon, was sein Betriebssystem im Hintergrund so alles macht? Ist man wirklich sicher, dass private Daten nicht heimlich an Dritte übertragen werden?

Neben der technischen Sicht der Sicherheit, also der *Verlässlichkeit* von Systemen, darf auch die Benutzer- (bzw. Betroffenen-)Sicht, also die Frage der *Beherrschbarkeit* von Systemen, nicht vergessen werden. In diesem Kurs geht es allerdings nicht um Fragen der Beherrschbarkeit, sondern zunächst um die Verlässlichkeit der Systeme.

An dieser Stelle soll auch erwähnt werden, dass die Wahrnehmung von Gefahren, d. h. letztlich die Sicherheit, auch von der Perspektive des Wahrnehmenden abhängt. Ein Netzbetreiber hält es eventuell für gefährlich, wenn jemand anonym Zugang zum Netz erhält. Wie könnte sich der Betreiber vor un-

Interessenkonflikt

liebsamen Teilnehmern schützen? Wer bezahlt den Betreiber für die in Anspruch genommenen Dienste?

Auf der anderen Seite möchten Sie als „Websurfer“ nicht unbedingt mit Ihrem wahren Namen auftreten. Dadurch verhindern Sie, dass ein Aktivitäts- und Interessenprofil von Ihnen erstellt wird. Ein aus Ihrer Sicht sicheres System erlaubt also Anonymität, ein aus Sicht des Betreibers sicheres System würde Anonymität lieber verbieten.

Solche widersprüchlichen Sichten ergeben sich nicht nur bei verschiedenen Beteiligten, sondern auch aus den unterschiedlichen Situationen, in denen ein Beteiligter agiert. Während einer Recherche oder bei anderer Informationsbeschaffung möchten Sie gerne anonym bleiben. Auf der anderen Seite wollen und erwarten Sie beispielsweise bei Bankgeschäften, dass Sie und die Bank keinesfalls anonym bleiben. Sie möchten wissen, dass Sie auch tatsächlich mit der Bank kommunizieren und von einer Bank erwarten Sie, dass die Bank keine anonymen Aufträge zu Lasten Ihres Kontos ausführt.

Übungsaufgabe 1.1 *Welches sind die vier zu schützenden Eigenschaften in einem System? Erklären Sie deren Bedeutung mit eigenen Worten.*

1.3 Netze

In diesem Abschnitt werden der Aufbau und die Struktur von Computernetzen, insbesondere des Internets diskutiert. Die im Internet verfügbaren Dienste und die dort benutzten Kommunikationsprotokolle werden vorgestellt. Es wird auf die einzelnen Bereiche jedoch nur so weit eingegangen, wie es für das Verständnis des Kurses erforderlich ist.

1.3.1 Lokale Netze

Schon seit Jahren werden Computer miteinander vernetzt. Alle Computer einer Abteilung, eines Lehrgebietes oder einer beliebigen anderen geschlossenen Benutzergruppe werden in einem lokalen Netz (engl. **Local Area Network (LAN)**) miteinander verbunden. So können die Benutzer bestimmte Ressourcen wie Drucker oder Dateien gemeinsam benutzen und sehr einfach Daten austauschen. In lokalen Netzen werden unterschiedliche Topologien eingesetzt. Im Folgenden stellen wir diese Topologien kurz vor und betrachten dabei insbesondere die Auswirkungen auf die Sicherheit.

Netztopologien

Sterntopologie: In einem Netz mit Sterntopologie werden alle Computer an einen zentralen Punkt angeschlossen (siehe Abbildung 1.10). Aufgrund der Ähnlichkeit mit einem Rad nennt man den Mittelpunkt auch Nabe (engl. **hub**). Ein Beispiel für ein Sternnetz ist das ATM (engl. **Asynchronous Transfer Mode**) der Telefongesellschaften. Der zentrale Punkt ist ein elektronischer Vermittler (engl. **switch**), der dedizierte Verbindungen zwischen den angeschlossenen Computern schaltet. Die Kommunikationsdaten fließen also nur vom Sender zum Switch und zum Empfänger. Auf die Sicherheit hat dies folgende Auswirkungen:

- Fällt ein Computer aus, so können die anderen Computer weiterhin kommunizieren (Verfügbarkeit).

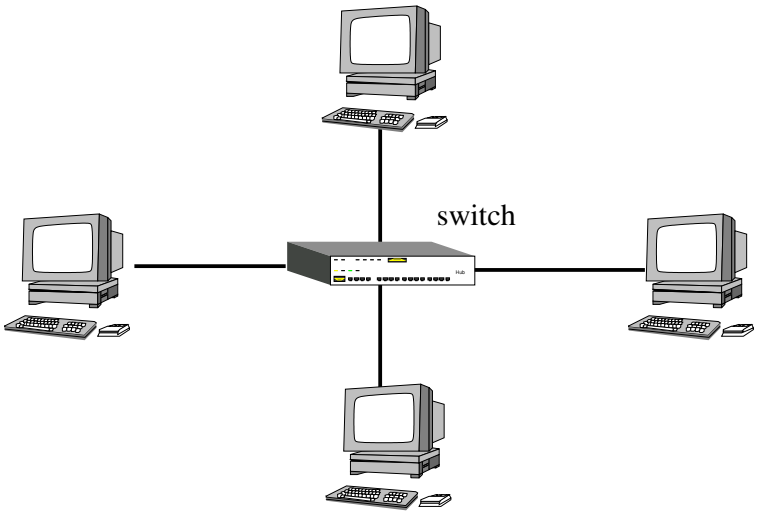

Abbildung 1.10: Prinzip der Sterntopologie

- Fällt der Switch aus, so ist keine Kommunikation mehr möglich (Verfügbarkeit).

- Da die Daten nur den Sender, den Switch und den Empfänger passieren, können andere Computer diese Kommunikation nicht stören oder abhören (Vertraulichkeit).

  Das setzt allerdings voraus, dass der Switch die Adressen der angeschlossenen Systeme kennt. Ethernet-Switches lernen diese Adressen im laufenden Betrieb. Trifft ein Paket ein, so steht darin die Adresse des Absenders. Soll der Switch anschließend ein anderes Paket an genau diese Adresse schicken, dann kennt der den zugehörigen Anschluss (engl. **port**). Der Switch ist also darauf angewiesen, dass die angeschlossenen Computer kooperativ sind und nicht beispielsweise gefälschte Absenderadressen benutzen.

Ringtopologie: Sind alle Computer eines lokalen Netzes in einer geschlossenen Schleife angeordnet, so spricht man von einer Ringtopologie (siehe Abbildung 1.11). Ein Beispiel für ein Ringnetz ist der *Token-Ring* von *IBM*. Die
Token Computer benutzen eine spezielle Nachricht, genannt **Token**, um die Nutzung zu koordinieren. Will ein Computer Daten verschicken, so wartet er auf das Token und sendet dann die Daten an seinen „rechten“ Nachbar. Die Nachricht wird von Computer zu Computer weitergereicht. Der Empfänger erstellt eine Kopie der Nachricht und der Absender erhält die Nachricht wieder zurück. Danach gibt der Sender das Token, und somit die Sendeerlaubnis, an seinen „rechten“ Nachbarn weiter. Will kein Computer senden, kreist das Token mit hoher Geschwindigkeit. Auf die Sicherheit hat dies folgende Auswirkungen:

- Fällt ein Computer oder *eine einzige* Verbindung aus, so ist die Kommunikation unterbrochen (Verfügbarkeit).

- Eine Nachricht passiert alle angeschlossenen Computer und kann im Prinzip auf jedem Computer gelesen werden (Vertraulichkeit). Jeder Computer könnte die Nachricht auch verfälschen (Integrität) oder unterdrücken.

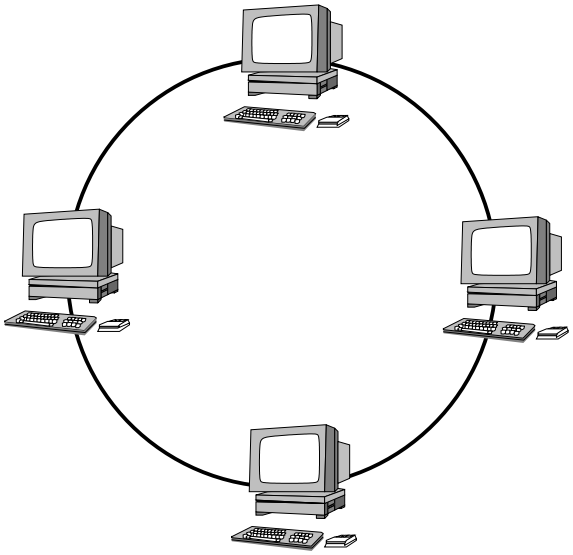

Abbildung 1.11: Prinzip der Ringtopologie

Bustopologie: Bei der Bustopologie sind alle Computer an das gleiche Kabel angeschlossen (siehe Abbildung 1.12). Ein Beispiel für ein Busnetz ist das *Ethernet*. Es kann in seiner Grundform die Daten mit 10 Mbit/s Geschwindigkeit übertragen. Inzwischen gibt es auch *Fast-Ethernet* mit 100 Mbit/s Übertragungsrate und auch *Gigabit-Ethernet* mit 1 Gbit/s Übertragungsrate. Wenn

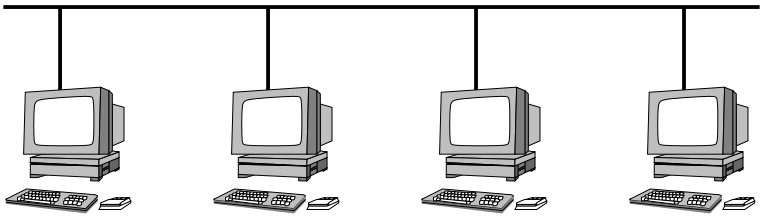

Abbildung 1.12: Prinzip der Bustopologie

ein Computer Daten versendet, dann können alle an das Kabel angeschlossenen Computer diese Nachricht lesen. Wollen zwei Computer gleichzeitig senden, so kommt es zu einer Kollision und die Computer müssen die Übertragung abbrechen und später wieder aufnehmen. Auf die Sicherheit hat dies folgende Auswirkungen:

- Fällt ein Computer aus, so können die anderen Computer weiterhin miteinander kommunizieren (Verfügbarkeit).
- Läuft ein Computer „Amok“, so können die anderen Computer u. U. nicht mehr kommunizieren, da die Leitung belegt ist (Verfügbarkeit).
- Jeder angeschlossene Computer kann *alle* Nachrichten auf dem Kabel mitlesen (Vertraulichkeit).

1.3.2 Vernetzte Netze

Lokale Netze können nicht beliebig groß werden. Außerdem gibt es nicht „die beste“ Netztechnologie für alle Anwendungsfälle. Es besteht also der

Bedarf unterschiedliche Netze miteinander zu verbinden. Dazu stehen folgende Gerätetechnologien zur Verfügung:

Repeater: Ein Repeater ist ein Gerät, das zwei gleichartige lokale Netze miteinander verbindet und zu einem Gesamtnetz macht. Dazu verstärkt und überträgt der Repeater *alle* Signale (also auch Störungen) zwischen den beiden Netzen.

Bridge: Eine Bridge arbeitet im Prinzip wie ein Repeater. Sie gibt aber nicht alle Signale weiter, sondern leitet nur vollständige und fehlerfreie Datenrahmen weiter. Daher kann eine Bridge anhand der Absenderadressen in den Rahmen mit der Zeit lernen, welcher Computer in welchem Netz liegt. Die Bridge leitet die Rahmen daher nur dann weiter, wenn der Zielcomputer in einem anderen Netz liegt.

Switch

Dies setzt voraus, dass die Struktur und die Datenrahmen in beiden Netzen gleich sind. Für ein Ethernet realisieren sogenannte Switches eine Menge von einelementigen Subnetzen für jeden angeschlossenen Computer, der über Bridges mit allen einelementigen Subnetzen verbunden ist. Dadurch kann die Bustechnologie Ethernet sternförmig verkabelt sein.

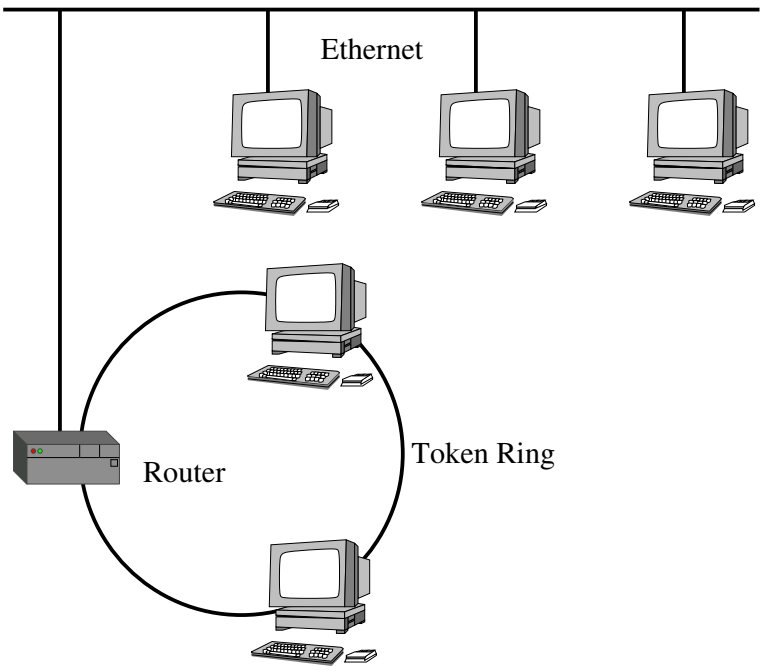

Abbildung 1.13: Verbindung von Ethernet und Token-Ring durch einen Router

Router: Ein Router ist ein dedizierter Computer für den Zusammenschluss von Netzen. Er kann Netze unterschiedlicher Technologien mit verschiedenen Medien, Adressschemata oder Rahmenformaten verbinden. Dazu hat er eine getrennte Schnittstelle für jeden Netzanschluss, d. h. eine Netzwerkkarte für jeden Netztyp. Ein Router kann also ein Busnetz wie das Ethernet mit einem Ringnetz wie dem Token-Ring verbinden (siehe Abbildung 1.13).

1.3.3 Das Internet-Protokoll

Damit man Daten zwischen unterschiedlichen Netzen austauschen kann, muss man sich vorher auf bestimmte Dinge wie Datenformate, Adressierung, Verbindungsaufbau usw. einigen. Diese Einigungen werden in einem sogenannten **Protokoll** festgehalten. Die am häufigsten implementierte Protokollfamilie ist das *TCP/IP*. Protokoll

Protokollschichten: Die Protokolle der TCP/IP-Familie sind hierarchisch in Schichten organisiert. In Abbildung 1.14 sind die Teile dargestellt, die bei einer Kommunikation beteiligt sind. Auf der obersten Ebene stehen die

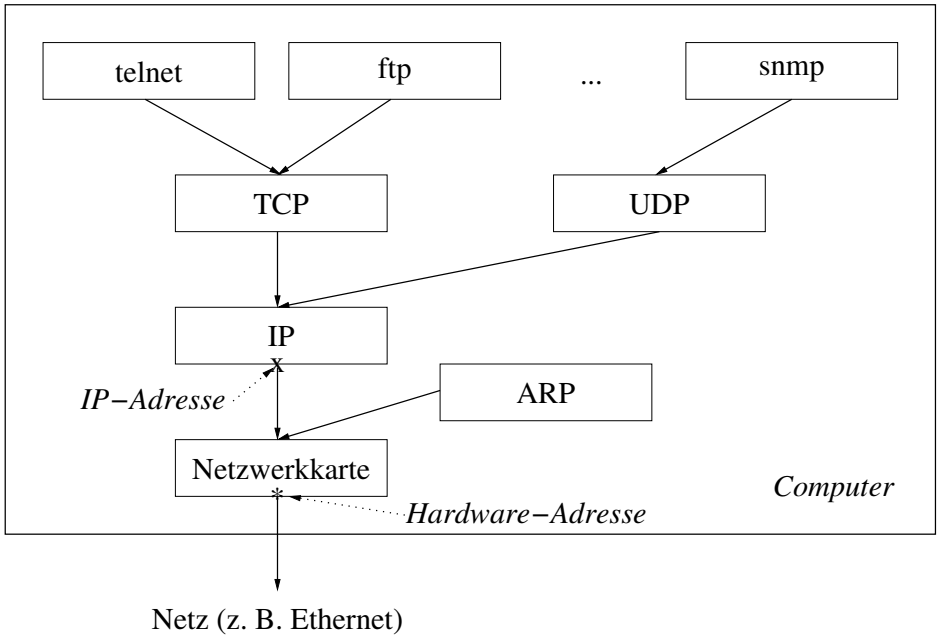

Abbildung 1.14: Ablauf der Kommunikation bei TCP/IP

Anwendungsprogramme (wie *telnet*, *ftp* oder andere), die den Bedarf haben, mit einem anderen Computer Daten auszutauschen. Diese Programme rufen dazu Funktionen aus der *Transmission-Control-Protocol*-Schicht (TCP-Schicht) bzw. der *User-Data-Protocol*-Schicht (UDP-Schicht) auf. Das Anwendungsprogramm übergibt eine Anwendungsnachricht. TCP UDP

Die Funktionen der TCP-Schicht zerlegen diese Nachricht und leiten TCP-Segmente an die darunterliegende *Internet-Protocol*-Schicht (IP-Schicht) weiter. In der IP-Schicht muss auch die IP-Adresse des Computers selbst bekannt sein. IP

Die IP-Schicht gibt ein IP-Paket (versehen mit einer IP-Zieladresse) an die Netzwerkkarte weiter. Mit Hilfe des *Address-Resolution-Protocol (ARP)* und der auf dem Computer gespeicherten Tabelle wird die Hardware-Adresse des Zielcomputers ermittelt. Damit versehen geht dann beispielsweise ein Ethernet-Frame auf den Weg vom Computer an das Netz. ARP

Adressierung: Jede Netzwerkkarte hat eine eigene eindeutige Adresse. Diese hängt von der verwendeten Netztechnologie ab und ist bei Ethernet beispielsweise 6 Bytes lang. Für das Internet abstrahiert man von den Hardwareadressen und benutzt ein eigenes Adressschema, die sogenannten **IP-Adressen**. Heute IP-Adressen

werden zwei Versionen von IP-Adressen verwendet, (1) IPv4-Adressen und (2) IPv6-Adressen.

Eine IPv4-Adresse ist eine eindeutige 4 Byte lange Binärzahl. Sie besteht aus zwei Teilen. Das *Präfix* identifiziert das Netz, an das der Computer angeschlossen ist. Das *Suffix* identifiziert den Computer innerhalb des Netzes. Da lokale Netze aus unterschiedlich vielen Computern bestehen können, ist die Länge des Suffixes nicht fest vorgegeben. IPv4-Adressen notiert man, indem jedes Byte als Dezimalzahl geschrieben wird. Zwischen die Zahlen wird ein Punkt geschrieben. Es gibt somit maximal 2^{32} viele IPv4-Adressen, das sind ca. vier Milliarden. Da heute aber nicht nur Computer, sondern auch Smartphones, Fernseher, Tablet-Computer und viele weitere Geräte an das Internet angeschlossen werden sollen, reichen die IPv4-Adressen nicht mehr aus.

Eine IPv6-Adresse ist 128 Bit lang, also 16 Bytes. Auch sie besteht aus zwei Teilen, einer Netzadresse (auch Präfix genannt) und einer Adresse innerhalb des Netzes (genannt Interface Identifier). Damit sind genügend Netze und genügend Geräte adressierbar. Man schreibt IPv6-Adressen als 32 hexadezimale Zahlen, gruppiert in acht durch Doppelpunkt getrennte Zifferngruppen. Eine IPv6-Beispieladresse sieht also so aus:

```
fe80:0000:0000:0000:3e07:54ff:fe4e:0e37
```

Während der Präfix der IPv6-Adresse i. d. R. vom Internet Service Provider vergeben wird, können Geräte in einem lokalen Netz den Interface Identifier auch automatisch selbst erzeugen. Dabei geht die Hardware-Adresse des Geräts in die IPv6-Adresse ein.

Das hat zur Folge, dass ein mobiles Gerät möglicherweise weltweit identifiziert werden kann. Auch wenn sich das Gerät in verschiedenen Netzen befindet und somit der Präfix der IPv6-Adresse variiert, kann der Interface-Identifier-Teil der IPv6-Adresse gleich bleiben. In der Spezifikation von IPv6 hat man daher gleich die sogenannten *Privacy Extensions* spezifiziert. Damit unterscheidet sich dann der Interface-Identifier-Teil der IPv6-Adresse derart, dass man ein Gerät nicht anhand der IPv6-Adresse wiedererkennen kann.

Privacy Extensions

Die IP-Adresse eines Computers wird bei der Konfiguration eingegeben und lokal gespeichert.[7] Zur ARP-Schicht gehört eine Tabelle der folgenden Form:

IPv4-Adresse	**Hardware-Adresse**
10.71.144.1	08-00-28-00-38-A9
10.71.144.2	08-00-39-00-2F-C3
10.71.144.3	...
...	

Der Inhalt dieser Tabelle wird jedoch nicht vom Administrator gepflegt, sondern sie füllt sich automatisch. Möchte der Computer ein Paket an eine IP-Adresse schicken, deren Hardware-Adresse nicht in der Tabelle steht, so wird das Paket zunächst zurückgestellt. Der Computer schickt eine Anfrage an alle angeschlossenen Computer (engl. **broadcast**) und fragt nach der Hardware-Adresse zu dieser IP-Adresse. Einer der angeschlossenen Computer erkennt nun seine IP-Adresse und antwortet mit seiner Hardware-Adresse. Anschließend

[7] Um diese aufwendige Arbeit zu sparen, benutzen viele Administratoren eine Technik mit der IP-Adressen automatisch an Rechner vergeben werden. Sie heißt *Dynamic-Host-Configuration-Protocol (DHCP)*. Es gibt DHCP für IPv4 und für IPv6.

kann das IP-Paket mit der richtigen Hardware-Adresse versehen in ein Netzpaket verwandelt und abgeschickt werden.

IPv4-Adressen sind für Benutzer nicht einfach zu merken, IPv6-Adressen noch sehr viel schwieriger. Daher benutzt man lieber Namen. Um Namen für Computer vergeben zu können, gibt es das **Domain-Name-System (DNS)**. Hierin werden symbolische Namen für Computer und deren zugehörige IP-Adressen verwaltet. Beispiele für DNS-Namen sind: `gremlin.fernuni-hagen.de` oder `www.deutsche-bank.de`. DNS-Namen sind hierarchisch aufgebaut. Ausgehend von einer leeren Wurzel (siehe Abbildung 1.15) setzt sich ein Name aus den Knoten eines Pfads durch den Baum zusammen. Dieser Baum gibt *nicht* die Netzstruktur wieder, sondern kann vom

Domain-Name-System (DNS)

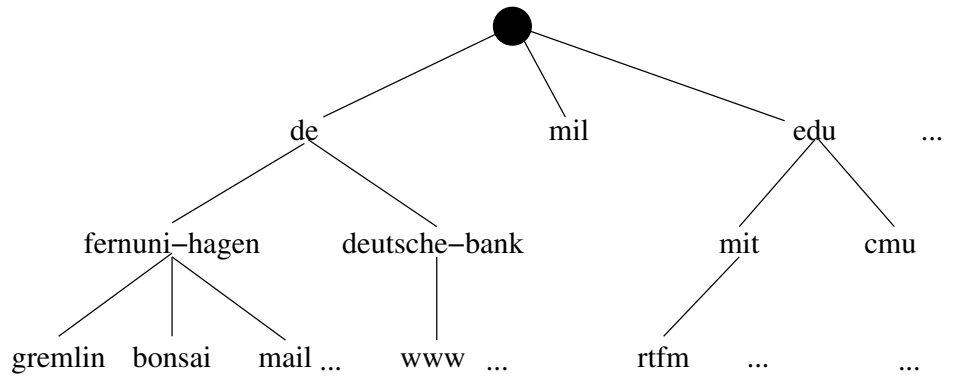

Abbildung 1.15: Beispiel für einen DNS-Namensraum

Verantwortlichen für einen Knoten beliebig unterhalb des Knotens gestaltet werden.

Früher wurde diese Zuordnung lokal auf jedem Computer gespeichert. Eine zentrale Stelle registrierte alle Veränderungen und verteilte die neue Zuordnungstabelle. Bei der derzeitigen Anzahl von Computern im Internet und der hohen Änderungsgeschwindigkeit ist dieser Mechanismus nicht mehr praktikabel.

Heute wird die Zuordnung von IP-Adresse zu Computername von einem **DNS-Server** erledigt. Dieser arbeitet in einem Verbund mit anderen DNS-Servern zusammen. Falls zu jedem inneren Knoten des Baums aus Abbildung 1.15 ein DNS-Server gehört, so muss dieser nur seine Kinder und seinen Vorgänger kennen. Wenn dann vom Computer `bonsai.fernuni-hagen.de` auf `www.deutsche-bank.de` zugegriffen werden soll, so entsteht eventuell folgende Abfragekette: `bonsai` fragt den DNS-Server von `fernuni-hagen`, der fragt den DNS-Server von `de`, der fragt den DNS-Server von `deutsche-bank` und der kennt die Adresse von `www`. Die IP-Adresse wird dann zurückgeschickt. Damit das Netz nicht ständig von solchen Anfragen belastet wird, merken sich die DNS-Server angefragte Adressen. Wenn also auch der Computer `gremlin.fernuni-hagen.de` die Adresse von `www.deutsche-bank.de` wissen will, so kann der DNS-Server der FernUniversität diese Anfrage aus seinem Zwischenspeicher (engl. **cache**) beantworten.

DNS-Server

Dieses Schema birgt leider auch gewisse Risiken. DNS-Nachrichten können einfach gefälscht werden. Außerdem kann man in einer DNS-Antwort neben der IP-Adresse des angefragten DNS-Namens auch weitere Informationen mitschicken. Der Empfänger dieser gefälschten Nachrichten kann diese Fälschungen nicht erkennen. Wenn Sie beispielsweise mit dem Computer `gremlin.fernuni-hagen.de` kommunizieren wollen, kann Ihr DNS-Server even-

tuell eine falsche IP-Adresse zurückgeben. Sie würden dann Daten an einen anderen Computer schicken, der nur vorgibt `gremlin.fernuni-hagen.de` zu sein. Man nennt das auch **DNS spoofing**.

DNS spoofing

DNS Security Extensions (DNSSEC)

Mit der Definition von *DNS Security Extensions (DNSSEC)* hat man DNS so erweitert, dass die Authentizität und Integrität von DNS-Nachrichten sichergestellt werden kann. Dazu werden *digitale Signaturen* (siehe auch Abschnitt 2.6) benutzt. DNSSEC ist in den RFCs 4033, 4034 und 4035 definiert.

Aber auch IP-Adressen sind kein sicherer Authentifizierungsmechanismus. Wie oben schon erwähnt, wird die Tabelle zum ARP durch eine Broadcast-Anfrage gefüllt. So kann auch ein anderer Computer in Ihrem lokalen Netz vorgeben, Ihre IP-Adresse zu haben. An Sie geschickte Daten landen dann auf einer anderen Maschine. Beachten Sie außerdem, dass der Administrator oder auch der Benutzer die IP-Adresse seines Computers selbst konfigurieren kann.

ARP spoofing

Für die verschiedenen Dienste bzw. Anwendungsprogramme ist es nicht nur erforderlich zu wissen, mit welchem Computer man kommunizieren möchte, sondern man muss auch mitschicken können, welchen Dienst man benutzen möchte. Dazu sind in TCP/IP die sogenannten *ports* vorgesehen. Ein Port ist eine Nummer. Empfängt ein Computer eine Nachricht, so kann die TCP/IP-Schicht bereits erkennen, welches Programm diese Nachricht erhalten muss. Der Anwendungsprogrammierer muss sich also nicht mit Datenpaketen befassen, die gar nicht für diese Anwendung gedacht sind. Die Portnummer wird an den Rechnernamen angehängt und durch einen Doppelpunkt vom Namen getrennt. Eine Rechneradresse inklusive Portnummer sieht dann beispielsweise so aus:

port

```
gremlin.fernuni-hagen.de:80
```

1.3.4 Die Internetdienste

In diesem Abschnitt werden einige typische Hilfsprogramme und Internetdienste vorgestellt. Es wird insbesondere auf die sicherheitsrelevanten Aspekte der Dienste eingegangen.

Nslookup: Mit dem Programm *nslookup* können sie einen DNS-Server abfragen. Sie können zu einem Rechnernamen die zugehörige IP-Adresse erfahren oder sie können zu einer IP-Adresse die zugehörigen Rechnernamen abfragen. Die Abbildung von Rechnernamen auf IP-Adressen ist nicht unbedingt bijektiv. Häufig sind mehrere Namen derselben IP-Adresse zugeordnet. Ein Beispiel für *nslookup*:

```
>nslookup -silent www.fernuni-hagen.de
Server:  141.71.30.1
Address: 141.71.30.1#53

Non-authoritative answer:
www.fernuni-hagen.de canonical name = cl-www.fernuni-hagen.de.
Name:   cl-www.fernuni-hagen.de
Address: 132.176.114.181
```

In diesem Beispiel kommt die Antwort vom DNS-Server mit der Adresse 141.71.30.1. Der Webserver der FernUniversität hat zwei Namen (`cl-www` und

`www`). Der Vorteil ist, dass man den Webserver bei Bedarf auf einer anderen Maschine installieren kann und dann nur den DNS-Eintrag ändern muss. Ihre lokal gespeicherten Verweise zeigen dann nach wie vor auf den richtigen Computer.

Unter UNIX ersetzt das Programm *dig* zukünftig *nslookup*. Die Option `-silent` in obigem Kommando unterdrückt die Erinnerungsmeldung von `nslookup`, dass zukünftig `dig` benutzt werden soll. Neben den Informationen über die IP-Adresse zu einem Namen gibt *dig* auch Informationen über den DNS-Server aus. Beispielsweise kann der DNS-Server eine IP-Adresse von einem anderen DNS-Server übermittelt bekommen haben oder er kann sie aus seinem lokalen Cache gelesen haben. In diesem Fall (Cache) könnte die Adresse evtl. nicht mehr gültig sein, da sie geändert wurde und sich diese Tatsache noch nicht bis zum Cache des DNS-Servers verbreitet hat.

Ping: Nachdem sie nun die IP-Adresse bzw. den Namen eines Computers kennen, können sie mit dem Programm *ping* testen, ob der Computer eingeschaltet ist und IP-Pakete empfangen und zurückschicken kann. Sie können den Computer entweder mit seiner IP-Adresse oder seinem DNS-Namen identifizieren.

```
>ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.207 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.041 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.098 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.090 ms

--- localhost ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.041/0.095/0.207/0.061 ms
```

In diesem Beispiel wird der lokale Rechner abgefragt. Beachten Sie, dass der Administrator eines Rechners den Rechner auch so konfigurieren kann, dass der Rechner niemals auf Pings antwortet.

Finger: Mit dem Programm *finger* können Sie herausfinden, wer an einem entfernten Computer angemeldet ist. Weiterhin können sie damit zusätzliche Informationen zu einem Benutzer erlangen. Da sich diese Informationen zur Vorbereitung von Angriffen benutzen lassen, wird der *finger*-Dienst i. d. R. deaktiviert.

Finger-Port: 79

Traceroute: Mit dem Programm *traceroute* können Sie den Weg verfolgen, den IP-Pakete von Ihrem lokalen Rechner zu einem entfernten Rechner (engl. **remote host**) nehmen. Es ist in erster Linie für Administratoren gedacht. Aber auch „normale“ Benutzer können es starten und dann etwas über die Struktur des Internets erfahren. Beachten Sie jedoch, dass sich diese Struktur ständig ändern kann und die Pakete beim nächsten Aufruf vielleicht einen ganz anderen Weg nehmen könnten. Das folgende Beispiel zeigt den Weg, den IP-Pakete aus der Abteilung Informatik der Hochschule Hannover zum Studentenrechner *bonsai* der FernUniversität in Hagen zurücklegen.

```
> traceroute bonsai.fernuni-hagen.de
traceroute to bonsai.fernuni-hagen.de (132.176.114.21), 30 hops max, 40 byte packets
 1  inner-gw-i.inform.fh-hannover.de (141.71.30.62)  0.288 ms   0.198 ms   0.128 ms
 2  outer-gw-i.inform.fh-hannover.de (141.71.31.30)  0.357 ms   0.366 ms   0.414 ms
 3  rzswitch10.rz.fh-hannover.de (141.71.1.240)  1.019 ms   0.754 ms   0.805 ms
 4  igwserv.rz.fh-hannover.de (141.71.7.2)  0.779 ms   0.628 ms   0.671 ms
 5  141.71.8.2  1.261 ms   1.093 ms   1.222 ms
 6  clustergate-907-rf.rrzn.uni-hannover.de (130.75.9.9)  2.752 ms   2.571 ms   2.703 ms
 7  gwingate-cgc.rrzn.uni-hannover.de (130.75.9.245)  3.675 ms   3.404 ms   3.629 ms
 8  ar-hannover1.g-win.dfn.de (188.1.46.1)  4.031 ms   3.962 ms   3.914 ms
 9  cr-hannover1-ge5-1.g-win.dfn.de (188.1.88.1)  4.452 ms   4.201 ms   4.061 ms
10  cr-essen1-po0-0.g-win.dfn.de (188.1.18.49)  9.775 ms   9.198 ms   9.005 ms
11  ar-essen1-ge0-0-0.g-win.dfn.de (188.1.86.2)  9.659 ms   8.775 ms   9.736 ms
12  C65-GWIN.fernuni-hagen.de (132.176.100.1)  11.467 ms   11.357 ms   11.261 ms
13  bonsai.fernuni-hagen.de (132.176.114.21)  11.465 ms   11.339 ms   11.520 ms
```

Man erkennt an den ersten beiden Zeilen (*inner gateway* und *outer gateway*), dass die Abteilung Informatik der Hochschule Hannover eine Firewall mit DMZ betreibt. Das Thema Firewall wird in Kurseinheit 4, Abschnitt 4.3 besprochen.

Einige Netzbetreiber konfigurieren ihre Router so, dass man auch mit *traceroute* keine Informationen darüber erhält, welchen Weg die Pakete nehmen. Alle an das Internet angeschlossene Firmennetze leiten die für *traceroute* erforderlichen Pakete nicht ins Internet. Man erhält somit keine Informationen über die interne Netzstruktur einer Firma.

Telnet-Port: 23

Telnet: Das Programm *telnet* ist ein Terminal-Programm, mit dem Sie sich mit einem beliebigen anderen Computer im Internet verbinden können. Dabei verhält sich Telnet so, als säßen Sie an einem Terminal, das direkt an den anderen Computer angeschlossen ist. Die Benutzerkennung und das Passwort werden über das Internet übertragen. Auch während der Sitzung werden die Benutzereingaben und die Antworten des Computers *unverschlüsselt* übertragen. Je nachdem welchen Weg die Pakete nehmen, können alle Computer auf diesem Weg die Daten mitschreiben. Statt *telnet* wird heute überwiegend *SSH* benutzt. Es wird in Abschnitt 3.4.1 vorgestellt.

Man kann Telnet aber einsetzen um Netzprotokolle zu testen. Dazu sagt man Telnet, mit welchem Port man verbunden werden will, indem man hinter dem Rechnernamen die Portnummer eingibt. Das Kommando `telnet www.fernuni-hagen.de 80` verbindet Sie direkt mit dem Webserver. Sie können dann die unten gezeigten HTTP-Nachrichten eintippen und die Antworten des Servers als Text angezeigt bekommen. Dasselbe macht Ihr Webbrowser im Prinzip auch, aus einem Klick wird eine Anfragenachricht und die Antwortnachricht wird im Fenster schön formatiert angezeigt.

ftp-Port: 21

Dateitransfer: Mit dem Programm *ftp* können Sie Dateien zwischen einem lokalen Computer und einem entfernten Computer hin und her kopieren. Das Programm erwartet beim Aufbau der Verbindung die Eingabe einer Benutzerkennung und eines Passworts. Diese werden, wie bei *telnet*, *unverschlüsselt* verschickt. Damit Sie nicht auf jedem ftp-Server eine Benutzerkennung einrichten müssen, gibt es die anonymen Kennungen `ftp` und `anonymous`. Diese brauchen kein Passwort (man bittet Sie i. d. R., Ihre eigene E-Mail-Adresse als Passwort anzugeben) und erlauben eingeschränkten Zugriff auf den Server.

Übungsaufgabe 1.2 *Welche Sicherheitsrisiken können bei* ftp *auftreten?*

World Wide Web: Waren die bisher vorgestellten Anwendungen und Dienste textbasiert und terminalorientiert, so ist das *World Wide Web (WWW)* multimediabasiert und an grafischen Benutzeroberflächen orientiert. Zusätzlich bietet das WWW Möglichkeiten des *Information Retrieval*. Als Benutzer brauchen sie ein Client-Programm, das auch als *Webbrowser* bezeichnet wird, um auf die Inhalte von Webservern zugreifen zu können. Die Kommunikation zwischen Webclient und Webserver ist durch das *Hypertext-Transfer-Protocol (HTTP)* definiert. Darin ist festgelegt, welche Form die Anfragen eines Webclients haben und wie ein Webserver darauf antwortet. Dokumente auf Webservern sind in der *Hypertext-Markup-Language (HTML)* geschrieben. HTML ist eine SGML-Anwendung, d. h. es gibt eine *Document-Type-Definition (DTD)* vom *World-Wide-Web-Consortium (W3C)* in der die zulässigen Dokumentbestandteile und ihre Organisation festgelegt sind. Für die Zukunft ist es geplant, nicht nur HTML-Dokumente, sondern allgemeine Dokumente in der *eXtensible-Markup-Language (XML)* zu erlauben. XML ist eine Weiterentwicklung von SGML, bei der man aus den Problemen mit SGML gelernt und die Anforderungen des WWW berücksichtigt hat. Weitere Informationen zu den Dateiformaten finden Sie auch im Kurs *(01873) Daten- und Dokumentformate*.

WWW

HTTP-Port: 80

HTML

XML

Webserver und die Dokumente auf ihnen werden durch *Uniform-Resource-Locators (URL)* adressiert. Eine URL besteht i. d. R. aus drei Teilen:

URL

1. dem Dienst, der benutzt werden soll,
2. dem Webserver, der kontaktiert werden soll und
3. dem Dokument, das angesprochen werden soll.

Die folgende Tabelle zeigt einige Beispiele von URLs.

Dienst	Webserver	Dokument
http	://www.fernuni-hagen.de	/FeU/Fachbereiche/fachbereiche_f.html
http	://www.deutsche-bank.de	
ftp	://ftp.fernuni-hagen.de	/

Die Kommunikation zwischen Client und Server basiert auf dem Frage-Antwort-Prinzip (engl. **request response**). Anfragen bestehen aus einem Anfragekopf und einem Anfragerumpf. Diese sind durch eine Leerzeile getrennt. Die erste Zeile eines Anfragekopfs enthält die Methode, das Objekt, auf das diese Methode angewendet werden soll, und die Versionsnummer des HTTP-Protokolls. Die weiteren Kopfzeilen enthalten Informationen wie beispielsweise das Datum, die Codierung der Zeichen usw. Ein Beispiel sieht wie folgt aus:

```
GET /index.html HTTP/1.1
Host: 10.71.144.4
```

In diesem Beispiel ist der Anfragerumpf leer. Hinter der Zeile `Host:` kommt also noch eine Leerzeile.

Die Antwort des Servers besteht aus einem Antwortkopf und einem Antwortrumpf, die wiederum durch eine Leerzeile getrennt sind. Die erste Zeile der Antwort ist eine Statuszeile. Die weiteren Kopfzeilen sind teilweise wie im Anfragekopf. Der Antwortrumpf enthält meistens die angeforderte HTML-Seite. Ein Beispiel:

```
HTTP/1.1 200 OK
Date: Fri, 06 Aug 1999 14:57:16 GMT
Server: mod_perl/1.18 Apache/1.3.4 (Unix)  (SuSE/Linux) ...
Last-Modified: Fri, 06 Aug 1999 14:27:12 GMT
Accept-Ranges: bytes
Content-Length: 4464
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML3.2//EN" "strict.dtd">
<HTML>
  <HEAD>
 ...
  </HEAD>
  <BODY>
 ...
  </BODY>
</HTML>
```

Die gesamte Kommunikation findet unverschlüsselt statt. Weiterhin kann weder der Client sicher sein, dass er tatsächlich mit dem Server kommuniziert, mit dem er kommunizieren möchte, noch kann der Server sicher sein, dass der Client derjenige ist, der er vorgibt zu sein (Authentizität).

Electronic Mail: Die Möglichkeit, schnell und einfach elektronische Nachrichten (engl. **email**) auszutauschen, hat sicherlich stark zur Popularität des Internets beigetragen. Die Art und Weise, wie eine E-Mail verschickt wird, ist im *Simple-Mail-Transfer-Protocol (SMTP)* festgehalten. Der Benutzer benötigt ein Client-Programm, in dem er seine E-Mail schreibt, verwaltet, empfängt und verschickt. Das Client-Programm kommuniziert dazu mit einem Mailserver, der E-Mails entgegen nimmt und entweder direkt zustellt oder an einen weiteren Mailserver weiterleitet. Die Übertragung einer E-Mail vom Mailclient zum Mailserver, bzw. von Mailserver zu Mailserver läuft wie folgt ab:

SMTP port: 25

Versand

1. Der Mailclient öffnet eine Verbindung zum Mailserver. Konkret bedeutet dies, dass an den Port 25 des Mailservers eine Nachricht geschickt wird.

   ```
   S: HELO gremlin.fernuni-hagen.de
   ```

   Das vorangestellte „`S:` “ kennzeichnet Nachrichten vom Mailclient an den Server. Antworten des Servers werden durch ein „`R:` “ gekennzeichnet. Diese Zeichen werden *nicht* übertragen, sondern sind zum besseren Verständnis der Beispiele von mir eingefügt worden. In der oben genannten Nachricht identifiziert sich der E-Mail verschickende Computer gegenüber dem Mailserver.

2. Als nächstes führt der Mailclient die eigentliche Übertragungstransaktion aus. Diese besteht aus drei Schritten:

   (a) Zunächst wird ein einleitendes Kommando geschickt.

   ```
   S: MAIL FROM: <wohlfeil@gremlin.fernuni-hagen.de>
   ```

Der Mailserver beginnt eine neue Transaktion und initialisiert seinen Zustand und seine internen Puffer.

(b) Anschließend schickt der Mailclient ein oder mehrere „Empfänger“-Kommandos:

```
S: RCPT TO: <rolf.klein@fernuni-hagen.de>
R: 250 OK
S: RCPT TO: <asdfg@fernuni-hagen.de>
R: 550 no such user here
S: RCPT TO: <swohlfeil@acm.org>
R: 250 OK
```

Der Mailserver überprüft und speichert die Empfängeradressen. Das Beispiel zeigt, dass ein Mailserver bestimmte Empfängeradressen, wie `asdfg`, u. U. sofort verwerfen kann. Der Server speichert diese Adressen, um die anschließend vom Client übergebene Nachricht an diese Adressen zu schicken.

(c) Zuletzt wird die eigentliche Nachricht übertragen. Diese Übertragung wird durch das DATA-Kommando eingeleitet. Das Ende der Nachricht wird durch eine Zeile gekennzeichnet, die nur einen Punkt enthält.

```
S: DATA
R: 354 start mail input; end with <CRLF>.<CRLF>
S: From stefan.wohlfeil@gremlin.fernuni-hagen.de Fri ...
S: Return-Path: <stefan.wohlfeil@fernuni-hagen.de>
S: Date: Fri, 11 Jun 1999 15:05:26 +0200 (MEST)
S: From: Stefan Wohlfeil <stefan.wohlfeil@gmx.net>
S: To: rolf.klein@fernuni-hagen.de
S: Cc: swohlfeil@acm.org
S: Subject: Mail Uebertragung mit SMTP
S: Content-Type: text/plain; charset="iso-8859-1"
S: Content-Transfer-Encoding: 8bit
S:
S: Hallo Herr Klein,
S: ...
S: MfG
S: Stefan Wohlfeil
S: .
S:
R: 250 message accepted for delivery
```

3. Der zweite Schritt kann nun mehrmals wiederholt werden. Hat der Mailclient alle E-Mails abgeliefert, so beendet er die Verbindung zum Mailserver.

```
S: QUIT
R: mail.fernuni-hagen.de closing connection
```

Die oben dargestellten Beispiele zeigen das Prinzip der Mailübermittlung, nicht unbedingt den Wortlaut der Antworten eines Mailservers. Vom Sicherheitsstandpunkt gibt es einige Anmerkungen. Sicherheit

Zunächst wird bei E-Mails im Internet zwischen einem Umschlag (engl. **envelope**) und der eigentlichen Nachricht unterschieden. Der Umschlag wird durch die Schritte 2.(a) und 2.(b) erstellt. Für den Mailserver ist nur der Umschlag für die Zustellung der E-Mail von Bedeutung.

Die Nachricht selbst besteht auch aus zwei Teilen. Alle Zeilen vor der ersten Leerzeile sind der Nachrichtenkopf (engl. **message header**), die Zeilen dahinter der Nachrichtenrumpf (engl. **message body**). Im Nachrichtenkopf werden Absender- und Empfängerangaben wiederholt. Diese sollten mit den Angaben auf dem Umschlag übereinstimmen, müssen es aber nicht. Unverlangt zugeschickte Werbe-E-Mails (engl. **spam**) haben im Nachrichtenkopf häufig gefälschte Einträge.

Weiterhin findet die komplette Kommunikation zwischen Mailclient und Mailserver *unverschlüsselt* statt. Jedermann auf dem Weg der Nachricht kann mitlesen und dadurch erfahren, mit wem und worüber Sie kommunizieren.

Bei der Weiterleitung einer E-Mail trägt jeder der beteiligten Mailserver in den Nachrichtenkopf eine Zeile `Received:` ein. Darin wird festgehalten, dass und wann diese E-Mail den Server erreicht hat. Damit kann bei Zustellungsproblemen der Weg einer E-Mail nachvollzogen werden. In den Nachrichtenkopf kann aber auch der Absender Informationen eintragen, beispielsweise auch eine Zeile `Received`. Damit kann der Weg der E-Mail verschleiert werden.

Empfang

Das gesamte E-Mail-System ist auch darauf eingerichtet, dass ein Empfänger nicht permanent online ist. In diesem Fall wird eine E-Mail auf dem letzten am Weg liegenden Mailserver gespeichert. Sie können mit Ihrem Mailclient dann direkt diesen Server kontaktieren, um angekommene E-Mails auf den eigenen Computer zu laden und neue E-Mails vom Computer an den Mailserver zu übertragen.

POP3
IMAP4

Diese Kommunikation ist im *Post-Office-Protocol Version 3* (*POP3*-Protokoll) bzw. auch im neuen *IMAP4*-Protokoll geregelt. Die Idee dieser Protokolle ist, dass der Client eine Verbindung zum Mailserver aufbaut. Im POP3-Protokoll schickt der Client dann einen Benutzernamen und ein Passwort an den Mailserver. Beides wird im Klartext über das Netz übertragen. Anschließend kann der Client die angekommenen E-Mails auf seinen Computer übertragen, auf dem Server dann löschen und neue E-Mails vom Computer zum Server schicken.

Da E-Mails sehr häufig mit POP3 abgeholt werden, unterstützen moderne E-Mail-Server auch durch SSL verschlüsselte Verbindungen. SSL wird in Abschnitt 3.3.1 genauer behandelt.

1.4 Konkrete Gefahren

In diesem Abschnitt lernen Sie einige der konkreten Bedrohungen kennen, denen Ihr Computer heutzutage ausgesetzt ist. Dazu gehören die auch in den Nachrichten immer öfter vorkommenden *Viren*, die sogenannten *Trojanischen Pferde* und der Missbrauch von Passwörtern. Sie werden in den folgenden Unterabschnitten vorgestellt.

Insgesamt ist dies allerdings nur ein erster Einblick. Vertiefende Informationen zu weiteren Angriffsmöglichkeiten auf Computer werden in Kurs *(01867) Sicherheit im Internet 2* behandelt.

1.4.1 Viren

Prinzip: Viren sind Computerprogramme, die sich selbst kopieren (vervielfältigen) können und sich auf diesem Weg vermehren. Neben dieser Grundfunktion enthalten Viren auch andere Funktionen. Diese anderen Funktionen können unterschiedlichen Zwecken dienen:

- Sie können Schäden aller Art auf Ihrem Computer anrichten, indem sie Dateien löschen oder deren Inhalt verändern.
- Viren können versuchen, sich selbst zu tarnen und zu verstecken.
- Die zusätzlichen Funktionen können aber auch „nur“ den Benutzer durch seltsame Ausgaben (Bildschirmmeldungen oder akustische Signale) verunsichern und erschrecken.

Schäden: Ein Virus ist für denjenigen, der ihn auf dem Computer hat, immer eine unangenehme Sache. Man kann eigentlich nie genau wissen, was ein Virus letztlich machen wird. Um das herauszufinden, müssen menschliche Experten das Virus genau analysieren. Aus der Theoretischen Informatik ist bekannt, dass Computerprogramme im Allgemeinen nicht einmal herausfinden können, ob ein gegebenes Programm bei einer gegebenen Eingabe terminiert (Unentscheidbarkeit des Halte-Problems). Es besteht also wenig Hoffnung, dass Virenscanner jemals genau herausfinden können, welchen Schaden ein Virus möglicherweise anrichten kann. Ein Benutzer hat also keine andere Wahl, als das Virus aus seinem System zu entfernen. Dies kostet auf jeden Fall Zeit und Mühe.

Schadensumfang

Von Viren verursachte Schäden können unterschiedlich groß sein. Im einfachsten Fall löst das Virus ein akustisches Signal bei jedem Tastendruck aus, falls gerade der 18. eines Monats ist. Ein größerer Schaden entsteht, wenn das Virus versucht Programmdateien zu löschen, die Sie ausführen möchten. Da diese Schadensfunktion für den Benutzer offensichtlich ist, kann man das Virus entfernen und die gelöschten Programme wieder installieren. Natürlich sind hierzu die originalen Datenträger oder die hoffentlich regelmäßig angelegten Sicherungskopien (engl. **backup**) erforderlich.

Zu den schlimmsten Schäden gehört es, wenn ein Virus unbemerkt Daten von Ihrem Computer liest und diese dann per E-Mail oder verpackt in HTTP-Requests verschickt. Sollte das Virus Ihre Benutzerkennung (engl. **account**) oder Ihr Passwort (engl. **password**) ausspioniert haben, kann sich ein Dritter auf Ihrem Computer anmelden und dann beliebigen Schaden anrichten.

Virentypen: Zu den ersten Viren überhaupt gehörten die sogenannten **Bootsektor-Viren (BSV)**. Wenn ein Computer eingeschaltet wird, dann muss als erstes das Betriebssystem geladen werden. Damit man nun auf einem Computer mehrere Betriebssysteme benutzen und ein Betriebssystem einfach aktualisieren (engl. **to update**) kann, ist das Betriebssystem i. d. R. nicht im *ROM (Read Only Memory)* vorhanden. Stattdessen enthält das ROM ein kleines Programm (Urlader), das die Aufgabe hat das Betriebssystem zu laden. Der Urlader sucht auf fest vorgegebenen Speichermedien nach dem eigentlichen Ladeprogramm (engl. **loader**). Welche Speichermedien das genau sind (Diskette, DVD, USB-Gerät, Festplatte) kann man im sogenannten Basic Input Output

Bootsektor-Viren (BSV)

System (BIOS) des Computers einstellen. Eine moderne Variante des BIOS heißt **Unified Extensible Firmware Interface (UEFI)**.

Unified Extensible Firmware Interface (UEFI)

Auf allen o. g. Speichermedien gibt es eine fest vorgegebene Stelle, an der ein Betriebssystem-Ladeprogramm stehen kann. Auf einer Festplatte nennt man diese Stelle **Master-Boot-Record (MBR)**. Bei Disketten (engl. **floppy disk**) ist das der erste Sektor auf der Diskette, bei Festplatten (engl. **hard disk**) ist es Sektor 1 auf Zylinder 0, Kopf 0.

Master-Boot-Record (MBR)

Der Master-Boot-Record einer Festplatte enthält weiterhin Informationen über die Aufteilung der Festplatte (engl. **partition table**). Das Ladeprogramm aus dem MBR startet nun das Betriebssystem, das auf einer der Partitionen gespeichert ist (siehe Abbildung 1.16). Falls mehrere Betriebssysteme auf der Festplatte installiert sind, bietet das Ladeprogramm eine Auswahl an und der Benutzer kann entscheiden, welches Betriebssystem gestartet werden soll. Zu Linux gehört das Ladeprogramm GRUB (GRand Unified Bootloader), es erlaubt das Laden unterschiedlicher Betriebssysteme.

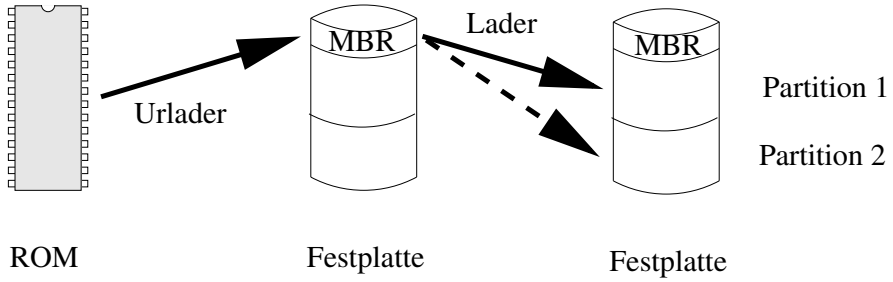

Abbildung 1.16: Bootvorgang bei einem Computer

Abbildung 1.16 zeigt den prinzipiellen Ablauf beim Starten eines Computers. In einem PC ist der Urlader so implementiert, dass er zunächst auf dem Diskettenlaufwerk nach einem Lader im Bootsektor sucht, anschließend auf der Festplatte und evtl. anschließend auch noch auf dem CD-ROM-Laufwerk. Im BIOS des PC kann man diese Reihenfolge evtl. auch ändern.

Wie kann nun ein PC mit einem Bootsektor-Virus infiziert werden? Am Anfang steht i. d. R. eine infizierte Diskette, eine infizierte CD/DVD oder ein infizierter USB-Stick, die irgendwelche Daten enthalten. Sie legen die Diskette/CD/DVD ein (schließen den USB-Stick an), kopieren die Daten, arbeiten mit dem Computer, vergessen die Diskette/CD/DVD bzw. den USB-Stick und schalten am Ende den Computer wieder aus. Bis jetzt ist noch nichts passiert. Beim nächsten Einschalten startet der Urlader nicht den Lader aus dem MBR der Festplatte, sondern den infizierten Lader aus dem Bootsektor der Diskette/CD/DVD bzw. dem USB-Stick. Das Virus führt dann die folgenden Schritte aus:

1. Es lädt sich in den Hauptspeicher und trägt sich in den Interrupt 13h (disk read/write) ein.

2. Es kopiert den Original-MBR der Festplatte an einen anderen freien Platz auf der Platte.

3. Es kopiert sich selbst in den MBR der Festplatte.

4. Es startet den Lader aus dem „Original“-MBR der Festplatte.

Für den Benutzer sieht es so aus, als starte der Computer wie immer. Schließt man nun einen neuen USB-Stick an und versucht den Inhalt zu lesen oder zu schreiben, so kommt das Virus ins Spiel. Zum Lesen oder Schreiben ruft das Programm das Betriebssystem auf (siehe auch Kurs *(01801) Betriebssysteme und Rechnernetze*) und das Betriebssystem löst den Interrupt aus, womit das Virus aktiviert ist. Das Virus überprüft, ob der USB-Stick bereits infiziert ist. Ist er es noch nicht, dann kopiert sich das Virus in den Bootsektor des USB-Sticks. Auf diese Art verbreitet sich das Virus.

Da das Virus gestartet wird bevor das Betriebssystem geladen ist, können die Schutzfunktionen des Betriebssystems nicht wirksam werden. Ein Virentest-Programm kann das Virus auch nachträglich nicht mehr erkennen. Dazu müsste es den Master-Boot-Record der Festplatte lesen. Dieser Lesezugriff läuft aber letztlich wieder über den Interrupt, in den sich das Virus „eingeklinkt“ hat. Das Virus liefert bei einer Leseanfrage also einfach den kopierten „Original“-MBR zurück. Viren, die sich auf diese oder eine andere Art tarnen, nennt man auch **Stealth-Virus**. Bei der Übertragung auf einen anderen Rechner schützt sich das Beagle-Virus beispielsweise dadurch, dass es sich nur verschlüsselt überträgt. Virenscanner auf dem Übertragungsweg können die Nachricht dann nicht entschlüsseln und das Virus erkennen. Erst der Empfänger entschlüsselt die Nachricht und startet möglicherweise das Virus.

Stealth-Virus

Ein PC wird also nur durch das Starten von einem infizierten Bootmedium mit einem Bootsektor-Virus infiziert. Um das Virus zu entdecken und zu entfernen, brauchen Sie eine *garantiert* virenfreie Bootdiskette. Diese enthält die entsprechenden Hilfsprogramme, die einen virenfreien MBR rekonstruieren und auf die Festplatte schreiben können. Von Klaus Knopper gibt es eine Linux-Distribution, genannt Knoppix, die man direkt von CD/DVD booten kann. Darauf basiert eine CD/DVD des *Heise Zeitschriften Verlags*, die Antivirenprogramme enthält. Bootet man dieses System und ist der Rechner ans Internet angeschlossen, dann bestehen gute Chancen mit Hilfe der aktuellen Virensignaturen die Schädlinge wieder zu entfernen.

Knoppix

Der sicherste Weg ein kompromittiertes System wieder in einen sauberen Zustand zu versetzen ist allerdings die komplette Neuinstallation mit einem „sauberen“ Installationsmedium. Nur mit aktuellen Sicherungskopien gelingt die Wiederherstellung ohne Datenverlust.

Den zweiten Virustyp bilden die sogenannten **Dateiviren**. Ein Dateivirus infiziert einzelne Dateien, vorzugsweise Programmdateien (also `.exe`- und `.com`-Dateien unter DOS). Startet ein Benutzer ein infiziertes Programm, so lädt sich das Virus in den Hauptspeicher und versucht dort sesshaft (resident) zu werden. Das heißt, dass auch nach der Beendigung des Wirtprogramms das Virus weiterhin im Speicher bleibt und weiter läuft. Startet der Benutzer weitere Programme, so versucht das Virus auch diese Programme zu infizieren. Dazu verändert das Virus die Programmdatei, beispielsweise indem der Viruscode an das Ende der Datei angehängt wird. An den Anfang der Programmdatei fügt das Virus dann auch einen Aufruf des Viruscodes am Ende der Datei ein. Siehe dazu auch Abbildung 1.17.

Dateiviren

Dabei verändert sich die Dateigröße des Programms und man kann daran die Infektion erkennen. Ein cleveres Virus versucht diese Veränderung zu verbergen. Dazu kann es beispielsweise in der Programmdatei einen Block zusammenhängender Nullen suchen und sich dort hinein kopieren. Beachten Sie, dass Abbildung 1.17 *nicht* maßstabsgetreu ist. Der Viruscode ist deutlich

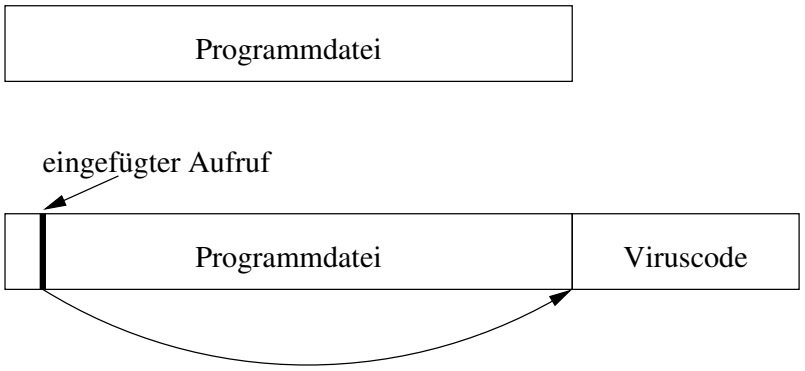

Abbildung 1.17: Infektion einer Programmdatei

kleiner als der Programmcode. Eine passende Folge von Nullen zu finden, ist also gar nicht so unwahrscheinlich. Beim Start des Programms wird dieses in den Hauptspeicher geladen und der Viruscode gestartet. Neben den oben bereits genannten Schritten überschreibt sich der Viruscode im geladenen Programm wieder mit Nullen, damit das Programm wie üblich funktioniert. Ein Dateivirus könnte nicht nur andere Programmdateien infizieren, sondern es könnte auch versuchen, sich selbst in den Bootsektor zu kopieren.

Wenn ein Dateivirus gestartet wird, dann ist das Betriebssystem bereits geladen und das Virus muss die Schutzmechanismen des Betriebssystems überwinden. Das Virus läuft mit den Berechtigungen des Benutzers, der das Virus (unabsichtlich oder absichtlich) gestartet hat. Normale Benutzer dürfen i. d. R. aber keine Änderungen am System, bzw. den Systemdateien vornehmen. Normale Benutzer haben nur das Ausführungsrecht (engl. **execute permission**) an diesen Dateien, aber kein Schreibrecht.

Da aber auch normale Benutzer häufig Sicherheitsaktualisierungen (engl. **security updates**) ihres Betriebssystems installieren müssen, können sie hierzu Administratorrechte anfordern. Dazu müssen die Benutzer in einem speziellen Dialog noch einmal ein Administrator-Passwort eingeben. Ein Virus kann auch diesen Dialog anzeigen und den Benutzer dazu verleiten, das Administrator-Passwort einzugeben. Dazu kann das Virus z. B. anzeigen „Programm XY möchte ein wichtiges Security Update installieren. Bitte Administrator-Passwort eingeben“. Viele Benutzer machen das dann auch und schon hat das Virus entsprechende Berechtigungen erlangt.

Vorhandene Dateiviren werden entfernt, indem man den Computer von einem garantiert virenfreien Medium startet und dann alle infizierten Programme wieder neu installiert. Im Zweifelsfall müssen alle Programme neu installiert werden.

Ein Dateivirus kann man durch den Start eines infizierten Programms auf dem eigenen Computer bekommen. Das infizierte Programm kann über ein USB-Gerät oder auch über das Netz (per E-Mail oder auch durch einfaches Surfen im Internet) auf den eigenen Computer kommen. Computer, die als Server für Programmdateien in einem Netz arbeiten, sind besonders gefährdet.

Makroviren

Den dritten Virentyp bilden die sogenannten **Makroviren**. Sie sind im Prinzip auch Dateiviren, denn sie befallen Dateien. Während „klassische“ Dateiviren i. d. R. Programmdateien infizieren, stecken Makroviren in Dokumentdateien, wie beispielsweise Textverarbeitungsdokumenten, Tabellenkalkulationsdokumenten oder Präsentationsdokumenten. Die zugehörigen Programme enthalten

Funktionen zum Erstellen von Makros. Makros sind kleine Programme, die dem Anwender die Arbeit erleichtern sollen. Zuerst bestand die Funktion von Makros darin, die vom Benutzer gedrückten Tasten zu speichern und diese Folge später wiederholt und automatisch abzuspielen. Moderne Anwendungsprogramme bieten spezielle Makrosprachen, die auf Programmiersprachen wie C oder BASIC basieren. Die Office-Programme der Firma *Microsoft* bieten beispielsweise die Sprache *VBA (Visual Basic for Applications)* zur Entwicklung von Makros an. Diese Makrosprachen erlauben nicht nur das Aufzeichnen von Tastendrücken, sondern auch komplexe Kontrollstrukturen (Sequenz, Auswahl, Wiederholung und Prozeduraufrufe) und den Zugriff auf Dateien.

Definition: Makro

Makros werden zusammen mit dem Dokumentinhalt in *einer* Datei gespeichert. Öffnet ein Benutzer diese Datei, so werden der Inhalt und der Makrocode geladen.

Inzwischen verwenden alle Office-Programme von *Microsoft* dieselbe Makrosprache. Makroviren können daher nicht nur in Textdokumenten, sondern auch in Tabellenkalkulationsdokumenten oder auch Präsentationen stecken. Im Gegensatz zu Bootsektor- oder „klassischen“ Dateiviren, die i. d. R. in Assembler oder einer anderen Low-Level-Programmiersprache geschrieben werden, werden Makroviren in einer einfacher zu lernenden höheren Programmiersprache geschrieben. Jedes Dokument, das Sie heute aus dem Internet laden, per E-Mail zugeschickt bekommen oder von einer Diskette/CD-ROM kopieren, könnte ein Makrovirus enthalten. Zum Schutz vor Makroviren bieten sich folgende Möglichkeiten:

Schutz vor Makroviren

1. Installation eines Virenscanners, der auch Makroviren erkennen und entfernen kann. Da immer wieder neue Makroviren auftauchen, reicht es nicht, den Scanner einmal zu installieren. Man muss den Virenscanner auch regelmäßig aktualisieren. Weitere Hinweise hierzu folgen in Abschnitt 3.5.

   Virenscanner einsetzen

2. Von *Microsoft* sind inzwischen auch Versionen ihrer Office-Programme verfügbar, die ausschließlich Dokumente anzeigen können. Sie können die Dokumente damit nicht bearbeiten. Weiterhin können diese Programme auch keine Makros ausführen. Ein Makrovirus kann in diesen Programmen also nicht aktiv werden.

   Reader-Programm benutzen

3. Wenn das Dokument von Ihnen nicht weiter bearbeitet werden soll, dann ist ein Dokumentformat, wie das *Portable-Document-Format (PDF)* von *Adobe*, die bessere Wahl.

4. Seit etwa 2003 können Makros auch digital signiert sein (vergleiche Abschnitt 2.6.2). Durch Prüfen der Signatur kann der Benutzer verifizieren, von wem das Makro stammt und dass es nicht verändert wurde. So kann man Makros von vertrauenswürdigen Personen zulassen, während Makros von anderen Autoren nicht ausgeführt werden. Konfigurieren Sie Ihre Office-Installation entsprechend.

Makroviren sind recht einfach zu schreiben und haben vielfältige Ausbreitungsmöglichkeiten. Sie sind unabhängig vom Betriebssystem und brauchen nur ihre zugehörige „Wirtsanwendung“. Vernünftigen Schutz vor Makroviren bieten derzeit Virenscanner, die a) regelmäßig aktualisiert werden und b) *alle* Dateien auf Viren überprüfen, sowie die sichere Konfiguration der Office-Programme, so dass unbekannte Makros nicht ausgeführt werden.

1.4.2 Würmer

Wirtsprogramme

Während Viren sich mit Hilfe anderer Programme, sog. **Wirtsprogramme**, verbreiten, verbreiten sich Würmer eigenständig. Sie müssen also vom Benutzer mindestens einmal explizit gestartet werden. Dann führen sie ihre Schadensfunktion aus und verbreiten sich weiter. Die Schadensfunktion kann nun auch darin bestehen, dass der Wurm dafür sorgt, dass er später automatisch immer wieder gestartet wird. Unter Microsoft Windows kann er sich in den Autostart-Ordner kopieren oder die Registry verändern.

Typische Verbreitungswege für Würmer sind E-Mails oder HTTP. Dazu enthält der Wurm beispielsweise seinen eigenen SMTP- oder HTTP-Server. Aber auch andere Verbreitungswege sind denkbar. In einem Microsoft Windows-Netz kann ein Wurm auch nach freigegebenen Laufwerken suchen und sich dort unter einem Tarnnamen installieren. Die Tarnnamen sind so gewählt, dass ein Benutzer dazu verleitet wird, einen Doppelklick auf die Datei auszuführen. Obwohl eigentlich ein Programm, gibt sich der Wurm dann auch einmal den Namen eines Bildes. Bei bestimmten Schwachstellen kann sich ein Wurm auch ganz ohne Unterstützung des Benutzers verbreiten. Der *Conficker*-Wurm ist so ein Beispiel, da er eine Schwachstelle im RPC-Dienst ausnutzte.

1.4.3 Trojanische Pferde

Ein Programm, das neben seiner eigentlichen Funktion auch weitere Funktionen ausführt, nennt man *Trojanisches Pferd*, falls die weiteren Funktionen dem Benutzer *nicht* bekannt sind und er deren Ausführung auch nicht bemerkt. Normalerweise stellt diese weitere Funktion für den Benutzer eine Gefahr dar. Ein Trojanisches Pferd kann in jedem Programm stecken, in einem Textverarbeitungssystem, einem Editor, einem Bildschirmschoner oder kleinen Hilfsprogrammen zur Datei- oder Netzwerkverwaltung. Aber auch im Betriebssystem oder der Systemsoftware können unbekannte und nicht dokumentierte Funktionen stecken.

Im Internet werden sehr viele Programme von den unterschiedlichsten Autoren für die unterschiedlichsten Zwecke angeboten. Seien Sie vorsichtig, wenn sie unbekannte Programme aus dem Netz laden und bei sich installieren wollen. Gerade wenn diese Programme nur in Binärform, also für Menschen unleserlich, verteilt werden, können sie über die Funktionen des Programms nur spekulieren. Anfang 2009 wurde beispielsweise über das BitTorrent-Netz eine trojanisierte Version der Bürosoftware *iWork2009* der Firma *Apple* verbreitet. Das Installationsprogramm hat neben der Bürosoftware auch einen Trojaner installiert.

Ein weiteres beliebtes Gebiet für Trojaner sind Multimedia-CODECs. Ein CODEC ist eine Software, die Multimediadaten wie Sound oder Filme in ein digitales Format wie MP3 für Musik oder MPEG für Filme COdiert oder DECodiert. Neben den beiden genannten Formaten gibt es auch viele andere Formate. Multimediasoftware ist daher modular programmiert, d. h. sie kann einfach durch Installation zusätzlicher CODECs erweitert werden und dann auch neue Formate abspielen. Auf einigen Internetseiten werden nun Multimediadaten angeboten, für die der Benutzer einen zusätzlichen CODEC aus dem Netz laden und installieren soll. Hier besteht die Gefahr, dass man neben einem CODEC auch weitere unerwünschte Funktionen installiert.

Dieses modulare Konzept findet sich auch bei den Webbrowsern wieder.

Sie können durch sogenannte Plug-Ins in ihren Funktionen erweitern werden. Ob ein Plug-In aber neben der offensichtlichen Funktion nicht auch die Benutzereingaben beim Internetbanking irgendwohin weiterleitet kann ein normaler Benutzer kaum und selbst Experten nur schwierig herausfinden.

Da man einem Programm nicht per se ansehen kann, was es letztlich machen wird[8], kann man nicht auf die automatische Entdeckung von trojanischen Pferden hoffen. Man kann als Benutzer nur versuchen, Veränderungen im System zu erkennen. Dazu erstellt man sich eine Datenbank, die zu allen installierten Programmen Informationen wie

- das Datum der letzten Änderung,
- die Größe der Programmdatei und
- spezielle Prüfsummen (Hash-Codes) der Programmdatei

enthält. Diese Daten werden dann regelmäßig geprüft. Stellt man Veränderungen fest[9], so müssen die veränderten Programme überprüft und gegebenenfalls neu installiert werden.

Trojanische Pferde finden sich nicht nur in Programmdateien, sondern unter Umständen auch in Dokumentdateien. Diese enthalten neben dem eigentlichen Dokumentinhalt weitere Informationen, die man als Anwender dort nicht vermutet. Dateien von *Microsoft Office* beispielsweise sind in einem codierten Format abgespeichert, das nur *Microsoft* selbst kennt. Neben dem jeweiligen Dokumentinhalt enthalten sie auch Informationen wie die Programmversion, mit der das Dokument erstellt wurde, den Pfadnamen, unter dem das Dokument auf der Festplatte gespeichert ist, Namen der Ersteller und Bearbeiter usw. Als Benutzer kann man nicht ausschließen, dass nicht auch Informationen über andere installierte Programme oder sonstige vertrauliche Daten in einem unverfänglichen Textdokument stecken. Verschickt man solche Dateien per E-Mail, so gibt man u. U. auch Informationen preis, die man eigentlich nicht preisgeben möchte.

Ab der Version *Microsoft Office 2007* gibt es auch das Werkzeug *Document Inspector*. Es ist dazu gedacht, vertrauliche und versteckte Informationen aus einem Dokument zu entfernen. Zu diesen Informationen gehören (1) persönliche Daten (z. B. Autor) und allgemeine Dokumenteigenschaften, (2) Kommentare und Informationen zur Versionsgeschichte, (3) Kopf- und Fußzeilen oder sog. Watermarks, (4) versteckter Text, versteckte Zeilen/Spalten in Tabellen, (5) unsichtbare Inhalte oder (6) Präsentationsnotizen.

Fazit: Viren, Würmer und Trojanische Pferde sind Beispiele für Schadsoftware (engl. **malware**). Aktuelle Schadsoftware kombiniert die Techniken der Viren, Würmer und Trojaner und ergänzt diese Techniken um automatische Aktualisierungsfunktionen. Dadurch kann die Schadsoftware weitere Komponenten nachladen und dadurch auch mutieren. Die Häufigkeit der unterschiedlichen Schadprogrammtypen zu zählen ist daher nicht mehr sinnvoll. Man kann jedoch davon ausgehen, dass die Zahl der Schadprogramme kontinuierlich steigen wird.

[8] Sie erinnern sich: Das Halte-Problem ist *nicht* entscheidbar!

[9] Ein Angreifer kann zwar das Datum der letzten Änderung oder die Dateigröße manipulieren, bei guten Prüfsummen kann der Angreifer die Veränderungen aber nicht mehr so einfach tarnen.

1.4.4 Passwortmissbrauch

Authentifikation: Passwörter (auch Kennwörter genannt) werden zur Authentifikation von Benutzern eingesetzt. Bei der Authentifikation geht es um die Prüfung, ob der Benutzer tatsächlich derjenige ist, der er vorgibt zu sein. Im Prinzip gibt es für diese Prüfung mehrere Möglichkeiten:

Biometrie
1. Überprüfen eines unverwechselbaren und schwer zu fälschenden biometrischen Merkmals, wie beispielsweise eines Fingerabdrucks. Dieses Thema wird in Kurs *(01868) Sicherheit im Internet 1 – Ergänzungen* vertieft.

Besitz
2. Kontrolle eines schwer zu fälschenden Gegenstandes, wie beispielsweise eines Personalausweises. Auf den neuen Personalausweis wird in Abschnitt 2.7.4 eingegangen.

Wissen
3. Überprüfung, ob die Person eine bestimmte Information, wie beispielsweise eine Geheimnummer, kennt.

Bei der Benutzer-Authentifikation an einem Geldautomaten werden die Merkmale *Besitz* und *Wissen* geprüft. Der Benutzer muss eine gültige EC-Karte besitzen und eine Geheimzahl (genannt PIN) kennen. Nur wenn beide Merkmale erfolgreich geprüft wurden, ist die Authentifikation gelungen.

Bei der Authentifikation gegenüber einem Computer greift man heute i. d. R. nur auf die Kontrolle des Merkmals *Wissen* zurück. Als Benutzer kann bzw.

Sicherheitsproblem

muss man ein Passwort auswählen. Das Problem mit Passwörtern ist, dass *jeder*, der das Passwort von Benutzer *xy* kennt, sich selbst als Benutzer *xy* ausgeben kann. Von einem fremden Passwort kann man auf verschiedenen Wegen erfahren:

Raten: Man kann beispielsweise prüfen, ob der Benutzer seinen Benutzernamen auch als Passwort eingetragen hat. Vielleicht hat ein Benutzer auch seine Telefonnummer, den Namen des Partners, der Kinder, der Eltern, sein Autokennzeichen oder eine andere öffentlich bekannte Information über sich selbst als Passwort verwendet.

Ausprobieren: Man kann Computer auch so programmieren, dass sie nacheinander alle Wörter eines Wörterbuchs, systematisch ausprobieren. Heute gibt es nicht nur typische Wörterbücher im Netz, sondern auch Listen mit bisher ausgespähten Passwörtern. Die Programme prüfen dann alle Wörter (klassische Wörter ebenso wie bereits bekannte Passwörter) und spezielle Modifikationen davon. Solche Modifikationen können das Anhängen von Ziffern(folgen) oder das Ersetzen einzelner Zeichen des Wortes (z. B. den Buchstaben l durch die Ziffer 1 ersetzen) sein.

Noch allgemeiner kann man hergehen und *alle* Zeichenketten über dem Alphabet der zulässigen Passwortzeichen generieren und als Passwort ausprobieren. Für ein Passwort der Länge n, bei dem jedes Zeichen aus einem Vorrat von x Zeichen stammen darf, gibt es jedoch x^n Möglichkeiten.

Ausspähen/Abhören: Bei der Eingabe des Passworts kann man beobachtet werden. Insbesondere bei Passwörtern, die man selbst nicht ändern kann oder darf, ist es besonders wichtig, bei der Eingabe nicht beobachtet zu werden. Falls ein unbeobachtet eingegebenes Passwort erst im Klartext

über ein Netz an einen entfernten Computer zur Prüfung geschickt wird, kann das Passwort auf diesem Weg ausgespäht und kopiert werden.

In diese Kategorie fallen auch alle Versuche von Hackern, mittels gefälschter Nachrichten jemanden zur Preisgabe eines Passwortes zu verleiten. Im Internet wird das häufig auch *Phishing* genannt. Die Nachrichten können Telefonanrufe oder aktuell auch E-Mails sein. Ein Angreifer gibt darin vor, ein Administrator von einer dem Benutzer bekannten Institution zu sein. Der Benutzer soll am Telefon sein Passwort dann direkt nennen.

Phishing

Alternativ wird in einer E-Mail behauptet, dass technische Probleme bei einem Dienstanbieter im Internet aufgetreten seien und der Benutzer seine Daten zur Aktualisierung bzw. Überprüfung erneut eingeben müsse. Dazu steht dann häufig eine lange URL in der Nachricht, die der Benutzer anklicken soll. Der Anbieter könnte dabei die eigene Bank, *eBay*, *Amazon* usw. sein. Jeder Anbieter, bei dem man eine Benutzerkennung mit Passwort besitzt, kann betroffen sein. Klickt man nun die URL an, dann erscheint eine Seite, die so aussieht als wäre sie vom Anbieter. In Wirklichkeit stammt sie vom Angreifer und dient nur dem Zweck, Geheimnisse des Benutzers (Passwörter, Kreditkartennummern, etc.) zu erlangen.

Man sollte sich also immer *wirklich* sicher sein, mit welchem Rechner man verbunden ist, wenn man im Internet surft. Hinweise, wie man das macht, werden in Abschnitt 3.3 vorgestellt.

Passwörter „knacken“: Beim Thema „Ausprobieren“ ist zu beachten, dass ein Hacker sich heute gar nicht mehr die Arbeit machen muss, ein eigenes Programm zum Passwörtersuchen oder -ausprobieren zu schreiben. Solche Programme kann man direkt aus dem Internet laden. Im Folgenden soll auf die prinzipielle Arbeitsweise solcher Programme etwas genauer eingegangen werden.

Dazu zunächst einige Hintergrundinformationen, wie Passwörter sinnvoll behandelt werden. Zunächst sollten Passwörter *niemals* im Klartext gespeichert werden. Es besteht immer die Gefahr, dass eine Liste mit Benutzerkennungen und den zugehörigen Passwörtern gestohlen wird oder anders abhanden kommt. Statt dessen wendet man eine nicht umkehrbare Funktion f auf das Passwort x an speichert nur den Funktionswert $f(x)$ als Passwort des Benutzers. Da f nicht umkehrbar ist, kann man aus $f(x)$ nicht zurück auf x schließen. Daneben sollte f weitere Eigenschaften besitzen:

Nie im Klartext speichern

Möglichst keine Kollisionen: Die Funktion f soll für zwei unterschiedliche Eingaben $x \neq x'$ auch unterschiedliche Funktionswerte berechnen, also $f(x) \neq f(x')$. In Abschnitt 2.5.1 wird diese Eigenschaft genauer besprochen.

Großer Wertebereich: Als Funktionswerte sollen sehr viele verschiedene Werte (z. B. mehr als 2^{256}) möglich sein.

Nicht zu schnell berechenbar: Da Angreifer versuchen werden, vorab Tabellen mit allen möglichen Paaren $(x, f(x))$ zu berechnen, sollte f nicht zu schnell berechenbar sein. Bei der Überprüfung eines eingegebenen Passworts darf die Berechnung eine halbe Sekunde dauern, ohne dass das den Benutzer merklich stört. Ein Angreifer kann dann aber auch nur zwei Passwörter pro Sekunde testen und nicht zwei Millionen.

Die in Abschnitt 2.5 vorgestellten Hash-Funktionen sind potentielle Kandidaten für f. Zusammengefasst kann man also sagen: Systeme speichern nicht die Passwörter selbst, sondern sogenannte *Passwort-Hashes*.

Passwort-Hashes

Bei der Überprüfung eines eingegebenen Passworts x' wird $f(x')$ berechnet und mit dem gespeicherten $f(x)$ verglichen. Sind die Funktionswerte gleich, dann war $x' = x$. Schafft es ein Angreifer, an die Liste mit Benutzerkennungen und den Funktionswerten der Passwörter zu kommen, so kann er die Passwörter der Benutzer trotzdem nicht rekonstruieren. Dieses Verfahren hat allerdings Nachteile:

Nachteile

- Wählen zwei Benutzer zufällig dasselbe Passwort, so steht bei beiden Benutzern derselbe Funktionswert in der Liste.
- Ein Angreifer kennt normalerweise die Funktion f. Er kann sich vorab (also offline) eine Liste mit Passwörtern und den zugehörigen Funktionswerten berechnen. Nun muss er eine „besorgte“ Liste mit (Kennung, Funktionswert) Paaren „nur“ noch mit seiner Liste vergleichen.

Die vom Angreifer zu erstellende Liste wird natürlich sehr groß. Sie enthält so viele Einträge wie es Passwörter, bzw. wie es Funktionswerte gibt. Gehen wir davon aus, dass es 2^n viele Funktionswerte gibt. Ohne Tabelle braucht ein Angreifer dann $O(2^n)$ Zeit für das „knacken“ des Funktionswerts. Er probiert einfach jedes mögliche Passwort aus. Man nennt das auch den *Brute Force* Ansatz.

Hat er diesen Aufwand vorab schon getrieben und ein sortierte Tabelle der Größe $O(2^n)$ berechnet, so braucht er $O(2^n)$ viel Speicherplatz für die Tabelle (und $O(2^n)$ viel Zeit für die Erstellung), kann aber anschließend einfach in $O(n)$ Zeit den passenden Wert finden. Dazu benutzt man binäre Suche. Folgende Tabelle zeigt den Platz- und Zeitbedarf dieser beiden Angriffsvarianten:

Variante	Zeitbedarf Vorbereitung	Platzbedarf	Zeitbedarf „Knacken“
Brute Force	$O(1)$	$O(1)$	$O(2^n)$
Komplette Tabelle	$O(2^n)$	$O(2^n)$	$O(n)$
Hellman (s. u.)	$O(2^n)$	$O(2^{n/2})$	$O(2^{n/2})$

Eine komplette Tabelle vorab zu berechnen erfordert zwar sehr viel Aufwand, ist aber machbar. Allerdings ist die erforderliche Menge an Speicherplatz noch zu groß für die Praxis.

Hellman [Hel80] hat ein Verfahren vorgestellt, mit dem man einen Mittelweg gehen kann. Die Idee dabei ist, Platzbedarf gegen Zeitbedarf zu tauschen. Hat man mehr Speicherplatz, so braucht man weniger Rechenzeit. Hellmans Idee besteht in der Vorabberechnung einer Tabelle der Größe $O(2^{n/2})$ mit einem Zeitaufwand der Größe $O(2^n)$[10]. Der anschließende Zeitbedarf für die Prüfung beträgt dann auch $O(2^{n/2})$. „Spendiert man also zusätzlichen Speicherplatz, so reduziert man den erforderlichen Rechenaufwand.“

time-memory trade-off

Die Idee dieses Prinzips ist wie folgt: Man wählt eine Liste von Startpasswörtern S_1, S_2 bis S_k. Für jedes S_i iteriert man nun t mal die Funktion f, d. h.

[10] Das kann in der Praxis durch parallele Nutzung von Rechenzeit auf sehr vielen PCs, die ansonsten kaum ausgelastet wären, erreicht werden. Oder man benutzt die vielen Recheneinheiten die auf modernen Grafikkarten vorhanden sind für diese Berechnungen.

man berechnet $S_{i,j} = f^j(S_i)$. Dabei ist $f^j(x)$ mit $j \leq t$ so definiert:

$$f^j(x) = \begin{cases} f(x) & \text{falls } j = 1 \\ f(f^{j-1}(x)) & \text{falls } j > 1 \end{cases}$$

Wenn $t = k = 2^n$ gilt, dann kann man eine Tabelle mit $2^{n/2}$ vielen Einträgen der Art $(S_i, f^t(S_i))$ erstellen. Man speichert also nur das Startpasswort und den zugehörigen Endwert der Berechnungskette von f.

Abbildung 1.18 zeigt die Berechnungsketten. Tatsächlich gespeichert werden also nur die erste Spalte und die letzte Spalte der Tabelle. Die Größe der gespeicherten Daten ist also $O(k)$. Würde man die komplette Matrix speichern wäre der Platzbedarf $O(k \cdot t)$ und bei $k = t = 2^{n/2}$ hätten wir exponentiellen Platzbedarf $O(2^n)$. Hat man nun einen Funktionswert $y = f(x)$ gegeben und

$$\begin{array}{l} S_1 \xrightarrow{f} S_{1,1} \xrightarrow{f} S_{1,2} \xrightarrow{f} \cdots \xrightarrow{f} S_{1,j} \\ S_2 \xrightarrow{f} S_{2,1} \xrightarrow{f} S_{2,2} \xrightarrow{f} \cdots \xrightarrow{f} S_{2,j} \\ \qquad\qquad \vdots \\ S_k \xrightarrow{f} S_{k,1} \xrightarrow{f} S_{k,2} \xrightarrow{f} \cdots \xrightarrow{f} S_{k,j} \end{array}$$

Abbildung 1.18: Prinzip des time-memory trade-off

sucht das zugehörige x dann gehen wir davon aus, dass dieses y irgendwo in der Matrix aus Abbildung 1.18 vorkommt. Das erste Ziel ist es dann, die Zeile in der y vorkommt zu finden. Entweder steht y schon ganz rechts, dann findet man es durch vergleichen mit den gespeicherten Werten. Falls man y nicht gefunden hat, berechnet man $f(y)$ und sucht das. Somit „bewegt man sich eine Spalte nach rechts“. Das macht man höchstens so oft, wie die Ketten lang sind.

Hat man dann die Zeile i gefunden, so startet man mit dem Startpasswort S_i dieser Zeile und wendet f so oft darauf an, bis man bei $f(x) = y$ angekommen ist und hat somit das Passwort x gefunden, das zu y gehört. Etwas formaler sieht der Algorithmus also so aus:

1. Suche y in der Tabelle, also ein $f^t(S_i) = y$.
2. Wenn $y \neq f^j(S_i)$ für alle i, dann berechne $y' = f(y)$ und benutze y' als neues y in 1.
3. Wenn ein S_i mit $y = f^t(S_i)$ gefunden wurde, dann muss das gesuchte x in der Berechnungskette die bei S_i beginnt stehen. Suche in der Kette.

Die Suche nach dem Urbild u von y in der Kette beginnend bei S_i erfolgt dann so:

1. Berechne $z = f(S_i)$ und setze $u = S_i$.
2. Falls $z = y$ dann gib u aus und beende die Suche, sonst setze $u = z$ und $z = f(z)$.
3. Falls noch nicht über das Kettenende hinaus gegangen, dann weiter bei 2.

Bei diesem Verfahren braucht man etwa $O(2^{n/2})$ Schritte. Das gilt allerdings nur, wenn tatsächlich alle möglichen Funktionswerte in einer der $O(2^{n/2})$ Ketten mit Länge $O(2^{n/2})$ vielen Einträgen auch tatsächlich vorkommen und alle Ketten disjunkt sind.

Oechslin [Oec03] hat dieses Prinzip erweitert und auf konkrete Passwörter und ihre Speicherung in aktuellen (Betriebs-)Systemen angewendet. Zur Vermeidung von Kollisionen zwischen den Ketten wird in einer Kette nicht die Funktion f benutzt, sondern eine andere Funktion $F = R \circ f$. Sie besteht aus f und einer anschließend ausgeführten Reduktionsfunktion R. Oechslin hat auch gezeigt, dass man nicht alle Ketten exakt gleich lang machen muss und dass die Zahl der erforderlichen Ketten nicht $O(2^{n/2})$ ist sondern etwas größer. Außerdem hat er ein Verfahren gefunden, wie man bei Kollisionen ein Verschmelzen der Ketten verhindern kann. Eine Kollision tritt auf, wenn es in zwei verschiedenen Ketten Werte $S_{i,k}$ und $S_{j,l}$ gibt mit $f(S_{i,k}) = f(S_{j,l})$. Dazu benutzt er eine Reduktionsfunktion, die auch von der Position in der Kette (also k bzw. l) abhängt. Oechslin nannte seine Datenstruktur *Regenbogenketten* (engl. **rainbow chains**). Die kompletten Tabellen, die dabei entstehen (wie in Abbildung 1.18), nennt man daher auch **Regenbogentabellen**, (engl. **rainbow tables**).

Regenbogentabellen

Um den Einsatz von vorab berechneten Regenbogentabellen signifikant aufwändiger zu machen, ändert man das Speicherschema für Passwörter wie folgt: Zu jedem Benutzer wird neben der Benutzerkennung B noch ein weiterer Zufallswert Bs erzeugt und gespeichert. Dieser Zufallswert wird Salz (engl. **salt**) genannt. Wählt der Benutzer nun ein Passwort Bp, so speichert das System nicht mehr $f(Bp)$, sondern $f(Bs, Bp)$. Das Salz geht damit in die Berechnung des Funktionswertes ein. Damit eine Passwortprüfung später noch möglich ist muss der *salt value* zusammen mit der Benutzerkennung und dem Funktionswert im System gespeichert werden.

Der Sinn des *salt value* liegt darin, dass nun zwei Benutzer dasselbe Passwort wählen können, *ohne* dass auch die Funktionswerte in der Benutzerliste gleich sind. Dazu müsste das System bei beiden Benutzern denselben Zufallswert erzeugt haben. Je nachdem wie viele verschiedenen Zufallswerte hier möglich sind, vergrößert sich der Platzbedarf für einen Angreifer, der vorab eine Liste mit Passwörtern und ihrem Funktionswert berechnen will. Für jedes Passwort müssen auch alle möglichen *salt values* betrachtet werden. Statt einer Regenbogentabelle braucht man für jeden *salt value* nun eine Regenbogentabelle.

crack
john

Die im Internet verfügbaren Programme *crack* und *john* probieren nach diesem Verfahren einfach eine große Menge von möglichen geheimen Passworten (mit jedem möglichen *salt value*) aus. *john* kann auch Microsoft Windows-Passwörter suchen. Dazu gibt es große Wörterbücher im Internet, die gerne gewählte Passwörter[11], Wörter aus unterschiedlichen Sprachen, Filmtitel, Namen usw. enthalten. Weiterhin prüft *crack* auch bestimmte Modifikationen der Wörter aus dem Wörterbuch. Beliebte Ersetzungen wie „i“ durch „1“ oder „o“ durch „0“ werden genauso geprüft wie unterschiedliche Groß-/Kleinschreibungen. Die Erfolgsquote solcher Angriffe kann bis zu 25% betragen!

[11] Viele Systemadministratoren sind auch Science-Fiction-Fans und kennen die einschlägige Literatur. Passwörter aus solchen Büchern sind also selbst als Passwort eines Administrators gar nicht so selten.

Eine weitere Methode zur Erzeugung potentieller Passwörter nutzt aus, dass Menschen keine echten Zufallszeichenketten als Passwörter wählen. Solche zufälligen Zeichenketten kann man sich nur schwer merken. Daher nehmen viele Benutzer bekannte Wörter und modifizieren sie. Diese Modifikationen folgen bestimmten Schemata und können mit statistischen Methoden bestimmt werden.

In einer echt zufälligen Zeichenkette ist die Wahrscheinlichkeit, dass das i-te Zeichen z. B. ein „a“ ist gleich $1/n$, wobei n die Zahl der möglichen Zeichen ist. Wörter in natürlicher Sprache und auch die von Menschen erfundenen Passwörter haben diese Eigenschaft nicht. Je nachdem welches Zeichen als (i-1)-tes in einer Zeichenkette steht, sind die Wahrscheinlichkeiten für das i-te Zeichen unterschiedlich. Das liegt daran, dass bestimmte Zeichenfolgen wie z. B. „ch“, „ck“ wesentlich wahrscheinlicher sind als z. B. „cq“ oder „ct“. Weiterhin kann man aus Listen bekannter Passwörter die Modifikationsalgorithmen bestimmen mit denen Benutzer ihre Passwörter schwerer zu knacken machen wollen. Eine gerne benutzte Methode ist beispielsweise das Anhängen einer Jahreszahl an ein Wort.

Während Regenbogentabellen davon ausgehen, dass alle Passwörter (und somit alle Funktionswerte) gleich wahrscheinlich sind gehen Angreifer heute nach einer anderen Methode vor. Sie erzeugen Listen von wahrscheinlicheren Passwörtern und berechnen die Funktionswerte dieser Passwörter. Im schlimmsten Fall dauert die Suche nach einem Passwort dann immer noch sehr lang ($O(n^l)$ wenn n die Zahl der möglichen Zeichen und l die Länge des Passworts ist), aber alle echten Passwörter die in der Liste der wahrscheinlichen Passwörter vorne stehen, können damit deutlich schneller gefunden werden.

Gute Passwörter finden: Auch wenn heutige Betriebssysteme selbst das „gehashte“ Passwort dem Zugriff der Benutzer entziehen[12], so sind Passwort-Crack-Programme nach wie vor eine ernste Bedrohung der Sicherheit. Bei der Auswahl eines Passwortes sollten Sie sich daher an die folgenden Regeln halten:

Gute Passwörter finden

- Das Passwort steht nicht in einem erkennbaren Zusammenhang mit Ihnen. Also keine Telefonnummern; keine Namen von Familienmitgliedern, Freunden, Kollegen; keine Autokennzeichen; keine Hobbys usw.
- Es gibt keine Einschränkungen bei den möglichen Zeichen im Passwort (wie z. B. nur lateinische Kleinbuchstaben). Es kommen Klein- und Großbuchstaben, Ziffern und Sonderzeichen im Passwort vor.
- Das Passwort ist ausreichend lang (mindestens 8 Zeichen, besser sogar 10), damit systematisches Ausprobieren zu lange dauert.
- Das Passwort stammt nicht aus einem Wörterbuch. Fügen Sie beispielsweise Sonderzeichen (Ziffern, Satzzeichen, o. ä.) mitten in ein Wort ein, so haben Sie ein längeres, nicht im Wörterbuch stehendes und trotzdem nicht zu schwer zu merkendes Passwort gefunden.
- Vom Hersteller voreingestellte Passwörter werden unbedingt geändert.

[12] Unter UNIX gibt es die sogenannten *Shadow-Passwords*. Die verschlüsselten Passwörter sind dann nicht mehr in der Datei `/etc/passwd` abgelegt, sondern in einer Datei, die nicht für jedermann lesbar ist.

- Passwörter sind nicht zu lange gültig. Irgendwann hat ein Angreifer beim Ausprobieren aller Möglichkeiten vielleicht doch Erfolg. Sie sollten Ihre Passwörter daher regelmäßig ändern. Auch wenn Sie das Gefühl haben, dass jemand Ihr Passwort geknackt hat, sollten Sie unbedingt sofort das Passwort ändern.

- Passwörter sollten nicht notiert werden. Falls es doch erforderlich sein sollte, dann nur in einem verschlossenen versiegelten Umschlag, der an einem sicheren Ort deponiert wird.

Regel

Leider gilt für Passwörter aber auch folgende Regel:

> Gut zu merkende Passwörter lassen sich i. d. R. einfach „knacken“. Schwer zu „knackende“ Passwörter können sich Benutzer nicht einfach merken.

Tipp!

Eine einfache Möglichkeit diese Regel zu brechen besteht darin, sich einen Satz an Stelle des Passwortes zu merken. Aus diesem Satz kann man dann das Passwort ableiten, beispielsweise indem das Passwort aus den Anfangsbuchstaben der Wörter und den Satzzeichen besteht. Aus dem Satz: „Das einfachste Passwort auf der Welt.“ würde dann die Gedankenstütze für das Passwort „DePadW.“ entstehen.

Ergänzend können Sie sich dann noch ein oder zwei Sonderzeichen überlegen und in das oben entstandene Passwort einbauen. Dann haben Sie ein sicheres und hoffentlich schwer zu knackendes Passwort gefunden, das Sie sich vermutlich auch gut merken können.

Ein weiteres Problem für viele Benutzer ist, dass man sich nicht nur ein Passwort merken muss sondern sehr viele. Am PC im Büro, am PC zu Hause, beim E-Mail-Anbieter, bei den sozialen Netzen in denen man aktiv ist, fürs Internet-Banking bei der Bank, beim Fonds-Anbieter, am privaten Smartphone, am Dienst-Smartphone, für die EC-Karte, für die Kreditkarte, bei eBay, Amazon und all den anderen E-Commerce-Anbietern, in allen Diskussionsforen, im WLAN zu Hause, unterwegs oder im Büro, usw. soll man sich ein geheimes Passwort (oder eine PIN) merken. Natürlich ist es einfach und bequem, überall dasselbe Passwort zu benutzen. Leider ist das auch sehr gefährlich. Bringt ein Angreifer dann dieses Passwort in Erfahrung, kann er sich komplett als der Angegriffene ausgeben, man nennt das auch **Identitätsdiebstahl**. Also braucht man unterschiedliche Passwörter für die einzelnen Gelegenheiten. Um diese Passwörter zu erstellen bieten sich zwei Wege an:

Identitätsdiebstahl

1. Anwendungs-Klassen bilden und jede Klasse bekommt ein Passwort. Man kann sich hierzu beispielsweise an den eigenen Sicherheitsanforderungen orientieren. Überall wo man eine Benutzerkennung und ein Passwort angeben muss, Ihnen ein möglicher Identitätsdiebstahl aber als unbedrohlich erscheint (z. B. Registrierung in Diskussionsforen aller Art) wählen Sie ein einfaches Standard-Passwort. Errät jemand dieses Passwort, kann er in Ihrem Namen höchstens an der „Star Trek“-Diskussion teilnehmen oder andere unbedrohliche Dinge tun.

   Klassifizieren Sie die weiteren Anwendungen nach mittel-wichtig, wichtig und sehr wichtig. Überlegen Sie sich für jede Klasse dann ein Passwort und benutzen es für alle Anwendungen dieser Klasse.

Diese Verfahren hat den Vorteil, dass Sie sich nur noch eine „Handvoll“ Passwörter merken müssen (und die Zuordnung zu den Klassen natürlich). Der Nachteil, dass ein Angreifer ein ausgespähtes Passwort mehrfach verwenden kann bleibt allerdings.

2. Überlegen Sie sich ein möglichst zufälliges Basis-Passwort und lernen Sie es auswendig. Für die einzelnen Anwendungen überlegen Sie sich eine Ergänzung, die eine Art „Abkürzung des Anwendungsnamens“ ist. Beispielsweise WLH für das WLAN zu Hause, AMA für den bekannten Internet Buchhändler, usw.

   Kombinieren Sie dann das Basis-Passwort und die Ergänzung zum Passwort für die Anwendung. Im einfachsten Fall nehmen sie die Ergänzung als Präfix oder als Suffix. Sicherer wird es, wenn Sie sich ein komplexeres Schema überlegen, wie beispielsweise 1. Buchstabe der Ergänzung an den Anfang, dann das Basis-Passwort, dann den Rest der Ergänzung. Oder sie fügen zwischen die Bestandteile noch Sonderzeichen ein.

   Ihre Tabelle mit den Ergänzungen müssen Sie nun nicht unbedingt geheim halten, sollten sie aber auch nicht veröffentlichen. Wichtig ist, dass das Basis-Passwort lang genug ist, nicht erraten werden kann und dass Ihr Kombinationsschema geheim bleibt. Ihr Passwort bei den einzelnen Anwendungen wird dadurch länger und schwieriger zu knacken.

1.5 Zusammenfassung

Nach dem Durcharbeiten dieser Kurseinheit sollten Sie Folgendes gelernt haben:

- Warum man sich mit dem Thema Computer und Sicherheit befassen sollte.
- Welche Bedeutung das Wort *Sicherheit* im Zusammenhang mit Computern eigentlich hat.
- Welche Systematik in den Bedrohungen steckt und welche Eigenschaften man von sicheren Systemen erwartet.
- Wie Rechnernetze im Prinzip funktionieren und welche Protokolle und Dienste im Internet benutzt bzw. angeboten werden.
- Wie Viren, Würmer und trojanische Pferde die Sicherheit ihres Rechners bedrohen.
- Wie Angriffe auf Passwörter durchgeführt werden und wie Sie sich ein sichereres Passwort überlegen können.

Lösungen der Übungsaufgaben

Übungsaufgabe 1.1 Die zu schützenden Eigenschaften in Systemen sind:

Vertraulichkeit: Dieses Schutzziel dient der Sicherung der Privatsphäre. Es bedeutet, dass der Inhalt bei einer Kommunikation nur dem Absender und dem beabsichtigten Empfänger bekannt sein darf. Man wird hier evtl. sogar fordern, dass auch die Tatsache, dass eine Kommunikation stattgefunden hat, nur den Beteiligten bekannt werden darf.

Integrität: Dieses Schutzziel dient der Sicherung des Vertrauens in den Inhalt der Kommunikation. Absender und Empfänger sollen sich sicher sein, dass die Nachricht auf dem Transportweg nicht verändert wurde.

Authentizität: Dieses Schutzziel dient der Sicherung des Vertrauens in die Identität der an einer Kommunikation Beteiligten. Konkret soll sich der Kommunikationspartner sicher sein, dass der andere Partner tatsächlich derjenige ist, der er vorgibt zu sein.

Verfügbarkeit: Dieses Schutzziel dient der Sicherung des Vertrauens in die „Technik“. Konkret erwartet man, dass die Dienste zur Verfügung stehen, wenn man sie benutzen möchte.

Übungsaufgabe 1.2 Bei *ftp* treten mindestens die folgenden Sicherheitsrisiken auf:

- Die Übertragung des Passwortes kann abgefangen werden, und ein Dritter kann dann mit diesem Passwort evtl. vertrauliche Daten vom *ftp*-Server laden.
- Der Dritte kann auch Daten, die durch Urheberrecht geschützt sind, auf den ftp-Server kopieren. Dadurch macht sich der Betreiber des *ftp*-Servers evtl. strafbar (Raubkopien).
- Der Account und das Passwort können nicht nur für *ftp* gelten, sondern auch ganz normale Benutzerkennungen sein. Dann kann sich ein Angreifer per Telnet auf dem *ftp*-Server anmelden und dort beliebige Kommandos ausführen.

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.