

## KAPITEL 4

# Welche Angriffe gibt es?

Nachdem wir eine grobe Vorstellung haben, wie das Internet eigentlich funktioniert, sollten wir uns nun einmal mit der Fragestellung auseinandersetzen, wo die Schwachstellen dieser Technik liegen. Wie wir sehen werden, ist das Internet zwar insgesamt erstaunlich robust gegen zufällige technische Störungen, es existieren aber einige grundlegende Schwächen, die für gezielte Angriffe genutzt werden können. Aus diesem Grund werden wir uns im folgenden einmal den gängigen Methoden zuwenden, mit denen Angreifer versuchen, die Kontrolle über fremde Rechner und Daten zu erlangen. Bitte beachten Sie, daß die folgende Aufstellung nicht vollständig ist und dies auch nicht sein kann. Regelmäßig werden neue Angriffe erfunden.

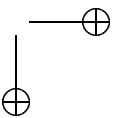
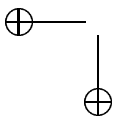
## Denial-of-Service-Angriffe

Wenn Sie Ihre Rechner an das Internet anschließen, sind Cracker, die versuchen, in Ihre Rechner einzubrechen und sie für ihre eigenen finsternen Pläne zu nutzen, nur eines Ihrer Probleme. Ein anderes besteht darin, daß ein gelangweilter Dreizehnjähriger beschließt, Ihnen zu beweisen, daß er Ihnen überlegen ist. Dieser Beweis ist schnell erbracht. Mit einem Mal bemerken Sie, daß Sie keinen Kontakt mehr zum Internet haben. Falls Sie einen Webserver betreiben, ist dieser aus dem Internet nicht mehr ansprechbar.

Solche Angriffe, die einen Benutzer von einem bestimmten Dienst ausschließen sollen, nennt man *Denial-of-Service-Angriffe* oder kurz *DoS-Angriffe*. Im folgenden werden wir einige der gängigeren Varianten kennenlernen.

## Flooding

Unter *Flooding* versteht man das Überlasten eines Rechners durch das Senden einer Vielzahl von Netzwerkpaketen [16]. Am effektivsten ist dabei das Senden von TCP-Paketen mit gesetztem SYN-Flag (*SYN-Flooding*). Im Gegensatz zu UDP und ICMP, bei denen jedes Paket den Rechner nur für eine recht kurze Zeit beschäftigt, löst ein TCP-SYN-Paket einen Verbindungsaufbau aus. Das bedeutet, daß zuerst einmal versucht wird, den Anfragenden zu erreichen. Gelingt dies nicht, so werden die dafür benötigten Ressourcen



erst bei Erreichen eines Timeouts, üblicherweise 75 Sekunden, freigegeben. Grundsätzlich ist die Anzahl der gleichzeitigen Anforderungen, eine Verbindung aufzubauen, auf einem Port beschränkt (Linux: 10, NT: 6). Kommen mehr SYN-Pakete gleichzeitig an, so werden die überzähligen entsorgt. Schickt man also eine ausreichend hohe Anzahl von Paketen, so kann man den angegriffenen Port für 75 Sekunden blockieren. Um z. B. einen Server eine Stunde unerreichbar zu machen, ist es daher nicht nötig, so viele Pakete zu schicken, daß seine Verbindung komplett ausgelastet ist. Vielmehr genügt es, 48 mal 10 Pakete an ihn zu schicken. Das sind etwa 32 Bit pro Sekunde. Programme, die genau dies leisten, sind im Netz verfügbar.

Sind auf dem Zielsystem keine frei zugänglichen Server installiert oder ist es gegen SYN-Floods immun, so kann der Angreifer versuchen, so viele Pakete zu senden, daß die Netzwerkanbindung des Zielsystems zusammenbricht. Ein Rechner allein wird dies aber in der Regel nicht schaffen, da zwischen ihm und dem Zielsystem wahrscheinlich Netzwerkgpässe liegen, die die Rate begrenzen, mit der er Pakete an sein Zielsystem schicken kann. Auch besitzen große Webserver oft eine deutlich bessere Netzwerkanbindung als die meisten Klientenrechner. Aus diesem Grund muß der Angreifer erreichen, daß mehrere Rechner gemeinsam Pakete an das Zielsystem schicken.

Ein solches Flooding, bei dem mehrere Rechner koordiniert ein gemeinsames Ziel angreifen, nennt man *verteilte DoS-Angriffe* oder *DDoS-Angriffe*<sup>1</sup>.

Eine Methode, um dieses Ziel zu erreichen, sind *Reflektor-Angriffe*. Dabei werden normale Rechner dazu gebracht, auf gefälschte Anfragen des Zielsystems zu antworten. Nimmt man eine hinreichende Anzahl von Rechnern und stellt Anfragen, bei denen die Antwort umfangreicher ausfällt als die gestellte Anfrage, so kann man mit geringem Aufwand eine große Menge von Netzwerkverkehr erzeugen.

Dies kann geschehen, indem Pakete mit gefälschter Absenderadresse an die Broadcast-Adresse eines größeren Teilnetzes geschickt werden. Jeder Rechner im Teilnetz wird die Antwort an den vorgeblichen Absender schicken, der dann regelrecht mit Paketen überschwemmt wird. Auf diese Weise kann ein einzelnes Paket des Angreifers leicht bis zu hundert Folgepakete an das eigentliche Opfer auslösen [17].

Man kann aber auch normale Server als Reflektoren benutzen. Dabei ist es egal, um welche Art Server es sich handelt. In [15] wird ein Angriff beschrieben, an dem Web-, SSH-, Telnet-, DNS- und IRC-Server sowie Backbone-Router teilnahmen, die zum Austausch von Routing-Informationen BGP<sup>2</sup> benutzen. In diesem Fall sah es für die Angegriffenen so aus, als ob sich das gesamte Internet verschworen hätte, um sie vom Rest der Welt abzuschneiden.

Der Angriff funktioniert, indem man ein SYN-Paket mit gefälschter Absenderadresse an einen Server mit einer möglichst guten Internet-Anbindung schickt. Dieser wird nun ein Antwortpaket an den vermeintlichen Klienten schicken. Schickt der Angreifer die Pa-

<sup>1</sup> DDoS steht für »Distributed Denial of Service«.

<sup>2</sup> Das Border Gateway Protocol (BGP) erlaubt es mehreren Teilnetzen, Informationen darüber auszutauschen, auf welchen Wegen andere Teilnetze erreichbar sind und wie gut die Übertragung auf einem bestimmten Weg funktioniert. Dies erlaubt es den zentralen Routern im Internet, den optimalen Weg für die zu übermittelnden Pakete zu finden.

kete nun in schneller Folge an eine Vielzahl von Servern, so kann er leicht eine große Menge von Paketen erzeugen, die auf verschiedenen Wegen zu seinem Opfer strömen. Dadurch ist die Wahrscheinlichkeit groß, daß diese Flut von Paketen nicht schon vorher auf einen Engpaß im Netz trifft, sondern daß sie erst bei der Anbindung des Opfers an das Internet zusammentreffen. Dort wird sich nun ein Stau bilden, der dazu führt, daß Pakete verlorengehen. Hat der Angreifer diesen Zustand erreicht, so werden die Server, die keine Antwort auf ihre Pakete erhalten, diese noch bis zu dreimal wiederholen. Dadurch gelingt es dem Angreifer, viermal soviel Verkehr zu erzeugen, wie er selbst sendet.

Schließlich existieren noch Fälle, in denen Angreifer in Rechner einbrechen und diese dazu benutzen, auf ein Kommando hin gemeinsam ein vom Angreifer vorgegebenes Ziel anzugreifen. Steve Gibson berichtet z. B. in [14], wie ein 13jähriger ohne tiefere Computerkenntnisse 474 Rechner in seine Gewalt brachte und damit Gibsons Netzwerkanbindung lahmlegte. In einem anderen Fall im Februar 2000 unterbrach eine Welle ähnlicher Angriffe unter anderem den Zugriff auf die Webseiten von Yahoo, Buy.com, eBay, CNN, Amazon und ZDNet [30]. Die Angriffe dauerten z. T. bis zu drei Stunden.

Gegen ein SYN-Flooding kann man einen Linux-Rechner schützen, indem man die sogenannten *Syncookies* aktiviert. Dieser Mechanismus bewirkt, daß der Kernel im Falle eines Überlaufens seiner Tabelle für unvollständig aufgebaute Verbindungen aufhört, Verbindungsanfragen weiter zu speichern. Statt dessen enthalten Antworten auf SYN-Pakete eine Folgenummer, die aus

- einem Zähler,
- Adresse und Port des Senders,
- Adresse und Port des Empfängers,
- der Folgenummer des SYN-Paketes

gebildet wird. Dabei wird zur Bildung dieses Wertes ein kryptographisches Verfahren mit einem geheimen Schlüssel eingesetzt [7].

Beantwortet der Klient nun so ein Paket, so enthält sein Paket auch die Bestätigung der Seriennummer des Servers. Der Server kann diese Seriennummer überprüfen und feststellen, ob sie zu den anderen Feldern im Paket des Klienten paßt. Auf diese Weise ist es nicht nötig, die Tabelle zu konsultieren. Obwohl die Tabelle also voll ist, können trotzdem neue Anfragen beantwortet werden.

Der Grund, warum man die Tabelle noch nicht ganz abgeschafft hat und vollständig auf Syncookies umgestiegen ist, liegt darin, daß sie es TCP erlaubt, schon im SYN-Paket einige Parameter für die Verbindung anzufordern. Diese Information kann aber nicht auch noch in der Folgenummer kodiert werden. Der Platz reicht dafür schlicht nicht aus. Aus diesem Grund müssen Anfragen, die bestimmte Parameter fordern, abgelehnt werden, während sie im Normalbetrieb problemlos hätten angenommen werden können. Deshalb werden Syncookies nur dann benutzt, wenn neue Verbindungen andernfalls immer abgelehnt werden müßten.

Gegen verteilte Angriffe, die darauf abzielen, die Netzwerkanbindung zu überlasten, kann man sich dagegen nicht schützen. Diese Art von Angriff wird in der Regel immer dazu führen, daß der Kontakt des Rechners zum Internet abbricht.

Soll ein Netz dagegen nicht angegriffen werden, sondern nur als »Verstärker« für einen Angriff dienen, so kann dies an der Firewall oder einem externen Router verhindert werden. Dort muß sichergestellt werden, daß keine Pakete angenommen oder weitergeleitet werden, die an Broadcast-Adressen (siehe Kapitel 3, Abschnitt *IP*, ab Seite 19) gerichtet sind. Dies gilt insbesondere, wenn diese Pakete von Adressen kommen, die nicht selbst zu dem Subnetz gehören, an das die Pakete gerichtet sind.

## Angriffe mittels ICMP

Neben dem Flooding, das auch mit ICMP-ECHO-Paketen (Ping) durchgeführt werden kann, existieren noch weitere Möglichkeiten, ICMP zu DoS-Angriffen zu benutzen. Ausgehend von dem Wissen, daß es eine Fehlermeldung »Destination Unreachable« gibt, kamen bössartige Zeitgenossen auf die Idee, dies für einen DoS-Angriff zu benutzen. Kennt man die Parameter einer Verbindung (Quell- und Zieladresse sowie die benutzten Ports), so ist es möglich, ein ICMP-Paket zu konstruieren, das die Verbindung beendet. Benutzt wurden Programme, die solche Pakete konstruierten, in erster Linie, um mißliebige Personen von IRC-Servern auszuschließen.

Eine andere Klasse von Angriffen mittels ICMP zielt darauf ab, das Routing eines Rechners zu manipulieren. Gelingt es dem Angreifer, einen Rechner dazu zu bringen, als Router einen nicht existenten Rechner zu benutzen, so werden alle Pakete, die nicht lokal zugestellt werden können, ins Leere geschickt. Der angegriffene Rechner ist damit vom Internet abgeschnitten.

Technisch gesehen existieren mindestens zwei Möglichkeiten, derartige Angriffe mittels ICMP durchzuführen. Es gibt z. B. eine Fehlermeldung »Redirect«, die ein Router sendet, wenn er feststellt, daß ein anderer Router besser geeignet wäre, um ein bestimmtes Paket weiterzuleiten. Der Sender des Paketes wird es dann noch einmal an den angegebenen Ersatzrouter senden und auch versuchen, Folgepakete der Verbindung ebenfalls über diesen Ersatzrouter zu verschicken.

Auch das ICMP Router Discovery Protocol läßt sich zu Angriffen nutzen. Hierbei sendet der Opferrechner Router Solicitation-Pakete an die Multicast-Adresse 224.0.0.2, um die Adresse des zuständigen Routers zu erfragen. Erreichen die Pakete einen Router, so antwortet dieser mit einem Router Advertisement-Paket, das eine oder mehrere Router-Adressen enthält. Da hier jeder Rechner im lokalen Netz antworten kann, kann ein Angreifer eine Antwort schicken, die eine beliebige Adresse enthält.

Dieser Angriff kann allerdings nicht über das Internet erfolgen, da die Router Solicitation-Pakete nicht durch die Router in das Internet vermittelt werden. Die anderen beiden Angriffe können und werden über das Internet eingeleitet. Hier hilft es, bei der Konfiguration der Firewall restriktive Regeln festzulegen, was das Akzeptieren von ICMP-Paketen angeht.

## Cracking

Unter Cracking versteht man den Versuch, einen Rechner zum Ausführen von Funktionen zu bringen, zu denen der Angreifer nicht autorisiert ist. Das kann bedeuten, daß ein Angreifer (Schreib-)Zugriff auf einige Dateien eines Rechners (z. B. Webseiten) erlangt, im schlimmsten Fall aber auch, daß er auf dem Rechner beliebige Befehle mit Rootrechten ausführt.

Der Vorgang des Crackings ist dabei in der Regel technisch deutlich komplizierter, als es im Fernsehen dargestellt wird. Dort sieht man üblicherweise, wie auf dem Bildschirm eine Aufforderung erscheint, ein Paßwort einzugeben. Hierauf startet der Angreifer ein Programm. Beginnend bei AAAAAAAAAA probiert das Programm alle Kombinationen durch, wobei nach einiger Zeit der erste Buchstabe auf einem Wert stehenbleibt, dann der zweite und so weiter.

Dies ist so natürlich lächerlich. Obwohl es Programme gibt, die in bestimmte Rechner automatisiert einbrechen können, geht es praktisch nie darum, Paßwörter zu erraten. Statt dessen arbeitet sich der Angreifer in der Regel Schritt für Schritt vor, wobei er eine Kombination diverser Techniken benutzt, bis er schließlich den gewünschten Zugriff auf den Rechner hat. Im folgenden wollen wir einen ausführlichen Blick auf die einzelnen Phasen eines solchen Angriffs werfen.

### Auswahl eines Ziels

Grundsätzlich existieren zwei Möglichkeiten, ein Ziel auszuwählen. Entweder hat der Angreifer vor, in einen bestimmten Rechner einzubrechen, oder er nimmt sich eine große Gruppe von Rechnern vor und probiert der Reihe nach an jedem ein paar Standardangriffe aus, bis er einen findet, der verwundbar ist.

Für Angriffe der zweiten Art existieren Programme, die den Vorgang des Crackings auf die Eingabe eines Bereiches von IP-Adressen beschränken. Anwender dieser Programme werden daher oft *Script Kiddies* genannt.

### Informationsbeschaffung

Bevor er tatsächlich einen Rechner angreift, wird ein erfahrener Cracker erst einmal Informationen über sein Ziel sammeln. Je mehr er über sein Opfer weiß, um so einfacher ist es für ihn, Schwachstellen zu finden.

#### Die Administratoren und Anwender

Für jede registrierte Netzwerkdomäne muß bei der zuständigen Registrierungsstelle ein Name eines technischen Ansprechpartners hinterlegt sein. Diese Information kann mit einem Dienst namens Whois abgefragt werden. Sucht man nach den gefundenen Namen in Newsgruppen und auf Webpages, so wird sich unter Umständen schon ein klareres Bild von den Personen ergeben, die für einen Rechner verantwortlich sind.

Hat z. B. ein Administrator gerade in eine Newsguppe Fragen gepostet, die an seiner Qualifikation zweifeln lassen, so ist dies für einen Angreifer eine gute Nachricht. Auch kann es sein, daß man zu freigiebig mit Angaben über das Zielsystem war oder daß die Homepage eines Administrators Begriffe enthält, die dieser als Paßwort für seinen Zugang benutzt hat.

Auch das Telefon kann dazu dienen, weitere Informationen zu sammeln. Geschickte Angreifer nutzen anderweitig gesammelte Informationen, um sich z. B. als Techniker auszugeben und Benutzer dazu zu bringen, Paßwörter oder technische Informationen zu verraten (*Social Engineering*).

### DNS-Abfragen

Eine weitere Informationsquelle findet der Angreifer im DNS. Sind die zuständigen Server unsicher konfiguriert, so erlauben sie Zone Transfers (siehe Kapitel 3, Abschnitt *DNS*, ab Seite 26). Diese kann er dazu benutzen, sich einen Überblick über ein anzugreifendes Netz zu verschaffen. Mit einem Befehl erhält er eine Liste aller eingetragenen Rechner in einem Netz samt ihrer Namen. Letztere erlauben oft schon einen Rückschluß auf ihre Funktion und Architektur (z. B. `poolpc14`, `fw`, `sparc15`, ...). Darüber hinaus können im DNS auch noch zusätzliche Informationen zum Rechner eingetragen sein, die diese Angaben sogar explizit enthalten (HINFO- und TXT-Records). Es empfiehlt sich daher, den Zugriff auf diese Funktion des DNS-Servers einzuschränken. Dadurch wird zwar nicht verhindert, daß ein Angreifer diese Informationen durch gezielte Zugriffe auf die Rechner erhält, sein Zeitaufwand ist aber deutlich höher, und er hinterläßt unter Umständen auch mehr Spuren.

### Ping Sweep

Auch wenn der zuständige DNS-Server keine Zone Transfers erlaubt, kann sich der Angreifer trotzdem einen Überblick verschaffen, welche Rechner in einem Netz momentan aktiv sind. Er sendet ICMP Echo Requests (Pings) an den Bereich von IP-Adressen, der dem zu untersuchenden Netz zugeordnet ist. Wann immer eine Anfrage von einem Rechner empfangen wird, wird dieser mit einem ICMP Echo Reply antworten.

Nun besitzt er eine Liste der Rechner im Zielnetz und kann gezielte DNS-Anfragen bezüglich dieser Systeme stellen. Abfragen zu einzelnen Rechnern muß der DNS-Server im Gegensatz zu Zone Transfer-Anfragen beantworten, denn das ist seine Aufgabe.

### Betriebssystemerkennung

Auf bestimmten Seiten im Internet werden regelmäßig die neuesten Sicherheitslücken bekannter Softwareprodukte diskutiert. Hierbei zeigt sich, daß fast jede Version der gängigen Betriebssysteme und die meisten Versionen der gängigen Serverdienste Schwachstellen haben, die Angriffe erlauben. Allerdings werden diese Sicherheitslücken im Normalfall in der nächsten Version des Betriebssystems oder des Servers behoben.

Manchmal sind diese Schwachstellen so gravierend, daß es für Angreifer lohnenswert erscheint, diese auszunutzen, um damit in einen Rechner einzudringen oder diesen zu-

mindest unbenutzbar zu machen. Um herauszufinden, auf welche Rechner ein Angriff Erfolg verspricht, wird oft ein großer Bereich von Internet-Adressen nach angreifbaren Systemen durchsucht. Dabei kommen oft zwei Methoden zur Anwendung, mit denen festgestellt werden soll, ob ein bestimmter Rechner verwundbar ist.

Die erste wird *Banner Grabbing* genannt. Dabei macht der Angreifer es sich zunutze, daß viele Dienste zur Begrüßung Serverversion und Betriebssystem bekanntgeben. Für SMTP kann dies z. B. so aussehen:

```
> nc localhost 25
220 dummy.local.net ESMTP Sendmail 8.9.3/8.9.3/SuSE Linux 8.9.3-01;
Sat, 19 Feb 2000 08:31:22 -0700
```

Banner Grabbing läßt sich technisch relativ einfach realisieren, es hat aus der Sicht des Angreifers aber auch Nachteile. So wird zu einem Serverdienst eines jeden zu untersuchenden Rechners eine Verbindung aufgebaut, womit der Zugriff unter Umständen mitprotokolliert wird. Auch ist der Angreifer darauf angewiesen, daß der fragliche Server ihm auch wirklich die Informationen gibt, die er sucht. Die Begrüßungsmeldungen der meisten Server sind aber frei konfigurierbar.

Aus diesen Gründen könnte es für unseren Angreifer sinnvoll sein, zuerst mit weniger auffälligen Mitteln herauszufinden, ob ein Rechner überhaupt das gewünschte Betriebssystem benutzt, bevor er gezielt untersucht, ob ein bestimmter Dienst installiert ist.

Hierzu bietet sich *TCP/IP Stack Fingerprinting*, im folgenden kurz Fingerprinting genannt, an [20]. Bei dieser Methode macht sich der Angreifer zunutze, daß nicht alle Implementationen des TCP/IP-Protokolls identisch sind. Es gibt Unterschiede in der Art, wie TCP-Folgenummern gebildet werden, wie auf ungültige Pakete reagiert wird, nach wie vielen Bytes einer Verbindung eine Bestätigung erwartet wird und welche zusätzlichen Informationen ICMP-Fehlermeldungen enthalten.

Mittlerweile existieren Programme, mit denen sich auf diese Weise nicht nur die einzelnen Betriebssysteme unterscheiden, sondern z. T. sogar auch noch die Versionsnummern bestimmen lassen. Da die Tests nur einzelne Pakete senden, aber keine vollständigen Verbindungen öffnen, bewirken sie auch nicht, daß die Anwendungsdienste des Zielrechners die Zugriffe protokollieren. Um sie zu entdecken, müßte der Zielrechner schon eine intensive Firewall-Protokollierung durchführen oder ein *Intrusion Detection System*<sup>3</sup> einsetzen.

## Port Scanning

*Port Scanning* wird der Vorgang genannt, wenn auf einem Rechner nach zugreifbaren Netzwerkdiensten gesucht wird. Der Vorgang ist per se nicht schädlich für den betroffenen Rechner. Es ist allerdings ein recht sicheres Zeichen für einen bevorstehenden Einbruchversuch.

<sup>3</sup> IDS (genaugenommen netzwerkbasierte IDS) überwachen den Verkehr in einem Netzwerk und versuchen anhand eines Regelwerks Angriffe festzustellen.

Die einfachste Methode, einen TCP-Port Scan durchzuführen, könnte darin bestehen, einfach eine Verbindung zu den zu untersuchenden Ports zu öffnen. Gelingt dies, so ist ein Server auf dem Port aktiv. Dies ist allerdings einfach zu entdecken. Hierzu genügt es schon, auf ungenutzten Ports Programme aufzusetzen, die auf Verbindungen warten und diese dann protokollieren.

Deshalb gehen modernere Port-Scanner dazu über, spezielle Pakete zu generieren und aus den Rückmeldungen ihre Schlüsse zu ziehen. So werden z. B. zwar TCP-SYN-Pakete<sup>4</sup> geschickt, nachdem aber das Bestätigungspaket empfangen wurde, wird der Verbindungsaufbau abgebrochen. Auf diese Weise kommt keine Verbindung zustande, und das Beobachtungsprogramm wird nicht ausgelöst.

Ein anderer Ansatz basiert darauf, daß Firewalls oft nur TCP-SYN-Pakete ausfiltern, andere Pakete aber durchlassen. Der Scanner sendet deshalb ein TCP-Paket mit gesetztem FIN-Bit. Residiert ein Server auf dem fraglichen Port, so wird das Paket in der Regel ignoriert. Ist dies nicht der Fall, so wird ein Paket mit gesetztem RST-Bit zurückgesendet.

UDP-Scans sind aus Sicht des Angreifers etwas komplizierter und unzuverlässiger. Dies liegt daran, daß UDP verbindungslos ist und damit nicht sicherstellt, daß Pakete auch beantwortet werden. Auch gibt es UDP-Dienste, die Pakete annehmen, aber selbst keine senden (z. B. syslog). Um dieses Problem zu umgehen, wartet ein UDP-Scanner in der Regel nicht auf Antworten von Servern, sondern auf die Fehlermeldungen durch Versuche, auf Ports zuzugreifen, denen kein Dienst zugeordnet ist. Im Umkehrschluß wird dann angenommen, daß all jene Ports benutzt werden, für die keine Fehlermeldungen eingetroffen sind.

Es existieren eine Reihe von Ansätzen, Port Scans zu erkennen. Die in unserem Rahmen einfachste Methode wäre die regelmäßige Untersuchung der Systemprotokolle. Um dort auch aussagekräftige Meldungen zu erzeugen, bietet es sich an, Paketfilterregeln zu benutzen, die einen Eintrag in `/var/log/messages` erzeugen. So ist es z. B. möglich, eine Filterregel zu konfigurieren, die Anwendung findet, wenn keine andere Regel greift. Häufen sich mit einem Male Meldungen über zurückgewiesene Pakete, kann ein Angriff vermutet werden.

Eine ausführliche Darstellung der unterschiedlichen Techniken des Port Scannings findet sich in [19].

## Einbruch in den Rechner

Durch die vorangegangenen Schritte weiß der Angreifer ziemlich genau, wie gut sein Ziel gesichert ist und an welchen Stellen es angegriffen werden kann. Im folgenden ist eine Reihe möglicher Angriffspunkte aufgeführt.

<sup>4</sup> Siehe Kapitel 3, Abschnitt *TCP*, ab Seite 23.

## Unsichere Protokolle

Auf Unix-Systemen ist oft eine Vielzahl von Diensten installiert. Darunter sind viele, die zu einer Zeit konzipiert wurden, als Sicherheit noch kaum ein relevantes Thema war. So erlauben es z. B. die R-Dienste (Rsh, Rlogin, Rexec), auf einem Rechner Befehle auszuführen, ohne ein Paßwort angeben zu müssen. Als einzige Authentifikation dient die Adresse des Rechners, von dem der Zugriff erfolgt. Diese kann aber gefälscht werden, wie wir in Kapitel 4, Unterabschnitt *Spoofing*, ab Seite 46 sehen werden.

TFTP wird von Klienten ohne eigene Festplatte benutzt, um die Systemdateien herunterzuladen. Dabei wird kein Paßwort verlangt, da das TFTP-Protokoll in den Klienten in einem ROM-Baustein implementiert ist<sup>5</sup>. Das eigentliche Betriebssystem wird ja erst noch heruntergeladen. Gelingt es nun einem Außenstehenden, auf diesen Dienst des Servers zuzugreifen, so wird er u. U. auch Zugriff auf die Paßwortdatei der Klienten erhalten.

Eine ähnliche Rolle spielt NIS, über das z. B. Paßwortdateien auf einem zentralen Server gehalten werden können. Auch hier findet keine nennenswerte Authentifikation statt. Ein einfaches `ypcat passwd` auf einem Rechner im lokalen Netz reicht in der Regel aus, um die Paßwortdatei auszulesen.<sup>6</sup>

Auch die graphische Oberfläche X stellt einen Netzwerkdienst dar, auf den prinzipiell von anderen Rechnern zugegriffen werden kann. Dies erlaubt es einem Angreifer, Tastatureingaben mitzulesen und zu manipulieren.

Schließlich können auch verteilte Dateisysteme einen Königsweg in einen Rechner darstellen. Unter Unix nimmt diese Rolle NFS ein. Kann auf vertrauliche Daten (z. B. Paßwort-Dateien) von anderen Rechnern über NFS zugegriffen werden, so muß auch davon ausgegangen werden, daß ein Angreifer diese lesen kann. Unter Windows ist insbesondere das Windows-Netzwerk<sup>7</sup> ein beliebtes Angriffsziel. Insbesondere Windows 9x-Benutzer, die auf ihrem Rechner Verzeichnisse freigegeben haben, um von anderen Rechnern im LAN auf sie zugreifen zu können, sind sich oft nicht bewußt, daß dies meistens auch von beliebigen Rechnern im Internet aus funktioniert.<sup>8</sup> Ist ein Schreibzugriff auf Programme oder Systemverzeichnisse möglich, so ist es eine Sache von Minuten, den Rechner zu kompromittieren.

Der beste Schutz gegen Angriffe dieser Art besteht sicherlich darin, alle Rechner so sicher zu konfigurieren, daß Angriffe nicht möglich sind. Dazu müßten alle nicht dringend benötigten Dienste abgeschaltet werden. Die verbleibenden Dienste würden dann so konfiguriert, daß sichergestellt ist, daß nur berechnete Benutzer zugreifen und nur diejenigen Aktionen ausführen dürfen, die vorher als sicher festgelegt wurden. Leider erweist sich dieser Ansatz in der Praxis oft als undurchführbar.

<sup>5</sup> Dies ähnelt dem BIOS auf PC-Motherboards.

<sup>6</sup> Darüber hinaus ist der Einsatz von NIS in der Regel nur möglich, wenn auf `shadow password` verzichtet wird. Es existiert eine verbesserte NIS-Version für Solaris-Rechner, in der die genannten Probleme behoben wurden, diese wird aber gegenwärtig nicht flächendeckend eingesetzt.

<sup>7</sup> Bekannt als NetBIOS over TCP/IP, SMB, CIFS, Netzwerkfreigaben . . .

<sup>8</sup> Technisch funktioniert dies dann, wenn eine Bindung des Freigabedienstes an das DFÜ-Netzwerk besteht oder der Rechner über das LAN mit dem Internet verbunden ist. Wenn Sie nicht wissen, ob dies auf Ihrem Rechner der Fall ist, so besteht eine sehr gute Chance, daß Sie angreifbar sind.

Hier können wir mit einer Firewall Abhilfe schaffen. Mit ihr können wir Zugriffe auf unsere Rechner im lokalen Netz unterbinden. Ohne die Möglichkeit, die Server anzusprechen, kann der Cracker sie auch nicht angreifen.

### Unsichere Applikationen

Neben unsicheren Protokollen kann auch die konkrete Implementation eines Serverdienstes ein Problem darstellen. So existieren in manchen Servern Möglichkeiten der Fernwartung, die mit einem Standardpaßwort geschützt sind, bis dieses nach der Installation geändert wird. Eine alte Version von Sendmail erlaubte es sogar, beliebige Befehle auszuführen, vorausgesetzt, man kannte das fest einkompilierte Paßwort.

Neben solchen beabsichtigten Hintertüren stellen aber vor allem unbeabsichtigte Programmierfehler einen steten Quell von Angriffsmöglichkeiten dar. Eine Variante sind Speicherüberläufe. Diese treten dann auf, wenn ein Programm mehr Daten erhält, als es in dem Speicherbereich ablegen kann, der dafür vorgesehen war. Wird dieser Fehler nicht abgefangen, so wird das Programm über den Speicherbereich hinaus Daten in nicht dafür vorgesehene Bereiche des Hauptspeichers schreiben. Dabei wird es irgendwann beginnen, seine eigenen Befehle zu überschreiben. Im besten Fall bedeutet dies, daß das Programm abstürzt, im schlimmsten Fall wird das Programm mit neuen Befehlen überschrieben, die dann anstelle der ursprünglichen Anweisungen ausgeführt werden. Obwohl die technische Realisierung eines solchen Angriffs nicht einfach ist, gibt es für viele Serverdienste fertige Angriffsprogramme, die von jedermann aus dem Internet heruntergeladen werden können. Eine tiefere technische Betrachtung des Themas finden Sie z. B. in [27].

Eine andere Variante von Programmierfehlern findet sich häufig auf Webservern. Diverse Angriffe benutzen schlecht geschriebene CGI-Skripte des Servers, um diesen dazu zu bringen, beliebige Anweisungen auszuführen. Das Problem besteht dabei in der Regel darin, daß Daten des Benutzers auf dem Server<sup>9</sup> ungeprüft in Betriebssystemaufrufe umgesetzt werden. Nehmen wir z. B. einmal folgendes Skript:

```
#!/bin/sh
#####
#
# pseudo
#
#   a good demonstration of bad programming style
#
#####

echo -n "Ihre Eingabe bitte: "
read EINGABE
sh -c "echo $EINGABE"
```

Dieses Skript nimmt eine Eingabe entgegen und schreibt diese auf den Bildschirm:

<sup>9</sup> Der Versuch, ungültige Eingaben abzufangen, indem man in das Formular JavaScript-Code einbaut, der dann durch den Browser ausgeführt wird, ist sinnlos. Es ist einfach, diese zu umgehen und beliebige Daten manuell an den Server zu schicken. Leider gibt es genug Websites, bei denen es genügt, JavaScript abzuschalten, um bestimmte Sicherheitsabfragen unwirksam zu machen.

```
> ./pseudo
Ihre Eingabe bitte: Hallo Du
Hallo Du
```

Interessant wird es, wenn wir unsere Eingabe leicht abwandeln:

```
> ./pseudo
Ihre Eingabe bitte: Hallo Du; du siehst gut aus
Hallo Du
du: siehst: No such file or directory
du: gut: No such file or directory
du: aus: No such file or directory
```

Hier wurde offensichtlich der Befehl `du` ausgeführt. Dies ist auch nicht weiter verwunderlich, da das Semikolon für `sh` das Ende eines Befehls bedeutet. »`du`« war damit ein neuer Befehl, der dann auch ordnungsgemäß ausgeführt wurde.

Auf dieselbe Weise können auch viele CGI-Skripte dazu gebracht werden, beliebige Befehle auszuführen. Beispiele reichen von Testskripten, die übergebene Argumente ausgeben, über Bestellsysteme, bei denen in ein Formular eingegebene Daten per E-Mail verschickt werden sollten, bis zu Servern, die die Daten direkt in SQL-Anfragen umsetzen. In letzterem Fall war ein ziemlich weitreichender Zugriff auf die Datenbank möglich [28]. Auch hier existieren Programme, die gezielt nach bestimmten unsicheren Skripten suchen und dem Angreifer dann die Möglichkeit geben, beliebige Befehle auszuführen.

Schützen kann man sich gegen diese Angriffe kaum. Theoretisch könnte man nur Software verwenden, die man selbst geschrieben oder deren Quelltext man ausgiebig analysiert hat. In der Realität wäre ein solches Vorgehen nur denkbar, wenn man überragende technische Kenntnisse und ein Übermaß an Freizeit besitzt. Es bleibt daher nur, sich ständig über die bekannten Sicherheitslücken auf dem laufenden zu halten und regelmäßig die Sicherheitsupdates der Hersteller einzuspielen (siehe Kapitel 16, Abschnitt *Updates*, ab Seite 535).

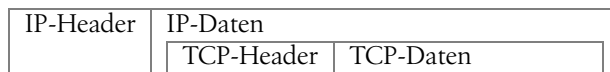
Wir können uns allerdings dagegen schützen, daß ein erfolgreicher Angriff auf einen öffentlichen Server dazu dienen kann, leichter unsere Rechner im internen Netz zu kompromittieren. Hierzu stellen wir die Server, die aus dem Internet zugänglich sein sollen, nicht in das interne Netz, sondern in ein eigenes Netz, die sogenannte *Demilitarized Zone* oder DMZ. Dieses Netz ist sowohl gegenüber dem internen Netz als auch dem Internet mittels Firewalls abgeschirmt. Auf diese Weise kann auf einen öffentlichen Server auch aus dem internen Netz zugegriffen werden. Der Server kann aber keine Rechner im internen Netz kontaktieren, womit er für den Angreifer keine bessere Ausgangsbasis für Angriffe gegen das interne Netz darstellt als ein beliebiger Rechner im Internet. Wie eine entsprechende Firewall-Architektur genau aussieht, werden wir in Kapitel 5 ab Seite 65 sehen.

### Fragment-Angriffe

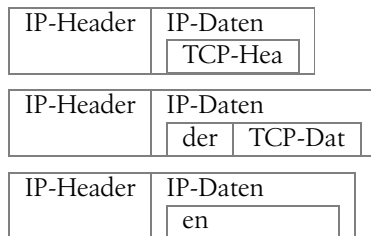
Stellt unser Angreifer fest, daß sich eine Firewall zwischen ihm und seinem Ziel befindet, so bedeutet dies nicht zwangsläufig, daß er schon verloren hat. Ist die Firewall nicht sorgfältig genug konfiguriert, so kann er sie unter Umständen durch die Benutzung frag-

mentierter IP-Pakete austricksen. Beispiele dafür finden sich in RFC 1858. Um dies zu verstehen, müssen wir noch einmal betrachten, wie TCP-Pakete über IP übertragen werden.<sup>10</sup>

Ein IP-Paket besteht aus einem Header, in dem z. B. die Quell- und die Zieladresse eingetragen sind, und einem Datenteil. Bei der Übertragung eines TCP-Paketes steht das eigentliche TCP-Paket im Datenteil des IP-Paketes. Auch das TCP-Paket hat wieder einen Header, in dem z. B. Quell- und Zielpport sowie die Folgenummern stehen, sowie einen Datenteil, in dem die Informationen der nächsthöheren Schicht stehen. Es ergibt sich also folgender Aufbau:



Soll nun ein IP-Paket fragmentiert werden, so wird sein Datenteil in mehrere Teilbereiche zerlegt, die jeweils mit einem IP-Header versehen und auf die Reise geschickt werden:



Wie wir an obigem Beispiel sehen können, kann es dabei vorkommen, daß der TCP-Header in mehrere Teile gesplittet wird. In ihm befinden sich aber wichtige Daten, die für die Filterung in der Firewall unerlässlich sind. Ihre Arbeit basiert u. a. auf Regeln der Art:

*»Wenn ein TCP-Paket einen Zielpport von 137 hat und das SYN-Flag gesetzt ist, das ACK-Flag aber nicht (d. h. ein Verbindungsaufbau zu Port 137), dann ist die Beförderung zu verweigern.«*

Derartige Regeln laufen ins Leere, wenn zum Zeitpunkt ihrer Auswertung entweder der Zielpport oder das CTL-Feld nicht vorhanden ist, da sich beide Felder in unterschiedlichen Teilpaketen befinden.

Ein anderer Angriff basiert darauf, wie Pakete wieder zusammgebaut werden. Jedes Teilpaket enthält im Header eine Angabe (Offset), an welcher Stelle des Originalpaketes sein Inhalt beginnt. Pakete der Art:

<sup>10</sup> Die folgenden Ausführungen gelten analog für UDP.

IP-Header Offset=0	IP-Daten Test,
-----------------------	-------------------

IP-Header Offset=5	IP-Daten 1,2,3,...
-----------------------	-----------------------

werden damit zu:

IP-Header	IP-Daten Test,1,2,3,...
-----------	----------------------------

Was passiert aber, wenn die Angabe »Offset« im zweiten Paket kleiner als 5 gewesen wäre?

Die Antwort hängt vom verwendeten Betriebssystem ab. Bei Linux würde das zweite Paket das erste überdecken und damit einen Teil der Information löschen. Aus

IP-Header Offset=0	IP-Daten Fragmentierung ist toll
-----------------------	-------------------------------------

IP-Header Offset=19	IP-Daten gefährlich!
------------------------	-------------------------

würde damit

IP-Header	IP-Daten Fragmentierung ist gefährlich!
-----------	--

Was in diesem Fall noch wie eine harmlose Spielerei erscheint, wird dann problematisch, wenn es sich bei den so manipulierten Daten um die Portangaben im TCP-Header handelt. Die Firewall sieht dann das erste Teilpaket, in welchem ein gültiger TCP-Header steht, der auf einen Zugriff auf einen erlaubten Port hinweist, und läßt dieses passieren. Die weiteren Teilpakete werden in der Regel nicht betrachtet. Erreichen die Pakete den Zielrechner, so werden sie dort wieder zusammengebaut, und das zweite Paket überlagert die Portangaben in einer Weise, daß aus dem Zugriff auf einen erlaubten Port mit einem Male ein Zugriff auf einen verbotenen Port wird. Damit wurde die Filterung in der Firewall unterlaufen.

Beide Angriffe funktionieren aus demselben Grund: Die Firewall sieht zu jedem Zeitpunkt nur einen Bruchteil der Informationen, die eigentlich nötig wären, um die Filterregeln richtig anzuwenden. Dies kann man vermeiden, indem man die Firewall anweist, Teilpakete erst zusammenzubauen, bevor sie untersucht und gegebenenfalls weitergeleitet werden. Wie dies praktisch geschieht, erfahren Sie in Kapitel 8, Unterabschnitt *Konfiguration des /proc-Dateisystems*, ab Seite 166.

### Unsichere Paßwörter

Selbst wenn keine inhärent unsicheren Dienste zugreifbar sind, unsere Server keine Programmierfehler aufweisen und die Firewall richtig konfiguriert ist, heißt dies nicht, daß damit alle Angriffspunkte beseitigt sind.

Betreiben wir einen Serverdienst, der die Angabe eines Passwortes verlangt, so könnte der Angreifer versuchen, sich als ein legitimer Benutzer auszugeben, indem er das Passwort errät. Da die die meisten Dienste bei mehrfachen Fehlversuchen entweder das benutzte Benutzerkonto sperren oder die Antwort verzögern, ist die Verwendung eines Programmes, das nacheinander alle Passwörter zwischen »a« und »ZZZZZZZZ« durchprobiert, nicht sinnvoll.

Unser Cracker kann sich aber die Tatsache zunutze machen, daß die meisten Benutzer recht bequem sind. Um zu vermeiden, daß sie ihr Passwort ständig vergessen, wählen sie eines, daß sie sich leicht merken können (Benutzername als Passwort, Name der Enkeltochter, Geburtsdatum, ...). Weiß der Angreifer, unter wessen Benutzerkonto er sich anmeldet, so hat er unter Umständen schon eine Reihe von Passwort-Kandidaten auf der Homepage des jeweiligen Benutzers gefunden. Ist dies nicht der Fall, kann er es immer noch mit bekannten Allerwelts-Passwörtern probieren (geheim, password, 4711, 0815, qwertzu, ...).

Schließlich kann er den Benutzer auch einfach fragen. Üblicherweise wird der Cracker sich dazu als Techniker ausgeben und behaupten, für Wartungsarbeiten das Passwort des Benutzers zu brauchen. Man sollte meinen, daß dies viel zu einfach ist, um funktionieren zu können, aber bei dem richtigen Auftreten ist diese *Social Engineering* genannte Technik erstaunlich wirkungsvoll.

Mit technischen Mitteln wird man dieser Angriffe nicht Herr. Hier hilft es nur, die Benutzer regelmäßig darauf hinzuweisen, ihre Passwörter absolut vertraulich zu behandeln, sie regelmäßig zu wechseln und Passwörter zu benutzen, die nicht einfach zu erraten sind.

## Rootrechte erlangen

Nachdem der Cracker in den Rechner eingedrungen ist, hat er unter Umständen zunächst nur begrenzte Rechte. Ist es ihm z. B. gelungen, einen Systemdienst in eine Hintertür umzuwandeln, so kann er nur Befehle mit den Benutzerrechten ausführen, unter denen der Dienst läuft. Für viele Aktionen braucht er aber Rootrechte.

Ein Weg, diese zu erlangen, ähnelt dem eigentlichen Einbruch in den Rechner. Es existieren Programme, die von normalen Benutzern aufgerufen werden können, die aber trotzdem Rootrechte besitzen (SUID-Programme). Gelingt es, so ein Programm so umzufunktionieren, daß es beliebige Befehle ausführt, so kann der Angreifer sich beliebige Privilegien verschaffen. Er kann z. B. ein neues Benutzerkonto anlegen, unter dem er über Rootrechte verfügt.

Es gab z. B. einmal den Fall, daß ein bekannter Editor auch E-Mails auf dem lokalen Rechner zustellen konnte. Da er dazu auf die Postfächer der Empfänger zugreifen mußte, brauchte er Rootrechte. Wie sich dann herausstellte, konnte er aber auch wichtige Systemdateien ändern, was es Angreifern erlaubte, durch das Ersetzen einer Datei selbst zu Root zu werden [34].

Ein klassischer Weg besteht im Knacken der Passwortdatei des Systems. Zwar sind die Passwörter dort in einer Einweg-Verschlüsselung abgelegt, die nicht dekodiert werden

kann, es ist aber möglich festzustellen, ob ein beliebiges Paßwort vom System in derselben Weise verschlüsselt wird wie eines, das in der Paßwortdatei eingetragen ist. Um nun ein Paßwort zu knacken, erstellt man eine lange Liste von Wörtern. Hierbei handelt es sich um Lexika, naheliegende Tastenkombinationen (z. B. qwertzu), Kombinationen daraus und Variationen der Groß- und Kleinschreibung. Heutzutage reicht selbst ein PC, um in erträglicher Zeit herauszufinden, ob ein Eintrag in der Liste als Paßwort verwendet wurde.

Aus diesem Grund verwenden moderne Unix-Systeme eine spezielle Datei (*Shadow-Datei*), die nur von Root lesbar ist. In der eigentlichen Paßwortdatei, die jedermann lesen darf<sup>11</sup>, ist das Paßwort durch einen ungültigen Wert ersetzt, während das richtige Paßwort in der Shadow-Datei steht.

Auch temporäre Dateien können für Angriffe genutzt werden. Legt z. B. ein Programm mit Rootrechten Daten in einer temporären Datei ab, deren Name der Angreifer erraten kann, so kann er im entsprechenden Verzeichnis einen symbolischen Link anlegen, der auf eine Systemdatei verweist. Viele schlecht programmierte Programme überschreiben dann nicht etwa den symbolischen Link, wenn sie ihre temporäre Datei anlegen, sondern die Datei, auf die er verweist.

Auf Systemen, auf denen Shellskripte ebenfalls als SUID-Programme<sup>12</sup> laufen können (Linux gehört nicht dazu), existiert auch noch die Möglichkeit, die Art und Weise zu manipulieren, wie die Shell Zeilen in einzelne Argumente aufspaltet. Die Zeichen, die einzelne Argumente eines Befehls trennen, sind nämlich nicht fest auf Leerzeichen beschränkt. Sie können durch das Setzen der Umgebungsvariablen IFS (Internal Field Separator) beliebig angepaßt werden. So könnte z. B. eine Konfigurationsdatei, die einzelne Felder kennt, welche durch »:« statt durch Leerzeichen getrennt werden, mit einem Shellskript eingelesen und korrekt verarbeitet werden.

Diese Fähigkeit der Shell läßt sich aber auch zu Angriffen heranziehen. Enthält ein Skript z. B. den Aufruf `/bin/date`, so könnte ein Angreifer den Feld-Separator (IFS) in `»/«` umdefinieren. Der Aufruf wäre dann gleichbedeutend mit `bin date`. Nun liegt es am Angreifer, ein Programm `bin` so zu plazieren, daß es auch tatsächlich ausgeführt wird.

Schließlich besteht für den Angreifer auch noch die Möglichkeit, einen Trojaner im System zu plazieren. Es wäre z. B. möglich, in einem Verzeichnis ein Programm namens `ls` abzulegen. Ein Benutzer, der sich das Verzeichnis auflisten lassen will, führt dann unwissentlich das Programm des Angreifers mit seinen eigenen Rechten aus.

Dies ist der Grund, warum Root normalerweise `»:«` nicht in seiner *PATH*-Variablen hat. Dadurch kann er Programme, die nicht in einem Standard-Verzeichnis liegen, nur ausführen, wenn er sie explizit mit Angabe ihres Pfades aufruft.

<sup>11</sup> Sie enthält neben dem Paßwort auch noch andere wichtige Informationen über den Benutzer.

<sup>12</sup> SUID steht für **S**et **U**ser **I**D. So markierte Programme werden immer mit den Rechten ihres Besitzers ausgeführt, unabhängig davon, wer sie aufruft.

## Sicherungsmaßnahmen

Will ein Angreifer den erworbenen Root-Zugang länger nutzen, so muß er nun beginnen, seine Spuren zu verwischen und das System so zu manipulieren, daß seine nächste Anmeldung sich einfacher gestaltet. Zuerst muß er dazu die Logdateien des Systems so manipulieren, daß möglichst keine Spuren seines Angriffs zurückbleiben. Welche Dateien sich dafür anbieten, werden wir in Kapitel 17 ab Seite 539 sehen, wenn es darum geht, wie man nach Spuren für einen erfolgten Einbruch sucht. Auch hier existieren fertige Programme, die es dem Angreifer erlauben, diese Manipulationen automatisiert durchzuführen.

Im nächsten Schritt geht es darum, erneute Besuche zu einem späteren Zeitpunkt vorzubereiten. Hier bietet sich eine Vielzahl von möglichen Ansatzpunkten. Zu den einfachsten gehört die Installation eines neuen Benutzerkontos mit der Benutzernummer 0. Obwohl dieses Konto einen völlig harmlosen Namen haben kann, ist es für das System dann doch mit root identisch. Es hat lediglich bei der Anmeldung ein anderes Paßwort.

Sind auf dem System R-Dienste installiert, so wäre es möglich, *.rhosts-* bzw. *host.equiv-* Dateien zu erzeugen, die es erlauben, sich von beliebigen Rechnern aus ohne Angabe eines Paßwortes als root einzuloggen.

Etwas unauffälliger ist die Installation eines Rootkits. Dieses enthält modifizierte Versionen von Systemprogrammen, die anstelle der Originale installiert werden. Prominente Beispiele für solche Programme wären z. B. *ls* und *ps*. Die modifizierten Varianten zeigen Prozesse und Dateien des Crackers nicht an. Dazu kommen noch Systemdienste, die sich im großen und ganzen wie die Originale verhalten, allerdings bei Eingabe eines bestimmten Paßwortes root-Zugang erlauben.

Ein derartig manipuliertes System wirkt für den unerfahrenen Betrachter absolut »sauber«, da alle Befehle ersetzt wurden, die ihm helfen könnten, veränderte Programme, verdächtige Prozesse oder ungewöhnliche Netzwerkverbindungen anzuzeigen. Hier hilft lediglich das Booten eines Rettungssystems von Diskette oder CD, das die nötigen Werkzeuge zur Spurensuche enthält.

Schließlich könnte der Cracker noch Vorkehrungen für den Fall treffen, daß er sich zu einem späteren Zeitpunkt zwar lokal anmelden kann, aber keine Rootrechte mehr besitzt. Hierfür kann er Sollbruchstellen in das System einbauen, indem er z. B. die Dateirechte ändert. Er könnte z. B. eine Kopie eines Kommandointerpreters erzeugen, ihr SUID-Bit setzen und als Besitzer root eintragen. Wenn dieses Programm für jedermann ausführbar ist, so reicht die Kenntnis ihres Pfades, um beliebige Befehle mit Rootrechten ausführen zu können. Setzt er dagegen die Schreibrechte auf die Paßwort- und u. U. Shadow-Datei entsprechend, so kann er jederzeit neue Benutzer einrichten, die dann bei Bedarf auch Rootrechte hätten.

Um solche Manipulationen erkennen zu können, ist es notwendig, eine Datenbank zu erstellen, die alle Dateien, ihre Rechte, Erstellungs-, Zugriffs- und Modifikationsdaten sowie kryptographische Checksummen über ihre Inhalte enthalten. Hierfür existieren Programme, *Checksummer* genannt, die in Kapitel 16, Abschnitt *Checksummer*, ab Seite 453 noch einmal genauer behandelt werden. Wichtig ist dabei allerdings, die Datenbank

vor einem Zugriff durch den Angreifer zu schützen. Sie sollte daher z. B. auf eine Diskette oder CD geschrieben werden, die dann an einem sicheren Ort aufgehoben wird. Hat man den Verdacht, das System könnte kompromittiert sein, so sollte man auch hier ein Rettungssystem starten, das neben einigen Systemprogrammen den Checksummer enthält.

Viele Systemadministratoren installieren zwar einen Checksummer und bilden dann eine Datenbank, halten diese aber auf dem System selbst. Dort wird sie dann dazu benutzt, das System regelmäßig zu prüfen. Dagegen ist im Prinzip nichts einzuwenden. Erfahrene Cracker bemerken dies aber und hebeln das System aus, indem sie nach ihren Manipulationen einfach eine neue Datenbank erzeugen oder den Checksummer durch eine Variante ersetzen, die besagte Manipulationen nicht meldet. Ohne eine Sicherheitskopie der Datenbank und ein geeignetes Rettungssystem mit nicht modifiziertem Checksummer sind solche Eingriffe nur schwer nachzuweisen.

## Lokale Aktivität

Der Angreifer ist jetzt absoluter Herrscher über das System. Obwohl es Cracker gibt, die sich mit dieser Tatsache zufriedengeben, liegt es doch nahe, den neu erworbenen Status auch zu nutzen. Beliebt ist hierbei das Verändern von Seiten eines Webservers. Viele Cracker sind begierig, die Öffentlichkeit an ihrem Erfolg teilhaben zu lassen, und möchten mit ihrer Tat Aufmerksamkeit erregen.

Diese Angriffe sind ziemlich offensichtlich und daher noch relativ harmlos. Problematischer wird es, wenn Cracker einen Rechner als Ressource für ihre eigenen dunklen Zwecke einsetzen. Es gab z. B. Fälle, in denen die Angreifer übernommene Rechner zu IRC-Servern umrüsteten, die sie zur Kommunikation mit anderen Mitgliedern ihrer Szene nutzten. Auch das Ablegen von Raubkopien zum freien Download wird immer wieder beobachtet. Da deren Verbreitung verboten ist, kann so der Besitzer des Rechners unfreiwillig eine Straftat unterstützen. Noch gravierender wird dieses Problem, wenn der/die Angreifer den Rechner als Ausgangsbasis für Angriffe auf andere Rechner benutzen. Ein Beispiel hierfür stellen die Angriffe auf Yahoo und eBay dar, bei denen bis zu 40 Rechner für koordinierte Denial-of-Service-Angriffe benutzt wurden [30].

Daß der Angreifer auch auf alle auf einem Rechner gelagerten Daten zugreifen und diese kopieren, manipulieren und löschen kann, sei nur der Vollständigkeit halber erwähnt. Dies versteht sich von selbst.

## Den nächsten Angriff vorbereiten

Unser Angreifer hat mittlerweile alles genutzt, was ihm der übernommene Rechner an Ressourcen zur Verfügung gestellt hat. Nun wird es Zeit, die Sachen zu packen und die Abreise zu anderen Rechnern des Netzes vorzubereiten. Ein Rechner, der sich im selben Netz wie das zukünftige Opfer befindet, bietet dabei eine günstige Ausgangsbasis.

### Lokales Ausspähen von Paßwörtern

Oft finden sich auf dem übernommenen Rechner Paßwörter für den Zugriff auf andere Systeme. Unter Unix bietet es sich an, unsichere Paßwörter in der Paßwort- oder Shadow-Datei zu knacken. Auch wenn ein Benutzer keine Rootrechte besitzt, so hat er vielleicht auf einem anderen Rechner im Netz ebenfalls ein Benutzerkonto, für das er das gleiche Paßwort benutzt.

In Windows-Systemen braucht sich ein Benutzer nur zu Beginn einer Sitzung anzumelden, wenn er dem System erlaubt, Paßwörter für den Zugriff auf andere Rechner lokal zu speichern. Diese gespeicherten Paßwörter können in der Regel problemlos ausgelesen werden. Auch speichern einige FTP-Klienten Paßwörter im Klartext in ihren Konfigurationsdateien.

Clifford Stoll berichtet in [34], wie ein Cracker auf seinem System E-Mails fand, in denen Dinge standen wie

*»Ich bin ein paar Wochen im Urlaub. Falls Du an meine Daten mußt, log Dich nur in meinen Account auf der VAX ein. Der Benutzername ist Wilson, das Paßwort Maryanna (das ist der Name meiner Frau). Viel Spaß!«*

*(Freie Übersetzung)*

Hilft dies alles nichts, so kann der Angreifer immer noch ein Programm installieren, das Tastatureingaben mitliest. Auf diese Weise kann er Paßwörter mitlesen, wenn sie vom Benutzer eingegeben werden.

### Ausnutzen von Vertrauensbeziehungen

In vielen Netzen existieren Vertrauensbeziehungen zwischen einzelnen Rechnern, so daß die erfolgreiche Authentisierung auf einem von ihnen zur Anmeldung auf einem anderen ausreicht. Die schon mehrfach geschmähten R-Dienste sind ein Beispiel dafür.

Ein anderes wäre ein Windows-Rechner mit freigegebenen Laufwerken. Solange er selbst keine Verbindung zum Internet hat, ist es nicht unbedingt nötig, diese Freigaben mit Paßwörtern zu schützen. Befindet sich im selben Netz ein zweiter Rechner, der mit dem Internet verbunden ist, aber keine Pakete für andere Rechner vermittelt (z. B. ein normaler Windows 98-Rechner), so ändert dies erst einmal nichts an der Situation für den ersten Rechner, da auf ihn immer noch nicht aus dem Internet zugegriffen werden kann. Wird jetzt aber der zweite Rechner von einem Angreifer übernommen, so kann der Cracker nun auf alle Rechner im lokalen Netz zugreifen, genauso wie dies auch ein Benutzer könnte, der physisch am zweiten Rechner säße. Die ungeschützten Freigaben des ersten Rechners, die vorher für ihn unerreichbar waren, präsentieren sich nun auf dem Silbertablett.

### Sniffing

Kaum jemand, der sich mittels FTP Dateien herunterlädt, sich mit Telnet auf anderen Rechnern anmeldet oder mit POP3 seine E-Mail abholt, stellt sich die Frage, in welcher

Weise Benutzerkennung und Paßwort zum Zielsystem übertragen werden. Dabei ist es normalerweise so, daß diese Angaben im Klartext über das Netz wandern.<sup>13</sup> Hier stellt sich für einen Angreifer natürlich die Frage, ob es nicht eine Möglichkeit gibt, die übertragenen Datenpakete mitzulesen und die in ihnen enthaltenen Daten für Angriffe zu nutzen.

Diese Methode, an sensible Daten zu gelangen, nennt sich *Sniffing*. Hierzu ist es notwendig, daß schon ein Rechner kompromittiert und so umkonfiguriert wurde, daß alle Daten auf einem Netzsegment mitprotokolliert werden. Ein Netzsegment ist dabei der Bereich des Netzes, in dem die angeschlossenen Rechner alle Pakete mitlesen können, die von einem anderen Rechner desselben Segmentes gesendet werden. Ein Beispiel dafür ist ein Koax-Ethernetkabel, an das alle Rechner des lokalen Netzes angeschlossen sind. Wenn ein Rechner sendet, so empfangen diese Signale alle Rechner am Kabel. Technisch nennt man dies eine *Bus-Verkabelung*.

Stellt man das Ethernet von 10 MBit auf 100 MBit um oder besitzt man moderne Netzwerkkarten ohne BNC-Anschluß, so liegt eine *Stern-Verkabelung* vor. Hier gehen alle Leitungen sternförmig von einer zentralen Komponente aus. Bei dieser handelt es sich normalerweise um einen *Switch* oder einen *Hub*. Ein Hub setzt dabei einfach nur alle Signale, die er auf einem Eingang empfängt, auf alle anderen Ausgänge um. Hierbei ist aus unserer Sicht der Unterschied zur Bus-Verkabelung nicht allzu groß.

Anders liegt der Fall beim Einsatz von Switches anstelle von normalen Hubs, da diese die Pakete direkt vom Quellrechner zum Zielrechner befördern, ohne daß weitere Rechner sie zu sehen bekommen. Hier wird ein Sniffing erst einmal verhindert. Zwar existieren immer noch Angriffe, die einen Rechner dazu bringen können, seine Pakete statt an den eigentlichen Zielrechner an den Rechner des Angreifers zu senden, diese sind aber deutlich aufwendiger und erfordern es, spezielle Datenpakete zu senden, womit der Angreifer seinen größten Vorteil, die Unsichtbarkeit seines Angriffes, aufgibt.

Der Einsatz von Switches bietet allerdings nur Schutz gegen Sniffer im eigenen LAN, vor Sniffen auf dem Rechner des Providers oder an einem Backbone können wir uns nicht schützen. Dies sollte auch bedacht werden, wenn im Internet Protokolle verwendet werden, bei denen Paßwörter unverschlüsselt übertragen werden.

### Spoofting

Unter Spoofting versteht man, wenn der Angreifer sich als jemand anderen ausgibt. In diesem Fall sind Angriffe gemeint, bei denen der Angreifer vortäuscht, der von ihm kontrollierte Rechner sei eigentlich ein ganz anderer.

**Spoofting der MAC-Adresse** Bei dem in [37] beschriebenen Angriff sendet der Angreifer Pakete mit gefälschter MAC-Adresse und gibt sich so z. B. als der lokale Mailserver aus.

<sup>13</sup> Natürlich existieren Protokollerweiterungen (z. B. APOP), Alternativprotokolle (z. B. SSH) und zusätzliche Sicherheitsmaßnahmen (z. B. Tunnelung mittels SSL), um dieses Problem zu beseitigen, praktisch werden diese aber immer noch nicht flächendeckend eingesetzt.

Sinn dieses Vorgehens ist, den Schutz auszuhebeln, den Switches vor dem Sniffing von Paßwörtern bieten. Ein Switch funktioniert, indem er Buch darüber führt, welche MAC-Adressen auf seinen jeweiligen Anschlüssen verwendet werden. Dabei können an einem Anschluß mehrere Adressen verwendet werden, wenn dort nicht ein einzelner Rechner, sondern wiederum ein Switch oder Hub angeschlossen ist. Der Switch muß also für jeden Anschluß eine Liste mit allen Adressen führen, die dort in letzter Zeit verwendet wurden.

Mit dieser Liste kann er Nachrichten gezielt auf den Anschluß weiterleiten, an dem der gemeinte Empfänger angeschlossen ist. Dies würde normalerweise bedeuten, daß ein Angreifer keine Chance hat, die Anmeldung an einem Server zu beobachten.

Um trotzdem zum Ziel zu kommen, beginnt der Angreifer Pakete zu senden, die als Absender die MAC-Adresse des zu beobachtenden Servers enthalten und an die reale Adresse des Rechners des Angreifers gerichtet sind. Der Switch wird nun feststellen, daß an einem seiner Eingänge sowohl der Mailserver als auch der Rechner des Angreifers angeschlossen sind. Auf den anderen Anschlüssen wird man von dem Angriff nichts merken, da die Pakete für einen Rechner bestimmt sind, der sich aus Sicht des Switches am selben Anschluß befindet. Die Pakete werden daher nicht an andere Anschlüsse weitergeleitet.

Sendet nun ein Klient eine Anfrage an den Server, so wird der Switch das Paket statt an den Server an den Angreifer weiterleiten, der nun den Klienten nach einem Paßwort fragt. Eine Weiterleitung der Anfrage zu dem eigentlichen Server ist nicht möglich, da dieser effektiv vom Rest des Netzes abgeschnitten wurde. Der Angreifer kann daher nicht den vom Server erbrachten Dienst vorspiegeln. Er kann aber nach der Paßwortabfrage mit einer Fehlermeldung abbrechen, ohne allzu verdächtig zu wirken. Nun besitzt der Angreifer das Paßwort, mit dem er später, wenn die Tabelle des Switches wieder zum Normalzustand zurückgefunden hat, selbst auf den Server zugreifen und sich mit dem erbeuteten Paßwort als sein Opfer ausgeben kann.

Ein anderer Ansatz zielt darauf ab, so viele gefälschte Pakete zu generieren, daß die Tabelle des Switches nicht ausreicht, um alle erhaltenen Adressen zu speichern. In diesem Fall wird der Switch darauf verzichten, Pakete gezielt zuzustellen, sondern grundsätzlich alle Pakete an alle Ausgänge weiterleiten. Er wird damit de facto zum Hub. Nun kann ein Angreifer problemlos den kompletten Verkehr im Netz mit einem Sniffer mitlesen.

Einige Switches bieten Möglichkeiten, Sicherheitschecks zu implementieren. Dies läuft aber oft darauf hinaus, fest zu definieren, welche MAC-Adressen auf welchen Anschlüssen des Switches erlaubt sind. In einem größeren Netz ist dies oft nicht praktikabel. Trotzdem sollten Sie sich vergewissern, welche Möglichkeiten Ihnen in Ihrem konkreten Fall zur Verfügung stehen, und abwägen, ob ihr Einsatz sinnvoll ist.

**IP-Spoofing und Source Routing** Ein weiteres Problem ist die Tatsache, daß es einem Angreifer prinzipiell möglich ist, Pakete mit einer beliebigen IP-Adresse zu verschicken. Dies wird dann gefährlich, wenn ein Dienst die Quelladresse eines Paketes zur Entscheidung dafür heranzieht, wie mit den übermittelten Daten zu verfahren ist. Ein Beispiel sind die R-Dienste unter Unix, die es einem Benutzer erlauben, sich ohne Angabe eines Paßwortes am System anzumelden. Sie vertrauen darauf, daß der Rechner, von dem aus die Anmeldung erfolgt, den Benutzer schon authentisiert hat.

Dieses Vertrauen beschränkt sich normalerweise nur auf eine kleine Gruppe von Rechnern. Gelingt es aber dem Angreifer, sich als einen von diesen auszugeben, so kann er sich an einem System anmelden, für dessen Benutzung er keine Berechtigung hat.

Dieser Angriff hat einen gravierenden Schönheitsfehler. Die Antwortpakete werden grundsätzlich an den vorgeblichen Sender und nicht an den Angreifer geschickt. Zwar gibt es durchaus Situationen, in denen die Antwort nicht so wichtig ist<sup>14</sup>, für TCP-Verbindungen gilt aber, daß beim Verbindungsaufbau die Folgenummer der Gegenseite bestätigt werden muß. Gegenwärtig sind drei Methoden bekannt, wie dies umgangen werden kann:

1. Sniffing
2. Raten
3. Source Routing

Sniffing haben wir ja bereits weiter vorne in diesem Kapitel kennengelernt. Natürlich ist es nicht darauf beschränkt, Namen und Paßwörter auszuspähen. Es kann genauso gut dazu verwendet werden, die Folgenummern aus Netzwerkpaketen zu lesen. Allerdings ist diese Methode nur dann wirklich praktikabel, wenn der Angreifer Zugang zu einem Rechner im Netzwerkstrang des Zielrechners hat oder falls er einen der Rechner kontrolliert, über den der Verkehr zu dem Rechner geroutet wird, dessen Identität er vorgibt.

Einfacher wird es für ihn, wenn er die Folgenummern nicht wirklich sehen müßte, sondern sie relativ einfach erraten könnte. Auch wenn es erschreckend ist, so war dies lange Zeit möglich, da die Mechanismen zur Erzeugung der Folgenummern recht einfach waren. Erst nach einigen Einbrüchen in prominente Systeme wurden Patches herausgebracht, die diese Lücke schlossen. Linux ist nicht mehr anfällig.

Aus Sicht des Angreifers bestünde die beste Lösung sicherlich darin, wenn er in der Lage wäre, die Antworten auf seine Pakete zu sehen, obwohl die Absenderangabe falsch ist. Hier kommt das schon erwähnte Source Routing ins Spiel. Da hierbei nicht nur Absender und Empfänger, sondern auch Zwischenstationen angegeben werden können, kann der Angreifer Pakete senden, deren Header folgende Information enthalten:

*»Dieses Paket stammt von rechner\_a.trusted.org, ist bestimmt für rechner\_b.victim.com und muß über big.bad.attacker.net geroutet werden.«*

Die Antwort auf so ein Paket wird zwar formal an *rechner\_a.trusted.org* geschickt, landet aber im ersten Schritt bei *big.bad.attacker.net*, von dem erwartet wird, daß es an *rechner\_a.trusted.org* weiterleitet. Da es sich hier aber um keinen normalen Router, sondern den Angreifer handelt, wird dies nicht geschehen. Statt dessen hat unser Angreifer sein Ziel erreicht.

<sup>14</sup> Angenommen, man weiß, daß ein syslog-Server für Speicherüberläufe anfällig ist, dann reicht es, ihm ein UDP-Paket zu schicken, um ihn dazu zu bringen einen Befehl auszuführen, der dem Angreifer einen Weg in den Rechner öffnet. Eine Antwortnachricht wird der Server nicht senden, da das syslog-Protokoll dies nicht vorsieht. Aber der Angreifer kann nun ausprobieren, ob sein Angriff erfolgreich war, indem er probiert, durch den von ihm geschaffenen Weg in den Rechner einzudringen.

Diese Problematik ist relativ bekannt. Viele Router im Internet leiten Pakete mit einer Source Route daher nicht weiter. Wir werden in Kapitel 8, Unterabschnitt *Kernelkompilation*, ab Seite 150 sehen, wie wir den Kernel entsprechend konfigurieren.

### Man-in-the-Middle-Attacks

Einen Schritt weiter gehen *Man-in-the-Middle-Attacks*. Bei diesen Angriffen versucht der Cracker sich in eine Verbindung zwischen zwei Kommunikationspartnern zu drängen. Dabei gibt er jedem Gesprächsteilnehmer gegenüber vor, der jeweils andere zu sein.

Um zu verstehen, welches Potential ein solcher Angriff bietet, stellen wir uns einmal vor, ein Kunde wollte bei einem Internethändler einkaufen. Diese Verbindung würde normalerweise so ablaufen, daß er mit dem Händler einen geheimen Schlüssel aushandelt, der dann dazu dient, eine »sichere« Verbindung aufzubauen. Gelingt es dem Angreifer, sich zwischen Händler und Kunde einzuschalten, so könnte er mit dem Kunden einen eigenen Schlüssel aushandeln, die Bestellung und die Zahlungsinformationen (z. B. Kreditkartendaten) entgegennehmen und die Bestellung an den Händler weiterleiten, gegenüber dem er sich dann als der Kunde ausgeben müßte. Auf diese Weise kann der Angreifer in den Besitz von Daten gelangen, die er normalerweise wegen der verwendeten Verschlüsselung nicht lesen könnte. Er könnte auch die Bestellung beliebig manipulieren.

**ARP Cache Poisoning** Eine Methode, die es erlaubt, Netzwerkverbindungen umzuleiten oder sich gar in bestehende Verbindungen einzuschalten, ist das *ARP Cache Poisoning* oder *ARP Spoofing*. Im Gegensatz zum vorher genannten Angriff funktioniert diese Methode unabhängig davon, ob eine Bus- oder eine Sternverkabelung eingesetzt bzw. Switches oder Hubs verwendet werden.

Hierzu setzt der Angreifer gefälschte ARP-Pakete ein. Um den Vorgang zu verstehen, müssen wir uns vergegenwärtigen, wie ARP funktioniert. Wenn IP ein Paket an einen anderen Rechner weiterleiten will, muß es dazu die IP-Adresse des Zielrechners in eine MAC-Adresse umsetzen. Hierzu verwendet es eine Tabelle, die beide Adressen einander zuordnet.

Diese Tabelle, der ARP Cache, muß aber erst einmal gefüllt werden. Um nicht jede MAC-Adresse auf jedem Rechner von Hand zu konfigurieren, wird der ARP Cache gefüllt, indem eine ARP-Anfrage an alle Rechner im lokalen Netz gesendet wird, wann immer eine IP-Adresse im Cache nicht gefunden werden kann. Der Rechner, dem die gesuchte IP-Adresse gehört, wird dann antworten und dem Fragenden seine MAC-Adresse mitteilen. Damit kann die Adresse im Cache eingetragen werden und steht beim nächsten Mal zur Verfügung, so daß nicht noch einmal gefragt werden muß. Da sich IP-Adressen auch einmal ändern können, werden Einträge im Cache nach einer bestimmten Zeit wieder gelöscht.

Damit stellt sich die Frage, was einen dritten Rechner davon abhält zu behaupten, er wäre der Besitzer der gesuchten IP-Adresse. Die Antwort lautet: »Nichts!« Tatsächlich ist es für einen Angreifer nicht einmal nötig zu warten, bis eine Anfrage gestellt wird. Er kann vielmehr jederzeit gefälschte ARP-Pakete schicken, und sein Opfer wird diese

Informationen in den ARP Cache aufnehmen. Damit ist die Grundlage für eine Reihe von Angriffen gelegt.

Sendet ein Angreifer von Rechner A aus ARP-Antwortpakete an einen Rechner B, in denen behauptet wird, seine MAC-Adresse gehöre zur IP-Adresse von Rechner C, so wird diese Information im ARP Cache von B gespeichert. Nun wird B jegliche für C bestimmten Pakete an A schicken. Schickt A Pakete, die als Absenderadresse die IP-Adresse von C enthalten, so wird B annehmen, sie kämen von C. Nun muß er den Vorgang nur noch mit C wiederholen, um so beide Richtungen des Datenaustauschs zwischen B und C zu kontrollieren.

Obwohl das ARP Cache Poisoning technisch nicht trivial ist, müssen wir damit rechnen, daß ein Angreifer es anwendet. Es existieren im Internet mehrere Programme, die den Vorgang automatisieren.

Aus Sicht des Angreifers besteht die wichtigste Einschränkung sicherlich darin, daß er nur die ARP Caches von Rechnern manipulieren kann, die sich im gleichen lokalen Netz wie er selbst befinden. Ist es ihm aber erst einmal gelungen, in einen Rechner eines bestimmten Netzes einzudringen, so kann er mit ARP Cache Poisoning die Kontrolle über jeglichen Verkehr zwischen den Rechnern gewinnen, die direkt mit dem von ihm kontrollierten verbunden sind. Auch der Einsatz von Switches hilft gegen diesen Angriff nicht.

Um sich zu schützen, könnte man theoretisch die ARP Caches der einzelnen Rechner von Hand mit statischen Zuordnungen versehen. Dies dürfte aber in größeren Netzen nicht praktikabel sein. Damit bleibt nur Intrusion Detection-Systeme (IDS) einzusetzen, d. h. spezielle Programme, die den Netzverkehr überwachen und nach verdächtigen Paketen absuchen. Mit ihnen ist es möglich, ARP-Angriffe zu erkennen.

Auch könnte man die ARP Caches auf verschiedenen Rechnern überwachen und miteinander vergleichen bzw. auf Abweichungen von erwarteten Werten überprüfen. Letztlich wird einem dies aber nur etwas nützen, wenn sich der Angreifer im gleichen lokalen Netz befindet wie man selbst. Ist er aber in das Netz eines Serverbetreibers im Internet eingebrochen und kontrolliert dort den Verkehr zwischen einem Webserver und dem für ihn zuständigen Router, so hat man keine Möglichkeit, dies festzustellen.

Es bleibt nur, für sicherheitsrelevante Vorgänge Protokolle zu verwenden, bei denen man sicher feststellen kann, mit welchem Kommunikationspartner man es zu tun hat. Wenn man z. B. Kreditkartendaten über eine SSL-gesicherte Verbindung sendet, dann sollte man sich nicht darauf verlassen, daß die Daten sicher sind, weil der Browser mit einem goldenen Schloß anzeigt, daß die Verbindung verschlüsselt wird. Erst wenn man auch überprüft hat, welcher Name in dem verwendeten Zertifikat steht, weiß man, ob man wirklich den Schlüssel des gemeinten Servers oder den eines Angreifers verwendet.

**Angriffe mittels ICMP** Eine andere Klasse von Man-in-the-Middle-Attacks zielt darauf ab, mittels ICMP das Routing eines Rechners zu manipulieren. Es existiert z. B. eine Fehlermeldung »Redirect«, die ein Router sendet, wenn er feststellt, daß ein anderer Router besser geeignet wäre, ein bestimmtes Paket weiterzuleiten. Der Sender des Paketes wird es dann noch einmal an den angegebenen Ersatzrouter senden und auch versuchen, Folgepakete der Verbindung ebenfalls über diesen Ersatzrouter zu verschicken. Handelt es

sich dabei um einen Rechner, den der Angreifer kontrolliert, so hat er schon sein Ziel erreicht. Er sieht jedes Paket der Verbindung und kann es auch bei Bedarf manipulieren.

Auch das ICMP Router Discovery Protocol lässt sich für Angriffe nutzen. Hierbei sendet der Opferrechner »Router Solicitation«-Pakete an die Multicast-Adresse 224.0.0.2, um die Adresse des zuständigen Routers zu erfragen. Erreichen die Pakete einen Router, so antwortet dieser mit einem »Router Advertisement«-Paket, das eine oder mehrere Router-Adressen enthält. Da hier jeder Rechner im lokalen Netz antworten kann, kann ein Angreifer vorgeben, ein von ihm kontrollierter Rechner wäre ein Router. Dies kann dazu führen, daß der Opferrechner alle Pakete, die nicht für Rechner im lokalen Netz bestimmt sind, an den Rechner des Angreifers sendet.

**Telnet-Hijacking** Beim Telnet-Hijacking versucht der Angreifer, eine bestehende TCP-Verbindung zu übernehmen. Ist er erfolgreich, so kann er Pakete an die Teilnehmer einer Verbindung senden, als wäre er die jeweilige Gegenstelle, während die beiden ursprünglichen Kommunikationspartner ihre Pakete gegenseitig nicht mehr annehmen. Es ist, als hätte der Angreifer eine Telefonleitung zwischen den beiden durchgeschnitten und an jedes der beiden nun offenen Enden ein eigenes Telefon angeschlossen. Die beiden Rechner können nur noch miteinander kommunizieren, wenn er die Nachrichten entgegennimmt und dann selbst an den Empfänger weiterleitet.

Die technischen Details werden ausführlich in [22] abgehandelt. Der Angriff basiert darauf, eigene Pakete in eine bestehende Verbindung einzuschleusen und so die Register für die TCP-Folgenummern der Kommunikationspartner zu desynchronisieren. Dadurch wird jedes echte Paket eines Kommunikationspartners als schon gesendet angesehen und verworfen.

Abbildung 4-1 verdeutlicht den Vorgang der Desynchronisation in einer Richtung. Zu Beginn sendet Rechner A ein Paket mit der Folgenummer 180 und 20 Bytes Inhalt. Rechner B erwartet nun als nächstes ein Paket mit der Folgenummer 200. Dies erhält er dann auch, allerdings nicht von Rechner A, sondern vom Angreifer. Rechner B erwartet nun als nächstes ein Paket mit der Folgenummer 230. Dies teilt er Rechner A mit. Dessen interner Zähler für Folgenummern steht aber erst auf 200. Er wird daher eine Bestätigung mit der Folgenummer 200 senden, um die Gegenstelle darauf hinzuweisen. Rechner B erwartet aber ein Paket mit der Folgenummer 230. Er wird also seine Bestätigung noch einmal senden, um A dazu zu veranlassen, neue Pakete zu schicken und nicht alte zu wiederholen. Dies führt zu einem Teufelskreis, den *ACK-Storms*, der erst durchbrochen wird, wenn das Netzwerk überlastet ist und ACK-Pakete verlorengehen.

Nun kann der Angreifer weitere Pakete an Rechner B senden. Da er im Gegensatz zu Rechner A die richtigen Folgenummern kennt, wird Rechner B sie entgegennehmen. Er wird sie allerdings auch bestätigen, wodurch erneute ACK-Storms ausgelöst werden. Damit ist die Übertragung zwischen den Rechnern allerdings erst in eine Richtung unterbrochen. Konsequenterweise müsste der Angreifer noch ein gefälschtes Paket von Rechner B an Rechner A senden, um auch die Zähler für die Gegenrichtung zu manipulieren. Danach können die Rechner nicht mehr kommunizieren. Pakete können nur ausgetauscht werden, wenn der Angreifer für jedes Paket einer der Parteien ein weiteres erzeugt, das tatsächlich die Folgenummer enthält, die der Empfänger erwartet.

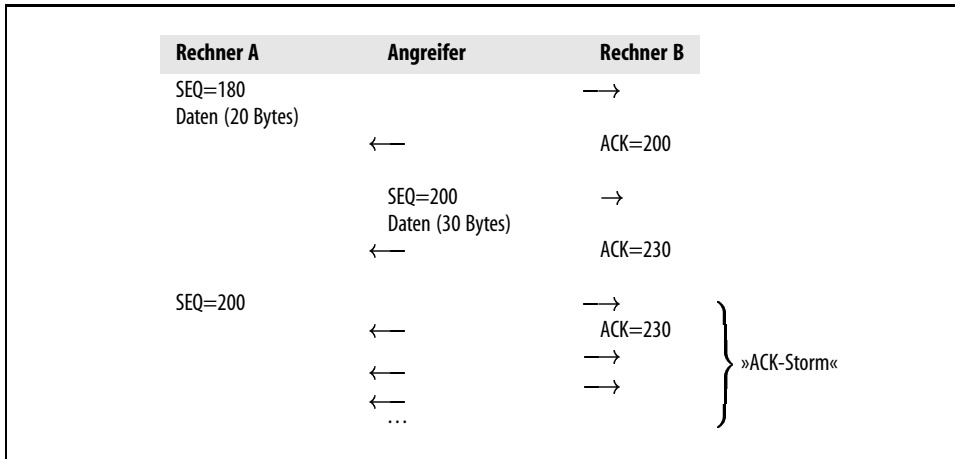


Abbildung 4-1: Telnet-Hijacking – Desynchronisation

Durchführbar ist so ein Angriff allerdings wie beim Sniffing nur, wenn der Angreifer direkten Zugriff auf einen Netzwerkstrang hat, über den die Pakete zwischen Rechner A und B geroutet werden. Auch die Gegenmaßnahmen entsprechen denen gegen Sniffing. Eine sternförmige Verkabelung mit Switches verhindert, daß ein Angreifer, der einen Rechner im LAN kompromittiert hat, die Verbindung zwischen zwei anderen Rechnern beobachten kann. Der Angriff ist damit nicht mehr möglich.

**DNS-Spoofing** Der letzte Man-in-the-Middle-Angriff, der hier vorgestellt werden soll, ist das DNS-Spoofing. Üblicherweise wird er nicht in lokalen Netzen, sondern im Internet eingesetzt. Da es sich aber um einen Man-in-the-Middle-Angriff handelt, erwähne ich ihn trotzdem an dieser Stelle.

Beim DNS-Spoofing versucht der Angreifer, den Cache eines DNS-Servers so zu verändern, daß dieser bei Anfragen nach einem bestimmten Rechner eine falsche Antwort gibt [31],[24]. Möglich wird dies, indem er eine Nachfrage des DNS-Servers nach einer bestimmten Adresse provoziert und dann eine Antwort schickt, bevor dies der eigentlich zuständige Server tut.

Im Extremfall kann der Angreifer auch einen Server aufsetzen, der neben der eigentlichen Antwort auf eine Anfrage auch noch eine weitere Adreßinformation mitliefert, die mit der eigentlichen Anfrage nicht das geringste zu tun hat. Nun wird er den zu manipulierenden Server nach einer Adresse, für die der von ihm kontrollierte Server zuständig ist, fragen. Der Opferrechner wird darauf den speziellen Server nach der Adresse befragen und erhält mit der Antwort gleich noch die Adresse für einen völlig anderen Rechner. Manche Server speichern nun diese zusätzliche Information und benutzen sie, um folgende Anfragen zu beantworten.

Handelt es sich bei dem so verfälschten Eintrag z. B. um die Adresse einer Online-Bank, so kann ein derartiger Angriff für den Angreifer profitabel sein. Leider gibt es für uns keine Möglichkeit, uns gegen diese Klasse von Angriffen zu schützen, da es sich bei dem

betroffenen DNS-Server in der Regel um einen Rechner handelt, der nicht von uns kontrolliert wird. Es bleibt uns also nur, die Benutzer unseres Netzes auf die prinzipiellen Unsicherheiten des Internets hinzuweisen.

## Würmer

Eine logische Weiterentwicklung der automatisierten Angriffskripte, die von weniger erfahrenen Crackern verwendet werden, sind Programme, die nicht nur einen kompletten Angriff durchführen, sondern sich auch auf dem Zielrechner installieren, um von dort aus Angriffe auf weitere Rechner zu starten. Ein solches Programm, das sich selbstständig über Computernetzwerke von Rechner zu Rechner weiterverbreitet, ohne daß ein Benutzer dabei eingreifen muß, nennt man *Wurm*.

Aus Firewall-Sicht besteht kein Unterschied zwischen dem Angriff eines Crackers und dem eines Wurms. In beiden Fällen wird versucht Sicherheitslücken, in Netzwerkdiensten auszunutzen. Dies ist genau die Sorte Angriff, gegen die eine Firewall schützt.

Allerdings ist der Kreis der möglichen Opfer für Würmer größer als beim Cracking. Dadurch, daß jeder infizierte Rechner selbst wieder als Ausgangsbasis für Angriffe dient, wird es möglich, das ganze Internet in relativ kurzer Zeit nach anfälligen Rechnern zu durchsuchen.

Dies macht Würmer auch für das organisierte Verbrechen interessant. Waren früher Würmer in erster Linie ein Mittel, um aufzufallen, so installieren heutige Würmer immer häufiger Hintertüren, die es ihrem Besitzer erlauben, die Kontrolle über den Rechner zu übernehmen [72]. Nach einer Zeit steht dem Angreifer ein ganzes Netzwerk von Rechnern (*Botnet*) zur Verfügung, das er nach Wunsch für Denial-of-Service-Angriffe, das Versenden von Spam oder das Hosting von Webseiten verwenden kann, die vorgeben, Banken zu gehören und den arglosen Benutzer nach Kontoinformationen fragen [70].

Aber auch das Ausspionieren von Daten ist durchaus eine beliebte Anwendung für Würmer. So sind bereits Würmer aufgetreten, die Sniffer enthalten, die den Netzwerkverkehr abhören und gezielt nach geldwerten Daten filtern. In einem Fall wurde z. B. nach Anmeldedaten für das Bezahlssystem PayPal gesucht [73]. Auch wurde ein Fall bekannt, in dem ein Wurm CD-Keys für Spiele auslas [72].

## Trojaner

Sind auf einem Rechner keine Serverdienste installiert, so ist normalerweise jeder Versuch des Crackings zum Scheitern verurteilt. In Ermangelung eines Programms, das er dazu bringen kann, für ihn tätig zu werden, könnte der Angreifer auf die Idee kommen, selbst ein Programm zu schreiben, das die Dinge tut, die er auf dem Rechner des Opfers tun möchte. Dieses könnte z. B.

- beleidigende Dialogboxen anzeigen [54],
- die Startseite des Browsers ändern [47],

- die Registry manipulieren, um diverse Systemfunktionen (z. B. Abmeldung, Taskmanager) unbenutzbar zu machen [54],
- im Hintergrund laufende Antivirusprogramme beenden [55],
- seinen Schöpfer darüber informieren, daß ein neuer Rechner infiziert wurde [46][53],
- sich so installieren, daß es bei jedem Rechnerstart oder bei der Anmeldung eines bestimmten Benutzers automatisch gestartet wird [46][47][49][50][51][55],
- beliebige Dateien aus dem Internet auf den Rechner des Opfers herunterladen und installieren (wird z. T. als Update-Mechanismus für den Trojaner genutzt) [45][53][55],
- sich selbst an weitere Opfer per E-Mail versenden (*Kettenbrief-Trojaner*, auch *E-Mail-Würmer* genannt) [48][51][55],
- Dateien in Verzeichnissen von File-Sharing-Klienten mit Kopien seiner selbst ersetzen, um sich so weiter zu verbreiten [51][54][55],
- Dateien ausspionieren und/oder löschen [46][54],
- Tastatureingaben mitprotokollieren [46][47][55],
- gecachte Paßwörter auslesen [46],
- eine eventuell an den Computer angeschlossene Kamera benutzen [46],
- versuchen, durch das Anzeigen manipulierter Webseiten an Authentisierungsdaten für das Online-Banking zu gelangen [50],
- kostenpflichtige Telefonnummern anrufen (*Dialer*),
- Systemdateien (z. B. hosts-Datei, Registry) manipulieren, um den Benutzer auf eigene Webseiten umzuleiten [52],
- bei Empfang eines bestimmten Netzwerkpaketes DoS-Angriffe gegen andere Rechner ausführen (*DDoS-Zombies*) [45][49],
- als Proxy dienen, um den tatsächlichen Urheber von Netzzugriffen zu verschleiern [53],
- von Ihrem Rechner aus Spam versenden [56][57],
- als Reverse Proxy Web-Anfragen entgegennehmen und unbemerkt vom Anfragenden an den eigentlichen Webserver weiterleiten. Auf diese Weise befindet sich auf Ihrem Rechner scheinbar eine Website, auf der man Mitglied bei kostenpflichtigen Pornoseiten werden kann [57].
- Kinderpornographie auf Ihren Rechner herunterladen<sup>15</sup> [58][59],
- sich nach getaner Arbeit selbst löschen, um die Spuren zu verwischen [53].
- einen Dienst installieren, der vom Angreifer beliebige Befehle entgegennimmt und ausführt (*Remote Shell*) [46][55].

<sup>15</sup> In den zitierten Fällen landete das Opfer vor Gericht. Inzwischen soll es schon Erpresser geben, die damit drohen, einen Rechner mit Kinderpornos zu versehen und dann die Polizei zu rufen.

Es ist allerdings relativ unwahrscheinlich, daß ein Anwender dieses Programm ausführt, wenn der Angreifer ihm offen sagt, wozu es dient. Deswegen ist eine Tarnung nötig. In meiner privaten Sammlung befindet sich z. B. ein Spiel, in dem man einen bekannten ehemaligen Vorsitzenden einer großen Softwarefirma mit Torten bewerfen kann. Dieses Spiel installiert im Hintergrund eine Remote Shell, die fortan bei jedem Systemstart automatisch ausgeführt wird. Wer sich mit ihr verbindet, kann den PC, auf dem sie installiert ist, fast so gut bedienen, als säße er direkt davor. Es ist kein Problem, Dateien zu transferieren, auf dem Bildschirm des ahnungslosen Benutzers Fehlermeldungen erscheinen zu lassen, Tastatureingaben mitzulesen oder das CD-ROM-Laufwerk in regelmäßigen Abständen zu öffnen und zu schließen.

Das beschriebene Programm arbeitet unter Windows, ähnliche Programme existieren aber auch unter Linux. In einigen Fällen gelang es böartigen Zeitgenossen sogar, modifizierte Versionen von Software in Umlauf zu bringen, die normalerweise dazu dient, den Zugriff auf Rechner zu kontrollieren. So wurde z. B. das Login-Programm so verändert, daß bei der Eingabe eines bestimmten Paßwortes grundsätzlich Zugang zum System gewährt wurde.

Solche Programme, die eine verborgene, vorsätzlich böartige Funktion enthalten, nennt man *trojanische Pferde* oder kurz *Trojaner*<sup>16</sup>.

Üblicherweise gelangen sie auf die folgenden Wegen auf Ihren Rechner:

- Sie erhalten eine E-Mail mit einer angehängten Datei. Beim Starten des Anhangs wird der Trojaner installiert. Dabei existieren insbesondere unter Windows diverse Tricks, um zu verschleiern, daß Sie ein Programm ausführen. So zeigt der Explorer bekannte Dateiendungen nicht an. Aus einer Datei *Beispiel.jpg.exe* wird so *Beispiel.jpg*, und es wird der Eindruck erweckt, es handle sich um eine Graphik. Auch existieren diverse Dateierarten, die geeignet sind, Programme zu enthalten. Sie kennen sicher die Endungen *.bat*, *.com* und *.exe*, aber wie steht es mit *.cmd*, *.pif* und *.scr*? Auch Dateien von Office-Anwendungen wie Word und Excel lassen sich dank der eingebauten Makro-Sprachen zu solchen Zwecken verwenden.
- Sie laden ein Programm von einem Server herunter, in den eingebrochen wurde. Die Angreifer haben die dort gelagerten Programme durch trojanisierte Versionen ersetzt [40].
- Sie besuchen eine Website, die ein ActiveX-Control<sup>17</sup> enthält, das den Trojaner automatisch installiert [42]. Diese Methode wird gerne von Dialern verwendet. Sie funktioniert allerdings nur, wenn die Webseite mit dem Internet-Explorer betrachtet wird.
- Sie betrachten eine Webseite mit einem Browser, der eine Sicherheitslücke besitzt, die dazu genutzt werden kann, Dateien automatisch herunterzuladen und auszuführen.
- Sie laden ein Programm herunter, dessen Hersteller eine Funktion eingebaut hat, die Ihnen schadet. Eine Reihe von Programmen wird z. B. beschuldigt, den ahnungslo-

<sup>16</sup> Mir ist bewußt, daß der Begriff Trojaner die Ilias ziemlich auf den Kopf stellt, er hat sich aber eingebürgert.

<sup>17</sup> Vgl. Kapitel 4, Abschnitt *Aktive Inhalte von HTML-Seiten*, ab Seite 59

sen Benutzer auszuspionieren [43][44]. Man hat daher für solche Programme den Begriff *Spyware* geprägt.

Ein Schutz vor Trojanern ist durch technische Maßnahmen allein nur bedingt möglich. Zwar erkennen moderne Virens Scanner<sup>18</sup> durchaus eine ganze Reihe von Kettenbrief-Trojanern, Remote Shells, DDoS-Zombies und Spionage-Programmen, wie sie von Crackern verwendet werden. Auch für Dialer und Spyware, die von Virens Scannern in der Regel nicht erkannt werden, existieren Schutzprogramme, die diese erkennen und u. U. auch entfernen. Es ist aber nicht schwierig, neue Trojaner zu erzeugen, die von keinem Scanner erkannt werden. Auch ist nicht von jedem kommerziellen Programm bekannt, ob es seine Benutzer ausspioniert.

Eine zentrale Firewall löst das Problem auch nur zum Teil. Sie kann nicht verhindern, daß der Trojaner auf einen angeschlossenen Rechner gelangt, da dies über normale Dateidownloads oder E-Mails geschieht. Lokale Aktionen (z. B. das Löschen von Dateien) oder Angriffe gegen Rechner im lokalen Netz kann sie ebenfalls nicht verhindern.

Allerdings macht sie den Anschluß von Modems oder den Einbau von ISDN-Karten in die Arbeitsplatzrechner überflüssig. Solange sie also selbst nicht kompromittiert ist, haben Dialer auf den Arbeitsplatzrechnern keine Möglichkeit, hinauszuwählen. Dies setzt allerdings voraus, daß darin keine Modems installiert sind, um z. B. Faxe zu versenden.

Eine Firewall kann im Einzelfall u. U. auch verhindern, daß Remote-Shell oder DDoS-Zombies von außen angesprochen werden oder daß DoS-Angriffe gegen Rechner außerhalb des lokalen Netzes gelingen. Dies hängt aber davon ab, wie die Remote-Shell oder der DDoS-Zombie gesteuert wird bzw. welche DoS-Angriffe der Zombie verwendet. Wenn die Shell oder der Zombie z. B. seine Befehle erhält, indem er eine Datei von einem Webserver herunterlädt, dann wird eine Firewall ihn normalerweise nicht daran hindern. Auch DoS-Angriffe, die darin bestehen, daß unzählige Verbindungen zu einem Webserver aufgebaut werden, sind aus Sicht einer Firewall nur massenhafte Downloads und damit in der Regel nicht verboten.

Eine Ausnahme bilden hier u. U. *Personal Firewalls*, wie sie von verschiedenen Herstellern für Windows-Rechner angeboten werden. Diese erlauben es oft, gezielt festzulegen, welches Programm mit welchen Rechnern im Internet auf welchen Ports kommunizieren darf. Dies erlaubt eine sehr viel feinere Kontrolle, ist aber auch aufwendiger und fehlerträchtiger. Hinzu kommt, daß die Personal Firewall auf jedem Rechner installiert und für die auf ihm installierte Software konfiguriert sein muß, um einen echten Vorteil gegenüber einer zentralen Firewall zu bieten. Auch schützen Personal Firewalls teilweise nicht vor Dialern, so daß zusätzlich auch ein Dialerschutz-Programm benötigt wird. Insgesamt sind Personal Firewalls insbesondere dann interessant, wenn nur ein einzelner Windows-Rechner an das Internet angeschlossen werden soll.

Schließlich soll auch noch erwähnt werden, daß Dialer nur funktionieren, wenn der Rechner an ein Modem oder eine ISDN-Anlage angeschlossen ist. Benutzt man DSL, so hat er keine Chance, wenn nicht noch zusätzlich ein Modem angeschlossen ist, um z. B. Faxe zu versenden.

<sup>18</sup> Eine Übersicht über Virenschutzlösungen unter Linux finden Sie unter <http://www.openantivirus.org>.

Neben allen technischen Maßnahmen kann man aber nicht genug betonen, bei den Benutzern ein Bewußtsein für die Problematik zu schaffen. So sollte man Software, die man per E-Mail erhalten hat, normalerweise nicht installieren. Auch beim Versand von Office-Dokumenten sollte man Formate verwenden, bei denen das Risiko, daß sie ausführbaren Code enthalten, gering ist, wie z. B. RTF- oder PDF-Dateien<sup>19</sup>.

Beim Download von Programmen aus dem Internet sollte man sich zurückhalten. Benötigt man ein Programm, so sollte man es unbedingt vom Server der Autoren oder einem offiziellen Mirror herunterladen. Geben die Autoren auf ihren Webseiten MD5-Checksummen an, so sollte man sie unbedingt überprüfen<sup>20</sup>. Sind die Dateien mit PGP oder GPG signiert, so sollte man sich den öffentlichen Schlüssel der Autoren besorgen und die Signatur überprüfen. Auch auf offiziellen Servern für bekannte Programme wurden schon trojanisierte Versionen gefunden.

## Angriffe auf die Privatsphäre: Referer-Header und Cookies

Wer sich im Internet aufhält, tut dies meistens in der Vorstellung, völlig anonym und unbeobachtet zu sein. Dem ist aber nicht so. Mindestens eine andere Stelle weiß in der Regel ganz genau, wer wir sind und was wir tun. Dabei handelt es sich um unseren Provider. Alle Verbindungsaufbauten laufen über ihn und werden in der Regel auch mitprotokolliert. Da es sich aber um eine Firma in Deutschland handelt und wir hier ein sehr gutes Datenschutzrecht haben, soll uns das erst einmal nicht weiter beunruhigen.

Die nächste Stelle, die sich für uns interessieren könnte, wäre der Betreiber des Servers, mit dem wir uns verbinden. Für ihn gestaltet sich die Feststellung, wer ihn da besucht, schon etwas schwieriger. Er hat nur wenige Daten, die ihm weiterhelfen. Als erstes wäre da die IP-Adresse, an die er die Daten schickt. Diese nützt ihm allerdings nur dann etwas, wenn wir bei jedem Besuch dieselbe IP-Adresse benutzen. Viele Provider vergeben IP-Adressen allerdings dynamisch, d. h., bei jeder neuen Einwahl erhält man eine neue Adresse. Dadurch wird die begrenzte Anzahl von Adressen, die ein Provider besitzt, besser ausgenutzt.

Die Industrie ist sich dieses Problems bewußt und hat auch eine Lösung gefunden. Wenn man Webseiten, Graphiken oder Dateien mittels HTTP herunterlädt, kann der Server zusätzlich ein »Cookie« senden. Dies ist eine kurze Zeichenkette, die von nun an zusammen mit jeder neuen Anfrage an den Server geschickt wird.

Auf diese Weise kann einem Benutzer eine Seriennummer zugewiesen werden, unter der er von nun an jedesmal auftritt, wenn er den Server besucht. Der Server kann damit alle Zugriffe des Benutzers eindeutig miteinander in Verbindung bringen.

<sup>19</sup> Ich behaupte nicht, daß diese Formate sicher sind. Benutzt man die Acrobat-Vollversion, so kann im Gegensatz zum Acrobat Reader im Dokument enthaltener JavaScript-Code ausgeführt werden. Auch für RTF-Dokumente soll es eine Methode geben, ausführbaren Code einzubetten. Im Gegensatz zu DOC- und XLS-Dateien ist mir allerdings noch kein tatsächlicher Vorfall zu Ohren gekommen.

<sup>20</sup> Dafür existiert das Programm `md5sum`, das in den meisten Linux-Distributionen enthalten ist.

Dies erlaubt allerdings nur das Beobachten der Bewegungen auf einem Server oder einigen wenigen Servern desselben Betreibers. Dies liegt daran, daß Cookies grundsätzlich nur an Server gesendet werden, wenn der Rechner, der das Cookie gesetzt hat, und der Rechner, von dem gerade eine Seite geholt werden soll, in derselben Netzwerkdomäne liegen. Würden also die Domänen *Gorilla.com* und *OrangUtan.com* derselben Firma gehören, so könnte *www.Gorilla.com* trotzdem kein Cookie setzen, das an *www.OrangUtan.com* gesendet würde.

Auch hier wird man in der Regel das Bedrohungspotential für die eigene Privatsphäre eher gering einstufen. Interessant wird es allerdings, wenn man feststellt, daß es eine Firma gibt, die Bewegungen von Benutzern in großen Teilen des Internets nachvollziehen kann und diese Daten auch zur kommerziellen Verwertung speichert.

Was sich auf den ersten Blick wie eine Verschwörungstheorie anhört, ist tatsächlich nur ganz normaler Geschäftsalltag. Auf vielen Webseiten finden sich Werbebanner, kleine, oft animierte Graphiken, die den Surfer dazu bringen sollen, auf den Sites ihrer Auftraggeber vorbeizuschauen. Diese Graphiken holt der Browser allerdings in der Regel nicht von dem Server, auf dem sich die eigentliche Webseite befindet, sondern vom Rechner eines speziellen Betreibers, der Werbeflächen vermittelt. Dieser hat damit die Kontrolle, welches Banner wo wie oft erscheint.

Die Anforderung der Werbebanner durch den Browser funktioniert genauso wie die Anforderung einer Webseite. Auch hier können Cookies gesetzt werden, was der Werbende dazu nutzen kann, dafür zu sorgen, daß derselbe Benutzer nicht ständig dieselbe Werbung erhält. Des weiteren wird bei der Anforderung von Webseiten durch den Browser auch mitgeteilt, wenn der Benutzer über einen Verweis von einer anderen Webseite kam. Diese Angabe (Referer-Header) wird auch gesendet, wenn eine Graphik mittels des HTTP-Protokolls geladen wird. Wird also ein Werbebanner geladen, so wird dem Server dabei mitgeteilt, in welche Webseite dieses eingebaut werden soll. Durch die Verbindung von Cookies und Referer-Header weiß der Werbevermittler genau, welcher Benutzer gerade welche Webseiten betrachtet. Dies wird dazu genutzt, Profile zu erstellen und den Benutzer gezielt mit Werbung zu versorgen, die seinen Interessen entspricht.

Neuerdings wird dieses Konzept noch erweitert. Sogenannte *Web Bugs*,  $1 \times 1$  Pixel große weiße Graphiken, werden wie Werbebanner eingesetzt, um Nutzerdaten zu sammeln [32]. Wegen ihrer Farbe und Größe sind sie im Gegensatz zu Werbebannern für den Benutzer unsichtbar. Auch können diese nicht nur in HTML-Seiten eingebaut werden. Wie sich herausgestellt hat [33], ist dies auch in Word-, Excel- und PowerPoint-Dokumenten möglich, seit diese die Möglichkeit bieten, Graphiken als Links einzubinden. Dies erlaubt es nicht nur festzustellen, daß auf einem Rechner mit einer bestimmten IP-Adresse gerade ein bestimmtes Word-Dokument gelesen wird. Es können auch Cookies gesetzt werden, da Word (Excel, PowerPoint) den Internet Explorer benutzt, um Graphiken aus dem Internet zu laden.

Allerdings kennt der Werbevermittler den Benutzer bisher nur unter einem Pseudonym, das er ihm selbst zugewiesen hat (z. B. »Benutzer 43501«). Um ihm auch einen Namen und eine Adresse zuordnen zu können, ist es nötig, die gewonnenen Daten mit einer weiteren Informationsquelle abzugleichen. Genau dies kündigte DoubleClick, der

größte Anbieter auf diesem Sektor, im Januar 2000 an. Die Firma hatte kurz vorher mit der Marktforschungsfirma Abacus Alliance fusioniert, die eine Datenbank mit über zwei Millionen Kundenprofilen besitzt, die aus E-Commerce-Einkäufen gewonnen wurden. Nachdem allerdings diverse Datenschutzorganisationen protestierten und sogar das amerikanische Kartellamt begann, Untersuchungen anzustellen, kündigte DoubleClick-Chef Kevin O’Connor im März 2000 an, diese Pläne vorerst auf Eis zu legen.<sup>21</sup>

Der Neugier der Werbeindustrie ist man allerdings nicht hilflos ausgeliefert. Als erste Maßnahme kann man damit beginnen, Cookies im Browser abzuschalten. Des Weiteren kann man einen speziellen Proxy einsetzen, der kompromittierende Header filtert. Beispiele dafür wären der Internet Junkbuster und der Squid, die wir in Kapitel 14, Abschnitt *Einrichten eines Web- oder FTP-Proxys*, ab Seite 399 noch genauer kennenlernen werden. Um ihre Wirkungsweise zu verstehen, müssen wir uns klarmachen, wie eine HTTP-Anfrage aufgebaut ist.

Fordert z. B. ein Internet Explorer die Seite */somewhere/page.html* vom Server *dummy.com* an, wohin er über einen Link von *http://foo.bar/links.htm* kam, so wird seine Anfrage unter Umständen folgendermaßen aussehen:

```
GET /somewhere/page.html HTTP/1.1
Accept: */*
Referer: http://foo.bar/links.htm
Accept-Language: de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)
Host: dummy.com
Connection: Keep-Alive
Cookies: ID=4711
```

Neben einem Cookie und dem schon erwähnten Referer-Header sehen wir, daß der Browser auch Daten über seine Version (IE 4.01), sein Betriebssystem (Windows 98) und die Muttersprache seines Benutzers (de) liefert. Alte Versionen (vor 3.0) von Navigator und Explorer benutzten sogar einen speziellen Header, um dem Zielsystem die E-Mail-Adresse des Benutzers mitzuteilen.

Header sind aber im Prinzip optional. Theoretisch würde es auch reichen, einem Server nur den GET-Befehl zu schicken. Dies funktioniert oft auch, es existieren aber Server, die unter diversen Namen ansprechbar sind. Diese benutzen den Host-Header, um zu entscheiden, unter welchem Namen sie angesprochen wurden. Auch der User-Agent-Header wird von einigen Servern gezielt ausgewertet, um die Webseiten auf den Browser des Benutzers zuzuschneiden. Dies merkt man, wenn man einen normalen Browser benutzt, alle Header filtert und plötzlich eine Meldung erhält: »Sorry, aber Ihr Browser ist zu alt und unterstützt keine Frames. Klicken Sie [hier](#), um einen aktuellen Browser herunterzuladen.« Man sollte daher zumindest den Host-Header von der Filterung ausnehmen. Ob man auch den User-Agent-Header weiterleitet, sollte man unter Berücksichtigung der eigenen Surfgeohnheiten entscheiden.

<sup>21</sup> Quelle: <http://www.heise.de/newsticker/data/hob-03.03.00-000/>

## Aktive Inhalte von HTML-Seiten

Die einzige Art der Interaktion mit dem Benutzer, die normale Webseiten kennen, sind Links auf andere Webseiten. Nachdem aber das Schlagwort E-Commerce aufkam, begann eine Nachfrage nach Möglichkeiten, interaktive Webseiten zu gestalten. Diese wurde durch mehrere Technologien befriedigt, die es erlauben, kleine Programme in Webseiten einzubauen, die beim Betrachten der Seite ausgeführt werden. Gebräuchlich sind drei Varianten:

**Java-Applets** Bei Java-Applets handelt es sich um Dateien, die in einem speziellen Binärformat, ausführbaren Programmen ähnlich, vorliegen. Im Gegensatz zu normalen Programmen werden sie aber nicht direkt ausgeführt, sondern von einer »virtuellen Maschine« interpretiert.

Java ist die einzige hier aufgeführte Technologie, die ein Sicherheitskonzept besitzt. Jeder Aufruf von sicherheitskritischen Funktionen (Dateizugriffe, Netzwerkzugriffe, Manipulation anderer laufender Prozesse) wird erst anhand einer definierten Policy geprüft, bevor er eventuell ausgeführt wird.

Um solche Funktionen ausführen zu können, muß ein Applet digital signiert sein. Zusätzlich wird der Browser in der Regel beim Benutzer nachfragen, ob so ein Zugriff überhaupt erwünscht ist. Auch wenn gelegentlich Fehler in den Implementationen der Browser bekannt werden, so kann die Benutzung von Java-Applets im E-Commerce doch als relativ beherrschbares Risiko angesehen werden.

**Skriptsprachen** Hier werden Befehle im Klartext in HTML-Seiten eingebettet und vom Browser interpretiert. Bekannte Vertreter sind JavaScript, JScript und VB-Script. Diese Sprachen kennen eigentlich keine Funktionen für den direkten Zugriff auf lokale Dateien. Unter Windows 98 werden dafür nötige Mechanismen aber vom Windows Scripting Host bereitgestellt. Auch können oft Funktionen des Browsers dazu benutzt werden, Dateien zumindest zu lesen. Dies kann zum Beispiel geschehen, indem ein Skript eine Seite in ein neues Browserfenster lädt und dann über Umwege darauf zugreift. Zwar ist das von den Browserherstellern so nicht gewollt und wird auch umgehend behoben, die Entdeckung neuer Implementationsfehler in den Browsern erfolgt aber mit erschreckender Regelmäßigkeit.

**ActiveX-Controls** Hierbei handelt es sich um spezielle dynamisch ladbare Bibliotheken (DLLs), die nur unter Windows benutzt werden können. In den Standardeinstellungen des Internet Explorers ist eine digitale Signatur eines Controls nötig, damit es heruntergeladen wird. Da dieselbe Technologie aber auch zur Realisierung wiederverwertbarer Programmkomponenten unter Windows eingesetzt wird, existieren oft diverse Controls, die schon im System installiert sind und zum Teil auch von Webseiten aus aufgerufen werden können.<sup>22</sup>

Einmal geladen und gestartet, werden Controls ausgeführt wie normale Programme. Eine besondere Überwachung durch das Betriebssystem oder den Browser fin-

<sup>22</sup> Es gab zum Beispiel einmal ein Control eines Herstellers von Programmen zur Behebung von Festplattenproblemen, das es erlaubte, die Festplatte neu zu formatieren. Es war so programmiert, daß es auch aus Webseiten heraus aufgerufen werden konnte.

det nicht statt. Dies bedeutet, daß ein Control auf dem Rechner alles tun kann, was auch der Benutzer, der es gestartet hat, tun darf. Unter Windows 95 und 98 bedeutet dies, daß das Control keinen Einschränkungen unterliegt. Dies kann z. B. dazu genutzt werden, Trojaner zu installieren (siehe Kapitel 4, Abschnitt *Aktive Inhalte von HTML-Seiten*, ab Seite 59).

ActiveX-Controls werden in den Standardeinstellungen nur heruntergeladen, wenn sie digital signiert sind. Praktisch bedeutet dies, daß jemand, der ein Control in eine Webseite einbauen will, zuerst eine spezielle Prüfsumme über das Control bildet. Diese wird dann mit dem geheimen Schlüssel eines Public-Key-Verfahrens<sup>23</sup> verschlüsselt. Der Empfänger entschlüsselt die Prüfsumme mit dem öffentlichen Schlüssel des Senders und überprüft, ob sie mit einer selbst generierten Prüfsumme über das Control übereinstimmt. Er weiß dann, wer das Control signiert hat.

Der Empfänger muß allerdings den öffentlichen Schlüssel des Senders kennen. Hierzu erhält er zusammen mit dem Control ein Zertifikat. Dies ist ein elektronisches Dokument, das den öffentlichen Schlüssel des Senders sowie Angaben zu seiner Person enthält. Es ist selbst wieder digital signiert. Damit der Browser das Zertifikat »anerkennt«, muß der öffentliche Schlüssel der zertifizierenden Stelle in seiner Datenbank eingetragen sein.

Wie es scheint, schreckt der Aufwand, ein Zertifikat von einer anerkannten Stelle zu bekommen, die meisten Programmierer bössartiger Controls ab. Mir ist nur ein Fall bekannt, wo ein signiertes bössartiges Control in eine Webseite eingebettet wurde. Dabei handelte es sich allerdings auch nur um eine Demonstration. Das Control fuhr nach einer Rückfrage den Rechner herunter. Nach der Androhung einer Klage durch die Zertifizierungsinstanz entfernte der Autor die Signatur von seinem Control.

Auch sonst werden ActiveX-Controls kaum in Webseiten eingesetzt. Es ist daher sinnvoll, ihre Benutzung im Browser generell abzuschalten oder sie mit einem geeigneten Proxy zu filtern. Man verliert dadurch keine Funktionalität, ist aber sicher, falls doch einmal ein Angreifer versucht, ActiveX als Einbruchswerkzeug zu mißbrauchen.

Java-Applets sind da schon beliebter. Sie werden vor allem im E-Commerce und beim Online-Banking eingesetzt. Dort dienen sie in der Regel als graphische Oberfläche und Mittel, einen sicheren Kommunikationskanal aufzubauen. Damit gibt es durchaus Webseiten, für die es sich lohnt, Java zu aktivieren.

Java generell zu erlauben ist keine so gute Idee. Zwar ist mir weder ein Angriff bekannt, der signierte Applets benutzte, noch sind die eher selten auftretenden Sicherheitslücken in Java jemals wirklich von Angreifern genutzt worden. Aber es ist problemlos möglich, Applets zu schreiben, die wichtige Ressourcen so auslasten, daß sie zu keinen anderen Zwecken mehr zur Verfügung stehen. Gerade auf Windows-Systemen reicht oft schon das versehentliche Laden einer ganz normalen Java-Anwendung, wie z. B. eines Bestellsystems für Pizza, um das System zum Stillstand zu bringen.

<sup>23</sup> Während normale Crypto-Verfahren nur einen Schlüssel zum Ver- und Entschlüsseln kennen, werden in Public-Key-Verfahren zwei unterschiedliche Schlüssel verwendet. Einer der beiden wird öffentlich bekanntgegeben, während der andere geheimgehalten wird. So ist es möglich, entweder eine Nachricht zu generieren, die nur einer verschlüsselt haben kann, die aber von allen entschlüsselt werden kann (wie hier), oder eine, die nur eine Person lesen kann, die aber von jedermann verschlüsselt werden sein kann.

Gezielt als Angriffswerkzeuge geschriebene Applets können diesen Effekt noch verstärken. So existieren bereits Applets, die den Rechner durch komplizierte mathematische Berechnungen verlangsamen, seltsame Geräusche ertönen lassen oder dafür sorgen, daß der Rechner plötzlich gar keine Töne mehr erzeugt. Daß dies kaum genutzt wird, liegt daran, daß Skripte einfacher und schneller zu realisieren sind als Java-Applets. Wer also Java nicht gerade für das Online-Banking oder Einkäufe im Internet braucht, sollte es abschalten.

Skriptsprachen sind das Lieblingswerkzeug der Gestalter interaktiver Webseiten. Immer öfter werden JavaScript-Funktionen aufgerufen, wo eigentlich ein einfacher Link genügt hätte. Dadurch werden immer mehr Webseiten ohne JavaScript unbenutzbar. Dies verführt dazu, JavaScript generell eingeschaltet zu lassen. Allerdings sollte man bei seiner Entscheidung berücksichtigen, daß auch Angreifer die Leichtigkeit zu schätzen wissen, mit der Skripte zu realisieren sind.

Wer viel surft, ist mit Sicherheit schon einmal einer jener Seiten begegnet, deren Besuch oder Verlassen zum Öffnen einer Vielzahl von Werbefenstern führt. Tatsächlich kann der Angreifer dies dadurch auf die Spitze treiben, daß er dafür sorgt, daß das Schließen von Fenstern zum Öffnen von neuen führt. Damit wird die Arbeitsfläche komplett von Dutzenden von Fenstern verdeckt, die der Benutzer de facto nicht mehr schließen kann.

Deutlich gefährlicher sind allerdings Angriffe, die auf Sicherheitslücken der Browser beruhen und mit steter Regelmäßigkeit wieder auftauchen. Eine ganze Reihe von Angriffen erlaubte es beispielsweise, lokale Dateien auszulesen und an Rechner im Internet zu versenden.

Man sollte auch nicht übersehen, daß HTML-Seiten nicht länger nur auf Webservern zu finden sind. Viele E-Mail-Programme kennen inzwischen E-Mails im HTML-Format. So warnte z. B. Network Associates im November 1999 vor dem E-Mail-Wurm »Bubbleboy«. Dieser bestand aus einer E-Mail im HTML-Format mit eingebettetem VBScript. Beim Betrachten mit dem Mailprogramm Outlook installierte er sich im System und fing damit an, infizierte E-Mails an alle Adressen in allen Adreßbüchern von Outlook zu schicken. Unter Outlook Express 98 war es nicht einmal nötig, die E-Mail tatsächlich zu öffnen. Es reichte schon die Anzeige der Vorschau beim Blättern in den empfangenen E-Mails, um die Infektion auszulösen. Diese Sicherheitslücke ist allerdings laut Microsoft inzwischen geschlossen.

Für die Infektion mußte Bubbleboy eine Datei in ein Systemverzeichnis schreiben. Daß ihm dies möglich war, zeigt, welches Potential in derartigen Angriffen steckt. Mir sind auch Skripte bekannt, die mit Hilfe des Windows Scripting Host Dateien löschen und Makroviren in Word-Dokumente einbauen. Man sollte sich daher die Entscheidung, wann man Skriptsprachen zuläßt, nicht zu einfach machen.

## Social Engineering und Phishing

Einer der berühmtesten Cracker war Kevin Mitnick[66]. Das FBI schätzte ihn seinerzeit als einen der gefährlichsten Vertreter seiner Spezies ein. Man sagt kein System sei vor ihm sicher gewesen und er sei das Vorbild für den Jugendlichen gewesen, der im Film »War Games« beinahe einen weltumspannenden Atomkrieg auslöst.

Nachdem er seine Gefängnisstrafe abgesessen hat, verdient er nun seinen Lebensunterhalt als Berater. Er hat auch ein Buch über seine Erfahrungen geschrieben. Darin schreibt er, daß sein wichtigstes Werkzeug keine komplizierte Technik oder Programmiertricks auf Assemblerebene waren. Vielmehr verdankt er seine Erfolge einer Technik namens *Social Engineering*. Diese richtet sich nicht gegen den Computer sondern gegen dessen Anwender.

Tatsächlich ist die einfachste Methode, an vertrauliche Daten eines Anwenders zu gelangen, ihn einfach zu fragen. In zwei Studien reichte es schon, Büroangestellten einen Plastik-Kugelschreiber [63] oder ein Schokoladen-Osterei [64] zu versprechen und eine Meinungsumfrage vorzutauschen, damit diese bereitwillig ihre Passwörter verrieten.

Dieses Vorgehen ist nur ein schwacher Abklatsch von dem, was wahre Experten auf diesem Sektor zu Stande bringen. Wenn diese in die Rechner einer Firma einbrechen wollen, bereiten sie sich erst einmal vor, indem sie aus frei zugänglichen Quellen so viel wie möglich über ihr Ziel in Erfahrung bringen. Dann rufen Sie eine der öffentlichen Nummern des Zielunternehmens an und geben vor, ein Kunde zu sein, der Hilfe benötigt. In dem nun folgenden Gespräch versuchen sie so viele Details wie möglich in Erfahrung zu bringen. Dabei geht es in diesem Schritt noch nicht um Geheimnisse, sondern nur um scheinbar unwichtige Details wie Namen, gebräuchliche Fachausdrücke und Details der internen Organisation der Firma.

Mit diesem Wissen bewaffnet wird dann ein anderer Supportmitarbeiter angerufen und es wird vorgegeben, ein Techniker zu sein, der gerade schnell und unbürokratisch Hilfe benötigt. Durch die geschickte Verwendung von Details wie Namen von Kollegen, internen Fachbegriffen und ein selbstsicheres Auftreten gelingt es dann schnell, die Person am anderen Ende dazu zu bringen, Konfigurationsänderungen am System vorzunehmen, Paßwörter zu nennen oder technische Unterlagen herauszugeben, die nicht für Außenstehende bestimmt sind. Indem diese Schritte mehrfach wiederholt werden, kann ein geschickter Angreifer genug Informationen erhalten, um sich weitreichende Zugriffsrechte auf die Systeme einer Firma zu beschaffen. [65][67]

Eine andere Variante des Social Engineering wird am Besten durch das folgende Sprichwort beschrieben:

*»Mit einem Helm und einem Klemmbrett kommt man überall hinein.«*

Unauffällige Kleidung und ein selbstsicheres Auftreten reichen oft aus, um auch in eigentlich gesicherte Bereiche zu kommen. Das bekannteste Beispiel für ein gelungenes Social Engineering ereignete sich am 16. Oktober 1906, als der Schuster Wilhelm Voigt sich

als Hauptmann verkleidete, das Kommando über ein paar Soldaten übernahm, denen er zufällig begegnetwar, und das Rathaus der Stadt Köpenick<sup>24</sup> übernahm. Er ließ den Bürgermeister verhaften und entkam mit der Stadtkasse. Zwar wurde er nach 10 Tagen gefaßt, aber hatte eindrucksvoll bewiesen, daß man durch das richtige Auftreten eine Menge erreichen kann.

Auch heute, fast hundert Jahre später<sup>25</sup>, gelingt es talentierten Individuen immer noch, auf diese Weise in streng gesicherte Serverräume oder die Büros von Führungskräften großer Firmen zu gelangen [65]. In einem Fall berichtet ein Social Engineer sogar, es sei ihm gelungen, mit 5 Computern unter den Augen der Angestellten aus einem Geschäft zu spazieren [68].

Solche Leistungen sind allerdings Spezialisten vorbehalten, deren Ziel in erster Linie große Firmen sind, die genug Personal haben, daß die Mitarbeiter sich untereinander nicht kennen. Die beschriebenen Social Engineers benötigen für Ihre Arbeit ein sicheres Auftreten, Menschenkenntnis und jahrelange Erfahrung.

Die große Masse der Angreifer benutzt daher weitaus einfachere Tricks. In der Regel richten sich Ihre Angriffe gegen normale Benutzer und zielen darauf ab, an Zugangsdaten zu Online-Diensten oder an Kreditkartendaten zu gelangen.

Der Angriff beginnt in der Regel damit, daß dem Opfer eine E-Mail geschickt wird, in der der Benutzer darüber informiert wird, daß ein technisches Problem aufgetreten ist und er seine Daten neu eingeben oder »verifizieren« muß.

Früher wurde der Benutzer aufgefordert, seine persönlichen Daten per E-Mail an eine vorgebliche Service-Adresse des Providers zu schicken, heutzutage ist in der Regel die Adresse einer Webseite angegeben, auf der man seine Daten in ein Formular eintragen muß. Die angegebene Adresse ist dabei natürlich keine offizielle Adresse des Unternehmens, sondern wird vom Angreifer kontrolliert. Für das Opfer ist dies oft aber nicht ohne weiteres zu erkennen.

Diese Klasse von Angriffen nennt man *Phishing* [69]. Während solche Angriffe oft dazu verwendet wurden, auf fremde Kosten an einen Internet-Zugang zu gelangen, sind heute Zugangsdaten zu Online-Banken oder Auktionshäusern das Ziel.

Auch der Kreis der Täter hat sich von Einzelgängern hin zur organisierten Kriminalität verschoben. Mittlerweile hat sich in der Szene eine Spezialisierung herausgebildet, bei der ein potentieller Phisher vor seinen Raubzügen in Internet-Foren Hostingkapazitäten in Netzwerken anmietet, die aus den kompromittierten Rechnern ahnungsloser Benutzer bestehen, denen z. B. ein Trojaner untergeschoben wurde (Botnets). Diese Rechner werden dann dazu benutzt, nachgemachte Webseiten bekannter Banken oder Auktionshäuser zu hosten. Entsprechende Vorlagen sind im Preis für die Nutzung des Botnets enthalten.

Der Phisher braucht nun nur noch E-Mails zu versenden, welche die Benutzer auf die von ihm gemieteten Server locken. Auch Listen mit gültigen E-Mail-Adressen kann er

<sup>24</sup> Seit 1920 ein Stadtteil von Berlin

<sup>25</sup> Während ich dies schreibe, ist es auf den Tag genau 98 Jahre und 11 Monate her.

dabei problemlos auf dem schwarzen Markt erwerben. Hat er auf diese Weise geldwerte Daten erhalten, so wird er sie in der Regel nicht selbst ausbeuten, sondern an Spezialisten verkaufen, die sie gegen Gewinnbeteiligung zu Geld machen [70][71].

Social Engineering zeigen recht deutlich auf, daß Technik allein nicht reicht, sich gegen Angriffe aus dem Internet zu schützen. Zwar kann man mit einer Firewall verhindern, daß Rechner im eigenen LAN in Webserver umfunktioniert werden, man kann auch auf den Mailservern Schutzprogramme installieren, die nach bekannten Phishing-E-Mails und Trojanern suchen. Spätestens aber wenn ein Angreifer zum Hörer greift und sein Opfer anruft, kommt man mit Technik nicht mehr weiter.

Hier können nur noch Schulungen der Anwender helfen. Erst wenn diese verstehen, worin die Probleme bestehen, und wissen, wie sie sich schützen können, hat man eine Chance, das Problem in den Griff zu bekommen.

Zwar wird man nie hundertprozentig sicher gegen erfahrene Social Engineers sein, aber den gemeinen Phisher kann man in der Regel schon durch die Befolgung von ein paar einfachen Grundsätzen das Handwerk deutlich erschweren. Am wichtigsten dabei ist sicherlich, sein Paßwort grundsätzlich niemandem mitzuteilen. Fragt einen ein Beauftragter einer Bank, eines Auktionshauses, eines Versandhändlers oder eines Online-Dienstes nach dem Paßwort, so kann man mit ziemlicher Sicherheit davon ausgehen, daß es sich um einen Phishing-Versuch handelt.

Wird man per E-Mail aufgefordert, sich zur Behebung von Problemen auf den Webseiten eines Anbieters anzumelden, so kann man Phishing-Versuche dadurch Schachmatt setzen, daß man nicht auf den Link in der E-mail klickt, sondern stattdessen die offizielle Adresse des Dienstes in den Webbrowser eintippt, wie man es auch sonst tut, wenn man den Dienst nutzt. Nur so kann man sicher sein, sich tatsächlich auf den Webseiten des Dienstes und nicht auf einer cleveren Fälschung zu befinden.