Inhaltsverzeichnis

1 Einleitung

Andy Warhol: "In the future, everybody will have 15 minutes of fame"
So beginnt der Artikel über den Warhol Wurm (aus [Weaver 01]), welcher als erste theoretische Überlegung zu den sogenannten Superwürmern gilt. Diese Diplomarbeit beschäftigt sich mit dieser speziellen Art von Computerwürmern, welche bis heute nicht aufgetreten sind. Trotzdem kann anhand die Superwürmer aufgezeigt werden, welchen Bedrohungen die technische Welt in Zukunft ausgesetzt sein könnte. Dafür werden die Superwürmer im ersten Kapitel in den großen Kontext der Bedrohungen für IT-Systeme eingeordnet. Danach werden im zweiten Kapitel einige Superwürmer als theoretische Konstrukte vorgestellt und es werden einige ausgewählte Beispiele zur Verdeutlichung der Arbeits- und Wirkungsweise von Würmern beschrieben und exemplarisch bewertet. Nachdem gezeigt wurde, was Computerwürmer sind, wie sie arbeiten und welchen Schaden sie anrichten können, wird der Begriff des Superwurms definiert. Außerdem wird im dritten Kapitel dargestellt, wie Superwürmer sich von herkömmlichen Würmern unterscheiden, indem ein Ausbreitungsmodell aufgestellt wird und die Charakteristika von Superwürmern beschrieben werden. Da Superwürmer bis heute nur in der Theorie existieren, wird im vierten Kapitel ein Szenario aufgestellt, um zu zeigen, was für Auswirkungen ein Superwurm auf eine konkrete Organisation und das Internet haben könnte. Weiterhin wird beschrieben, wie Superwürmer entdeckt und welche Gegenmaßnahmen getroffen werden können.

1.1 Motivation

Nicholas Weaver beschreibt einen Wurm, welcher innerhalb von 15 Minuten das gesamte Internet infizieren soll (vgl. [Weaver 01a]). Dieser, als Warhol Wurm bekannte, erste theoretische Ansatz eines Superwurms dient als Ausgangspunkt der Diplomarbeit. Als Nächstes haben Stuart Staniford, Gary Grim und Roelof Jonkman auf der Grundlage des Warhol Wurms und der

Rücksprache mit Nicholas Weaver den Flash Wurm entwickelt (vgl. [Staniford 01]). Der Flash Wurm ist ein zweites theoretisches Konstrukt für einen Superwurm. Aus beiden Publikationen geht hervor, dass diese Superwürmer sich sehr schnell ausbreiten und dabei einen verheerenden Schaden im Internet anrichten können. Aus den Artikeln geht jedoch nicht hervor, was genau einen Superwurm von herkömmlichen Würmern unterscheidet. Deshalb beschäftigt sich diese Diplomarbeit im dritten Kapitel mit der Frage: Was macht einen Wurm zum Superwurm?

Doch als erstes sollen die Computerwürmer, von welchen die Superwürmer eine spezielle Art darstellen, in den Kontext der Bedrohungen für IT-Systeme eingeordnet werden.

1.2 Bedrohungen für IT-Systeme

Für Computersysteme gibt es eine Vielzahl von Bedrohungen, für die es bis heute keine gängige Taxonomie gibt. Es wird deshalb versucht Bedrohungen für IT-Systeme zum Einen aus der Sicht des BSI (Bundesamt für Sicherheit in der Informationstechnik) und dem gegenüberstellend aus der Sicht von Klaus Brunnstein darzustellen.

Gefährdungen bzw. Bedrohungen können laut dem IT-Grundschutzhandbuch des BSI von:

- höherer Gewalt,
- organisatorischen Mängeln,
- menschlichen Fehlhandlungen,
- technischem Versagen und
- vorsätzlichen Handlungen

ausgehen. Beispiele für höhere Gewalt sind Überflutung, Blitzeinschlag oder Stürme. Organisatorische Mängel sind unter anderem mangelnde Wartung, unzureichende Schulung der Mitarbeiter oder die zu hohe Komplexität der IT-Systeme. Unter menschlichen Fehlhandlungen werden nicht Fehler

verstanden, die sofort wieder behoben werden können, sondern grobe Fehlhandlungen, wie zum Beispiel von Administratoren, welche die Sicherheitsmaßnahmen umgehen, oder die fahrlässige Zerstörung von Daten oder Datenträgern. Für technisches Versagen gibt es eine Vielzahl von Möglichkeiten: einige Beispiele sind der Ausfall der Stromversorgung, defekte Datenträger oder der Verlust von Nachrichten bei der Übertragung. Die meisten Schäden in der Wirtschaft entstehen durch vorsätzliche Handlungen und gehen meist von den eigenen Mitarbeitern aus und zwar angefangen beim Diebstahl von Disketten bis hin zur Weitergabe von vertraulichen Informationen. Eine andere vorsätzliche Handlung ist das Schreiben und Verbreiten von Malware. Zur Malware gehören unter anderem Viren, Trojanische Pferde und auch Würmer.

Problematisch bei der Klassifizierung des BSI ist, dass die Aufteilung der Bedrohungen nicht klar voneinander abgegrenzt werden kann. So könnten zum Beispiel alle vorsätzlichen Handlungen bei den menschlichen Fehlhandlungen eingeordnet werden. Ein anderes Beispiel ist, dass technisches Versagen häufig auf mangelnde Wartung zurückgeführt werden kann, welches nach der Aufteilung des BSI unter organisatorischen Mängeln steht. Deshalb wird im Folgenden eine andere Sicht auf die Bedrohungen für IT-Systeme vorgestellt.

Nach Klaus Brunnstein [Brunnstein 02] werden Bedrohungen in versehentliche Bedrohungen und absichtliche Bedrohungen eingeteilt. Versehentliche Bedrohungen sind natürliche Ereignisse wie Feuer- oder Wasserschaden oder Wettereinflüsse, menschliche Fehler verursacht durch Programmierer, Operateure oder Benutzer und das Nichtfunktionieren von Ausrüstung und Software wie zum Beispiel von Prozessoren, Speichern oder Netzwerk. Unter absichtlichen Bedrohungen werden

- die Systeminfiltration,
- der Missbrauch von Ressourcen,
- vorsätzliche Schädigung und
- Diebstahl

verstanden. Eine Systeminfiltration ist der absichtliche Versuch einer unautorisierten Person, Zugriff auf Daten zu bekommen, mit dem Ziel die Daten zu verändern, zu zerstören oder unerlaubt zu veröffentlichen. Der Missbrauch von Ressourcen geschieht durch autorisierte Benutzer meist für private Zwecke und führt zum Fehlen der missbrauchten Ressourcen für die eigentlichen Zwecke. Ein Beispiel hierfür ist das unerlaubte Surfen im Internet am Arbeitsplatz. Während des Surfens im Internet stehen weder der Arbeitsplatz in Form eines Computers noch die Arbeitskraft des Mitarbeiters für Tätigkeiten in der Organisation zur Verfügung. Dadurch entstehen der Organisation zusätzliche Kosten für die Benutzung des Internets und unnötige Gehaltskosten. Vorsätzliche Schädigung kann beispielsweise das Zerstören eines Computers mit einem Vorschlaghammer sein, die vorsätzliche Zerstörung von Daten durch Formatierung einer Festplatte oder auch Brandstiftung. Der Diebstahl kann das Stehlen von Daten oder von Hardware beinhalten.

Im Rahmen dieser Diplomarbeit sind insbesondere die Bedrohungen für Software relevant, weswegen die Schwächen von Software als Grundlage für die Bedrohungen näher betrachtet werden müssen:
Solche Schwächen können bei der Spezifikation, der Implementation sowie der Konfiguration von Software auftreten. Bei der Spezifikation einer Software wird festgelegt, welche Aufgaben sie übernehmen und wie sie funktionieren soll. Schwächen der Spezifikation entstehen durch Konzept-, Denk-, und Architekturfehler bei der Softwareherstellung. Während der Implementation einer Software wird die Spezifikation mittels Quellcode und Kompilierung für ein IT-System übersetzt. Programmier- und Kompilierungsfehler sind die Schwächen, die bei der Implementation auftreten. Sie sind auch als sogenannte „bugs“ bekannt. Unter Schwächen der Konfiguration werden Fehler bei der Installation, der Nutzung oder der Wartung der Software verstanden ([Brunnstein 01]).

Die Schwächen in einer Software können dafür ausgenutzt werden, um Systeme mit Malware zu infizieren.

1.3 Malware

In diesem Abschnitt wird der Begriff der Malware definiert sowie ihre möglichen Eigenschaften und ihre verschiedenen Arten dargestellt.

> Definition 1: *Malware* bzw. *maliziöser Code* ist der Oberbegriff für unerwartete oder unerwünschte Effekte in Programmen oder Programmteilen verursacht von einer Person, die absichtlich damit Schaden anrichten will ([Pfleeger 00]).

Diese Definition ist nicht ausreichend, denn eine absichtliche Fehlkonfiguration einer Firewall durch einen verärgerten Administrator, die zum Ausfall derselben führt, macht nach [Pfleeger 00] die Firewall zur Malware. Dabei richtet die Firewall selbst keinen Schaden an, sondern die eigentliche Schutzfunktion der Firewall geht verloren.

Um besser zu verstehen, was Malware ist, werden nun weitere Begriffe sowie Eigenschaften von Malware eingeführt. Diese werden danach für eine genauere Definition von Malware nach [Brunnstein 99] verwendet.

Die *Funktionalität* eines Programms, eines Moduls oder eines Objektes wird durch die Menge aller formalen und nicht formalen Spezifikationen charakterisiert. Durch diese können Informationen über die beabsichtigten Funktionen eines Programmes geschlossen und bestimmte unerwünschte Funktionen ausgeschlossen werden (vgl. [Brunnstein 99]).

Eine Software oder ein Modul wird *dysfunktional* genannt, wenn mindestens eine Funktion von der Spezifikation abweicht (vgl. [Brunnstein 99]). Nach Brunnstein können Dysfunktionen schon bei der Softwareherstellung durch Konzept-, Denk-, Architektur- oder Programmierfehler entstehen. Später können durch Fehler bei der Installation, der Benutzung oder der Wartung Dysfunktionen in eine Software gelangen. Diese Dysfunktionen sind Schwachstellen in der Software, welche von Malware ausgenutzt werden

können, um auf Systeme zu gelangen und sie zu schädigen. Wenn Dysfunktionen absichtlich in eine Software gebracht werden, dann wird diese Software als Malware bezeichnet.

Im Zusammenhang mit Malware spielt die *Selbstreplikation* eine wichtige Rolle. Sie ist die Fähigkeit einer Software, sich selbst zu reproduzieren, indem sie von sich selbst eine Kopie anfertigt. Dabei werden entweder ein vollständiges Programm oder Programmteile in ein anderes Programm kopiert. Der gesamte Umfang der Funktionalität bzw. der Dyfunktionilität des kopierten Programms bzw. der Programmteile muss bei dem Kopierprozess erhalten bleiben.

Eine Software oder ein Modul ist *kontaminiert*, wenn es durch einen Prozess in Malware transformiert wurde (vgl. [Brunnstein 99]). Spezielle Kontaminationsprozesse sind die Infektion, die Ausbreitung und die Trojanisierung:

- *Infektion* ist der Prozess, durch welchen selbstreplizierende Software in eine andere Software eingefügt wird.
- Bei der *Ausbreitung* wird ein Modul über ein Netzwerk zum Beispiel durch einen Wurm infiziert.
- Die *Trojanisierung* ist ein Prozess, bei dem unspezifizierte Funktionen in eine Software eingefügt werden.

Ein Kontaminationsprozess kann getriggert sein, dass heisst er hängt von bestimmten Bedingungen ab.

Die *Schadfunktion* von Malware, auch *payload* genannt, ist eine Funktion, die ein System schädigen soll. Dabei wird es als Schaden angesehen, wenn Dateien verändert oder gelöscht werden, Daten ausspioniert oder Arbeitsprozesse mit dem IT-System behindert bzw. verhindert werden.
Als weitere Schadfunktionen werden die unbefugte Benutzung von Systemressourcen sowie die unbefugte Beanspruchung von Netzwerkkapazitäten angesehen.

Allgemein lässt sich der Schadensbegriff aus der Betriebswirtschaftslehre und den Rechtswissenschaften übernehmen: Ein Schaden liegt vor, wenn eine vom Inhaber ungewünschte Wertminderung eintritt.

Mithilfe der oben eingeführten Begriffe definiert [Brunnstein 99] den Begriff der Malware:

> Definition 2: Eine Software oder ein Modul wird „*maliziös*“ („*Malware*“) genannt, wenn sie mit Absicht dysfunktional ist und es ausreichend Beweise dafür gibt, dass die Dysfunktion einen nachteiligen Einfluss auf die Benutzung oder das Verhalten der Originalsoftware hat.

Malware kann nach verschiedenen Kriterien eingeteilt werden. Abbildung 1 teilt Malware zunächst nach vernetzten Systemen und Einzelsystemen ein, und danach, ob sie selbst replizierend ist oder nicht.

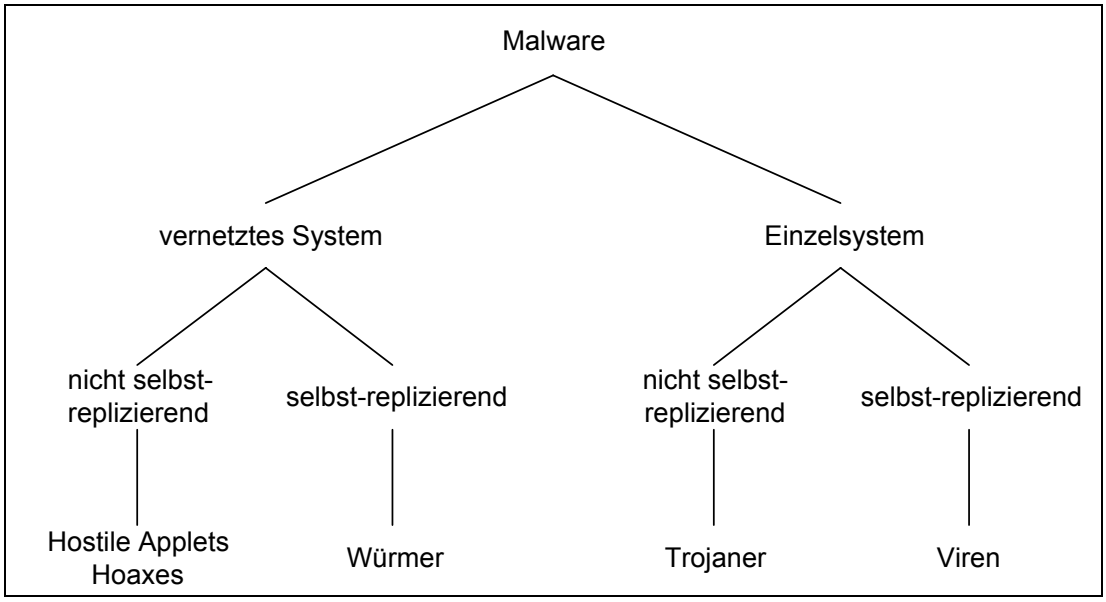

Abbildung 1: „Viren und Malware“ aus der Broschüre des antiVirus Test Center

In den nächsten Abschnitten werden nun einzelne Arten von Malware näher betrachtet.

1.3.1 Viren

Nach [Shirey 00] ist ein Virus ein verborgener, selbst-replizierender Teil einer Software, welcher normalerweise maliziös ist. Die Infektion geschieht in der Regel durch das Einfügen der eigenen Kopie in ein anderes Programm, wodurch es ein Teil des Programms wird. Ein Virus kann nicht selbständig arbeiten. Das Wirtsprogramm des Virus muss ausgeführt werden, um den Virus zu aktivieren. Die Definition von [Shirey 00] lautet wie folgt:

> Definition 3: A hidden, self-replicating section of computer software, usually malicious logic, that propagates by infecting--i.e., inserting a copy of itself into and becoming part of--another program. A virus cannot run by itself; it requires that its host program be run to make the virus active.

[Brunnstein 99] richtet in seiner Definition von Viren den Schwerpunkt auf die Selbstreplikation:

> Definition 4: Jede Software, die selbst-replizierend ist, und sich mindestens zweimal hintereinander selbst-replizieren kann, wird Virus genannt. Dabei wird ein Programm oder eine Datei jeweils mit dem Virus befallen. Dieses Programm bzw. diese Datei heißt Wirt [Brunnstein 99].

Die Infektion eines Programmes geschieht durch Einfügen des Virus in das Programm. Dadurch kann die Software völlig zerstört werden. Der Virus im infizierten Programm beginnt weitere Programme zu infizieren, sobald das Programm ausgeführt werden soll.

Es wird zwischen transienten und residenten Viren unterschieden. Transiente Viren starten und enden mit dem infizierten Programm, und können somit nur während der Laufzeit des Wirtsprogramms andere Programme infizieren. Residente Viren befinden sich im Hauptspeicher und können sich auch weiter verbreiten, wenn der Prozess des infizierten Programms bereits beendet ist.

Für das Einfügen des Viruscodes in ein Programm sind drei Methoden bekannt. Die erste Methode ist das Voranhängen des Virus an ein Programm (vgl. Abbildung 2). Dies ist die einfachste und effektivste Methode (vgl. [Pfleeger 00]). Vor die erste Instruktion des Programms wird der Viruscode eingefügt, so dass beim Aufruf des Programms zuerst der Virus ausgeführt wird. Danach kann das Programm ganz normal weiter laufen, ohne dass der Virus sofort bemerkt wird. Der Virenschreiber muss den Aufbau des Programms nicht kennen.

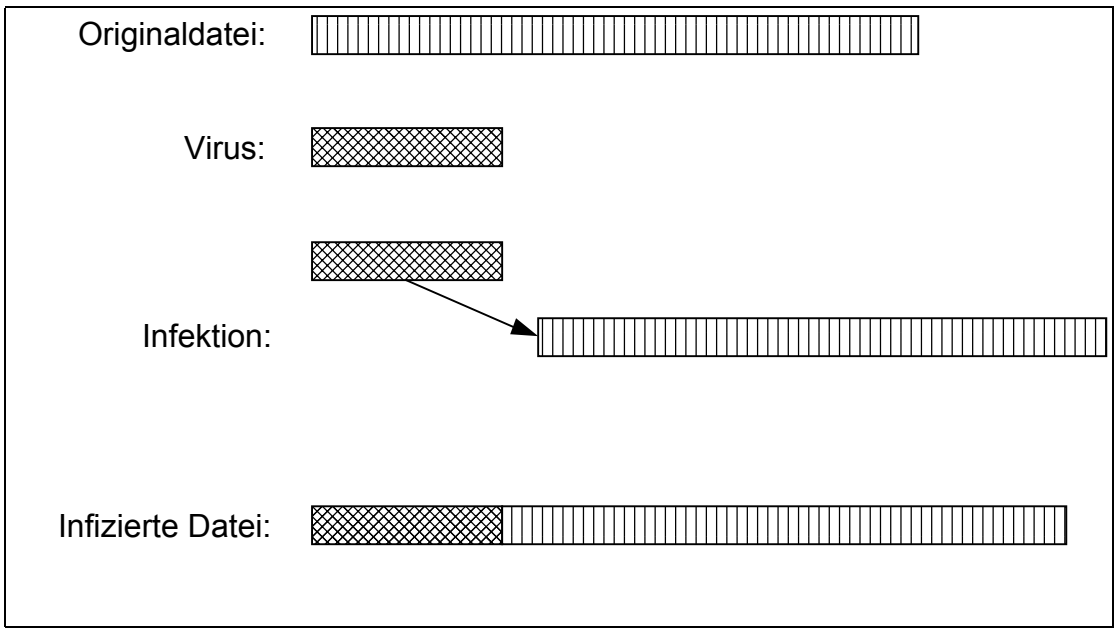

Abbildung 2: Voranhängen eines Virus an eine Datei

Eine zweite Methode ist das Umschließen des Programms durch den Virus. Dadurch hat der Virus die Kontrolle vor und nach der Ausführung des ursprünglichen Programmes ([Pfleeger 00]). Ein Teil des Virus wird wie bei der ersten Methode vor das Programm gehängt. Der zweite Teil des Programms wird an das Programm angehängt und verwischt bzw. löscht alle Spuren, die der Virus bei der Ausführung hinterlassen hat. Solche Spuren sind zum Beispiel eine veränderte Dateigröße oder ein neues Änderungsdatum. Das Umschließen des Programms kann dadurch die Entdeckung des Virus erschweren.

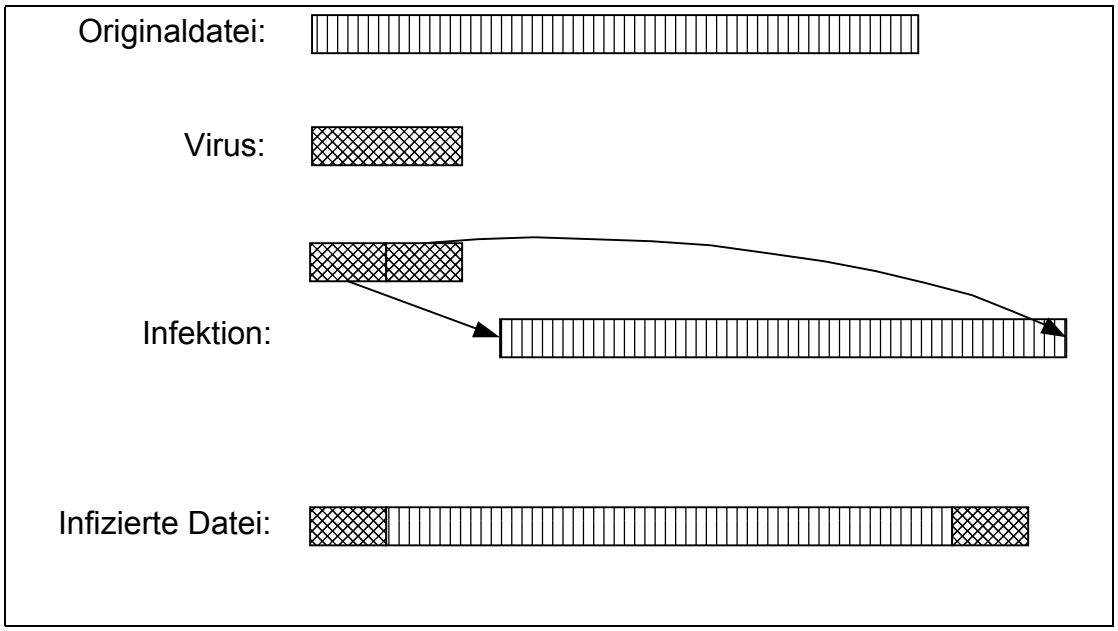

Abbildung 3: Umschließen einer Datei durch einen Virus

Eine weitere Methode ist die Integration des Virus in das Programm. Für diese Methode gibt es zwei Vorgehensweisen. Für die erste Vorgehensweise muss der Virenautor die Struktur des Programmes kennen bzw. der Virus befähigt sein, die Struktur zu erkennen, damit der Virus einige Teile des Programms durch den eigenen Code an geeigneter Stelle ersetzen kann.

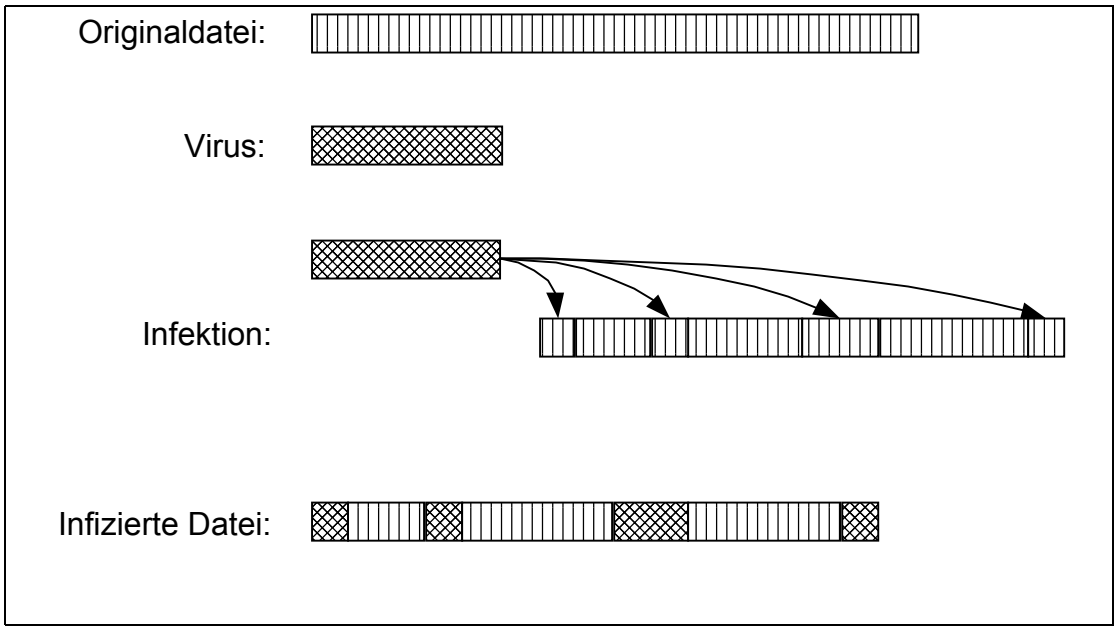

Abbildung 4: Integration eines Virus in eine Datei

Die zweite Vorgehensweise ist besonders schädigend für ein Programm, weil das Programm durch den Virus vollständig ersetzt wird. Dabei kann der Virus versuchen, die Funktionen des Programms zu imitieren, um sich zu tarnen oder seine Funktionen anstelle des Programms durchführen, ohne sich zu verstecken.

Bei Viren fallen Schadfunktion und Infektion häufig zusammen, weil durch die Infektion die Programme unbrauchbar gemacht werden können. Weiterer Schaden entsteht zum Beispiel, wenn der Virus bestimmte Dateien und Programme löscht oder die Festplatte formatiert.

1.3.2 Würmer

Ein Wurm ist nach [Shirey 00] ein selbständig laufendes Computerprogramm. Es verbreitet sich durch Infektion anderer Einzelsysteme in einem Netzwerk mit einer vollständigen, arbeitsfähigen Kopie von sich selbst. Dabei kann es Computerressourcen destruktiv nutzen. Die Definition eines Wurmes von [Shirey 00] ist die Folgende:

> Definition 5: A computer program that can run independently, can propagate a complete working version of itself onto other hosts on a network, and may consume computer resources destructively.

Die Infektion von Einzelsystemen in einem Netzwerk wird bei einem Wurm Ausbreitung genannt. Die Ausbreitung mit Hilfe einer vollständigen und arbeitsfähigen Kopie von sich selbst wird als die Fähigkeit der Selbstreplikation von bestimmter Malware wie Würmern und Viren bezeichnet. Damit können Würmer wie folgt definiert werden:

> Definition 6: Ein *Wurm* ist ein selbständiges Programm, welches sich durch Selbstreplikation in Netzwerken ausbreitet.

Findet die Verbreitung eines Wurmes über das Internet bzw. durch einen der Internetdienste statt, so wird von einem *Internetwurm* gesprochen.

Würmer verbrauchen Netz- und Systemressourcen. Sie können Viren in ein System transportieren und dort Hintertüren und Trojanische Pferde installieren.

Der erste Internetwurm

Der erste Wurm, der sich über das Internet ausbreitete, war der Morris Wurm. Der Morris Wurm, auch Internetwurm genannt, begann seine Ausbreitung am 2. November 1988. Der Autor des Wurmes war Robert Tappan Morris Jr., Student der Cornell University. Der Wurm konnte Rechner der Firmen DEC und SUN befallen, die mit BSD Unix liefen (vgl. [Kossakowski03]).
Es gab vier Wege der Ausbreitung: über `fingerd`, `sendmail`, `rsh` und Passwörter. Bei der `fingerd` Methode wurde mit Hilfe eines Pufferüberlaufs eine Shell aufgerufen, die ihrerseits benötigte Dateien für den Start des Wurms übertrug und ausführte. `Sendmail` ist ein Emailprogramm, welches eine Hintertür besitzt, so dass ein Wurm-Prozess wiederum eine Shell auf dem Zielsystem öffnen konnte (zu Hintertüren vgl. Abschnitt 1.2.4). Die Ausbreitung über `rsh` beruhte darauf, dass einzelne Maschinen sich gegenseitig vertrauten. Dadurch wurde den Maschinen ohne Passwortabfrage der Zugriff erlaubt. Dies ermöglichte wiederum eine Shell auf dem angegriffenen System zu öffnen, um die Wurmdateien zu übertragen. Die letzte Methode bestand darin, die Passwörter für die Benutzeridentifikation auf der Maschine zu erraten. Gelang dies, konnten sie benutzt werden, um auf einen anderen Rechner zu gelangen, wenn die verwendeten Passwörter identisch waren. Hierdurch konnte ebenfalls eine Shell geöffnet werden, um die Wurmdateien zu übertragen. Das Erraten von Passwörtern wird auch als Brute Force Angriff bezeichnet. Hierbei wird überprüft, ob bestimmte häufig verwendete Passwörter wie zum Beispiel „Paßwort“ oder „12345“ benutzt wurden.

Grundsätzlich wurde zunächst ein kleines Programm durch eine Shell auf den Opferrechner übertragen, welches dann den eigentlichen Wurm auf den Opferrechner übertrug. Der Wurm infizierte nur Rechner, die noch nicht befallen wurden. Damit der Wurm aber nicht aufhörte sich auszubreiten, sollte einer von fünfzehn Rechnern noch einmal infiziert werden. Dabei unterlief Robert T. Morris ein Programmierfehler, so dass vierzehn von fünfzehn Rechnern noch einmal befallen wurden. Somit wurden viele Rechner überlastet, was zu einer schnellen Entdeckung des Wurmes führte.
Insgesamt wurden nach Schätzungen 2000 bis 6000 Rechner von damals etwa 60.000 Rechnern im Internet infiziert. Robert T. Morris konnte der Tat überführt werden und wurde zu 3 Jahren Haft auf Bewährung, 10.000 Dollar Geldstrafe und 400 Stunden gemeinnütziger Arbeit verurteilt.

Nach dem ersten Internetwurm breiteten sich viele weitere Würmer im Internet aus. Zu den gefährlichsten und schädlichsten zählen Loveletter, Melissa, Code Red, Nimda und Sapphire/ SQL Slammer.

1.3.3 Trojanische Pferde

Nach [Shirey 00] ist ein Trojanisches Pferd ein Computerprogramm, welches so aussieht, als habe es eine nützliche Funktion. In Wirklichkeit hat es eine versteckte und potenziell maliziöse Funktion, welche manchmal die Sicherheitsmechanismen des Systems umgeht, in dem es die Autorisierung des Systems ausnutzt. Die Definition von [Shirey 00] ist die Folgende:

> Definition 7: A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the program.

Diese Definition ist zu speziell, weil diese die maliziöse Funktion eines Trojanischen Pferdes zu stark auf die Umgehung der

Sicherheitsmechanismen für die Autorisierung beschränkt. Mit einem Trojanischen Pferd können zwar Passwörter für die Autorisierung ausspioniert werden, aber auch sämtliche anderen sensiblen Daten, die sich auf dem System befinden, können an unbefugte Personen weitergegeben werden. Deswegen soll ein Trojanisches Pferd wie folgt definiert werden:

> Definition 8: Ein *Trojanisches Pferd* ist ein maliziöses Programm, das neben der eigentlichen Funktion eine weitere nicht offensichtliche maliziöse Funktion besitzt. Die maliziöse Funktion eines Trojanischen Pferdes ist seine Schadfunktion.

Es handelt sich um eine nicht offensichtlich maliziöse Funktion, wenn ein Großteil der Benutzer die maliziöse Funktion nicht erkennen können.

1.3.4 Hintertüren

Hintertüren sind besser bekannt unter den englischen Begriffen „backdoor“ oder „trapdoor“. Aus dem eigenen Verständnis werden Hintertüren wie folgt definiert:

> Definition 9: Eine *Hintertür* ist eine Zusatzfunktion in einem Programm, die einen verdeckten Zugang in das Programm bzw. System ermöglicht. Der Zugang kann mit besonderen Privilegien wie Administratorrechten verbunden sein.

[Shirey 00] unterscheidet die Hintertüren noch nach „trapdoors“ und „backdoors“. Seine Definitionen lauten wie folgt:

> Definition 10: Eine „*trapdoor*“ (zu deutsch Falltür) ist eine versteckte Computerschwäche, die einem Eindringling bekannt ist oder ein versteckter Computermechanismus, der von einem Eindringling installiert wurde. Der Eindringling kann die „trapdoor“ aktivieren, um Zugriff auf den

Computer zu bekommen, ohne dass ein Sicherheitsmechanismus ihn darin hindern kann ([Shirey 00]).

Definition 11: Eine „*backdoor*“ (zu deutsch Hintertür) ist ein Hardware- oder Softwaremechanismus, der erstens einen Zugriff auf ein System und seine Ressourcen erlaubt durch ein anderes Verfahren als dem Üblichen, zweitens von einem Systemgestalter oder Administrator absichtlich hinterlassen wurde und drittens nicht veröffentlicht wurde ([Shirey 00]).

Hintertüren in Programmen dienen ursprünglich dem Zweck, auch im Notfall schnell in ein Programm zu gelangen. Die Hintertüren können auch von unbefugten Personen dazu benutzt werden, in das Programm zu gelangen, oder Administratorrechte für ein System zu erhalten und damit das System zu kontrollieren.

Hintertüren sind nicht selbst replizierend und sie befinden sich immer nur auf Einzelsystemen. Sie können mit Hilfe von anderer Malware wie zum Beispiel Würmern in vernetzten Systemen verbreitet werden, aber sie können sich nicht eigenständig ausbreiten. Sie liegen deshalb in einer Klasse mit den Trojanischen Pferden.

Ein Beispiel für eine bekannte Hintertür ist diejenige, die der Wurm Code Red II nach der Infektion eines Systems installiert hat und die später durch den Wurm Nimda unter anderem zur Ausbreitung genutzt wurde.

1.3.5 Hoaxes

„Hoaxes“ bedeutet „böse Scherze“. Sie werden per Email an ahnungslose Computernutzer verschickt und enthalten Warnungen vor zum Beispiel „gefährlichen Viren“, die jedoch nicht existieren. Die Benutzer werden in der Email aufgefordert, die Warnung an ihre Bekannten weiter zu verschicken. Durch Hoaxes werden die Computernutzer irritiert und verunsichert.

Manchmal fordern die Warnungen auch auf, bestimmte Dateien auf ihrem System zu löschen, um den „gefährlichen Virus“ zu entfernen oder andere Maßnahmen zu treffen, um sich zu schützen. In Wirklichkeit schaden sich die Benutzer dadurch nur selbst, weil die Warnung und die dazu gehörigen Schutzmaßnahmen frei erfunden sind.

1.3.6 Hostile Applets

Ein Applet ist ein kleines Programm, welches Instruktionen für den Aufbau einer zu übertragenden Webseite an ein empfangenes System enthält. Damit sie keinen Schaden auf einem System anrichten können, laufen sie in einer sogenannten „Sandbox“, die verhindert, dass das Applet Zugriff auf die Betriebssystemebene bekommt. Andere Applets werden eingesetzt, um die zu übertragenen Datenmengen einer Webseite möglichst klein zu halten, indem zum Beispiel die Applets Bilder vor Ort auf dem System errechnen.
Hostile Applets sind „feindliche“ Applets. Sie nutzen Fehlimplementationen in den „Sandboxes“, um Schaden anzurichten. So können sie geheime Daten wie Passwörter ausspähen oder sie können Daten verändern oder löschen. Des Weiteren ist es ihnen möglich, Viren und andere Malware auf einem System zu installieren.

1.4 Zusammenfassung

Für IT-Systeme gibt es eine Reihe von Bedrohungen. Eine der größten Bedrohungen ist Malware, weil ihre Existenz nur dem Zweck dient, Schaden anzurichten. Im laufe der vergangenen Jahre hat sich eine große Vielfalt von unterschiedlicher Malware entwickelt. Durch das Internet kann Malware schneller auf eine Vielzahl von IT-Systemen verteilt werden. Vor allem Internetwürmer richteten in den letzten Jahren große Schäden an. Die Weiterentwicklung von Internetwürmern könnten Superwürmer sein.

2 Existierende Würmer und theoretische Superwürmer

In diesem Kapitel werden die als Superwürmer angekündigten bzw. die Vorläufer der Superwürmer betrachtet. Als erstes werden der Warhol Wurm und der Flash Wurm dargestellt. Sie existieren bisher nur als theoretische Konstrukte von Superwürmern. Als nächstes werden die Würmer Code Red I, Code Red II und Nimda beschrieben, die wegen ihrer schnellen Ausbreitung als Vorläufer der Superwürmer aufgefasst werden können. Zuletzt werden Lioten und Sapphire analysiert, für die zumindest der Anspruch erhoben wurde, dass es sich um Superwürmer handelt bzw. die einem Superwurm sehr nahe kamen.
Bei allen in diesem Kapitel dargestellten Würmern werden vor allem die Ausbreitungsmechanismen betrachtet und bewertet, um sie später weiter zu verwenden.

2.1 Der Warhol Wurm

Der Warhol Wurm ist der erste Ansatz für einen Superwurm, deshalb soll er hier als erstes erläutert werden. Der Name „Warhol Wurm“ bezieht sich auf Andy Warhols Aussage: "In the future, everybody will have 15 minutes of fame" (vgl. [Weaver 01]). Nicholas C. Weaver von der University of California in Berkeley beschreibt in seiner Arbeit „Warhol Worms: The Potential for Very Fast Internet Plagues“ wie sich solch ein Wurm innerhalb von 15 Minuten im gesamten Internet ausbreiten könnte.

Um das Internet schnell zu infizieren, muss der Wurm sehr effizient neue Ziele in Form von verwundbaren Systemen finden. Hierzu werden neue Ausbreitungsmechanismen benötigt. Die Suche nach neuen Zielen während der Ausbreitung wird als Scannen bezeichnet. Für das Scannen generiert der Wurm eine IP-Adresse. Findet er unter dieser Adresse einen Rechner mit zum Beispiel dem benötigten Dienst oder einer nutzbaren Schwachstelle,

verschickt er eine Kopie von sich dorthin. Eine andere Möglichkeit ist das direkte Verschicken der Wurmkopie an die erzeugte IP-Adresse ohne eine vorherige Analyse des Zielsystems. Dadurch brauchen keine Pakete ausgetauscht zu werden, und wenn das System infizierbar ist, wird es sofort durch die verschickte Wurmkopie infiziert.
Im Folgenden werden das Scannen nach einer Hitliste, das zufällige Scannen nach IP-Adressen, das permutierte Scannen und das verteilte permutierte Scannen betrachtet.

2.1.1 Scannen nach einer Hitliste

Beim Scannen nach einer Hitliste erstellt der Wurmautor eine Liste von verwundbaren Systemen im Internet, bevor er den Wurm schreibt. Eine solche Hitliste kann etwa 10.000 bis 50.000 Einträge mit guten Netzwerkverbindungen enthalten. Es sollten laut David Moore ([Caida.org]) mindestens 10.000 Einträge sein, weil die ersten 10.000 Infektionen die Infektionsrate bestimmen. Abbildung 5 zeigt die Simulation der Infektionsraten für einen zufälligen Scan, der einmal mit einer Hitliste von 10.000 und ein anderes Mal mit einer Hitliste von 10 Einträgen startet. Die Kurven enden, wenn 99% der hier angenommenen 1.000.000 verwundbaren Systeme infiziert sind.

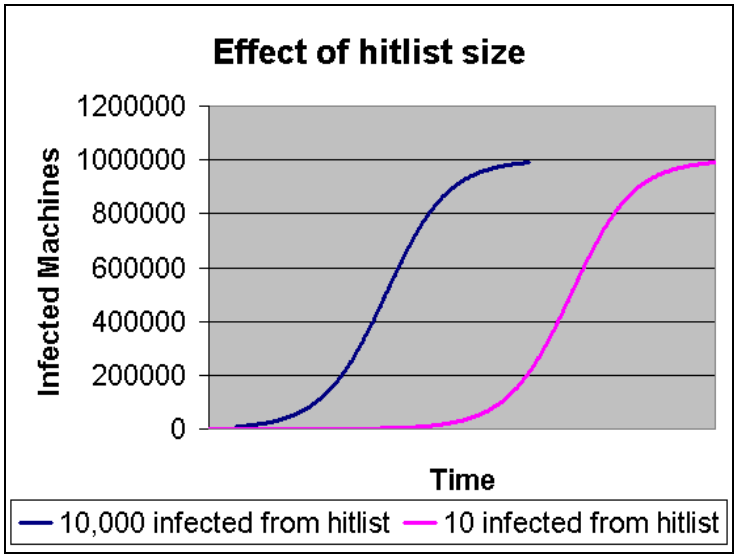

Abbildung 5: Vergleich der Ausbreitung eines Wurms durch eine Hitliste mit 10 und mit 10.000 Einträgen (aus [Weaver 01])

Wenn der Wurm initialisiert wird, scannt er die Einträge der Liste durch. Da die Hitliste vor der Initialisierung des Wurms erstellt wurde, müssen nicht mehr alle eingetragenen Systeme infizierbar sein. Einige Systeme können inzwischen gepatcht worden sein oder sie wurden durch andere Maßnahmen geschützt oder sie sind während des Scanprozesses ausgeschaltet. Findet der Wurm ein verwundbares System, wird die Liste in zwei Hälften geteilt. Die eine Hälfte wird der Wurmkopie mitgeschickt. Die andere Hälfte verbleibt bei der Originalkopie und wird dort weiter abgearbeitet. Beide Würmer wenden dieses Verfahren nun parallel an. Der Wurm wird durch das Aufteilen der Liste insgesamt kleiner und kann so schneller über das Netz verteilt werden. Nachdem die Hitliste abgearbeitet ist, muss ein anderer Mechanismus die Verbreitung des Wurmes übernehmen.

Nach [Weaver 01] ist das Erstellen einer Hitliste einfach. Ein Scan nach allen laufenden Diensten auf den Systemen (nicht nach bestimmten Sicherheitslücken) fällt nicht besonders auf und kann mit dem später geschriebenen Wurm nicht in Zusammenhang gebracht werden. [Weaver 01] schlägt als Beispiel vor, dass auch öffentliche Server wie der Netcraft Survey dazu benutzt werden können, die Liste der verwundbaren Systeme zu erstellen, indem sie nach bestimmten Diensten suchen, ohne dass jemand aufmerksam wird.

Es reicht aus, wenn noch 10 bis 20% der Einträge in der Hitliste auf verwundbare Systeme verweisen, wenn der Wurm gestartet wird. Der Wurm kann nach [Weaver 01] dann alle noch verwundbaren Systeme, die auf der Hitliste stehen, in unter einer Minute infizieren.
Durch eine Hitliste kann also die Zeitspanne, bis zu der die Anzahl der infizierten Systeme exponential ansteigt, wesentlich verkürzt werden. Somit kann durch eine Hitliste die gesamte Ausbreitung eines Wurms beschleunigt werden.

2.1.2 Zufälliges Scannen

Nachdem die Liste vollständig gescannt wurde, übernimmt ein anderer Mechanismus die Ausbreitung des Wurmes. Dies könnte die zufällige Auswahl von IP-Adressen im Adressraum sein. Ein Beispiel für zufälliges Scannen (auch „random scanning“) ist im Abschnitt 2.7 beschrieben. Das zufällige Scannen ist, wie in Abbildung 6 zu sehen, ein guter Ausbreitungsmechanismus. Die Zufälligkeit führt aber auch zu Nachteilen. Systeme können nach [Staniford 02] mehrmals durch den Zufallsalgorithmus ausgewählt werden, wodurch sie mehrfach gescannt werden. Da beim zufälligen Scannen in der Regel nicht überprüft wird, ob ein System bereits infiziert ist, kommt es hierbei zu Mehrfachinfektionen. Mehrfache Scans und Infektionen eines Systems verbrauchen unnötig viel Zeit und Ressourcen. Ein weiterer Nachteil ist, dass es keine Garantie gibt, dass der Wurm terminiert, wenn alle verwundbaren Systeme infiziert sind. Dies führt wiederum zu unnötigen Mehrfachinfektionen, welche die Ausbreitungsgeschwindigkeit verlangsamen.
Insgesamt läuft das zufällige Scannen unkoordiniert ab, wodurch es unkontrollierbar für den Wurmautor wird.

2.1.3 (Verteiltes) Permutiertes Scannen

Besser als das zufällige Scannen ist das permutierte Scannen, weil der Wurm mit diesem Verfahren erkennen kann, ob ein System schon infiziert ist ([Staniford 02]). Beim permutierten Scannen wird der IP-Adressraum in Permutationen aufgeteilt. Jedem Wurm wird eine Permutation, d.h. ein Bereich des IP-Adressraumes, durch einen Zufallsalgorithmus zugewiesen, wobei mehrere Wurminstanzen dieselbe Permutation zugewiesen bekommen können. Jede Wurmkopie wählt einen zufälligen Startpunkt in der Permutation, von wo er mit dem Scan beginnt. Findet der Wurm hinter einer IP-Adresse ein verwundbares System, versucht er, es zu infizieren. Nach erfolgreicher Infektion des Systems beginnt die dortige Wurmkopie ebenfalls mit dem permutierten Scannen nach verwundbaren Systemen und deren Infektion. Steht hinter einer IP-Adresse kein verwundbares System oder hat der Wurm

ein System infiziert, setzt er den Scan mit der nächsten IP-Adresse in der Permutation fort. Findet der Wurm während seiner Suche durch die Permutation ein bereits infiziertes System, initialisiert er sich mit einem neuen Startpunkt.

Nach [Staniford 02] hat dieses Verfahren den Vorteil, dass der Wurm sich beim Scannen nach verwundbaren Systemen selbst koordinieren kann. Auf den ersten Blick sieht es so aus, als würden die Würmer ein zufälliges Scannen durchführen. Dabei werden beim permutierten Scannen die IP-Adressen nicht mehrmals gescannt und damit ein System nicht mehrfach infiziert. Sobald eine Instanz des Wurmes ein bereits infiziertes System findet, braucht es von hier aus nicht weiter nach verwundbaren Systemen zu scannen, weil bereits eine andere Instanz des Wurmes dabei ist, alle nachfolgenden IP-Adressen zu scannen. Aus diesem Grund wählt die Instanz des Wurms einen neuen Startpunkt in der Permutation, um weitere Systeme zu infizieren. Nach einer bestimmten Anzahl von bereits infizierten Systemen hintereinander kann der Wurm das Scannen stoppen, weil davon auszugehen ist, dass in der Permutation alle verwundbaren Systeme infiziert sind. Durch einen Timer könnte der Wurm nach einen bestimmten Zeit den Scanprozess wieder aufnehmen, wobei er eine andere Permutation, d.h. einen anderen Bereich des IP-Adressraumes wählt. Dort werden alle IP-Adressen noch einmal gescannt, so dass alle neu hinzu gekommenen verwundbaren Systeme ebenfalls infiziert werden. Auch ehemals infizierte Systeme, die in der Zwischenzeit gereinigt wurden, aber immer noch verwundbar sind, werden wieder infiziert.

Ein Verbesserung des Scanprozesses kann durch eine Aufteilung der Permutation (partitoned permutation) erreicht werden. Sobald der Wurm ein System infiziert hat, teilt er die Permutation in zwei Hälften. Die eine Hälfte bekommt die neue Wurmkopie. Die andere Hälfte scannt er selbst. Die Instanzen des Wurmes teilen wiederum ihren Teil der Permutation, sobald sie ein System infizieren. Dies geschieht solange bis der Permutationsteil einer Instanz eine gewisse Größe unterschreitet, dann scannt die Instanz mit dem

einfachen permutierten Scannen weiter. Das verteilte permutierte Scannen ist, wie in Abbildung 6 zu sehen ist, nicht sehr viel aber durchaus sichtbar schneller als das permutierte Scannen.

Permutiertes Scannen als Ausbreitungsmechanismus hat den Vorteil, dass dieser Mechanismus sich selbst koordinieren kann, was zu einer höheren Infektionsrate führt im Gegensatz zu Ausbreitungsmechanismen, die sich nicht selbst koordinieren können wie das zufällige Scannen.
Abbildung 6 zeigt die unterschiedlichen Infektionsraten für zufälliges, permutiertes und verteiltes permutiertes Scannen:

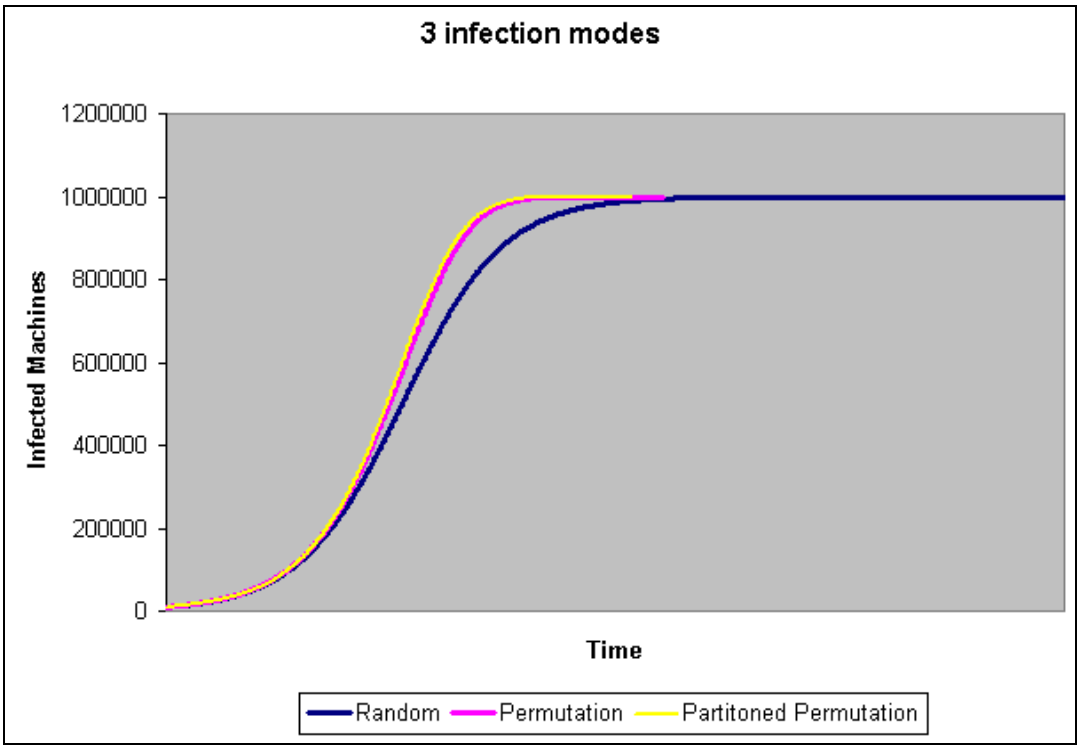

Abbildung 6: die Infektionraten von zufälligen, permutierten und verteilten permutierten Scannen im Vergleich (aus [Weaver 01]).

Der Warhol Wurm soll eine Kombination aus Scannen nach einer Hitliste und permutiertem Scannen für seine Ausbreitung benutzen. Eine genaue Schadfunktion wurde in [Weaver 01] nicht geschildert, wobei vom Stehlen und Löschen von Daten bis hin zu DDOS-Angriffen (distributed denial of service) alles denkbar wäre. Bei einem DDOS-Angriff wird ein Dienst oder ein System von mehreren Rechnern aus dem Internet mit einer großen Anzahl von

Anfragen attackiert. Der Dienst oder das System kann keine Anfragen mehr abarbeiten und steht somit für die eigentlichen Aufgaben nicht mehr zur Verfügung.
Ob der Warhol Wurm es wirklich schaffen würde, alle verwundbaren Systeme im Internet innerhalb von 15 Minuten zu infizieren, ist ungewiss. Damit er es schaffen könnte, bräuchte der Warhol Wurm eine signifikant größere Anzahl von schnelle Verbindungen im Internet für seine Ausbreitung als zur Zeit verfügbar ist.

2.2 Der Flash Wurm

Stuart Staniford, Gary Grim und Roelof Jonkman von Silicon Defense haben die Idee des Warhol Wurms von Nicholas C. Weaver von der UC Berkeley aufgegriffen (vgl. [Staniford 01]) und den Flash Wurm entwickelt.

Der Warhol Wurm sollte das Internet innerhalb von 15 Minuten infizieren. Ein Flash Wurm soll dies in 30 Sekunden schaffen. Aus diesem Grund muss die Effizienz und die Effektivität des Wurms im Vergleich zum Warhol Wurm noch einmal gesteigert werden.

Die Idee des Flash Wurmes ist, alle verwundbaren Systeme im vornherein ausfindig zu machen. Im Gegensatz dazu hat der Warhol Wurm nur einen Teil der verwundbaren Systeme vorher in einer Hitliste gesammelt. Für die Suche nach verwundbaren Systemen gibt es eine Reihe von Strategien:

- Über eine entsprechend schnelle Verbindung kann nach Auffassung von [Staniford 01] das Internet innerhalb von 2 Stunden von einer Site aus gescannt werden und alle verwundbaren Systeme in einer Liste erfassen.
- Wie bei einer distributed denial of service (DDOS) Attacke kann auch das Internet verteilt durch eine große Anzahl von Systemen gescannt werden.

- Da Portscans alltäglich sind, kann durch sie Scans nach verwundbaren Systemen durchgeführt werden, ohne dass sie auffallen.
- Spam Mail Listen können dazu benutzt werden, um eine Liste von Domänen zusammenzustellen. Domain Name Server (DNS) können damit nach IP-Adressen von Mail- und Web-Servern durchsucht werden.
- Schließlich können sogenannte „web-crawling Techniken“, welche den Suchmaschinen ähnlich sind, dazu benutzt werden, häufig mit dem Internet verbundene Systeme zu finden.

Die auf diese Art erstellte Hitliste, kann in *n* Blöcke geteilt werden. Der Wurm sucht und infiziert die ersten Adressen in jedem Block oder wählt Adressen aus, die eine Verbindung mit einer hohen Datenrate haben und infiziert diese als erstes. Bei der Infektion gibt er jedem Folgewurm den entsprechenden Block der Liste mit. Der Prozess wird iterativ fortgesetzt. Entweder gibt der Wurm alle Adressblöcke an die Folgewürmer weiter und beginnt danach mit einem alternativen Ausbreitungsmechanismus (z.B.: zufälliges Scannen) oder er übergibt die ersten *n-1* Adressblöcke an die Folgewürmer und führt den gleichen Scanprozess mit dem letzten Adressblock fort. Durch einen alternativen Ausbreitungsmechanismus kann der Wurm weitere Systeme infizieren, welche nicht in der Hitliste erfasst sind, weil sie neu hinzugekommen sind.

Zwei Probleme müssen bei der Ausbreitung des Flash Wurmes beachtet werden. Der Wurm und seine Liste der verwundbaren Systeme sind am Anfang sehr groß (Schätzungsweise: 48 MB). Da die Liste immer weiter aufgeteilt wird, werden die Wurmkopien immer kleiner und ihre Größe ist nach ein paar Iterationen für die Übertragung über Netzwerke mit langsamen Verbindungen kein Problem mehr. Nur am Anfang der Ausbreitung sollten Verbindungen mit einer hohen Datenrate benutzt werden. Auch macht das Komprimieren der Liste den Wurm wesentlich kleiner, wobei andererseits die Komprimierung und Dekomprimierung der Hitliste Zeit in Anspruch nehmen.

Das andere Problem ist die Verzögerungszeit, die sich ergibt, wenn der Wurm seine Liste durchgeht, seine Kopien erstellt und diese auf die verwundbaren Systeme überträgt. Die Verzögerungszeit durch die Übertragung des Wurms könnte auch hier durch schnelle Verbindungen verkleinert werden. Allerdings hängt die Verzögerungszeit für die Infektion eines Systems von dem jeweiligen System selbst ab und kann nur schwer abgeschätzt werden.

Beim Flash Wurm ist es noch fraglicher als beim Warhol Wurm, ob er es schaffen würde, in der angegeben Zeit alle verwundbaren Systeme im Internet zu infizieren. Zwar verwendet der Flash Wurm einen abgewandelten Ausbreitungsmechanismus, doch auch bei diesem würden ähnliche Probleme auftreten wie beim Warhol Wurm. Zusätzlich kommt das Problem hinzu, dass nachdem die Hitliste abgearbeitet wurde, keine weitere Ausbreitung stattfindet. Vermutlich werden um so weniger Systeme infiziert, um so älter die Einträge auf der Hitliste sind. Dadurch ist der Flash Wurm in seiner Ausbreitung stark begrenzt und der hier beschriebene Mechanismus könnte eher als Teil der Ausbreitung eines anderen Superwurms Verwendung finden.

Als nächstes werden drei Würmer betrachtet, die in der Praxis sehr großen Schaden angerichtet haben, insbesondere wegen ihrer schnellen Ausbreitung. Innerhalb von drei Monaten traten Code Red I, Code Red II und Nimda das erste Mal auf und beeinflussten sich sogar gegenseitig.

2.3 Code Red I

Die erste Version von Code Red (CRv1) wurde am 13. Juli 2001 entdeckt ([Staniford 02]). Der Wurm nutzt die .ida Schwäche in Microsoft IIS Webservern beschrieben in CVE-2001-0500. Der Wurm wird nicht auf der Festplatte gespeichert, sondern agiert allein aus dem Hauptspeicher heraus ([McAfee 01]). CRv1 startet 99 Prozesse, wobei jeweils eine zufällige IP-Adresse generiert wird. Jeder der 99 Prozesse versucht dann, unter der jeweiligen Adresse ein System zu infizieren. Ist die Infektion erfolgreich, sucht

er nach der Datei C:\notworm. Wenn C:\notworm existiert, geht der Wurm zunächst in einen Schlafmodus über, um später zu einer bestimmten Zeit wieder aktiviert zu werden. Existiert die Datei nicht und ist das Datum vor dem 20. eines Monats, werden wieder 99 Prozesse für weitere Infektionen gestartet. Ein 100. Prozeß gestaltet den Webserver selbst um, wenn die benutzte Sprache auf dem System U.S.-Englisch ist. Anstatt der normalen Webseite steht dort:

Abbildung 7: Eine durch Code Red veränderte Webseite (aus [Erdelyi 01])

Dem Autor von CRv1 ist jedoch ein Fehler unterlaufen. Der Zahlengenerator für die zu infizierenden IP-Adressen besitzt einen festen Kern. Dadurch generieren alle Wurmkopien auf allen Systemen in den einzelnen Prozesse dieselben IP-Adressen und versuchen diese dann zu infizieren. Es wurden deshalb nicht besonders viele Systeme durch CRv1 infiziert.

Am 19. Juli 2001 breitete sich eine 2. Version von Code Red aus. Diese wurde als Code Red I oder CRv2 bekannt. Der Code von Code Red I unterscheidet sich nur geringfügig von CRv1. So wurde der Fehler im Zahlengenerator behoben. Es gibt keine Webseitenverschandelungen, ein so genanntes „Defacement“, mehr. Stattdessen wird die IP-Adresse von www.whitehouse.gov über eine DDOS-Attacke angegriffen. Die DDOS-Attacke findet zwischen dem 20. und dem 28. eines Monats statt. Dann

senden alle aktiven Prozesse, anstatt weitere Systeme zu infizieren, große Mengen an unnützen Daten an den Port 80 der IP-Adresse 198.137.240.91, welche zu www.whitehouse.gov gehörte. Die IP-Adresse wurde inzwischen geändert (vgl. [Chien 02]). Des Weiteren verbraucht das Erzeugen der vielen Prozesse auf einem System eine Menge Systemressourcen und macht es instabil.

Wenn das Datum nach dem 28. eines Monats ist, geht Code Red I in einen Schlafmodus über. Am 1. eines Monats beginnt die Infektionsphase von neuem.

Um Code Red I von einem System zu entfernen, reicht ein Neustarten des Systems aus, da damit das Löschen der Inhalte des Hauptspeichers einher geht. Durch das Einspielen der entsprechenden Patches wird eine Neuinfektion verhindert.
Dieser Wurm richtete noch relativ wenig Schaden an, dies sollte sich mit Code Red II ändern.

2.4 Code Red II

Am 4. August 2001 trat zum ersten Mal der Wurm Code Red II auf, der sich anschließend sehr schnell verbreitete (vgl. [Staniford 02]). Er benutzt dieselbe Schwäche im IIS Webserver wie CRv1 und CRv2. Wenn er ein System erfolgreich infiziert hat, installiert er eine Hintertür (backdoor), die einen unbemerkten Zugriff auf das System erlaubt. Code Red II konnte ein System nur infizieren, wenn ein IIS Server auf einem Windows 2000 Betriebssystem läuft. Auf Windows NT-Systemen verursacht Code Red II sofort einen Systemabsturz.

Auch Code Red II befindet sich nur im Hauptspeicher und wählt zufällig IP-Adressen aus, um weitere Systeme zu infizieren. Die zugrunde liegende

Wahrscheinlichkeitsverteilung sorgt für einen lokal ausgerichteten Ausbreitungsmechanismus: Er wählt mit einer Wahrscheinlichkeit von 3/8 eine zufällige IP-Adresse aus dem Klasse B Adressraum, um Systeme zu infizieren. Mit einer Wahrscheinlichkeit von 1/2 wählt er eine IP-Adresse von seinem derzeitigen Klasse A Adressraum. Vom gesamten Internet sucht er sich mit einer Wahrscheinlichkeit von 1/8 zufällige IP-Adressen aus. Dadurch befällt er vor allem Systeme mit benachbarten IP-Adressen (vgl. [Erdelyi 01]). Dieser Ausbreitungsmechanismus wird auch als *lokales Scannen* bezeichnet. Der Wurm kann durch diese Strategie sehr schnell ganze Intranets infizieren, sobald er durch eine ggf. existierende Firewall gedrungen ist.

Hat der Wurm ein System infiziert, startet er 300 Prozesse, um nach weiteren verwundbaren Systemen zu suchen. Er scannt genau 24 Stunden lang. Danach startet er das System neu. Der Hauptprozess des Wurmes plaziert im System ein Trojanisches Pferd, welches auch nach dem Neustart im System verbleibt. Der Rest des Wurms wird durch den Neustart gelöscht. Das Trojanische Pferd schafft eine Hintertür, die einen Fernzugriff auf das System erlaubt. Außerdem prüfen alle Prozesse des Wurmes, ob es Oktober ist oder das Jahr 2002. Dann wird das System ebenfalls neu gestartet (vgl. [Szor 02]).

Eine Besonderheit besteht bei chinesischen Systemen. Bei Code Red II prüft der Wurm, ob das befallene System die chinesische Sprache verwendet. Ist dies so, dann werden anstatt 300 Prozessen 600 Prozesse gestartet, die anstatt 24 Stunden 48 Stunden nach verwundbaren Systemen suchen.

Code Red II konnte sich durch das lokale Scannen in Teilnetzen stark ausbreiten. Das lokale Scannen in Teilnetzen ermöglicht im Vergleich zum normalen zufälligen Scannen eine schnellere Ausbreitung von Würmern und wurde später auch für die Ausbreitung von Nimda verwendet.

2.5 Nimda

Nimda wurde das erste Mal am 18. September 2001 entdeckt. Der Wurmname Nimda leitet sich von der zur Ausbreitung genutzten Datei Admin.dll ab, da Nimda rückwärts gelesen Admin ergibt. Der Wurm befällt Systeme mit Windows 95, Windows 98, Windows Me, Windows NT 4 und Windows 2000. Der Lebenszyklus von Nimda kann in 4 Phasen eingeteilt werden (vgl. [Tocheva 01]):

Die 1. Phase ist die Dateiinfektion. Nimda sucht auf dem System nach allen .exe Dateien und assimiliert sie in den eigenen Code, d.h. die .exe Datei wird in den Programmcode des Wurmes als Ressource übernommen. Wird eine der Dateien ausgeführt, führt dies zur weiteren Verbreitung von Nimda.

In der 2. Phase benutzt Nimda Email zur Verbreitung. Der Wurm sucht sämtliche Adressen aus den lokalen Adressbüchern heraus und verschickt an jede Adresse eine Email mit dem Anhang README.EXE, in dem sich der Wurm befindet. Wenn der Empfänger den Anhang öffnet, wird sein System infiziert. Manche Systemen sind durch ihre Administratoren bzw. Benutzer so konfiguriert, dass Email Anhänge und somit die README.EXE automatisch ausgeführt werden.

In der 3. Phase durchsucht der Wurm das Internet nach Webservern. Findet er einen Webserver, versucht er, ihn zu infizieren. Die Webseiten werden so modifiziert, dass ein Anwender, der diese Webseiten anschaut, sein System automatisch infiziert.

Die 4. Phase ist die Verbreitung in lokalen Netzwerken. Dies geschieht über gemeinsam genutzte Laufwerke (shared drives). Nimda plaziert die RICHED20.DLL in jedes Verzeichnis, welches .DOC und .EML Dateien enthält. Wird versucht .DOC oder .EML Dateien aus diesen Verzeichnissen mit Word, WordPad oder Outlook zu öffnen, dann wird die RICHED20.DLL ausgeführt und eine weitere Verbreitung von Nimda ausgelöst.

Durch die verschiedenen Dateien, mit denen Nimda gestartet wird, und die Art des Systems ergibt sich ein unterschiedliches Verhalten bei der Ausbreitung und Infektion durch den Wurm, wobei der Lebenszyklus immer identisch ist. Die verschiedenen typischen Systemvarianten werden in den folgenden Abschnitten beschrieben.

2.5.1 Nimda auf einem Server

Wird Nimda auf einem Server gestartet, dann heißt die Wurmdatei ADMIN.DLL ([Tocheva 01]). Diese bewirkt, dass Nimda sich selbst als MMC.EXE in das Windowsverzeichnis kopiert und ausführt. Danach scannt und infiziert der Wurm alle erreichbaren lokalen Laufwerke und Netzlaufwerke. Insbesondere alle .EXE Dateien werden dadurch infiziert, dass Nimda sie zunächst als Ressource in den eigenen Code übernimmt. Werden diese .EXE Dateien ausgeführt, stellt Nimda die .EXE Datei mit infektiösem Code wieder her, führt sie aus und löscht sie anschließend. Kann die Datei nicht gelöscht werden, erzeugt Nimda die Datei WININIT.INI, die die infizierte Datei beim nächsten Start des Windowssystems löscht.

Des Weiteren sucht der Wurm auf der Festplatte nach HTML, HTM und ASP Dateien. Wenn er welche findet, erzeugt er die Datei README.EML und fügt an die gefundenen Dateien ein Stück JavaScript Code an. Dieses Programm öffnet die README.EML, wenn eine der modifizierten HTML Dateien von einem Webbrowser geladen wird. Dadurch wird das System, das die HTML Datei lädt, mit Nimda infiziert. Der Wurm läuft dann in einem minimierten Fenster auf dem System. Dieser Verbreitungsweg funktioniert nur, wenn der Internet Explorer 5.0 oder 5.01 benutzt wird.

Nimda legt in alle Ordner des Systems, auf die er Zugriff hat, .EML und .NWS Dateien. Außerdem wird die RICHED20.DLL in alle Ordner gelegt, in denen sich .DOC oder .EML Dateien befinden. Nimda versucht außerdem, die originale Windows RICHED20.DLL durch seine eigene Kopie zu ersetzen.

Damit erhöht sich die Wahrscheinlichkeit, dass eine mit Nimda infizierte Datei ausgeführt wird.

2.5.2 Nimda auf einer Workstation

Wenn der Wurm auf einer Workstation gestartet wird, dann heißt die Wurmdatei meist README.EXE. Alternativ befindet sich Nimda in einer Datei, deren Dateiname mehr als 5 Zeichen hat und mit .EXE endet. Nimda kopiert sich selbst in einen temporären Ordner mit einem zufälligen Namen in der Form „MEP*.TMP“ und führt sich aus. Wenn Nimda startet, lädt er sich selbst als eine DLL Bibliothek. Dafür sucht er nach einer bestimmten Ressource in der DLL und ermittelt deren Größe. Ist die Größe kleiner als 100, entpackt er sich selbst. Ansonsten erstellt er aus der Ressource eine neue Datei und führt sie aus. Die Größe der Ressource gibt Aufschluss darüber, ob Nimda bereits auf dem System läuft ([Tocheva 01]).

Danach generiert der Wurm eine Zufallszahl, führt mit ihr ein paar arithmetische Operationen durch und überprüft das Ergebnis. Ist das Ergebnis größer als der Zähler des Wurmes, beginnt Nimda alle README*.EXE Dateien im temporären Ordner zu suchen und zu löschen.

Danach erzeugt Nimda wiederum eine Kopie von sich mit einem zufälligen Namen und legt diese in einen temporären Ordner. Dann sucht der Wurm nach Explorer Prozessen, öffnet einen und weist ihn als Fernzugriffsprozess des Explorers aus. Auf manchen Systemen schafft er es nicht, sich als Explorer Prozess auszugeben. In diesem Fall wird Nimda für einige Zeit inaktiv. Wenn Nimda wieder in eine aktive Phase eintritt, ermittelt er als erstes das Betriebssystem, welches auf dem infizierten System läuft. Wenn ein Windows NT-basiertes System läuft, dann kopiert er sich als LOAD.EXE in das Windows Systemverzeichnis. Danach verändert Nimda die Datei SYSTEM.INI so, dass die Wurmkopie immer dann startet, wenn Windows startet. Dann kopiert er sich als RICHED20.DLL in den Systemordner und beginnt, gemeinsam genutzte Netzressourcen zu sammeln und nach Dateien

auf anderen Systemen zu scannen. Dort sucht er wie bei Servern nach .DOC und .EML Dateien und kopiert wieder die RICHED20.DLL in die zugehörigen Ordner. Dies erhöht die Wahrscheinlichkeit, dass anstatt der originalen RICHED20.DLL Datei die Nimdakopie gestartet wird, wodurch Nimda sich weiter verbreiten kann. Nimda versucht nicht, .EXE Dateien auf Workstations zu infizieren.

Bei der Verbreitung über Email sucht Nimda nach Emailadressen in den .HTM und .HTML Dateien. Nimda sammelt außerdem alle Senderadressen, die zu finden sind. Wenn Nimda die Adressbücher vollständig durchsucht hat, wird eine Adressliste fertig gestellt, mit welcher der Wurm über ein eigenes SMTP Programm infizierte Emailnachrichten versendet.

Für die Ausbreitung auf die Internet Information Server (IIS) nutzt Nimda die durch Code Red II installierte Hintertür. Zu diesem Zweck scannt Nimda per Zufall IP-Adressen. Wenn ein entsprechender IIS Server gefunden wird, veranlasst Nimda, dass der Wurmcode zum Beispiel als ADMIN.DLL auf das System von dem scannenden System herauf geladen wird. Hier geht Nimda dann wie schon beschrieben vor.

Von Nimda gibt es mittlerweile eine große Anzahl von Varianten.
Nimda verwendet für seine Ausbreitung insgesamt 19 Infektionskanäle und konnte sich damit im Internet weit verbreiten. Nimda benutzte nicht nur den Mechanismus des lokalen Scannens in Teilnetzen, er gebraucht auch die Informationen, welche sich auf den infizierten Systemen über andere Systeme befinden, um diese zu infizieren. Zum Beispiel nutzt Nimda diese Informationen für die Infektion über Email und über gemeinsam genutzte Laufwerke. Die Ausbreitung über mehrere Infektionskanäle hat sich bei Nimda als äußerst erfolgreich erwiesen. Immerhin konnte Nimda geschätzte 4,5 bis 5 Millionen Systeme infizieren.

2.6 Lioten

Der Wurm Lioten wurde am 16.12.2002 das erste Mal entdeckt. Der Name Lioten wird rückwärts gelesen zu Netoil und steht in Zusammenhang mit der Datei Iraq_oil.exe, die zur Verbreitung des Wurms genutzt wird. Der Wurm sollte die Internetkommunikation durch das Erzeugen einer hohen Netzlast mit einer Vielzahl von Paketen stören.

Der Wurm befällt Windows 2000- und Windows XP-Systeme. Er ist in Visual C programmiert ([Kytojoki 02]).

Er verbreitet sich über mit anderen Benutzern gemeinsam genutzte Laufwerke (shared drives) und den Port 445 des SMB (Server Message Block) Dienstes. Hat der Wurm ein System infiziert, erzeugt er 100 Prozesse, die zufällige IP-Adressen generieren. Dann versucht er sich über den Port 445 mit den Systemen hinter den IP-Adressen zu verbinden. Gelingt dies, versucht der Wurm die Passwörter aller Benutzer auf dem Zielsystem zu erraten ([Carrera 02]). Die Passwörter befinden sich in einer internen Liste des Wurms, die folgendermaßen aussieht (darunter das „leere Passwort“):

admin
root
111
123
1234
123456
654321
1

!@#$
asdf
asdfgh
!@#$%

!@#$%^
!@#$%^&
!@#$%^&*
server

Ist das Erraten des Passwortes erfolgreich, lokalisiert der Wurm den Systemordner und kopiert die Iraq_oil.exe dorthin. Die Grundeinstellung hierfür ist C:\Winnt\System32 (Windows NT/2000) oder C:\Windows\System32 (Windows XP) ([Kytojoki 02]). Daraufhin beginnt die weitere Verbreitung des Wurms.

Der Wurm nutzt laut [Kytojoki 02] die NetScheduleJobAdd Funktion in netapi32, um zu einem bestimmten Zeitpunkt ausgeführt zu werden. Dies führt dazu, dass Windows 95/98/ME-Systeme nicht von dem Wurm betroffen sind, weil sie diese Funktion nicht unterstützen.

Die Payload des Wurmes fällt gering aus. Die einzigen Effekte sind, dass die Performance des Systems leidet und dass es einen erhöhten Netzverkehr gibt.

Laut den Antivirenherstellern ist der Wurm Lioten nicht gefährlich und lässt sich leicht vom System wieder entfernen. Zum Beispiel vergibt McAfee sowohl für Heimanwender als auch für Unternehmen eine „low-profiled“ Risikoeinschätzung ab und TrendMicro spricht von „low overall risk rating“. Lioten konnte sich allerdings im Internet weit verbreiten.

Lioten ist als Iraq_oil, Datrix, W32.Lioten, W32/Lioten, I-Worm.Lioten, W32/Lioten.worm, Win32.Lioten, WORM_LIOTEN.A, W32/Lioten-A, Worm.Win32.Lioten, W32.HLLW.Lioten und W32/Lioten.A@mm bekannt.

Wenn der Autor diesen Wurm als Superwurm konzipiert hat, dann hat er versagt, denn Lioten fällt wohl kaum in die Kategorie Superwurm. Er breitet

sich nicht schnell und weit genug aus. Lioten hat keine payload, sondern versucht durch Überlastung von Netzen den Datenaustausch zu stören.
Die Idee beim Warhol Wurm und beim Flash Wurm war es, im vornherein nach verwundbaren Systemen zu suchen und diese zu einer Hitliste zusammenzustellen. Die Liste wird bei der Initialisierung abgearbeitet, damit sich der Wurm möglichst schnell ausbreiten kann. Bei Lioten wurde weder eine Hitliste oder ein Teil davon gefunden, noch ein andersartiger neuer Ausbreitungsmechanismus benutzt.
Die Ausbreitungsgeschwindigkeit ist verhältnismäßig gering, weil Lioten darauf angewiesen ist, dass der Port 445 auf den Zielsystemen offen ist und diese keine geeigneten Schutzmaßnahmen wie etwa eine Firewall haben. Wenn dies einmal gelingt, muss noch das Erraten des Passwortes erfolgreich sein, um das System zu infizieren. Dazu kommt, dass nur Windows 2000/NT/XP-Systeme erfolgreich infiziert werden können. Der Wurm braucht zu viel Zeit, ein verwundbares System zu finden, um wirklich die Internetkommunikation zu stören.
Dass der Wurm shared drives mit anderen Systemen ausnutzt, hilft nur wenig bei der schnelleren Verbreitung. Es müssen weiterhin shared drives zu anderen Systemen existieren und diese müssen auch noch ein infizierbares Betriebssystem besitzen.
Liotens Mechanismen zur Ausbreitung sind also zu langsam, um wirklich Schaden in der Klasse eines Superwurms anzurichten. Da er auch keine Payload besitzt, kann jeglicher Schaden nur durch das Lahmlegen von Netzen durch zu viel Netzverkehr angerichtet werden, was nicht gelingt. Die 100 von dem Wurm erzeugten Prozesse sind ein guter Hinweis, dass genau dies beabsichtigt war, sind jedoch in Verbindung mit dem Ausbreitungs-mechanismus zu wenig.
Lioten ist ein anfänglicher Versuch, einen Superwurm zu schreiben.

2.7 SQL Slammer/ Sapphire

Der Sapphire Wurm ist laut der Analyse von [Moore 03] der schnellste Computerwurm in der Geschichte.

Die Ausbreitung von Sapphire erfolgt über zufälliges Scannen (random scanning) von IP-Adressen. Er verschickt eine 376 Byte große Kopie von sich an den UDP Port 1434 an alle erzeugten Adressen. Dabei nutzt Sapphire die Buffer Overflow Verletzlichkeit in Microsoft SQL Servern aus, welche seit 1996 bekannt ist und für die es seit Juli 2002 einen Patch gibt. Hat ein Microsoft SQL Server diesen Patch nicht, kann er durch den Wurm infiziert werden.
Der Wurm kopiert sich nur in den Hauptspeicher. Dies bedeutet, dass er nicht auf der Festplatte zu finden ist und dass ein Neustart des Systems den Wurm vernichtet.

Sapphire nutzt einen Pseudo Random Number Generation (PRNG) Algorithmus für die Erzeugung von zufälligen IP-Adressen. Dieser Algorithmus hat die Form: x´ = (x * a + b) mod m. Hierbei ist *x´* die neue, zu generierende IP-Adresse, *x* die letzte erzeugte IP-Adresse, *a* und *b* sind Konstanten und *m* ist der Raum, in dem sich das Ergebnis bewegen soll. Der Autor des Wurmes nutzte folgende von Microsoft veröffentlichte Parametrisierung des Algorithmus:

x´ = (x * 214013 + 2531011) mod 2^32.

Der Autor des Wurms machte aber bei der Implementation des Algorithmus zwei Fehler. Erstens benutzte er anstatt der SUB-Instruktion die ADD-Instruktion für das Inkrement eines Wertes und zweitens die OR-Instruktion anstatt der XOR-Instruktion für das Leeren eines Schlüsselregisters. Dies führt dazu, dass das 25. und 26. Bit der erzeugten IP-Adresse immer 0 ist, d.h. der gescannte Adressraum wesentlich kleiner ist als der zur Verfügung stehende IP-Adressraum (vgl. [Moore 03]).

Das zufällige Scannen führt zu einer exponentiellen Ausbreitung von Sapphire. Ein solcher Ausbreitungsverlauf läßt sich im zufälligen und konstanten Ausbreitungsmodell („the random constant spread (RCS) model“) darstellen. Der typische Verlauf der exponentiellen Ausbreitung ist bei dem Wurm Code Red, gut zu erkennen. Abbildung 8 zeigt das Ergebnis einer Untersuchung des Chemical Abstract Service vom 01. August 2001. Hierbei wurde der Verlauf der Neuinfektionen durch Code Red an diesem Tag im Vergleich zum Verlauf der Kurve bei einem Wurm mit dem Verhalten des RCS Modells dargestellt. Die Kurven sind fast deckungsgleich.

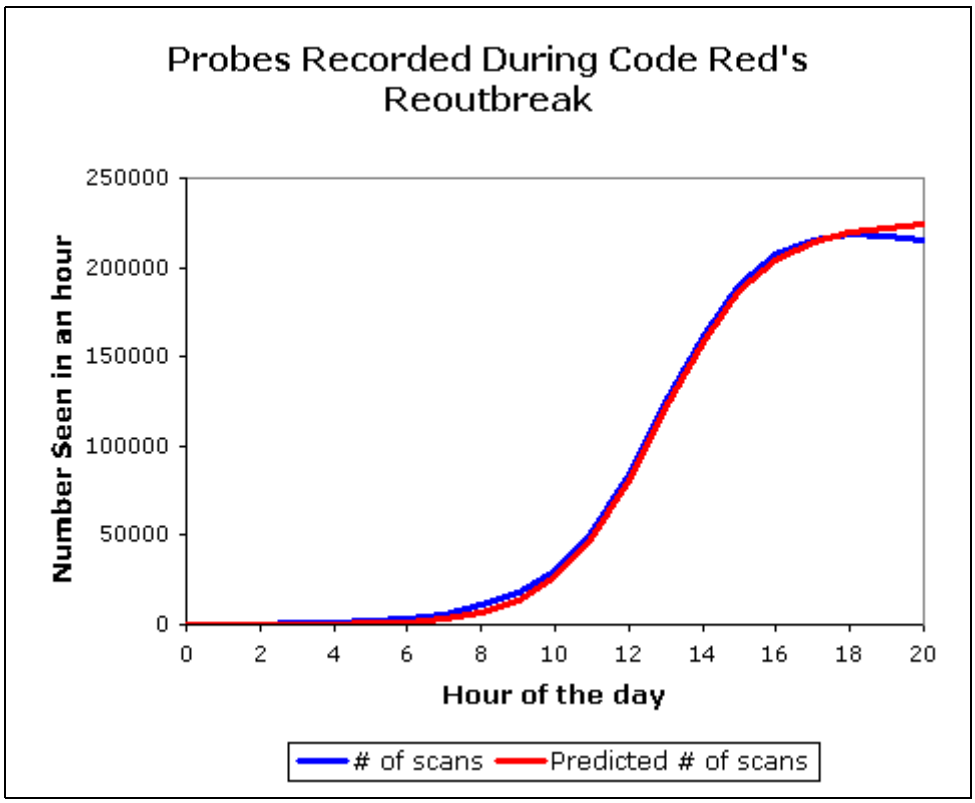

Abbildung 8: Vergleich der Neuinfektionen durch Code Red am 01. August 2001 mit dem Verhalten eines Wurms nach dem RCS Modell (aus [Moore 03]).

Auch bei Sapphire ist am Anfang eine exponentielle Ausbreitung gemessen worden. Später hat Sapphire die Netzwerke mit seinen Scans und der Ausnutzung der Datenrate so gesättigt, dass er vom RCS Modell stark abweicht. Der Wurm verschickt so viele Kopien von sich beim Scannen der Netze über die Leitungen, dass deren Kapazität voll ausgeschöpft wird und einige Netzwerke unter der hohen Last zusammenbrechen. Deshalb wird die

Sättigung der Kurve bereits vorher erreicht. Abbildung 9 zeigt die gemessenen Daten von Sapphire im Vergleich zum normalen Verhalten eines Wurmes des RCS Modells.

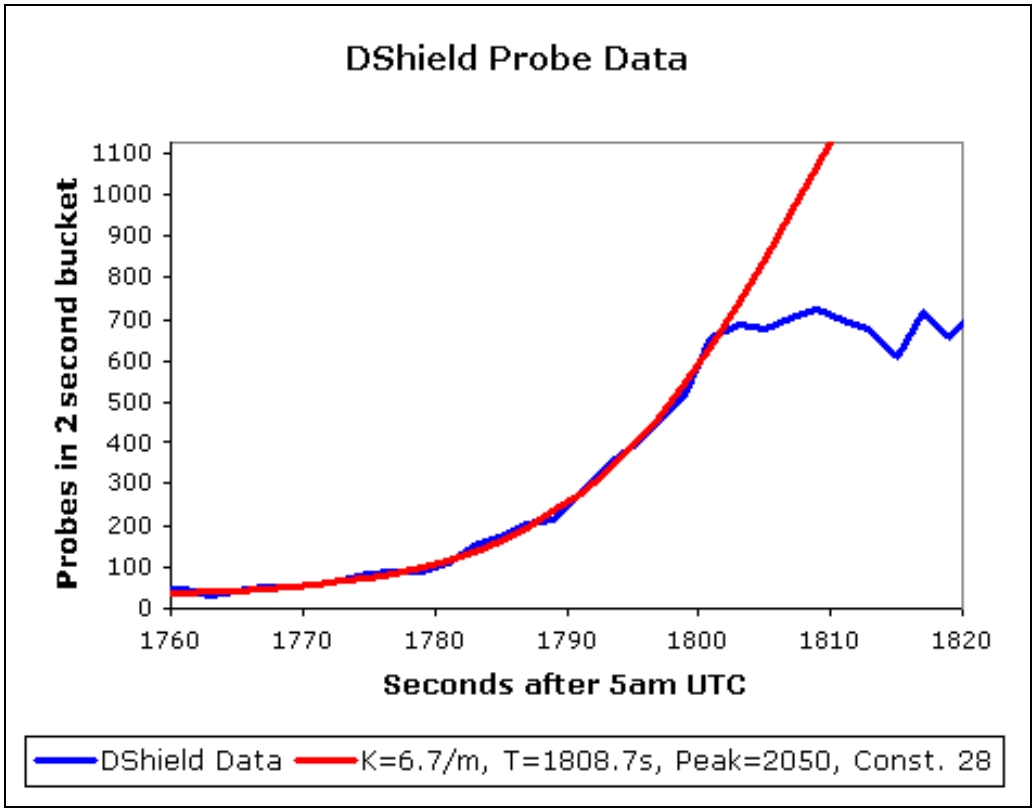

Abbildung 9: Vergleich des Verhaltens von Sapphire mit dem Verhalten eines Wurms nach dem RCS Modell (aus [Moore 03]).

Am Anfang verdoppelte sich die Anzahl der infizierten Systeme alle 8,5 (±1) Sekunden. Nach etwa 3 Minuten entfaltete Sapphire seine volle Scannrate von über 55 Millionen Scans pro Sekunde durch alle Instanzen zusammen.

Sapphire nutzt mit dem UDP Port 1434 keinen kritischen Port für die Internetkommunikation. Deshalb ist es kein Problem, den gesamten Netzverkehr über den Port zu filtern. Wenn der Port geschlossen ist, braucht das System nur neu gestartet zu werden, um Sapphire zu entfernen. Danach ist es sinnvoll, den Patch für den SQL Server einzuspielen, damit bei einer eventuellen Freigabe von Port 1434 keine neue Infektion mit Sapphire stattfinden kann.

Auch bei diesem Wurm könnte es sich um einen Versuch handeln, einen Superwurm zu erschaffen. Dabei wurde an Code Red angeknüpft.
Sapphire hält sich nicht damit auf, Systeme zu übernehmen und dort Schäden anzurichten, wie zum Beispiel Daten zu stehlen oder zu löschen. Das Ziel des Wurmes ist, möglichst viele Systeme möglichst schnell zu infizieren und dann durch den erzeugten Netzverkehr ganze Netze lahm zu legen.
Sapphire hat es geschafft zahlreiche Unternehmensnetze zu befallen und sehr große Schäden anzurichten. Insbesondere Intranets waren betroffen, d.h. die Netze brachen unter der Last des Netzverkehrs, der von Sapphire produziert wurde, zusammen. Obwohl Sapphire eine Schwäche ausnutzt, die seit langem bekannt war, ist er sehr erfolgreich gewesen und konnte eine Vielzahl von Systemen infizieren. Die gesamte Internetkommunikation konnte er aber nicht lahm legen.
Von Vorteil für den Wurm sind seine geringe Größe sowie seine Scanstrategie. Zufälliges Scannen ist zwar nicht die optimale Strategie, jedoch eine recht gute und vor allem einfache Methode. Trotzdem hat der Wurmautor bei der Implementation des Zufallsalgorithmus Fehler gemacht. Ein weiterer Nachteil ist, dass der Wurm so viel Netzverkehr erzeugt, dass er sich bei seiner Ausbreitung selbst behindert.
Eine Auffälligkeit ist, dass Sapphire nur eine Schwäche zur Infizierung nutzt. Seit Nimda ist bekannt, wie verheerend ein Wurm sein kann, wenn er mehrere Schwächen ausnutzt. Hätte Sapphire noch mehr Schwachstellen zur Infizierung genutzt, hätte er noch viel mehr Schaden anrichten können, wobei jedoch seine eigene Größe angestiegen wäre.
Insgesamt ist Sapphire im Vergleich zu Lioten eine enorme Steigerung beim Versuch, einen Superwurm zu programmieren.

2.8 Zusammenfassung

Der Warhol und der Flash Wurm sind theoretische Ansätze für Superwürmer, welche vor allem neue Ideen für die schnelle und effektive Ausbreitung von Würmern liefern. Bereits existierende Würmer wie Code Red II, Nimda oder Sapphire konnten durch die Verwendung bestimmter Ausbreitungsmechanismen eine Vielzahl von Systemen infizieren. Code Red II kann sich durch lokales Scannen in vielen Netzwerken stark ausbreiten. Nimda gelingt dies durch die Ausnutzung mehrerer Schwächen in Kombination mit verschiedenen Ausbreitungsmechanismen. Sapphire ist der wohl schnellste Computerwurm in der Geschichte, weil er sehr aggressiv nach neuen Zielen scannt und dabei selbst relativ klein ist. Dahingegen sind Code Red I und Lioten eher Versuche besonders gefährliche Würmer zu implementieren.

3 Superwürmer

In diesem Kapitel wird eine Definition von Superwürmern aufgestellt. Danach werden die sich daraus ergebenen Charakteristika diskutiert. Das dominierende Charakteristikum von Superwürmern ist die hohe Infektionsrate, welche im dritten Abschnitt dieses Kapitels näher betrachtet wird. Zum Schluß dieses Kapitels die Frage der Realisierbarkeit von Superwürmern diskutuiert.

3.1 Was ist ein Superwurm?

In den Abschnitten 2.1 und 2.2 wurden bereits der Warhol und der Flash Wurm vorgestellt. Der Warhol Wurm soll in 15 Minuten alle verwundbaren Systeme im Internet infizieren. Der Flash Wurm soll dies in 30 Sekunden bewerkstelligen. Beide Würmer existieren aber lediglich als theoretische Konstrukte, die bisher noch nicht realisiert wurden. Aus diesem Grund ist es schwierig, eine vernünftige Definition für Superwürmer zu erstellen. Deshalb ist die folgende Definition von Superwürmern rein quantitativ gehalten. Die benutzten Werte für die Definition stammen aus der Besprechung mit Klaus Brunnstein vom 03.07.2003.

Definition*:* Ein *Superwurm* ist ein Internetwurm, der mindestens zehn Prozent der Systeme im Internet innerhalb von 24 Stunden infiziert.

Nach dieser Definition zählen Code Red II, Nimda und Sapphire nicht zu den Superwürmern. Die Anforderungen, um in die Klasse der Superwürmer eingeordnet zu werden, sind hoch und können durch die benutzten Ausbreitungsmechanismen so nicht erreicht werden. Zum Einen haben die genannten Würmer es nicht geschafft zehn Prozent der Systeme im Internet (zur Zeit also etwa 40 Millionen von geschätzten 400 Millionen) zu infizieren, und Code Red II und Nimda haben zum Anderen mehrere Tage gebraucht, um sich merklich auszubreiten. Ein Superwurm müsste eine sehr hohe

Infektionsrate haben, um viele Systeme in kurzer Zeit zu infizieren, um die Kriterien zu erfüllen.

Die *Infektionsrate* eines Wurmes ist die Anzahl der Systeme, die in einem bestimmten Zeitraum durch den Wurm infiziert werden. Die Infektionsrate wird in Anzahl der Systeme pro Zeiteinheit angegeben.

Die Infektionsrate eines Wurms wird durch seine Ausbreitungsmechanismen bestimmt. Code Red II, Nimda und Sapphire waren, als sie sich ausbreiteten, so gefährlich und schädlich, weil sie effektive Ausbreitungsmechanismen benutzt haben (vgl. Kapitel 2). Für die Kriterien eines Superwurms reichen sie aber nicht aus. In den Beschreibungen über den Warhol und den Flash Wurm wurden neue Ausbreitungsmechanismen angedacht, welche für eine hohe Infektionsrate sorgen sollen. Allerdings werfen auch sie Probleme auf und es ist fraglich, ob sie überhaupt realistisch bzw. realisierbar sind. Die Infektionsrate hängt aber auch von anderen Dingen, wie zum Beispiel von der Übertragungszeit einer Wurmkopie über ein Netz oder der Zeitdauer, die ein Wurm für die Infektion eines Systems braucht ab. Die Infektion eines Systems wird also durch Übertragungswege und die Beschaffenheit der Systeme selbst mehr oder minder verzögert.

Ein weiteres Problem ist, dass ein Superwurm erst einmal zehn Prozent der Systeme im Internet infizieren muss, d.h. mindestens zehn Prozent der Systeme müssen verwundbar sein und der Superwurm muss diese Schwächen ausnutzen können. Benutzt der Superwurm für die Ausbreitung über eine Schwäche einen bestimmten Mechanismus ergibt dies einen Infektionskanal. Ein Superwurm könnte zusätzlich versuchen andere Infektionskanäle, wie zum Beispiel HTML oder IRC zu benutzen. Für eine weite Ausbreitung wären mehrere Infektionskanäle förderlich. Am Beispiel von Nimda, welcher sich unter anderen über Email, Wegserver und gemeinsam genutzte Laufwerke ausbreitet (vgl. Kapitel 2), wird klar, wie gut diese Strategie funktionieren kann.

Aus all diesen Erkenntnissen ergeben sich die nun folgenden Charakteristika von Superwürmern.

3.2 Charakteristika eines Superwurms

In diesem Abschnitt werden die besonderen Eigenschaften und Eigenarten von Superwürmern beschrieben. Dazu zählen die Lebens- und Ausbreitungsphasen, die Ausnutzung von Schwächen und anderen Infektionskanälen sowie die Auswirkungen von Superwürmern.

3.2.1 Die Phasen eines Superwurms

Ein Superwurm kann zuerst eine Vorbereitungsphase, danach eine Aktionsphase und zuletzt eine Nachbereitungsphase durchschreiten.

Die Vorbereitungsphase
Während der Vorbereitungsphase werden für den Superwurm wichtige Informationen gesammelt, wie zum Beispiel über Schwächen von Systemen. In dieser Phase entsteht ein Grobentwurf für den Superwurm. In dem Grobentwurf wird festgelegt, welche Schwächen und welche Ausbreitungsmechanismen verwendet werden sollen. Soll eine Hitliste die initiale Ausbreitung steuern, wird sie zu diesem Zeitpunkt erstellt.

Die Aktionsphase
Die Aktionsphase des Superwurms ist die Hauptphase, weil hier der Superwurm seine volle Wirkung entfaltet. In der Aktionsphase wird der Superwurm implementiert, gestartet, und er breitet sich aus. Für die Implementation können die gewonnenen Informationen aus der Vorbereitungsphase genutzt werden. Nach der Initialisierung des Superwurms beginnt die Ausbreitung über die Infektionskanäle. Die Ausbreitung selbst

kann wiederum in verschiedene Phasen geteilt werden und soll das Thema des nächsten Abschnittes 3.2.2 sein.

Die Nachbereitungsphase
In einer Nachbereitungsphase kann der Superwurmautor in Verbindung mit dem Superwurm bleiben, zum Beispiel durch das Versenden von Nachrichten von jedem neu infizierten System. Diese enthalten Informationen über den Status des Superwurms und des infizierten Systems. Dadurch kann der Wurmautor einerseits Erkenntnisse gewinnen, um diesen oder einen weiteren Superwurm noch effizienter und effektiver zu machen. Andererseits kann der Superwurmautor seine Verbindung mit dem Superwurm nutzen, um einzelne Systeme gezielt auszuspionieren. Das Problem der Nachbereitungsphase ist, dass gerade einige Millionen Systeme versuchen, einige Millionen andere Systeme zu infizieren. Durch den hohen Netzverkehr könnte es schwierig werden, den Kontakt mit dem Superwurm aufrecht zu erhalten. Gelöst werden könnte dieses Problem dadurch, dass der Superwurm unter bestimmten Bedingungen die Ausbreitung beendet, um danach Kontakt mit dem Superwurmautor aufnimmt. Dieser könnte den Superwurm nach dem Datenaustausch wieder starten und der Superwurm seine Ausbreitung fortsetzen. Der Neustart des Superwurms könnte auch unter Verwendung eines Timers erfolgen.

Die Vorbereitungsphase kann stark verkürzt sein, wenn bekannte Schwachstellen verwendet werden oder keine andere aufwendige Vorbereitung wie bei einer Hitliste betrieben wird. Die Nachbereitungsphase kann auch ganz wegfallen, wenn keine Informationen gesammelt werden sollen.

3.2.2 Die Ausbreitungsphasen

Um zu erkennen, was einen Wurm zu einem Superwurm macht, muss vor allem die Ausbreitung eines Superwurms betrachtet werden. Die Ausbreitung

eines Superwurms erfolgt durch die höhere Infektionsrate in einem wesentlich kleineren Zeitraum als bei herkömmlichen Würmern wie Nimda oder Code Red II. Die Ausbreitung kann in mehrere Phasen eingeteilt werden (vgl. Abbildung 10).

Die erste Phase ist die Vorinfektion $t_{vor\text{-}inf}$. Die Vorinfektionsphase ist nach Brunnstein der Zeitraum bei der Ausbreitung eines Wurms von der Initialisierung des Wurms bis ein Prozent der Systeme infiziert sind. Die Vorinfektion wird auch als Inkubationszeit bezeichnet.

Die wichtigste Phase ist der Hauptinfektionszeitraum t_{hinf}. Die Hauptinfektionsphase schließt direkt an die Vorinfektion an. Das heißt, es ist nach Brunnstein der Infektionszeitraum, wo ein Prozent bis zehn Prozent der Systeme infiziert werden. Manche Würmer verlassen niemals die Phase der Vorinfektion, weil sie nicht genügend Systeme infizieren. In der Hauptphase findet in der Regel die stärkste Ausbreitung mit der höchsten Infektionsrate statt.

Die dritte und letzte Phase ist die Nachinfektion $t_{nach\text{-}inf}$. Sie schließt direkt an die Hauptinfektion an. In dieser Phase können noch weitere Systeme infiziert werden. Im schlimmsten Fall werden 100 Prozent der Systeme infiziert. In dieser Phase kann es zu einer Sättigung der verwundbaren Systeme mit dem Wurm kommen, d.h. die Anzahl der infizierten Systeme durch einen Wurm bleibt etwa auf dem gleichen Niveau (vgl. Abbildung 10). Ebenfalls in dieser Phase können Gegenmaßnahmen wie Reinigung und Patchen der Systeme zum Tragen kommen.

Durch die Benutzung eines Timers kann es zu mehrfachen Ausbreitungen eines Superwurms kommen. Dabei kommt es zu gegenseitigen Beeinflussungen der Phasen der aufeinander folgenden Ausbreitungen. In diesem Fall können die einzelnen Phasen nicht mehr unterschieden werden.

Die Ausbreitungsphasen eines Superwurms sind in Abbildung 10 dargestellt.

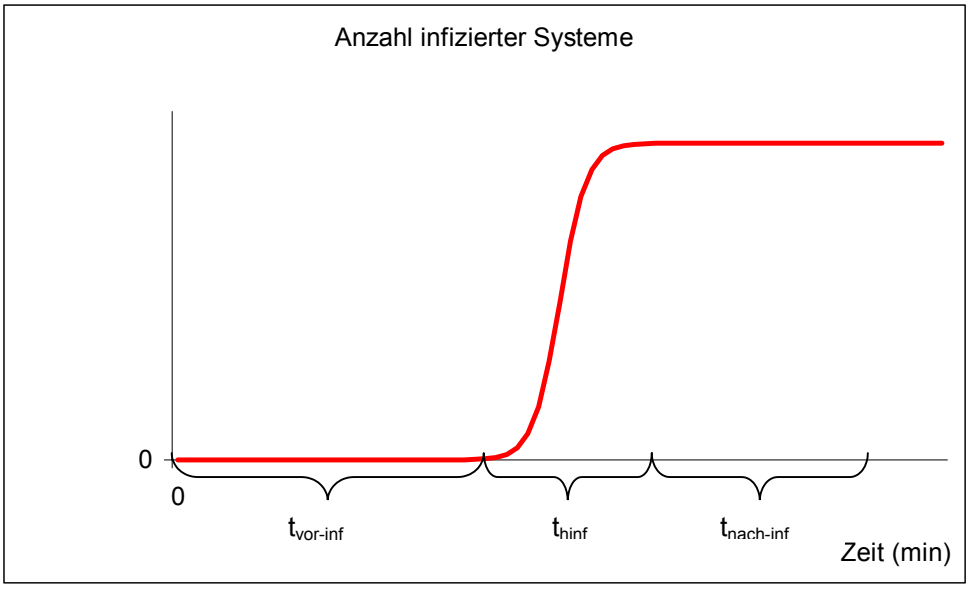

Abbildung 10: Die Phasen der Ausbreitung eines Superwurms

Wird ein herkömmlicher Wurm mit einem Superwurm verglichen, lassen sich folgende Unterschiede feststellen:
Die Vorinfektionsphase bei Superwürmern ist wesentlich kleiner als bei herkömmlichen Würmern. Sie beträgt nach Brunnstein bei Superwürmern weniger als einen halben Tag, während herkömmliche Würmer eine Inkubationszeit von mehr als einem Tag haben.
Weil die Infektionsrate bei Superwürmern wesentlich größer ist, ist die Hauptinfektionsphase bei ihnen entscheidend verkürzt.

3.2.3 Die Ausnutzung von Schwachstellen

Ein Superwurm kann eine Schwäche der Spezifikation, Implementation oder Konfiguration zur Verbreitung über das Internet nutzen. Schwächen der Spezifikation sind Konzept-, Denk- und Architekturfehler. Programmierfehler erzeugen Schwächen der Implementation. Wenn Fehler bei der Installation, der Benutzung oder der Wartung passieren, sind dies Schwächen der Konfiguration.

Für einen Superwurm sind alle Schwachstellen geeignet, die auf weit verbreiteten Systemen vorkommen und deren Ausnutzung auf weit verbreiteten Techniken basieren. In diesem Zusammenhang ist System der Oberbegriff für Betriebssysteme, Software, Hardware, Treiber und auch Firmware. Die Schwachstelle für einen Superwurm besitzen entweder viele Systeme, weil sie zum Beispiel durch einen Patch nicht behoben wurde oder weil die Schwachstelle noch nicht erkannt wurde. Im letzten Fall ist der Wurm besonders effektiv, weil nicht nur die Funktionsweise des Wurms ermittelt, sondern auch die Schwachstelle gefunden werden muss. Dafür muss der Wurmautor vorher einen erheblichen Aufwand leisten, um die Schwachstelle zu finden. Heutzutage sind die meisten Schwachstellen, die von Viren- und Wurmautoren genutzt werden, schon lange bekannt. Die Schwachstellen werden meistens von den zuständigen Administratoren nicht behoben, obwohl die Hersteller auf die Schwachstelle und Möglichkeiten zu ihrer Behebung hinweisen. Deshalb richtet Malware nach wie vor sehr viel Schaden an.
Ein Superwurm soll sich schnell ausbreiten und dadurch möglichst viele Systeme infizieren. Zu diesem Zweck ist es sinnvoll, wenn der Superwurm mehrere Schwachstellen ausnutzt. Durch mehrere unterschiedliche Schwachstellen können mehr Systeme infiziert werden, weil die Wahrscheinlichkeit höher ist, dass ein System zumindest eine dieser Schwachstellen besitzt. Insgesamt kann die Anzahl der infizierten Systeme durch mehrere Schwachstellen vergrößert werden. Andererseits ist es für die Effizienz auch von Vorteil, wenn der Superwurm möglichst klein ist. Dies war beim Wurm Sapphire der Fall. Durch seine geringe Größe von 376 Byte konnte er sich schnell ausbreiten, allerdings nur auf Microsoft SQL Servern.

Es ist wichtig, welche Auswirkungen die genutzte Schwachstelle auf die Ausbreitung des Wurmes hat. Die Infektion eines Systems über eine Schwachstelle muss schnell erfolgen und sollte nicht behindert oder verzögert werden können. Ein Beispiel für Verzögerung ist eine Schwachstelle, die nur über das Verschicken von Email ausgenutzt werden kann. Bei diesem Infektionsweg ist die Email der Träger des Wurmes. Dies kann zwar erfolgreich zur Infektion des Systems führen, aber Emails werden nicht sofort

vom Sender zum Adressaten geschickt, sondern liegen eine gewisse Zeit auf Emailservern, bis sie abgeholt werden. Emails sind relativ zeitunkritisch, weil es für die Benutzer in der Regel irrelevant ist, ob eine Email 5 Minuten früher oder später ankommt. Deshalb würden die Benutzer die Ausbreitung am meisten verzögern, wenn sie sich die Email nicht sofort nach Eingang auf dem Server auf ihr System holen und anschauen. Es gibt auch die Möglichkeit durch die Ausnutzung von entsprechenden Schwächen, die Emailserver selbst zu infizieren. Dies hätte andere Auswirkungen auf die Infektionsrate, da Mailserver normalerweise keine Mails lesen und aktive Inhalte wie ActiveX, JavaScript usw. ausführen, sondern lediglich an die Zieladresse weiterleiten, bis der Empfänger durch Eingaben eine Mail öffnet. Die Zeitverzögerung bei der Infektion durch Emails allein ist für die Verbreitung eines Superwurms zu groß, und deshalb können nicht Emails alleine für die schnelle Ausbreitung eines Superwurms genutzt werden.

Für die Erfüllung der Kriterien eines Superwurms ist also eine einzelne Schwachstelle zu wenig. Bei der Ausbreitung über mehrere Schwachstellen muss mindestens durch eine Schwachstelle eine schnelle Ausbreitung möglich sein. Die Ausnutzung weiterer Schwachstellen sorgt dann für eine noch weitere Verbreitung des Superwurms im Internet. Besitzt ein System eine Schwachstelle nicht oder ist davor geschützt, so kann es aber über die Ausnutzung einer anderen Schwäche infiziert werden. Die Ausbreitung über eine Vielzahl von Schwachstellen kann sogar dazu führen, dass Gateways, die ein dahinter liegendes Netz vor einer Schwachstelle schützen, mit einer anderen Schwachstelle überlistet werden können.

Eine Schwachstelle und ein Ausbreitungsmechanismus bilden zusammen einen Infektionskanal. Dabei können mehrere Schwachstellen den gleichen Ausbreitungsmechanismus benutzen. Der Ausbreitungsmechanismus kann sogar dafür sorgen, dass ein System auf jede dieser Schwächen überprüft wird. Sobald das System eine dieser Schwächen aufweist, wird es darüber infiziert. Durch die Nutzung von mehreren Ausbreitungsmechanismen in Verbindung mit verschiedenen Schwachstellen wird die Wahrscheinlichkeit

erhöht, dass wenigstens über einen Infektionskanal ein System infiziert werden kann. Während der eine Infektionskanal an einer Schutzmaßnahme scheitert, kann ein anderer Infektionskanal diese umgehen und somit ein System oder ein Netz infizieren. Wenig sinnvoll hingegen ist die Verwendung von mehreren Ausbreitungsmechanismen für die Ausnutzung einer Schwäche, weil der Mechanismus eigentlich nur Einfluss aus die Infektionsrate hat.
Eine Vielzahl von Infektionskanälen kann also die Infektionsrate erhöhen.

3.2.4 Die Auswirkungen

Die genauen Auswirkungen eines Superwurms sind nur schwer abzuschätzen, weil sich noch kein konkreter Superwurm im Internet ausgebreitet hat. Trotzdem haben bereits einige Würmer verheerenden Schaden angerichtet, so dass zumindest Anhaltspunkte bestehen, welche Wirkung ein Superwurm haben könnte.

Zunächst einmal infiziert ein Superwurm gemäß Definition mindestens zehn Prozent der Systeme im Internet, d.h. eine große Anzahl von Unternehmen, Behörden, Privatnutzern und vielen Anderen werden von dem Superwurm betroffen sein. Was für einen Schaden der Superwurm anrichtet, hängt wiederum von der Implementation des Superwurms selbst ab. Der Superwurm könnte Daten stehlen oder löschen. Er könnte zum Beispiel Passwörter ausspionieren oder alle Bilddateien auf einem System löschen. Darüber hinaus besteht auch die Möglichkeit, dass er Hintertüren auf dem System hinterlässt, um dieses später zu kontrollieren. Der Superwurm könnte auch sogenannte Zombies auf den infizierten Systemen installieren. Mit diesen Zombies werden dann so genannte DDOS-Angriffe ausgeführt. Neben der bereits beschriebenen Malware könnte der Superwurm auch alle anderen Arten von Malware wie Viren und Trojanische Pferde auf den Systemen verteilen.
Es wurde auch eine andere Schadfunktion für Würmern entwickelt, welche bei Superwürmern besonders verheerend wäre. Allein durch die Ausbreitung

eines Wurms kann so viel Netzverkehr entstehen, dass der eigentliche Zweck von Netzen, der Austausch von Daten bzw. Datenpakete, kaum mehr möglich ist. Dies war beim Wurm Sapphire alias SQL Slammer besonders gut zu sehen. Durch Sapphire konnten über einige Unternehmensnetze stundenlang keine Daten ausgetauscht werden. Durch die Überlastung einiger Netze hat Sapphire sich bei seiner Ausbreitung sogar selbst behindert (vgl. Abschnitt 2.7). Bei der Ausbreitung eines Superwurms könnte dies als Nebenwirkung auftreten oder es könnte auch als Schadfunktion beabsichtigt sein. Um diese mögliche Auswirkung eines Superwurms zu verdeutlichen, folgt nun eine kurze Erläuterung der Netzüberlastung:

- Die Datenrate r_s gibt das Datenvolumen V an, welches über eine Stelle s einer Datenleitung transferiert wird, während eines beobachteten Zeitraums t_1 bis t_2.

$$r_s = \frac{V}{t_2 - t_1} \text{ an der Stelle } s$$

- Die Auslastung A_s beschreibt, in welchem Verhältnis die konkrete Datenrate r_s zur leitungsbedingten maximalen Datenrate r_{max} an der Stelle s steht. Dabei gilt, dass r_s immer kleiner oder gleich r_{max} ist.

$$A_s = \frac{r_s}{r_{max}}$$

- Die Netzauslastung A_{Netz} ist die Auslastung an allen X Stellen des betrachteten Netzes, wobei ein Engpass an der Stelle mit der höchsten Auslastung existiert.

$$A_{Netz} = \max\left(A_{s_1}, \ldots A_{s_X}\right)$$

Ist die Netzauslastung so hoch, dass die eigentliche Aufgabe eines Netzes, nutzbare Daten zu übertragen, nicht mehr erfüllt werden kann, dann wird von einer Überlastung des Netzes gesprochen.
Ist ein Netz zum Beispiel bei normalen Gebrauch mit 70 Prozent ausgelastet, kann ein Superwurm die Netzauslastung auf 100 Prozent erhöhen. Dadurch

wird der Anteil der Datenpakete für den normalen Gebrauch gesenkt, so dass die eigentlichen Dienste des Netzes nicht mehr vollständig oder verzögert zur Verfügung stehen.

All die bis jetzt beschriebenen Auswirkungen haben zusätzlich auch noch weitere negative Folgewirkungen. Ein Superwurm kann den Ausfall von vielen Millionen Systemen und einer Vielzahl von Netzen verursachen. Dadurch können diese ihre Aufgaben nicht mehr erfüllen, was wiederum zu hohen finanziellen Schäden und Imageschäden führt.

3.3 Die Bestimmung der Infektionsrate

Die Definition von Superwürmern verlangt, dass eine Infektionsrate von zehn Prozent in 24 Stunden erreicht wird. In diesem Abschnitt wird ein Modell beschrieben, welches erlaubt, die Infektionsraten von Würmern abzuschätzen und miteinander zu vergleichen.

3.3.1 Plausibilitätsmodell

Um die Infektionsrate und die Anzahl der infizierten Systeme durch einen Superwurm bestimmen zu können, wird das zufällige, konstante Ausbreitungsmodell (*Random Constant Spread Model* oder auch *RCS Model*) aus [Staniford 02] verwendet.

Im RCS Modell werden folgende Parameter benutzt:

- *N* bezeichnet die Anzahl aller Systeme, die im Internet durch den Wurm infiziert werden können. Es handelt sich also um einen Parameter, der angibt, wie viele Systeme eine der Schwächen besitzen, die der untersuchte Wurm ausnutzt. In der Literatur wird davon ausgegangen, dass im betrachteten Zeitraum die Anzahl der Systeme relativ konstant

ist. Vor allem bei Superwürmern, die sich in sehr kurzer Zeit ausbreiten, unterliegt *N* durch Patchen der Systeme bzw. deren Ein- und Ausschaltung nur minimalen Schwankungen. Alle beispielsweise durch eine entsprechend konfigurierte Firewall geschützten Systeme werden nicht beachtet, da ihre Schwachstellen durch die Schutzmaßnahme nicht ausgenutzt werden können.

- *K* bezeichnet die initiale Infektionsrate, die in Anzahl von Systemen pro Zeiteinheit gemessen wird. *K* gibt die Anzahl der verwundbaren Systeme in einem bestimmten Zeitraum wieder, welche durch ein infiziertes System am Anfang der Ausbreitung des Wurms infiziert werden können. Da die Infektionsrate abhängig von der jeweiligen Hardware ist, gibt es üblicherweise keinen einheitlichen Wert für *K*, der für alle Systeme gilt. Viel mehr werden hier für das verwendete *K* genäherte bzw. geschätzte Werte benutzt. Außerdem wird für *K* angenommen, dass ein infiziertes System nicht noch einmal infiziert wird, bzw. dass Mehrfachinfektionen von Systemen die Infektionsrate nicht beeinflussen.

Des Weiteren werden die folgenden Variablen für das RCS Modell gebraucht:

- Der Prozentsatz der bereits infizierten Systeme wird mit *a* gekennzeichnet, wobei *a* später als Funktionswert verwendet wird.
- Die Zeit wird durch *t* dargestellt.

Zum besseren Verständnis werden nun die einzelnen Terme im Kontext erläutert, die in der Formel weiterhin benutzt werden:

- Im Zusammenhang mit *t* drückt *dt* den Abstand zwischen zwei Zeitpunkten, also einen Zeitraum bzw. eine Zeitdauer aus.
- *da* bezeichnet den Abstand zwischen zwei Prozentangaben also deren Differenz.
- *Na* bezeichnet die bereits infizierten Systeme.
- $1-a$ bezeichnet die noch nicht infizierten Systeme.
- $K(1-a)$ bezeichnet die Infektionen, die im Zeitraum *dt* stattfinden.

- Unter Berücksichtigung der bereits infizierten Systeme *Na* ergibt sich *Nda*, wobei *Nda* die Anzahl der Systeme ist, welche im Zeitraum *dt* infiziert werden.

Angenommen, zu einer bestimmten Zeit *t* wurde *a* Prozent der verwundbaren Systeme infiziert. Nun stellt sich die Frage: Wie viele Systeme *Nda* werden im nächsten Zeitabschnitt infiziert?
Die Antwort auf die Frage ergibt sich aus folgender Formel von Stuart Staniford, Vern Paxson und Nicholas Weaver (aus [Staniford 02]):

$$Nda = (Na)K(1-a)dt$$

Zu Beginn der Ausbreitung befällt jedes infizierte System *K* andere Systeme pro Zeiteinheit. Im Verlauf der weiteren Ausbreitung werden aber nur noch *K (*1 – *a)* Systeme durch ein System infiziert, weil *a* Prozent der Systeme bereits befallen ist.
Die Anzahl der infizierten Systeme, welche bei der nächsten Vergrößerung des Zeitraumes *dt* infiziert wird, verhält sich proportional zu der Anzahl der bereits infizierten Systeme *Na*, wobei jedes infizierte System *K (*1 – *a)* Systeme infizieren kann.

Nach Umstellung der oberen Gleichung ergibt sich folgende Differentialgleichung:

$$\frac{da}{dt} = Ka(1-a)$$

Diese Gleichung hat die Lösung:

$$a = \frac{e^{K(t-T)}}{1+e^{K(t-T)}}$$

T ist dabei der Zeitpunkt, an dem die Hälfte der verwundbaren Systeme *N* infiziert ist und an dem die meisten Neuinfektionen stattfinden. Außerdem ist *T* eine Konstante, die den Wendepunkt einer Kurve vom exponentiellen Bereich in den Bereich der asymptotischen Annäherung angibt. Für $t<T$ wächst *a* exponentiell. Größere *t* führen dazu, dass *a* gegen 1 geht, d.h. dass so gut wie alle verwundbaren Systeme infiziert sind.

Weiterhin geht aus der Formel hervor, dass die Infektionsrate ausschließlich von *K* abhängt. Die initiale Infektionsrate *K* kann durch die Nutzung einer Hitliste (vgl. Abschnitt 2.1.1) für die anfängliche Ausbreitung wesentlich erhöht werden.

Das RCS Modell schätzt die Infektionsrate nur recht grob ab, weil es mit vielen Vereinfachungen arbeitet. Dennoch reicht es aus, um die Infektionsrate eines normalen Wurms von der Infektionsrate eines Superwurms abzugrenzen. Der Parameter *N* zeigt, wie viele Systeme potentiell infiziert werden können, und kann deshalb zur Abschätzung der totalen Anzahl der am Ende infizierten Systeme benutzt werden. *Na* gibt an, wie viele der verwundbaren Systeme tatsächlich infiziert werden. Wenn *Na* am Ende der Ausbreitung mit der Anzahl aller Systeme im Internet in das Verhältnis gesetzt wird, ergibt sich der Prozentsatz der infizierten Systeme im Internet zu diesem Zeitpunkt.

3.3.2 Simulation der Ausbreitung von Würmern zur Überprüfung der Korrektheit des RCS Models

Um die Korrektheit des RCS Modells zu überprüfen, wurde es auf die Daten einiger Würmer angewendet (vgl. [Staniford 02]). Zu diesem Zweck wurde ein Wurm simuliert, der 10 Scans pro Sekunde bei N = 300.000 verwundbaren Systemen durchführt. Die Ergebnisse der Simulation sind mit dem Modell für K = 2,6 Systeme/ Stunde und T = 5,52 Stunden vergleichbar (vgl. Abbildung 11):

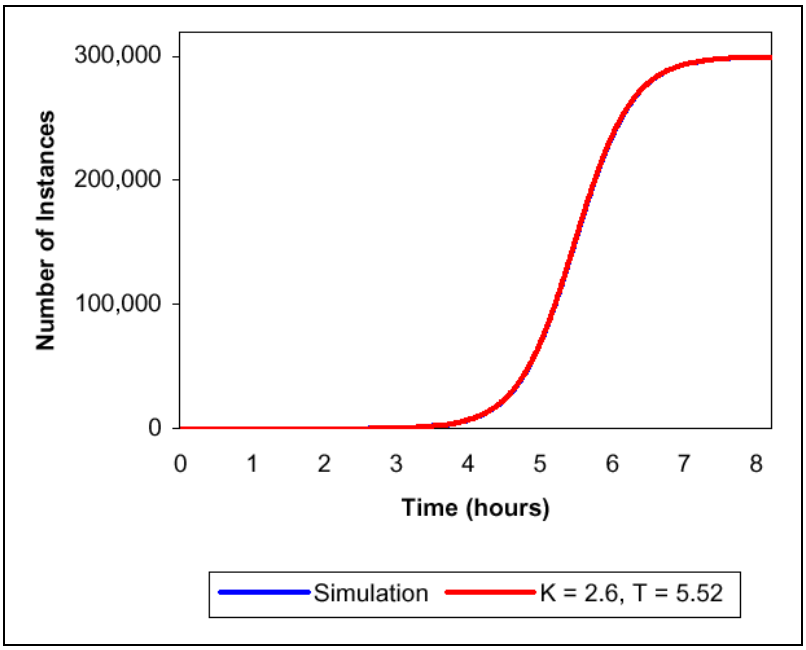

Abbildung 11: Vergleich der Ausbreitung eines simulierten Wurms mit einem Wurm nach dem RCS Modell (aus [Staniford 02]), wobei sich die Ergebnisse vollständig überlappen.

Nachdem durch diese Simulation gezeigt wurde, dass das RCS Modells korrekt sein könnte, wurden drei weitere Simulationen für N = 300.000 verwundbare Systeme durchgeführt.

Die erste Simulation ist für einen Wurm, der 10 Scans pro Sekunde durchführt und welcher mit Code Red II vergleichbar ist ([Staniford 02]). In der zweiten Simulation scannte der Wurm schneller mit 100 Scans pro Sekunde. Schließlich wurde der Warhol Wurm simuliert. Dieser benutzt in der Beschreibung eine Hitliste mit 10.000 Einträgen für die anfängliche Ausbreitung und danach permutiertes Scannen für die weitere Ausbreitung. Es wurden ebenfalls 100 Scans pro Sekunde angenommen. Das permutierte Scannen eines infizierten Systems wurde beendet, sobald zwei bereits infizierte Systeme entdeckt wurden, ohne ein weiteres verwundbares System zu finden (vgl. Abbildung 12).

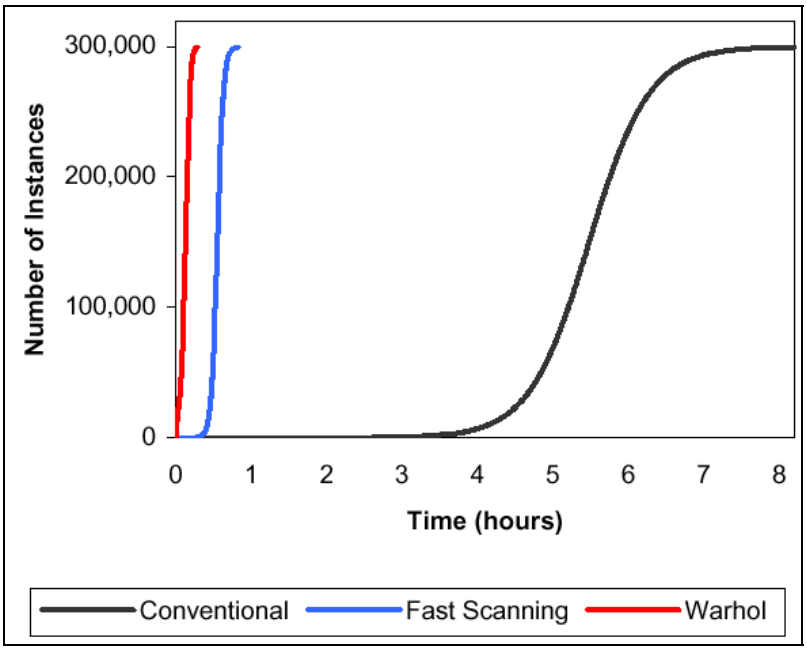

Abbildung 12: Vergleich der Ausbreitung dreier simulierter Würmer: eines herkömmlichen Wurms, einen schnell scannenden Wurms und des Warhol Wurms (aus [Staniford 02]). Alle drei Graphen enden, wenn 99,99% der verwundbaren Systeme infiziert sind.

Deutlich zu erkennen ist, dass der Warhol Wurm es schafft, alle verwundbaren Systeme innerhalb von 15 Minuten zu infizieren, wie in [Weaver 01] beschrieben wurde (vgl. Abschnitt 2.1).

Das Besondere an dieser Simulation des Warhol Wurms ist, dass eine Wurminstanz inaktiv (dormant) wird, sobald sie zwei bereits infizierte Systeme findet, ohne ein weiteres System zu infizieren. Deswegen werden nach einiger Zeit immer mehr Wurminstanzen inaktiv. Durch die Hitliste gibt es in den ersten Minuten einen sehr starken exponentiellen Anstieg im Vergleich zu den anderen simulierten Würmern. Während dieser Phase sind die meisten Wurminstanzen aktiv. Nachdem die Kurve den Wendepunkt zu der asymptotischen Annäherung erreicht hat, nimmt die Zahl der inaktiven Wurminstanzen stark zu (vgl. Abbildung 13).

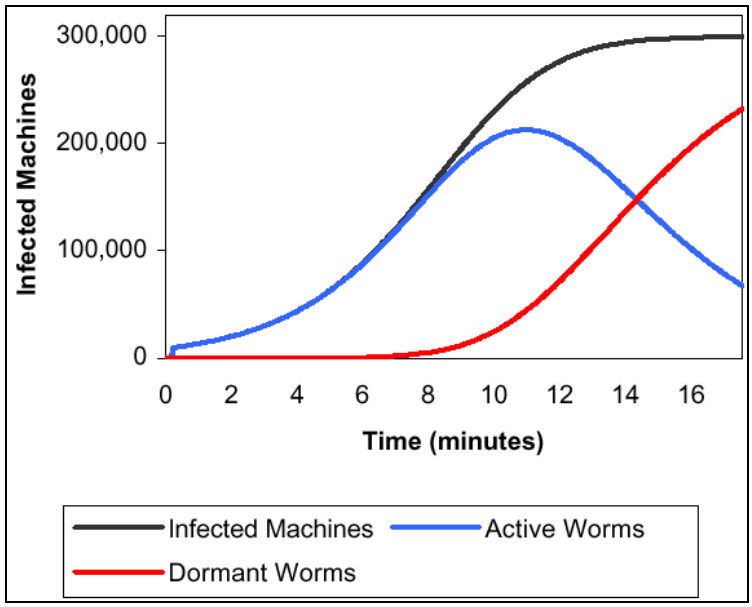

Abbildung 13: Detailliertere Abbildung zum Verhalten des Warhol Wurms aus Abbildung 12 (nach [Staniford 02]).

Problematisch wird die Anwendung des RCS Modells auf Würmer, die sich schon im Internet ausgebreitet haben. Denn es ist schwer, die erforderlichen Parameter für Würmer wie Nimda, Code Red I bzw. II oder Sapphire zu bestimmen. Allein die Bestimmung der Anzahl der verwundbaren Systeme *N* kann mangels Daten nicht genau erfolgen. Zudem ist das Internet konstruktionsbedingt unübersichtlich und unterliegt ständigen Veränderungen, so dass ermittelte Daten bestenfalls einen kurzlebigen Ist-Zustand repräsentieren können.

3.4 Schätzungen über die Realisierbarkeit von Superwürmern nach heutigem Stand der Technik

In den vorherigen Abschnitten wurde gezeigt, wie ein Superwurm aussehen müsste, um die Kriterien der Definition zu erfüllen. Dieser Abschnitt beschäftigt sich mit der Frage, ob ein Superwurm zum jetzigen Zeitpunkt überhaupt realistisch bzw. realisierbar wäre.

Das Internet ist ein großes und komplexes Netzwerk aus einer großen Anzahl von unterschiedlichen Teilnetzen, Servern und Clients. Diese unterscheiden

sich nicht nur in Hardware, Betriebssystem, installierter Software und Aufgabenstellung, sondern auch durch ihre installierten Schutzmechanismen. Oftmals sind genaue Daten über die Systeme oder Teilnetze nicht verfügbar oder werden bewusst geheim gehalten, um Angreifern keine Hilfestellung zu leisten. Deshalb existieren keine zuverlässigen Daten über die Anzahl der Systeme im Internet insgesamt und der Architektur dieser Systeme. Da entsprechende Daten nur abgeschätzt werden können, kann nicht mit absoluter Genauigkeit analysiert werden, ob Superwürmer zurzeit realisierbar wären. Im Folgenden soll die Realisierbarkeit von Superwürmern diskutiert werden, wobei die Daten über das Internet, wie dargelegt, auf Schätzungen basieren.

Das Internet besteht schätzungsweise aus 400 Millionen Servern und Clients, im Folgenden Hosts genannt. Ein Superwurm müsste demnach mindestens 40 Millionen Hosts innerhalb von 24 Stunden infizieren. Nach [ISC 03] befanden sich im Januar 2003 genau 171.638.297 permanent im Internet. Demnach befindet sich die restliche Anzahl der Hosts nur temporär im Internet, wodurch die Zahl der durch einen Superwurm infizierbaren Hosts noch schwieriger zu bestimmen ist.

Die bereits existierenden Würmer, welche im zweiten Kapitel beschrieben wurden, befallen immer nur eine Plattform. Würmer, die mehrere Plattformen wie zum Beispiel Windows 2000 und Linux befallen, sind kaum im Umlauf, weil es zu aufwendig ist, sie zu implementieren. Deshalb beschäftigt sich diese Analyse nur mit Superwürmern, welche lediglich eine Plattform infizieren.

Nach [Brunnstein 03] werden etwa 70 Prozent der Hosts im Internet mit Microsoft Systemen betrieben werden. Auf den restlichen 30 Prozent der Hosts befinden sich Linux-Derivate, Unix, SUN, Macintosh und andere Betriebssysteme. Auf Grund des hohen Anteils an Microsoft Betriebsystemen wird mit diesen Systemen weiter verfahren.

Microsoft hat eine ganze Reihe von Betriebssystemen auf den Markt gebracht. Im Folgenden werden alle Betriebssystemversionen wie DOS, Windows 95 und Windows Millennium wegen ihrer schätzungsweise geringen Anzahl vernachlässigt.
Betrachtet werden hingegen Windows 98, Windows NT 4.0, Windows 2000 und Windows XP. Es existieren keine verlässlichen Angaben über die gesamte Anzahl der Systeme im Internet, deshalb ist es unmöglich, die Anzahl der einzelnen Microsoft Betriebssysteme genau zu beziffern. Im Folgenden wird von einer geschätzten Gesamtanzahl von 250 Millionen Hosts ausgegangen.
Diese verteilen sich nach Schätzungen auf 30 Millionen Hosts mit Windows 98, 80 Millionen Hosts mit Windows NT 4.0, 100 Millionen Hosts mit Windows 2000 und 40 Millionen Hosts mit Windows XP. Von diesen Hosts müssen für die Erfüllung der Kriterien der Superwurmdefinition mindestens 40 Millionen Hosts innerhalb von 24 Stunden infiziert werden.
Dafür muss der Superwurm die Schwächen dieser Betriebssysteme ausnutzen. Es muss dabei aber auch beachtet werden, dass viele dieser Schwächen schon erkannt wurden und durch Patches behoben werden können. Für Windows NT 4.0 gibt es bereits einen Service Pack 6a (SP6a) und für Windows 2000 den SP4, während für Windows XP SP1a existiert. Ein gewisser Anteil von Hosts wird also nicht infizierbar sein. Dieser Anteil liegt schätzungsweise bei 80 Prozent der Hosts mit Windows 98, 90 Prozent der Hosts mit Windows NT 4.0, 60 Prozent der Hosts mit Windows 2000 und 40 Prozent der Windows XP. Damit verbleiben für die Infektion durch einen Superwurm 78 Millionen Hosts.

Von diesen 78 Millionen Hosts müsste der Superwurm mindestens 40 Millionen infizieren. Viele Schwächen von Microsoft Betriebssystemen wurden bereits für die Kontaminationsprozesse durch verschiedener Malware ausgenutzt. Einige dieser Infektionsmechanismen könnte der Superwurm ebenfalls für seine Ausbreitung unter den ungeschützten Systemen verwenden.

Nach diesen Schätzungen bzw. Annahmen wären also genug infizierbare Hosts für einen Superwurm vorhanden. Ein weiteres Problem ist die Übertragung des Superwurms über das Internet zu den Hosts (vgl. Abschnitt 3.2.3).
Die Übertragungsmedien im Internet haben Raten bei Hochgeschwindigkeitsverbindungen von Gigabits pro Sekunde wie zum Beispiel das Deutsche Forschungsnetz bis hin zu wenigen Kilobits pro Sekunde bei Modemverbindungen. Da ein Superwurm nicht die benötigte Anzahl von Hosts über Hochgeschwindigkeitsverbindungen erreichen kann, weil es von derartigen Verbindungen nicht genügend gibt, muss die Größe des Superwurms selbst beachtet werden, weil diese die Übertragung über langsame Verbindungen erheblich verzögern kann. Zu der Größe des Superwurms muss noch die Größe der IP-Header, welche zwischen 20 und 60 Byte groß sind, sowie des TCP-Headers von 20 Byte oder des UDP-Headers von 8 Byte addiert werden (vgl. [Tanenbaum 98]). Insgesamt ist eine Größe von bis zu 20 Kilobyte auch für langsame Modems, wie zum Beispiel „28k Modems“ kein Problem. Auch Zwei- bis Dreihundert Kilobyte große Superwürmer könnten noch relativ schnell übertragen werden. Bei einer Übertragung von Paketen mittels TCP muss eine Zeitverzögerung für den Vebindungsaufbau beachtet werden.

Ein Superwurm würde nach den Beschreibungen in diesem Kapitel mehrere Infektionskanäle besitzen. Außerdem könnte er ein eigenes SMTP Programm besitzen, um sich zusätzlich über Email auszubreiten. Nimda besitzt ebenfalls ein eigenes SMTP Programm und breitet sich über weitere 18 Kanäle aus. Damit kann Nimdas Größe von 57344 Byte (vgl. [Tocheva 01]) als Vergleichswert herangezogen werden. Angenommen ein Superwurm hätte noch mehr Infektionskanäle als Nimda, dann könnte er trotzdem kleiner als 100 Kilobyte sein.
Ein 100 Kilobyte großer Superwurm würde auf zwei TCP- und IP-Pakete aufgeteilt werden, wodurch etwa 80 Byte Code durch die Header dazukommen. Der zu übertragene Code wäre somit 102480 Byte groß. In diesem Beispiel beträgt die Verzögerungszeit eine halbe Sekunde. Wird der Superwurm in einem lokalen Netzwerk mittels TCP und IP mit einer

Übertragungsgeschwindigkeit von 10 Megabyte pro Sekunde verschickt, wird er in etwa 0,5098 Sekunden auf das Zielsystem übertragen.

Der letzte Punkt, der diskutiert werden muss, ist, dass niemand sich die Mühe machen wird, einen so komplexen und aufwendigen Wurm zu schreiben. Selbst wenn jemand versucht einen Superwurm zu schreiben, fällt dieser nicht unbedingt in die Klasse der Superwürmer, weil der Wurm nicht genug Systeme infiziert oder nicht schnell genug ist. Das Problem bei der Implementation eines Superwurms besteht darin, dass auch ein Superwurmautor Fehler macht. Ein Superwurm ist auch nur ein Programm, welches wie jede Software Fehler enthält, deren Anzahl mit der Komplexität des Programms steigt.

Die Realisierung eines Superwurms ist also technisch machbar, aber sehr aufwendig und fehleranfällig. Deshalb könnte nach dem heutigen Stand der Technik zwar ein Wurm geschrieben werden, der das Potential zu einem Superwurm hat. Aber die tatsächliche Ausbreitungsgeschwindigkeit und die Anzahl der infizierten Systeme hängen nicht nur vom Wurmcode selber ab, sondern auch von der aktuellen Architektur des Internets, der momentanen Netzlast und vom Zufall. Somit muss auch ein effektiv programmierter Wurm nicht zwangsläufig zu einem Superwurm werden. Tatsächlich ist dies nach dem heutigen Stand sehr unwahrscheinlich, aber möglich.

3.5 Zusammenfassung

Die Superwürmer wurden quantitativ definiert. Die Kriterien der Definition erfordern die Erreichung einer bestimmten Infektionsrate. Diese hängt von den genutzten Ausbreitungsmechanismen ab. Da die existierenden Würmer die Kriterien der Definition nicht erfüllen, müssten neue Mechanismen für die Ausbreitung von Superwürmern entwickelt werden. Des Weiteren könnte die Verwendung von mehreren Infektionskanälen die Infektionsrate steigern. Die Auswirkungen eines Superwurms könnten verheerend sein, aber ein

Superwurm könnte sich bei seiner Ausbreitung auch selbst behindern. Der Vergleich der Infektionsraten von normalen Würmern und Superwürmern ist schwierig. Damit ist eine Einstufung von Würmern in die Klasse der Superwürmer ein sehr aufwendiges Unterfangen. Superwürmer sind zwar technisch realisierbar, aber die Implementation wäre sehr aufwendig und fehleranfällig. Deshalb ist es unwahrscheinlich, dass ein Wurm zum heutigen Zeitpunkt die Kriterien eines Superwurms erfüllt.

4 Entdeckung und Gegenmaßnahmen

In diesem Kapitel wird erst beispielhaft in einem Szenario die Wirkungsweise eines Superwurms beschrieben. Danach wird mit Hilfe des Szenarios auf die Entdeckung eines Superwurms eingegangen, um schließlich darzustellen, welche Gegenmaßnahmen zur Bekämpfung von Superwürmern ergriffen werden könnten.

4.1 Das Szenario

Im dritten Kapitel wurde dargestellt, dass ein Superwurm technisch machbar wäre, eine Realisierung aber eher unwahrscheinlich ist. Trotzdem soll ein Superwurm beispielhaft in einem Szenario beschrieben werden, um diesen theoretischen Wurm etwas greifbarer zu machen.
Das Szenario beschreibt als erstes einen möglichen Superwurm. In den Ausführungen über den Superwurm wird versucht, möglichst wenige Details für eine Implementation zu liefern, und trotzdem eine gute Vorstellung von der Wirkungsweise zu vermitteln. Die möglichen Auswirkungen werden danach für ein konkretes Netz beschrieben sowie für das Internet allgemein. Diese Auswirkungen werden später für die Entdeckung und Gegenmaßnahmen benutzt.

Um die Wahrscheinlichkeit der Realisierbarkeit von Superwürmern in diesem Szenario zu erhöhen, wird die fiktive Betriebssystemfamilie Doors des Softwareherstellers Nanosofter eingeführt. Die Betriebssysteme Doors 1900, Doors 1902 und Doors XXL sowie der Doors XO Server sind demnach mit einem Anteil von 80 Prozent im Internet vertreten. Von diesen Systemen sind etwa die Hälfte ungeschützt und verletzlich. Bei angenommenen 400 Millionen Hosts insgesamt im Internet sind also 160 Millionen Hosts infizierbar.

4.1.1 Der Superwurm „Sturmflut“

Der fiktive Superwurm „Sturmflut“, der in diesem Szenario erläutert werden soll, breitet sich über vier Infektionskanäle aus:

1. Ein Infektionskanal nutzt einen Pufferüberlauf in den fiktiven Betriebssystemen Doors 1900, Doors 1902 und Doors XXL aus. Gegen diese Schwachstelle gibt es bereits einen funktionierenden Patch. Die Ausbreitung findet mit Hilfe von lokalem Scannen statt (vgl. Abschnitt 2.4). Mit einer Wahrscheinlichkeit von 1/3 wird eine IP-Adresse aus dem derzeitigen Klasse A IP-Adressraum gewählt. Des Weiteren werden IP-Adressen mit einer Wahrscheinlichkeit von 1/6 aus Klasse B, 1/6 aus Klasse C und 1/4 aus Klasse D erzeugt. Eine zufällige IP-Adresse aus dem gesamten IP-Adressraum wird mit 1/12 Wahrscheinlichkeit generiert.
2. Für den zweiten Infektionskanal werden Schwächen in Webservern genutzt. Dadurch werden HTML-Webseiten modifiziert. Wird eine solche Webseite besucht, wird der Superwurm vom besuchenden System mit heruntergeladen. Die Webseiten werden durch zufälliges Scannen (vgl. Abschnitt 2.1.2) gefunden und danach infiziert.
3. Ein eigenes SMTP-Programm zum Versenden von Emails, mit dem Superwurm als Anhang, ist ein weiterer Infektionskanal (vgl. Abschnitt 2.5). Der Superwurm sucht sämtliche Sendeadressen aus allen Adressbüchern für Email auf einem infizierten System zusammen. Danach verschickt er an alle Sendeadressen eine Email mit dem Betreff: „Sachen zum Lachen“ und dem Superwurm selbst als Anhang. Auf den Systemen, wo der Anhang automatisch ausgeführt wird, findet die Infektion sofort nach dem Öffnen der Email statt und ansonsten erst nach der Ausführung des Anhangs durch den Benutzer.
4. Schließlich nutzt der Superwurm als Infektionskanal die offene.Scheunentor-Schwäche auf Doors XO Servern aus. Auch zu dieser Schwäche gibt es einen Patch. Der Ausbreitungsmechanismus für diesen Kanal ist permutiertes Scannen (vgl. Abschnitt 2.1.3), wobei der IP-Adressraum in 128 Blöcke aufgeteilt ist.

Jedes infizierte System erzeugt Prozesse für jeden Infektionskanal, d.h. für den ersten Infektionskanal werden 100 Prozesse generiert, die parallel nach neuen Zielen Suchen, für den zweiten Kanal tun dies 20 Prozesse, welche nach infizierbaren Webservern suchen, für den dritten Kanal nur ein Prozess und für den vierten Kanal noch einmal 100 Prozesse. Dadurch werden einem infizierten System sehr viele Ressourcen für die eigentlichen Aufgaben entzogen. Die Prozesse können sich auf leistungsschwachen Systemen sogar gegenseitig behindern.

Durch die 220 parallel laufenden Scanprozesse eines Systems werden eine große Menge Pakete über das Internet bzw. dessen Teilnetze verschickt, insbesondere da eine Vielzahl von Systemen parallel nach verwundbaren Systemen scannt. Die Auswirkung ist eine erhöhte Netzlast in vielen Teilnetzen des Internets. Eine andere Auswirkung hat die eigentliche Schadfunktion des Superwurms. Sie startet einen DDOS-Angriff, wenn der Wochentag auf einem System als Montag angezeigt wird. Dann bricht der Superwurm sämtliche von ihm eingeleitete Scanprozesse ab und beginnt einen DDOS-Angriff auf die DNS-Server (domain name service). An allen anderen Wochentagen nutzt der Superwurm den ersten, zweiten und vierten Infektionskanal zu seiner Ausbreitung. Der dritte Infektionskanal über Email wird immer bei der Neuinfektion eines Systems gestartet und genau einmal durchlaufen. Die anderen Infektionskanäle scannen ständig nach neuen Zielsystemen weiter.

Nach dem RCS Modell (vgl. Abschnitt 3.3) soll der Superwurm in dieser Modellierung folgende Parameter aufweisen:

- N = 42.000.000
- K = 1,1 Systeme pro Stunde
- T = 10 Stunden

Der Superwurm scannt recht schnell, d.h. er führt 100 Scans pro Sekunde durch, wodurch die hohe anfängliche Infektionsrate von 1,1 Systemen pro Stunde zustande kommt. Nach 10 Stunden sind bereits 21 Millionen Systeme

mit dem Superwurm infiziert. Aus diesen Werten ergibt sich das in Abbildung 14 dargestellte Ausbreitungsverhalten:

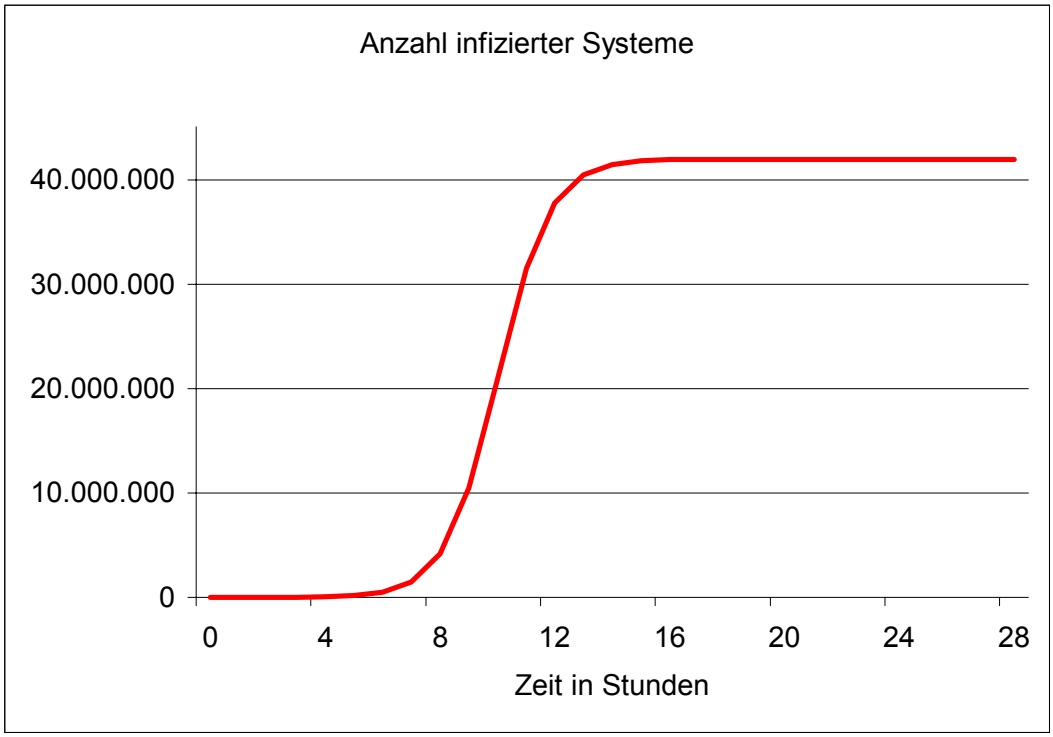

Abbildung 14: Ausbreitung des Superwurms Sturmflut nach dem RCS Modell

Zur Zeit besteht das Internet aus etwa 400 Millionen Systemen, d.h. wenn zehn Prozent des Internets durch einen Superwurm infiziert sind, handelt es sich um etwa 40 Millionen Systeme. Der Superwurm Sturmflut erfüllt die Kriterien der Superwurmdefinition (vgl. Abschnitt 3.1), weil er innerhalb von 24 Stunden 42 Millionen Systeme infiziert.

4.1.2 Auswirkungen von Sturmflut auf ein Firmennetz

Das Firmennetz der Pleitegeier GmbH und Co. KG besteht aus mehreren Standortnetzen, welche durch ein Backbone verbunden sind (vgl. Abbildung 15). Jedes Standortnetz bildet eine eigene Sicherheitszone bzw. ist in unterschiedliche Sicherheitszonen unterteilt, weil in den Standortnetzen mit mehr oder weniger sensiblen Daten gearbeitet wird. Es existieren vier

Standortnetze. Eines der Standortnetze ist die Firmenleitung, deren Netz in unterschiedliche Sicherheitszonen für die Abteilungen für das Personal, die Buchhaltung, die Finanzen und so weiter eingeteilt ist. Alle Sicherheitszonen sind voreinander durch Firewalls geschützt. Alle Standortnetze können über das Backbone auf das Internet zugreifen. Zum Schutz des Internetzugangs existiert ebenfalls eine Firewall. Jedes Teilnetz sowie das Backbone besitzen einen eigenen DNS Server. Der Datenverkehr zwischen den Standortnetzen wird durch ein netzwerkbasiertes Eindringlingsentdeckungssystem (NIDS) überwacht.

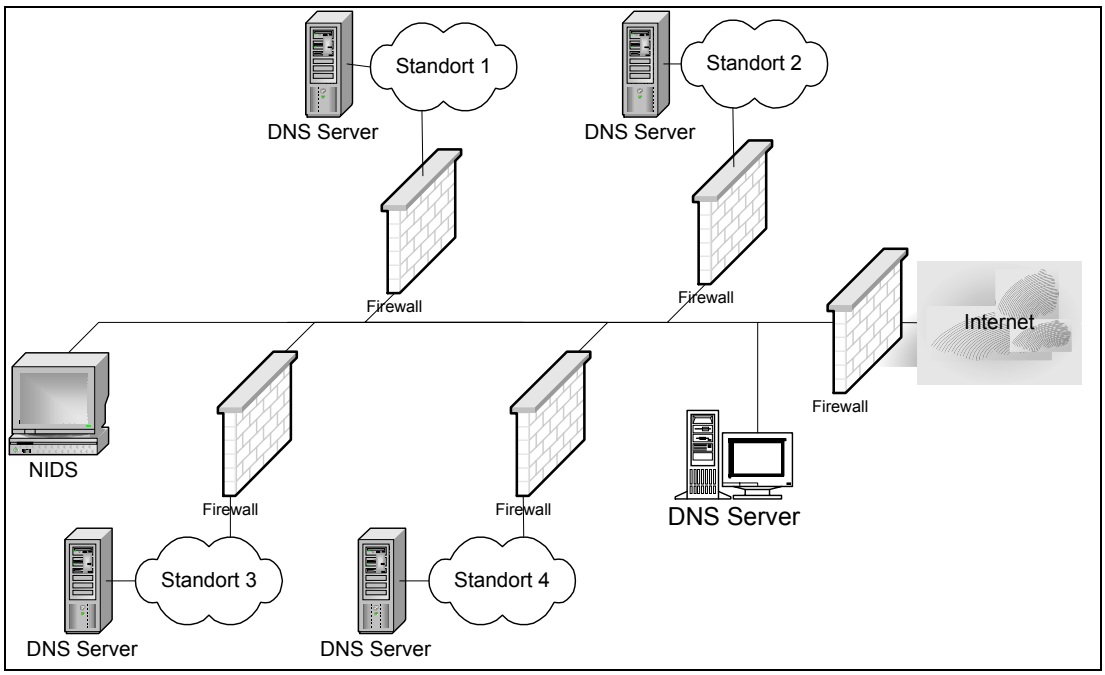

Abbildung 15: Backbone der Pleitegeier GmbH und Co. KG

Einige Mitarbeiter bekommen infizierte Emails von außerhalb des Netzes geschickt. Dadurch werden alle Zonen des Firmennetzes mit einem niedrigen Sicherheitsniveau infiziert, weil hier die Sicherheitseinstellungen nicht sehr hoch sind. Die Sicherheitszonen mit einem hohen Sicherheitsanspruch, wie zum Beispiel das Finanzwesen der Firma, wurden nicht betroffen, weil Emails mit Anhängen durch spezielle Firewalls, den Appliction-Level Gateways, gefiltert werden. Diese verhinderten ebenfalls eine Infektion der Systeme durch befallene Webserver. Die beiden anderen Infektionskanäle scheiterten direkt an der Firewall beim Internetzugang. Trotzdem konnten durch sie

Teilnetze mit sehr niedrigem Sicherheitsniveau infiziert werden, weil der Superwurm durch HTML bzw. Email dort eindringen kann und die Systeme nicht gegen die genutzte offene.Scheunentor-Schwäche und den Pufferüberlauf in Doors 1900/1902/XXL Schwächen gepatcht sind.

In den infizierten Teilnetzen kommt es zu einer erhöhten Netzlast, weil die Superwurminstanzen versuchen, von hier aus noch mehr Systeme zu infizieren. Des Weiteren wird eine Vielzahl von Emails an die Mitarbeiter der Firma sowie an Personen außerhalb verschickt. Dies bleibt von den Administratoren nicht unbemerkt. Zum Einen überschreitet der Emailverkehr im Firmennetz den eingestellten Schwellwert des NIDS. Zum Anderen verschicken die infizierten Systeme so viele Pakete im Backbone der Firma, dass auch hier der Schwellwert des NIDS überschritten wird und es eine Alarmmeldung ausgibt. Zu diesem Zeitpunkt sind bereits alle Systeme, die der Superwurm im Netz der Firma infizieren kann, befallen. Eine automatische Schließung der Firewalls durch das NIDS ist nicht vorgesehen. Bis diese Maßnahme durch die Administratoren durchgeführt wurde, ist es bereits zu spät.

4.1.3 Auswirkungen von Sturmflut auf das Internet

Um die Auswirkungen eines Superwurms auf das Internet zu verdeutlichen, wird das Szenario mit der Pleitegeier GmbH und Co. KG erweitert.
Ein Standort beschäftigt sich in der Hauptsache mit der Weiterentwicklung von Produkten der Firma. Dies ist ein sehr innovativer, teurer aber auch gewinnbringender Teil der Firma, welcher ständig durch Wirtschaftsspionage bedroht ist. Deshalb ist das Standortnetz eine Hochsicherheitszone, in welche der Superwurm nicht eindringen konnte. Da die Produktweiterentwicklung sehr kostenintensiv ist, wird mit ausgesuchten Partnern zusammen gearbeitet. Zu diesem Zweck werden Entwicklungsdaten über ein Virtual Private Network (VPN), welches durch Tunneln des Internets realisiert ist, ausgetauscht. Diese virtuellen Tunnel werden nicht ständig gebraucht, und deshalb erst bei Bedarf

aufgebaut. Der Superwurm erzeugt für seine Ausbreitung sehr viele Pakete, so dass die Router mit einer großen Vielzahl von Paketen fertig werden müssen. Für den Aufbau eines virtuellen Tunnels zwischen den Entwicklungspartnern bedeutet dies, dass ein fester Weg durch das Internet nicht aufgebaut werden kann. Dadurch kommt es bei der Weiterentwicklung von Produkten der Firma zu Verzögerungen und schließlich auch zu finanziellen Verlusten.

Am nächsten Montag nach der Initialisierung von Sturmflut zeigt das erste Mal die Schadfunktion ihre Wirkung. Auch nach ein paar Tagen sind noch einige Millionen Systeme mit Sturmflut infiziert. Diese beginnen nun nutzlose Anfragen über nicht existierende Domänennamen an die DNS Server zu stellen. Dies führt dazu, dass die eigentlichen Anfragen nicht beantwortet oder an einen anderen DNS Server weitergeleitet werden können. Bei der Pleitegeier GmbH und Co. KG kann mit sämtlichen Geschäftspartnern einen Tag lang nicht kommuniziert werden, wodurch das gesamte Tagesgeschäft ausfällt. Die finanziellen Verluste der Firma sind immens.

Sämtliche Geschäftstätigkeiten von Unternehmen über das Internet könnten durch den Superwurm behindert bzw. zeitweise verhindert werden. Alle anderen Nutzer des Internets würden ebenfalls in Mitleidenschaft gezogen werden. Allein der Arbeitsaufwand, um den Superwurm von allen Systemen zu entfernen, wäre beträchtlich. Der finanzielle Schaden sowie der Image-schaden vieler Organisationen bei Ausbreitung eines Superwurms wären kaum zu beziffern.

4.2 Maßnahmen zur Entdeckung von Superwürmern

Durch das vorangegangene Szenario wurden einige Aspekte der Entdeckung von Superwürmern angedeutet. In diesem Abschnitt kann ebenfalls nur angedeutet werden, was noch zur Entdeckung von Superwürmern führen könnte. Eine Ausbreitung eines Superwurms fand bis heute nicht statt, so

dass daraus auch keine Lehren in Form von Maßnahmen hätten gezogen werden können. Trotzdem ist es entscheidend zur Bekämpfung eines Superwurms im Speziellen und von Malware im Allgemeinen, diese schnell zu entdecken.

Da sich Superwürmer über das Internet ausbreiten und versuchen, alle an das Internet angeschlossenen Teilnetze bzw. Intranets zu befallen, ist die ständige Überwachung des Netzverkehrs zumindest in den Teilnetzen unabdingbar. Hierzu bietet sich der Einsatz von netzwerkbasierten Intrusion Detection Systemen (NIDS) an. NIDS können die Anzahl der in einem Netzsegment übertragenden Pakete pro Zeitperiode zählen, um die Netzwerkbelastung zu bestimmen. Ist die Netzwerkbelastung höher als der dafür eingestellte Schwellwert, wird eine Warnmeldung ausgegeben. NIDS können sich auch den Inhalt der Pakete auf dem Netz anschauen. Entdecken sie dabei in den Paketen sich selbst-replizierenden Code, also einen sicheren Hinweis auf Viren oder Würmer, können sie ebenfalls davor warnen. Die Entwicklung von NIDS ist aber noch nicht so weit, dass Paketinhalte in vertretbarer Zeit und mit geringem Aufwand auf spezielle Malware analysiert werden können. Um die Ausbreitung eines Superwurms im eigenen Netzwerk zu verhindern, könnte ein NIDS vor dem Internetzugang des Netzes alle eingehenden Pakete untersuchen. Weitere Informationen zu NIDS befinden sich in der Arbeit [Hoherz 01].

Eine weitere Möglichkeit der Entdeckung ist der Einsatz von Application-Level Gateways. Dies sind spezielle Firewalls, welche den Inhalt von Paketen auf der Anwendungsschicht überprüfen können. Wird zum Beispiel selbst-replizierender Code in den Paketen gefunden, werden sie von dem Application-Level Gateway gestoppt und der Vorfall gemeldet. Auch hier, wie beim NIDS, besteht das Problem, dies in kurzer Zeit mit geringem Aufwand zu bewerkstelligen, damit der Datenfluss nicht zu stark verlangsamt wird. Als Lösung des Problems wäre eine stufenweise Verarbeitung der Pakete durch ein Sicherheitssystem denkbar. Zuerst müssen die Pakete durch eine Firewall mit Paketfilterung, welche nach bestimmten Regeln die Pakete durchlässt

oder gleich abblockt. Dabei sollte die Firewall nach dem Grundsatz des generellen Verbots und mit speziellen Erlaubnisregeln arbeiten, um auch vor unbekannten Bedrohungen zu schützen. Die Pakete, welche die Firewall passiert haben, werden noch durch einen Virenscanner überprüft. Gibt es dann noch Pakete, die nicht mit Bestimmtheit als sicher oder unsicher eingeordnet werden können, werden sie an ein Application-Level Gateway weitergeleitet, welches die Pakete genauer analysiert. So können die meisten Pakete die Firewalls relativ schnell passieren, wenn sie als unkritisch eingestuft wurden.

Bei der Erkennung von Superwürmern können auch Virenscanner helfen. Allerdings sind sie bei der ersten Ausbreitung des Superwurms in der Regel nutzlos, weil sie für die Erkennung des Superwurms keine Signatur besitzen. Einzig eine sehr gute heuristische Erkennung eines Virenscanners könnte einen Superwurm finden. Sobald eine Signatur zur Erkennung des Superwurms existiert, können Virenscanner entscheidend zur Entdeckung von Superwürmern beitragen. Zum Einen können on-demand Scanner feststellen, ob ein System bereits infiziert ist. Zum Anderen können on-access Scanner auch bei einer erneuten Ausbreitung den Superwurm entdecken, bevor er ein System infiziert.

Die rechtzeitige Entdeckung von Superwürmern ist nach heutigem Stand der Technik sehr schwierig. Allein eine ständige Überwachung des Netzverkehrs auf ungewöhnliche Veränderungen könnte zumindest zu einer schnelleren Entdeckung führen. Sobald eine hohe Netzlast festgestellt wird, könnte es bereits zu spät sein, einen Superwurm aufzuhalten, weil dieser dann bereits viele Systeme infiziert hat. Die Entdeckung von Superwürmern wäre zu langsam.

4.3 Gegenmaßnahmen zum Schutz vor Superwürmern

In dem Szenario wurden bereits einige Gegenmaßnahmen zum Schutz vor einem Superwurm vorgestellt. Nun sollen diese verallgemeinert und ergänzt werden.

Das effektivste Mittel gegen einen Superwurm ist keine Verbindung zum Internet zu haben. Da ein Superwurm sich ausschließlich über das Internet ausbreitet, kann der Superwurm nicht auf andere Art und Weise ein isoliertes System infizieren.
Ist ein Internetzugang nötig, so sollte der Rechner, welcher mit dem Internet verbunden ist, nicht mit dem Intranet verbunden sein. So wird schlimmsten Falls nur dieser Internetrechner infiziert. Alle anderen Rechner stehen weiterhin zur Verfügung. Diese Maßnahme ist aber etwas unrealistisch, denn es gibt eine Vielzahl von Organisationen, die nicht darauf verzichten können oder wollen, dass das gesamte bzw. Teile des Organisationsnetzes Zugriff auf das Internet erhält.
Für diese Organisationen ist es unter anderem wichtig, dass sie eine sichere Netzstruktur aufgebaut haben. Dies kann durch die Einteilung in verschieden Sicherheitszonen geschehen, wie zum Beispiel bei der Pleitegeier GmbH und Co. KG. Den stark schutzbedürftigen Netzen wird gegebenenfalls kein Internetzugang gewährt, während auf den kaum schutzbedürftigen Systemen eines Netzes nur Virenscanner installiert sind und ansonsten ein freier Zugang ins Internet besteht. Dies sind zwei Extreme. Natürlich sind die Schutzmaßnahmen von der Sicherheitspolitik der Organisation abhängig und müssen den ermittelten Schutzanforderungen genügen.

Zum Schutz von Netzen bieten sich Firewalls an. Es sollte sich auf jeden Fall eine Firewall am Internetzugang befinden. Diese Firewall sollte unter anderem ein Application-Level Gateway sein, denn damit ist die Wahrscheinlichkeit höher, dass sie Malware sofort erkennt und sie erst gar nicht in das Netz lässt. Des Weiteren sollten auf schützenswerten Einzelsystemen Firewalls installiert werden. Besteht ein Organisationsnetz aus mehreren Teilnetzen, dann sollte

jedes Teilnetz zusätzlich durch eine Firewall und anderen Maßnahmen geschützt werden, damit bei Ausfall der Internetfirewall die Teilnetze trotzdem relativ sicher sind. Firewalls besitzen allerdings, wie jede Software, Schwachstellen, und können deshalb selbst Ziele von Angriffen werden. Weil Firewalls immer bestimmte Dienste passieren lassen müssen, können sie die Bedrohungen nicht auf Null reduzieren.

Eine gute Administration und Wartung aller Systeme ist unentbehrlich. Es sollten alle notwendigen Patches installiert sein, die Signaturen der Virenscanner immer auf dem neuesten Stand gehalten werden und alle beteiligten Personen sollten ein gewisses Sicherheitsbewusstsein entwickelt haben. Des Weiteren sollten andere organisatorische Maßnahmen implementiert sein, wie die Verantwortlichkeiten bei Vorfällen und der Ablauf bei Warnungen.

Alle diese Maßnahmen zur Entdeckung wie zum Schutz vor Superwürmern sind nichts Neues. Sie dienen generell zum Schutz vor Malware aus dem Internet. Deshalb werfen sie auch dieselben Probleme auf. Es ist fraglich, ob ein NIDS oder ein Application-Level Gateway nach dem heutigen Stand der Technik den Superwurm überhaupt entdeckt. Alle Schutz- und Entdeckungs-maßnahmen helfen nur wenig, wenn der Superwurm eine bis dahin unbekannte Schwäche ausnutzt. Des Weiteren spielen Mitglieder der Organisationen, die aus mangelndem Sicherheitsbewusstsein oder reiner Unwissenheit die Sicherheitsmaßnahmen umgehen oder außer Kraft setzen, einem Superwurm direkt in die „Hände“.
Gegen Superwürmer wie den Warhol oder den Flash Wurm könnte kaum etwas ausgerichtet werden, weil sie eine so hohe Netzlast erzeugen, dass keine vernünftige Kommunikation über das Internet mehr möglich ist. Das heißt, dass auch vor der Infektion durch die Superwürmer geschützte Systeme bzw. Netze in Mitleidenschaft gezogen werden, weil sie nicht mehr ihren Tätigkeiten im oder über das Internet nachgehen können. Dazu kommt, dass die Antivirenhersteller keine Warnmeldungen über das Internet verschicken

und die dringend benötigten Antiwurmsignaturen und Patches nicht von den Webseiten im Internet heruntergeladen werden können.

4.4 Zusammenfassung

In diesem Kapitel wurde gezeigt, wie ein Superwurm, grob aussehen könnte. Insbesondere die Auswirkungen auf einzelne Teilnetze, aber auch auf das gesamte Internet können verheerend sein. Superwürmer könnten zwar relativ schnell entdeckt werden, aber dann hätten sie bereits eine Vielzahl von Systemen infiziert. Um die eigenen Systeme und Netze vor einem Superwurm zu schützen, helfen die bekannten Sicherheitsmaßnahmen genauso gut bzw. schlecht wie bei anderer Malware.

5 Zusammenfassung

In dieser Diplomarbeit wurde gezeigt, dass konkrete Superwürmer spezielle, äußerst gefährliche Internetwürmer sein können, welche zur Malware zählen und damit den Bedrohungen von IT-Systemen zuzuordnen sind.
Es wurde deutlich, dass Würmer wie Code Red II, Nimda oder SQL Slammer/ Sapphire bereits sehr gefährlich sind, ohne zu den Superwürmern zu zählen. Durch neue Ausbreitungsmechanismen, wie sie beim Warhol und Flash Wurm erläutert wurden, stellen Superwürmer eine sehr große Bedrohung für den Betrieb des Internets dar.
Es konnte nur eine quantitative Definition von Superwürmern aufgestellt werden, weil für eine qualitative Definition reale Superwürmer fehlen. Dies könnte auch so bleiben, weil eine Erkenntnis dieser Diplomarbeit ist, dass die bis heute benutzten Ausbreitungsmechanismen und damit auch Infektionskanäle nicht reichen, um einem existierenden Wurm das Prädikat „super“ zu verleihen. Die Implementation eines Superwurms wäre zum heutigen Zeitpunkt zwar möglich, aber unwahrscheinlich, weil sie zu aufwendig und fehleranfällig wäre.

Die bis jetzt benutzten Entdeckungsverfahren für Malware und andere Bedrohungen von IT-Systemen sind zu langsam, um einen Superwurm rechtzeitig zu erkennen. Auch die Gegenmaßnahmen sind nicht unbedingt ausreichend, wenn ein Superwurm sich ausbreitet. Doch im schlimmsten Fall werden trotz aller Sicherheitsmaßnahmen alle Systeme im Internet durch einen Superwurm in Mitleidenschaft gezogen, weil die Ausbreitung des Superwurms große, wenn nicht sogar alle Teile des Internets durch Blockade der Kommunikationsverbindungen außer Betrieb nimmt.

Superwürmer sind bis jetzt nur Theorie. Es gibt aber Anzeichen, dass versucht wird Superwürmer zu implementieren, wie die Beispiele von Nimda, Code Red II und Sapphire zeigen.

Abbildungsverzeichnis

Literaturverzeichnis

[Brunnstein 99] Klaus Brunnstein: „From AntiVirus to AntiMalware Software and Beyond: Another Approach to the Protection of Customers from Dysfunctional System Behaviour“; Paper submitted to *22nd National Systems Security Conference*, Status: July 23, 1999

[Brunnstein 01] Klaus Brunnstein: Vorlesungsskript: *Gestaltbarkeit und Beherrschbarkeit von Informatiksystemen*; Universität Hamburg; Sommersemester 2001

[Brunnstein 02] Klaus Brunnstein: Vorlesungsskript: *Risikoanalyse, Risikomanagement und Forensische Informatik*; Universität Hamburg; Wintersemester 2002/ 2003

[Brunnstein 03] Klaus Brunnstein: *Diplomarbeitsbesprechung*, 16.07.2003; persönliches Gespräch

[Carrera 02] Ero Carrera, Mikko Hypponen: *F-Secure Computer Virus Information Pages: Lioten*; http://www.europe.f-secure.com/v-descs/lioten.shtml, 08.01.2003

[Chien 02] Eric Chien: *Symantec Security Response - CodeRed Worm*; http://securityresponse.symantec.com/avcenter/venc/data/codered.worm.html, 17.02.2003

[Erdelyi 01] Gergely Erdelyi, Sami Rautiainen, Mikko Hypponen: *F-Secure Computer Virus Information Pages: Code Red*; http://www.europe.f-secure.com/v-descs/bady.shtml, 16.02.2003

[Hoherz 01] Benjamin Hoherz, Silvio Krüger, Jan Menne, Nils Michaelsen: *Intrusion Detection Systeme in Firewalls*; Hamburg, Juli 2001

[Hypponen 03] Mikko Hypponen, Gergely Erdelyi: *F-Secure Computer Virus Information Pages: Slammer*; http://www.europe.f-secure.com/v-descs/mssqlm.shtml, 16.02.2003

[ISC 03] Internet Software Consortium (http://www.isc.org/): *Internet Software Consortium - Number of Internet Hosts*; http://www.isc.org/ds/host-count-history.html, 27.07.2003

[Kossakowski 03] Klaus-Peter Kossakowski: *Klaus-Peter Kossakowski: Computer-Würmer*; http://www.kossakowski.de/kap222.htm, 24.04.2003

[Kytojoki 02] Jari Kytojoki: *Symantec Security Response - W32.HLLW.Lioten*; http://securityresponse.symantec.com/avcenter/venc/data/w32.hllw.lioten.html, 08.01.2003

[McAfee 01] McAfee: McAfee – AVERT; http://vil.nai.com/vil/content/v_99142.htm, 16.02.2003

[McAfee 01a] McAfee: McAfee – AVERT; http://vil.nai.com/vil/content/v_99177.htm, 16.02.2003

[McAfee 02] McAfee: McAfee – AVERT; http://vil.nai.com/vil/content/v_99897.htm, 08.01.2003

[McAfee 02a] McAfee: McAfee – AVERT; http://vil.nai.com/vil/content/v_99209.htm, 16.02.2003

[McAfee 03] McAfee: McAfee – AVERT; http://vil.nai.com/vil/content/v_99992.htm, 16.02.2003

[Moore 03] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, Nicholas Weaver: *The Spread of the Sapphire/Slammer Worm*; http://www.cs.berkeley.edu/~nweaver/sapphire/, 08.02.2003

[Pfleeger 00] Charles P. Pfleeger: *Security in Computing*; 2. Edition; Prentice Hall; 2000

[Shirey 00] R. Shirey: *rfc2828.txt*; http://www.rfc-editor.org/rfc/rfc2828.txt, 10.04.2003

[Staniford 01] Stuart Staniford, Gary Grim, Roelof Jonkman: *SILICON DEFENSE - Flash Worm Analysis*; http://www.silicondefense.com/flash/, 23.11.2002

[Staniford 02] Stuart Staniford, Vern Paxson, Nicholas Weaver: *How to 0wn the Internet in Your Spare Time*; http://www.cs.berkeley.edu/~nweaver/cdc.web/, 08.02.2003

[Szor 02] Peter Szor, Eric Chien: *Symantec Security Response - CodeRed II*; http://securityresponse.symantec.com/avcenter/venc/data/codered.ii.html, 17.02.03

[Tanenbaum 98] Andrew S. Tanenbaum: *Computernetzwerke*; 3., revidierte Auflage; Prentice Hall; 1998

[Tocheva 01] K. Tocheva, G. Erdelyi, A. Podrezov, S. Rautiainen and M. Hypponen: *F-Secure Computer Virus Information Pages: Nimda*; http://www.europe.f-secure.com/v-descs/nimda.shtml, 16.02.2003

[Weaver 01] Nicholas C Weaver: *Warhol Worms: The Potential for Very Fast Internet Plagues*; http://www.cs.berkeley.edu/~nweaver/warhol.html, 23.11.2002

[Weaver 01a] Nicholas C Weaver: *A Warhol Worm: An Internet plague in 15 minutes!*; http://www.cs.berkeley.edu/~nweaver/warhol.old.html, 05.12.2002