```
          Interrupt:11 Base address:0x1000
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:316085 errors:0 dropped:0 overruns:0 frame:0
          TX packets:316085 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:26842574 (25.5 MiB)  TX bytes:26842574 (25.5 MiB)
debian:~# ifconfig eth0 promisc
debian:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:01:02:74:FD:C2
          inet addr:10.10.1.117  Bcast:10.255.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:30117219 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26477505 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1831172230 (1.7 GiB)  TX bytes:3409616241 (3.1 GiB)
          Interrupt:11 Base address:0x1000
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:316095 errors:0 dropped:0 overruns:0 frame:0
          TX packets:316095 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:26843414 (25.5 MiB)  TX bytes:26843414 (25.5 MiB)
debian:~#
```

Erfolgreiche Angriffe mittels Juggernaut können durch die Umsetzung eines geswitchten Netzwerkdesigns erschwert werden. In solchen Umgebungen muss der Angreifer zusätzlich die Switches mittels ARP-Spoofing penetrieren, um die gewünschten Verbindungen sehen und manipulieren zu können. Dies ist jedoch keineswegs die ultimative Lösung und sollte keinen Administrator in Sicherheit wiegen lassen. Einzig und allein der Einsatz von kryptografischen Methoden macht die Möglichkeiten von Juggernaut zunichte. Da TCP/IP in der aktuellen Version 4 keine Verschlüsselung auf Transport- oder Link-Ebene vorsieht, müssen zusätzlich Programme für den gesicherten Transport sorgen. Lösungen wie SSH (**S**ecure **Sh**ell) und IPsec bieten sich hierfür natürlich an.

4.3 Erdrückt vom IP: von DoS und DDoS-Attacken

Zu Beginn des Kapitels 4.2 haben wir gesehen, welche Gefahr von den Protokollen der Netzwerkschicht ausgehen kann. Die dadurch entstehenden Möglichkeiten sind sehr vielfältig. Zieht man jedoch eine statistische Auswertung zurate, sind Denial of Service-Attacken aufgrund ihrer Einfachheit die meistgenutzte Form, Computersysteme auf der Netzwerkebene zu manipulieren.

Denial of Service-Attacken haben das Ziel, Ressourcen zu überlasten, um legitimen Benutzern den Zugriff zu verwehren. Das im Januar 1989 durch das Internet Activities Board herausgegebene RFC 1087 (Ethics and the Internet) beschäftigt sich mit den ethischen Aspekten der Internetnutzung. Darin wird festgehalten, dass die Störung des Gebrauchs des Internets sowie der Verbrauch von Ressourcen als Verstoß gegen die ethischen Richtlinien anzusehen sind. Als Ressourcen werden in diesem historischen Dokument „Personen, Leistungen und Computer" festgehalten.

Jeder, der schon einmal ein wissenschaftlich fundiertes Buch über die Entstehung und Entwicklung des Internets gelesen hat, wurde im Eingangs-Kapitel damit konfrontiert, dass das Internet ursprünglich vom US-Militär für jenen Zweck entwickelt wurde, ein möglichst stabiles dezentrales Netzwerk zu errichten, das sogar einen atomaren Schlag überstehen könnte. Gegen Ende des Jahres 1962 lieferte Paul Baran von der Rand Corporation einen ersten Entwurf, der 1969 in Tat und Wahrheit umgesetzt wurde. Während über 30-jährigen Geschichte des Internets traten jedoch vereinzelte Situationen in Kraft, die einem gesamten Zusammenbruch sehr nahe kamen. Der wohl mitunter bekannteste Vorfall dieser Sparte offenbarte 1988 seine destruktive Absicht, als der berühmt berüchtigte Internetwurm von Robert Tappan Morris mit seinem bösartigen Code rund 50.000 Rechner zum Stillstand brachte. Um jedoch zu den Wurzeln und indirekt zur Entstehung von Denial of Service-Attacken zurückzugehen, müssen wir uns in unserer Fantasie einer Zeitreise ins Jahr 1980 bemächtigen. Als Ronald Reagan und Jimmy Carter sich im Oktober in den Präsidentschaftswahlen verstrikten, umfasste der Vorgänger des Internets, das ARPANET, nur rund 200 Hosts und wurde vorwiegend von Forschern und Wissenschaftlern für ihre Arbeit genutzt. Am 27. Oktober 1980 gingen diese besagten Gelehrten an ihre Terminals, um mit Schrecken festzustellen, dass ein Großteil des ARPANET zum Erliegen gekommen war.

Eine Welle von Anrufen überschwemmte das **N**etwork **C**ontrol **C**enter (NCC), und so wurde schnell klar, dass das Problem kein lokales war, sondern praktisch jedes Subnetz betreffen würde. Die Ursache für dieses Problem wurde auf mikroskopischer Ebene gefunden: Ein ziemlich seltsamer Hardwarefehler generierte fehlerhafte Sequenzen von Netzwerk-Kontrollpaketen. Diese fehlerhaften Konstrukte beeinflussten die Distribution der Softwareressourcen der Subnetze. Dies führte dazu, dass Prozesse zu viele Ressourcen belegten, sodass für andere Prozesse keine mehr verblieben. Nähere Informationen zu diesem historischen Vorfall finden sich in RFC 789 (Vulnerabilities of Network Control Protocols: An Example).

Sie erinnern sich bestimmt an die Sage, in der der kleine David gegen den großen Goliath kämpfen musste. Der schmächtige David zieht jedoch bei der nun vorgestellten Angriffsmethode 99,9%ig den Kürzeren: Leute mit einer langsamen Netzanbindung (z. B. Privatpersonen mit einem 56k-Modem) laufen Gefahr, in ihrer Arbeitsweise negativ beeinträchtigt zu werden. Ein Besitzer einer T3-Standlei-

tung wird ohne Probleme die Dial-Up-Verbindung seines Opfers so krass überfluten können, dass durch diese enorme Auslastung kein effizientes Arbeiten des Opfers über sein Modem mehr möglich ist.

Das Aufbrauchen der Bandbreite zielt auf das Überlasten der Netzwerkressourcen ab – das Aufbrauchen der Ressourcen hingegen auf das Überlasten der Systemressourcen. Im Allgemeinen unterscheidet man hierbei zwischen Attacken auf die CPU-, Speicher- und Festplatten-Auslastung. In den meisten Fällen wird dem Angreifer ein gewisses Maß dieser Ressourcen zugeschrieben, die er dann hemmungslos für sein destruktives Treiben missbraucht. Dadurch können wichtige Teile des Systems den rechtmäßigen Benutzern entzogen oder das Nutzen derer verhindert werden. Meistens gipfeln solche Angriffe darin, dass das System oder einzelne Komponenten temporär unbrauchbar werden, da ein Absturz oder Einfrieren eintritt.

Als Programmierfehler bezeichnet man den Zustand der unfähigen Verarbeitung einer Eingabe. Dies kann eine Anwendung, das Betriebssystem oder ein Mikrochip betreffen. Solche Ausnahmebedingungen entstehen in der Regel auf der Netzwerkebene bei der Übergabe von nicht RFC-konformen Paketen. Anwendungsprogramme dagegen reagieren typischerweise allergisch auf übertrieben lange oder kuriose Eingaben.

Denial of Service-Attacken haben stets einen destruktiven Charakter. Doch nicht alle Denial of Service-Angriffe bleiben auf dieser Ebene stehen und sehen sich selbst als das höchste Ziel des Unternehmens. Wie wir in Kapitel 4.2 erfahren haben, ist bei vielen Spoofing-Attacken für ihre erfolgreiche Durchführung Denial of Service ein tragendes Element. IP-Spoofing würde aufgrund der doppelt eintreffenden Datagramme und dadurch asynchron verlaufenden TCP-Kommunikation seine Gültigkeit und Effizienz verlieren.

Denial of Service ist und bleibt die Lieblingsdisziplin kindischer Script-Kiddies, die in ihrem jugendlichen Leichtsinn fremde Systeme beeinträchtigen wollen. Diese Angriffsform ist vor allem deshalb beliebt, weil sie kein großes Know-how zur erfolgreichen Durchführung benötigt. Sein Denial of Service-Tool laden und die IP-Adresse des Opfers eingeben, das kann nun wirklich jeder. Erfahrene Angreifer, mit einem konstruktiven Ziel, werden nur in Notsituationen Denial of Service-Angriffe durchführen. Denn solche Zugriffe werden schnell erkannt und vielleicht könnten aus Versehen wichtige Systeme negativ beeinträchtig werden, die für das weitere Vorgehen von Relevanz gewesen wären. Bestes Beispiel ist hierfür, wenn ein Angreifer aus Versehen die Perimeter-Firewall einer Organisation in die Knie zwingt, sodass keine Daten mehr aus dem oder in das Netzwerk geschickt werden können. In diesem Fall hat der Angreifer, sofern das Resultat nicht beabsichtigt war, verloren, denn nun ist die Zielumgebung aus der Netzwerksicht 100 % sicher und ein konstruktiver Angriff nicht mehr möglich.

AT-Attacke gegen Modems

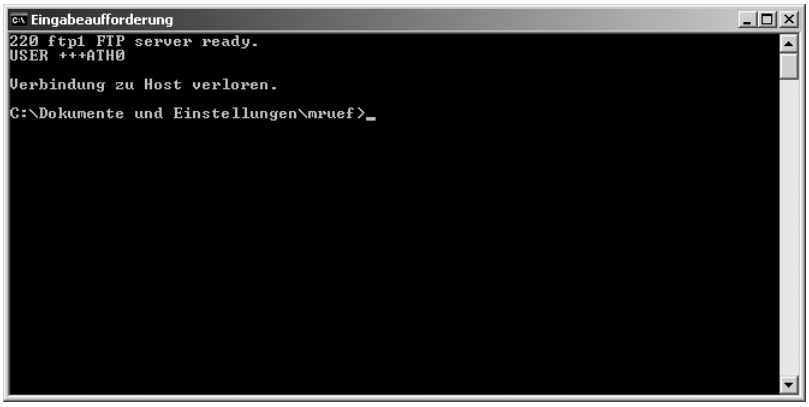
Der amerikanische Modem-Hersteller hat Ende der 70er Jahre eine einheitliche Befehlssprache für Modems entwickelt. Durch diesen Standard lassen sich Geräte, die die Hayes-Kommandos (z. B. *ATZ*, *ATDT* und *ATH0*) unterstützen, ansprechen. Nahezu alle modernen Modems sind mit dem AT-Befehlssatz kompatibel, können Anforderungen annehmen und korrekt verarbeiten.

Normalerweise ist ein solches Modem nur im Offlinebetrieb über den Kommandomodus, also die jeweiligen AT-Befehle, ansprechbar. Wird eine Verbindung aufgebaut, geht das Modem in den Übertragungsmodus über und ist in der Zeit nicht über AT-Befehle ansprechbar, es sei denn, man übergibt dem Modem drei Escape-Zeichen (mit +++ gekennzeichnet), die das Modem als Befehl zum Umschalten in den Kommandomodus interpretiert. Aus Sicherheitsgründen muss zwischen diesem Umschaltkommando in den Kommandomodus und dem ersten AT-Befehl mindestens eine Pause von einer Sekunde vorhanden sein.

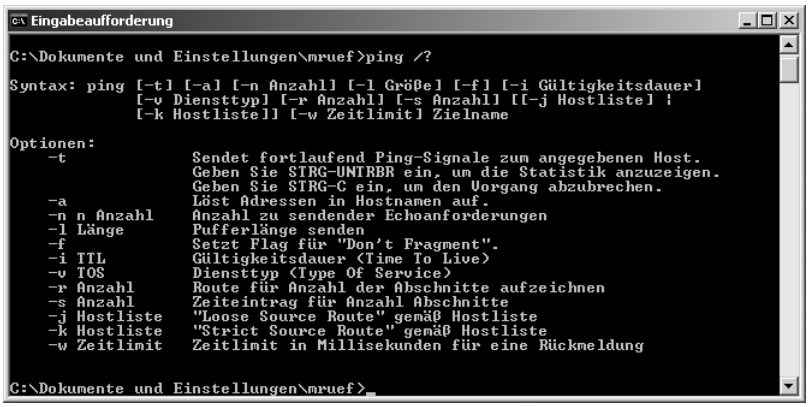
Leider verzichten einige Modemhersteller aus patentrechtlichen Gründen auf diese Pause, sodass bei diesen Modellen der Umschaltbefehl in den Kommandomodus und ein kompletter AT-Befehl direkt hintereinander ohne Zeitverzug eingegeben werden können. Ein Absender schickt an einen Empfänger über das Internet ein spezielles Ping-Paket, das z. B. die Sequenz *+++ATH0* (Umschalten in den Kommandomodus und Beenden der Verbindung) enthält. Laut den ICMP-Spezifizierungen für ICMP echo request-Nachrichten antwortet der Rechner des Empfängers auf die Ping-Anfrage mit der Spiegelung des ICMP-Paketinhalts bei der ICMP echo reply-Rückantwort. Kennt das Modem nun keine Pause zwischen dem Umschalten in den Kommandomodus und dem ersten AT-Befehl, wird es den Paketinhalt des Antwort-Pings als abzuarbeitende Sequenz interpretieren und die Verbindung beenden. Verlief der Angriff erfolgreich, dann wird keine ICMP echo reply-Nachricht zurückkommen und das Zielsystem von nun an bis zur nächsten Einwahl nicht mehr erreichbar sein. Ist das Zielsystem jedoch nicht verwundbar gegen diese Attacke, dann werden im Normallfall ICMP echo reply-Nachrichten zurückkommen und der Host auch weiterhin über das Netzwerk ansprechbar sein.

```
debian:~# ping -p 2b2b2b415448300d -c 3 195.65.88.12
PATTERN: 0x2b2b2b415448300d
PING www.computec.ch (195.65.88.12): 56 data bytes
--- 195.65.88.12 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
debian:~#
```

Dieser spezifische Angriff mit der Hilfe des Ping-Kommandos wird Ping-AT-Attacken genannt. Es sind aber durchaus auch andere interaktive Protokolle, zum Beispiel Telnet oder FTP, für diese Angriffsform nutzbar.



Gerade wenn AT-Angriffe von einem Windows-Betriebssystem ausgeführt werden sollen, muss auf das Verwenden der Microsoft-spezifischen Ping-Implementierung verzichtet werden. Diese hält nämlich den Parameter *-p* nicht bereit, mit dem für die Denial of Service-Attacke zwingend erforderliche Spezifizierung des AT-Patterns hätte getätigt werden müssen.



Die Gegenmaßnahme gegen AT-Attacken ist, die verdächtigen AT-Zeichenketten frühzeitig zu filtern oder gar nicht erst vom verwundbaren Modem verarbeiten zu lassen. Hierzu kann die Initialisierungs-Zeichenkette *s2=255* beim Modem aktiviert werden. Diese sorgt dafür, dass das Modem nicht mehr in den Kommandomodus geschaltet werden kann. Eine andere Lösung könnte sein, nicht mehr die Pluszeichen als die Escape-Zeichen des Modems gelten zu lassen. Diese Umkonfiguration von verwundbaren Modems kann remote durchgeführt werden. Hierfür ist lediglich auf einem UNIX-System die Ping-Eingabe „ping -c 1 -p 2b2b2 b415453323d32353526574f310d <zielsystem>“ nötig. Dieses Kommando verschickt eine ICMP echo request-Anfrage mit der AT-Zeichenkette *+++ ATS2=255&WO1* an das Zielsystem. Das Modem wird intern automatisch die Umkonfiguration vornehmen.

Lokale ICMP-Redirects mittels WinFreeze

Windows NT-Systeme sind gegen eine Denial of Service-Attacke mit dem Namen WinFreeze verwundbar. Wie die folgende tcpdump-Ausgabe zeigt, wird das System des Opfers mit ICMP redirect-Nachrichten, die angeblich vom Router stammen, zugeschüttet. Der Windows-Rechner ist so beschäftigt, die neuen Einträge in seine Routing-Tabelle zu speichern, dass er gar keine Ressourcen mehr für andere Aufgaben bereitzustellen in der Lage ist:

```
07.45.00.140000 router.ch > opfer.ch: icmp: redirect 192.168.0.1 to opfer.ch
07.45.00.200000 router.ch > opfer.ch: icmp: redirect 192.168.0.2 to opfer.ch
07.45.00.260000 router.ch > opfer.ch: icmp: redirect 192.168.0.3 to opfer.ch
07.45.00.330000 router.ch > opfer.ch: icmp: redirect 192.168.0.4 to opfer.ch
07.45.00.400000 router.ch > opfer.ch: icmp: redirect 192.168.0.5 to opfer.ch
07.45.00.480000 router.ch > opfer.ch: icmp: redirect 192.168.0.6 to opfer.ch
07.45.00.590000 router.ch > opfer.ch: icmp: redirect 192.168.0.7 to opfer.ch
```

Es ist gut zu erkennen, dass die WinFreeze-Attacke noch zusätzlich optimiert wurde: Neben den vielen Anfragen in kurzer Zeit (siehe Zeitstempel), wird der Redirect auf das Opfer selbst durchgeführt. Mit anderen Worten: Wenn das Opfer eine IP-Adresse mit verfälschtem Eintrag konntaktieren will, werden die Pakete an sich selbst geschickt. Dadurch wird auf dem angegriffenen System die Auslastung noch einen Tick in die Höhe getrieben.

Die Statistik für Angriffsversuche mit dem WinFreeze-Tool wurde bei der Veröffentlichung des Angriffs-Tools durch Paul Gregoire am 8. März 1999 publiziert. Sie sieht wie folgt aus:

Architektur	CPU	Takt-frequenz	RAM	Betriebssystem	Resultat	Zeitspanne
Intel x86	Pentium	200 MHz	16 MByte	Microsoft Windows 95 OSR 2	Applikationen werden sehr langsam ausgeführt.	nach ca. 20 Sekunden
Intel x86	Pentium	233 MHz	96 MByte	Microsoft Windows NT 4.0 Service Pack 4	Applikationen werden sehr langsam ausgeführt.	nach ca. 30 Sekunden
Intel x86	Pentium II	266 MHz	64 MByte	Microsoft Windows 95	Applikationen werden zuerst sehr langsam ausgeführt und das System friert schlussendlich ein.	nach ca. 20 Sekunden

Es ist zu erkennen, dass in erster Linie das Windows-Betriebssystem an sich ausschlaggebend für die Stabilität bei einer WinFreeze-Attacke ist – und nicht, wie zuerst vermutet, die CPU und das RAM. Windows NT 4.0 ist da eindeutig resis-

tenter als die älteren Windows 95-Varianten, trotzdem kann ein „Angriffserfolg“ im Schnitt nach einer halben Minute verzeichnet werden.

Das Vorhandensein und der Fortschritt dieser Attacke über ICMP redirect-Nachrichten kann auf den betroffenen Windows-Host mit der Eingabe von „route print“ in der Eingabeaufforderung bemerkt werden. Verdächtige und verdächtig viele Einträge deuten auf einen solchen Angriff hin.

```
===========================================================================
Schnittstellenliste
0x1 ........................... MS TCP Loopback interface
0x2 ...00 50 04 67 4f da ...... 3Com EtherLink PCI
===========================================================================
===========================================================================
Aktive Routen:
Netzwerk  Ziel             Netzmaske        Gateway       Schnittst. Metrik
          0.0.0.0          0.0.0.0    192.168.0.254   192.168.0.100   1
      192.168.0.0    255.255.255.0    192.168.0.100   192.168.0.100   1
    192.168.0.100  255.255.255.255        127.0.0.1       127.0.0.1   1
  192.255.255.255  255.255.255.255    192.168.0.100   192.168.0.100   1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1   1
        224.0.0.0        224.0.0.0    192.168.0.100   192.168.0.100   1
  255.255.255.255  255.255.255.255    192.168.0.100   192.168.0.100   1
      192.168.0.1  255.255.255.255    192.168.0.100   192.168.0.100   1
      192.168.0.2  255.255.255.255    192.168.0.100   192.168.0.100   1
      192.168.0.3  255.255.255.255    192.168.0.100   192.168.0.100   1
      192.168.0.4  255.255.255.255    192.168.0.100   192.168.0.100   1
      192.168.0.5  255.255.255.255    192.168.0.100   192.168.0.100   1
      192.168.0.6  255.255.255.255    192.168.0.100   192.168.0.100   1
      192.168.0.7  255.255.255.255    192.168.0.100   192.168.0.100   1
===========================================================================
```

IP-Fragmentierung mittels Ping of Death

Ein einzelnes IP-Paket ist inklusive Header maximal 65.535 Byte lang, Ethernet-Pakete können jedoch maximal 1.500 Byte Daten übertragen. Größere Pakete werden fragmentiert und beim Empfänger wieder defragmentiert, wobei die Zusammensetzung anhand eines Offset-Werts erfolgt. Dies wird gemacht, um Netzwerkabschnitte zu überwinden, die lediglich eine maximale Paketlänge unterstützen. Jedes Paketfragment erhält neben dem Offset-Wert auch noch eine Identifikationsnummer, aber nur das erste enthält den TCP-Header und damit die Portnummer. Dieser Offset-Wert bestimmt für jedes Fragment, wohin es gehört oder wohin es soll. Dadurch ist es möglich, dem letzten Fragment einen Offset zu geben, der inklusive Fragmentgröße einen größeren Wert als die maximalen 65.535 Byte ergibt. Dieses übergroße Ping-Paket erzeugt anschließend einen Buffer-Overflow. Dieser Angriff funktioniert nicht nur mit ICMP und Ping, sondern auch mit UDP und TCP. Obwohl ein ordentlicher Ping-Befehl keine Pakete größer als 65.507 Byte (65.535 Byte abzüglich 20 Byte IP-Header und 8 Byte ICMP-Header) zulässt, bot bei den ersten Versionen von Windows 95 der dort implementierte Ping-Be-

fehl das entsprechende Feature in Form eines Parameters. Einfach „ping -l 65510 zielhost“ eingeben, und der Todes-Ping wird ausgeführt.

Eine der erfolgreichsten Angriffsmöglichkeiten gegen Sniffer sind solche mit fragmentierten IP-Paketen. TCP-Filter können auch in der Regel nicht mehr sicher unterscheiden, ob die Pakete ins interne Netz gelassen werden dürfen, da die Portinformation bei den meisten Paketen fehlt. Wenn die Zielstation nun auch noch unvollständige Paketfragmentfolgen auswertet – und jene auch nicht verwirft – kann eine Firewall ohne größere Probleme umgangen werden.

Fragment-Overlapping mittels Teardrop

Ein einfacher, aber wirkungsvoller Angriff, der IP-Fragmentierung ausnutzt, ist die so genannte Overlapping Fragment Attack nach RFC 1858. Die derzeitige Internetprotokoll Spezifikation RFC 791 beschreibt einen Reassemblierungs-Algorithmus, der neue Fragmente produziert und dabei jeden überlappenden Teil der zuvor erhaltenen Fragmente überschreibt. Wird ein solcher Algorithmus angewendet, kann ein Angreifer eine Folge von Paketen konstruieren, in denen das erste Fragment (mit einem Offset der Länge 0) harmlose Daten beinhaltet (und dadurch von einem Paketfilter weitergeleitet werden kann). Ein beliebiges nachfolgendes Paket mit einem Offset, der größer als 0 ist, könnte TCP-Header-Informationen (z. B. destination port) überlappen. Diese würden durch den Algorithmus modifiziert (überschrieben). Dieses zweite Paket wird von vielen Paketfiltern nicht gefiltert. Die Gegenmaßnahme hierzu ist, Paketfilter zu verwenden, die ein Minimum an Fragment Offset für Fragmente verlangen. Nur wenige neuere TCP/IP-Stacks erkennen dieses Problem und korrigieren dieses. Ältere Router lassen sich mit diesem Trick einfach durchtunneln, sie bieten keinen Schutz. Besonders aber Firewalls, die auf der Basis der **S**tateful **P**aket-**F**ilterung (SPF) arbeiten, wie z. B. Raptor Eagle (heute Symantec Raptor Firewall) und Checkpoint Firewall-1, ließen sich so durchtunneln. Content-Anbieter im Internet und ISPs, die mit diesen Firewalls NT-Server schützen wollten, wurden so Ziel der unzähligen Angreifer, die neue Exploits mal testen wollten.

Anfang 1997 war von dieser Attacke so ziemlich alles betroffen, was einen IP-Stack hatte. Von der Workstation bis zum Drucker war damals praktisch nichts gegen eine fehlerhafte Fragmentierung gewappnet. Abhilfe schafft nur eine vollständige Reassemblierung der TCP/IP-Pakete oder der Einsatz eines Proxy-Elements. Der Nachteil dieser Lösung ist ein enormer Einbruch in der Performance. Dies zeigt aber, dass Firewalls keineswegs perfekt sind. Will man solchen Angriffen zuvorkommen, ist man als Betreiber eines Hochsicherheitssystems auf die ständige Betreuung eines Experten angewiesen. Da von diesem Angriff auch Versionen existieren, die gefälschte Absender-Adressen (IP-Spoofing) ermöglichen, können die Täter oft nicht aufgespürt werden.

ICMP-Stürme mittels Smurf

Wenn man auf eine Broadcast-Adresse einen Ping schickt, hat laut den Spezifizierungen ein jedes System im betroffenen Subnetz mit einer ICMP echo reply-Rückmeldung zu antworten. Je nach Größe des Adressbereichs und der Anzahl antwortender Rechner kann eine beachtliche Anzahl an Rückmeldungen generiert werden. Der Trick besteht darin, die Absenderadresse zu fälschen, sodass das Opfer die Antworten erhält. Sendet man nun rund 1.000 Pakete pro Sekunde und 1.000 Rechner antworten darauf, bedeutet dies also, dass 1.000.000 Pakete pro Sekunde beim Opfer eintreffen. Die verursacht ein solch hohes Datenaufkommen, dass der Rechner des Opfers unter dem enormen Verkehr zusammenbricht.

Dieses Problem war unter dem Namen ICMP-Sturm (engl. ICMP storm) schon länger bekannt. Aber erst durch das im Oktober 1997 erschienene Tool smurf.c gewann diese Angriffsart an Bedeutung. Die Software speichert die Broadcast-Adressen in einem Array und verschickt ICMP-Pakete mit gefälschter Absenderadresse – nämlich diejenige des Opfers – an diese. Smurf-Attacken weisen die folgenden Merkmale auf:

- Eine hohe Auslastung der Netzwerkanbindung kann bemerkt werden. Dies äußert sich in einer spürbar längeren Latenzzeit sowie dem Verlust und der daraus resultierenden Übertragungswiederholung von Paketen. Besonders dann muss von einer Denial of Service-Attacke ausgegangen werden, wenn dieser Umstand innerhalb weniger Sekunden eintritt und seinen Zenith erreicht.
- Das Opfer-System erhält eine Vielzahl von unangeforderten ICMP echo reply-Rückantworten, die von verschiedenen Quellsystemen stammen.
- Zusätzlich kann irgendwo im Netzwerk beobachtet werden, dass jemand (ein Smurf-Angreifer) fortwährend versucht, ICMP echo request-Anfragen an die Broadcast-Adressen des Netzwerks zu schicken.
- Auf betroffenen und verwundbaren Systemen kann eine hohe CPU-Auslastung beobachtet werden.

Zahlreiche Provider wurden durch solche Attacken tagelang arg bedrängt. Dieses Problem lässt sich jedoch durch einen einfachen Trick beheben, indem man an den Routern die IP-Broadcasts nicht mehr in Ethernet-Broadcasts umsetzen lässt und jene somit außen vor bleiben. Außerdem lassen sich viele Systeme so umkonfigurieren, dass sie nicht mehr auf ICMP echo request-Anforderungen antworten. Dadurch kann das Risiko einer Smurf-Attacke in einem Netzwerk minimiert werden, da die Anzahl der auf das falsche ICMP-Datagramm antwortenden Rechner verkleinert wird.

Eine Attacke, die mit dem klassischen Smurf vergleichbar ist, ist unter dem Namen nquake-dos bekannt geworden. Im Gegensatz zu Smurf, das ICMP als Trä-

gerprotokoll benutzt, wird bei nquake-dos das **N**et**Q**uake **P**rotocol (NQP) angewandt. Durch den Versand von NQP-Anforderungen mit gefälschter Absenderadresse an Quake-Server kann das Opfer-System mit einer Welle von Rückantworten bombadiert werden. Wurde ein nquake-Angriff erkannt, konnte er dadurch beendet werden, indem das Opfer jedem Quake-Server, der sich bei ihm unaufgefordert meldet, eine Disconnect-Meldung schickt. Die flutenden Quake-Server werden dann aufgeben, eine Verbindung mit dem Opfer-System herzustellen.

Diese Art des Denial of Service-Angriffs, bei dem mehrere Systeme überlistet werden können, ein Opfer-System mit unsinnigen Meldungen zu bombardieren, kann auch in Zukunft wieder zu einem wichtigen Thema werden. Grundsätzlich müssen die Programmierer von Netzwerkanwendungen darum bemüht sein, dass ihre Entwicklungen nicht einfach blind auf eine beliebige Anfrage antworten. Überall dort, wo automatisiert Rückmeldungen generiert werden, ohne die Herkunft und den Wahrheitsgehalt der ankommenden Nachricht zu prüfen, kann eine Smurf-ähnliche Attacke umgesetzt werden. Einmal mehr birgt die blinde Automatisierung von Prozessen eine Gefahr für die Sicherheit von Computersystemen in sich.

Sockets aufbrauchen mittels SYN- und TCP-Connect-Flooding

Verbindungen mit der Hilfe von TCP werden durch einen Drei-Wege-Handschlag aufgebaut. Der Client initiiert die Verbindung, indem er ein TCP-Datagramm mit gesetzter SYN-Flagge an den TCP-Port des Servers schickt. Stellt das Zielsystem am angesprochenen Port einen Netzwerkdienst zur Verfügung, antwortet der Server mit einem TCP-Datagramm mit gesetzter SYN/ACK-Flagge. Der Client wiederum quittiert die Rückantwort des Servers, indem er abschließend ein TCP-Datagramm mit gesetzter ACK-Flagge zurückschickt. Nun können im Rahmen dieser TCP-Sitzung Daten ausgetauscht werden.

Nun existiert in diesem Szenario die Möglichkeit einer Denial of Service-Attacke. Die Grundlage derer basiert darauf, dass der Initiator den dritten und letzten Schritt nicht durchführt. Dies bedeutet, dass in unserem Beispiel der Client den Drei-Wege-Handschlag nicht mit dem obligaten ACK-Datagramm abschließt:

- Der Angreifer sendet SYNa1 an sein Opfer.
- Das Opfer antwortet automatisch auf SYNa1 mit SYNo1/ACKo1 und wartet auf ACKa1.
- Der Angreifer sendet SYNa2 an sein Opfer.
- Das Opfer antwortet automatisch auf SYNa2 mit SYNo2/ACKo2 und wartet auf ACKa2.
- usw.