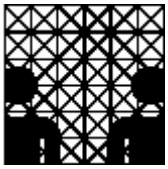
Universität Hamburg
Fachbereich Informatik



Incident Response Labor

Dokumentation zum Studienprojekt
Leitung: Prof. Dr. Klaus Brunnstein

Jörg Ziemann

Wintersemester 2002/2003

Inhaltsverzeichnis:

1 Einleitung

Im Wintersemester 2002/2003 wurde an der Universität Hamburg, Fachbereich Informatik/AGN das Projekt „Incident Response“ initiiert. Die 11 Teilnehmer dieses Projekts teilten sich in eine dreiköpfige Linux- und eine Windowsgruppe, in der die anderen acht Teilnehmer arbeiteten, auf. Weil ich zu der Windowsgruppe gehöre, wird der praktische Teil dieser Arbeit nur die von der Windowsgruppe geleistete Arbeit dokumentieren, während der Theorieteil auf beide Betriebssysteme anwendbar ist. Zu Beginn des Projektes wussten die meisten Teilnehmer wenig über die theoretischen Aspekte von Incident Response, weshalb die ersten Sitzungen unserer Gruppe dafür genutzt wurden, dieses Wissen zu erarbeiten. Dieses Wissen wird im zweiten Kapitel dokumentiert. Abschnitt drei beschreibt das Labor zur Untersuchung der Incidents, und Abschnitt vier beschreibt als Beispiel für einen Incident den Wurm Nimda, sowie die Möglichkeiten ihn in dem Labor zu untersuchen.

2 Incident Response

Die Wichtigkeit eines „Incident Response“ Vorgangs, bzw. einer „Antwort auf Vorfälle“ ist im Bereich der IT-Sicherheit offensichtlich: Einerseits werden die Werte, die von funktionierenden IuK-System abhängig sind, immer größer, andererseits steigt die Anzahl von Vorfällen die eine Beeinträchtigung der Sicherheit solcher Systeme darstellen. Banken, Regierungen und Krankenhäuser verlassen sich auf Computer und auf Internet gestützte Kommunikation, sie könnten durch einen Incident, sofern nicht angemessen auf ihn reagiert wird, immense und irreparable Schäden erleiden. Es gibt eine Vielzahl von Möglichkeiten solchen Vorfällen präventiv entgegenzuwirken, doch die immer schneller werdende Entwicklung von Angriffstechniken erhöht die Wahrscheinlichkeit, dass diese vorbeugenden Maßnahmen durchbrochen werden. Die Tatsache, dass die Anzahl der erfolgreichen Einbrüche in IuK-Systeme steigt, macht die Existenz eines Planes für so einen Fall zwingend notwendig: Einen Vorfall-Beantwortungsmechanismus bzw. eine Incident Response Funktion.

2.1 Definitionen

Um die Arbeit eines Incident Response Teams eingrenzen zu können, ist es nötig den Begriff „Incident“ zu definieren. In der folgenden Aufzählung erfolgt ein Überblick über die Definitionen von Fachautoren und wichtigen Institutionen aus dem Gebiet der IT-Sicherheit:

I. E.E. Schulz[1]: „A ‚computer security incident’ is an adverse event in a computing system or network caused by a failure of security mechanisms, or an attempted or threatened breech of these mechanisms”.
II. J.P.Wack[2]: “A computer security incident, for purposes of this guide, is any adverse event whereby some aspect of computer security could be threatened: loss of data confidentiality, disruption of data or system integrity, or disruption or denial of availability”. Er schreibt außerdem: „the definition of an incident may vary for each agency depending on many factors; however, the following categories and example are generally applicable:”, worauf folgende Liste von Schulz (1990) zitiert wird:
    1. Comprise of integrity, wenn z.B. ein Programm von einem Virus infiziert wird oder eine ernsthafte Verletzlichkeit eines Systems bemerkt wird;

[1] [SCHULTZ91]
[2] [WACK91]

2. Denial of Service, wenn z.B. durch einen Wurm die Netzwerk-Kapazität ausgelastet wird;
3. Misuse, z.B. wenn eine externe oder interne Person unautorisiert Nutzen aus einem Account zieht;
4. Damage, z.B. wenn ein Virus Daten zerstört;
5. Intrusions, z.B. wenn ein Eindringling die Sicherheitsschranken eines Systems durchbricht.

III. K.P. Kossakowski[1]: „Tritt eine Bedrohung ein, d.h. wird eine existierende Schwachstelle oder Sicherheitslücke tatsächlich ausgenutzt, stellt dies in einem konkreten Fall einen Angriff dar. Die Tatsache des Angriffs oder das Eintreten einer beliebigen anderen Bedrohung stellt einen Vorfall (Incident) dar". Kossakowski unterteilt Incidents außerdem in Deliberate/Malicious Incidents und Accidental Incidents, d.h. in Vorfälle bedingt durch mutwilligen Missbrauch und in nicht-intentionale Unfälle (die auch dem Bereich der Computer-Safety zugeordnet werden).

Der Begriff Incident Response wird von dem gleichen Autor folgendermaßen definiert: „unter den Begriff ‚Incident Response' (im engeren Sinne) fallen alle Aufgaben und Funktionen, die mit der Reaktion auf Vorfälle in einem konkreten technischen oder organisatorischen Zusammenhang stehen. IR umfasst von der Analyse eines Vorfalls über die verschiedenen Entscheidungsprozesse während der Bewältigung bis hin zur Beseitigung durch den Vorfall hervorgerufenen Veränderungen alle Aktivitäten, die ohne das Eintreten des jeweiligen Vorfalls nicht notwendig gewesen wären. Das gilt insbesondere für das Management dieser Aktivitäten und für alle Kommunikations- und Koordinationsaufgaben"[2].
Diese Definition korrespondiert mit der Graphik die im Vortrag des IR-Teams (WS2002/03) gezeigt wurde:

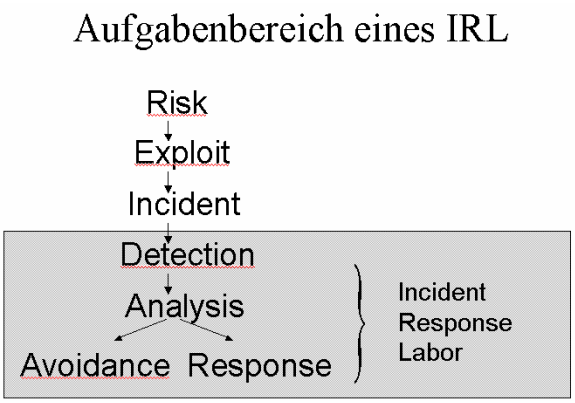


Abbildung: Aufgabenbereich eines IRL

Am Anfang der Kette steht ein Risiko (z.B. eine Schwachstelle in einem Serverprogramm), welches durch einen Exploit (z.B. eine Technik die eine Schwachstelle ausnutzt um unbefugt Zutritt zu einem Server zu bekommen) zu einem Vorfall wird: Ein Angreifer hat z.B. unbefugt Zutritt zu einem Server bekommen. Der Incident-Response Prozess beginnt also erst mit der Entdeckung („Detection") eines sicherheitsrelevanten Vorfalls. Nach der Entdeckung wird der Incident analysiert, daraufhin werden Schritte unternommen um den Schaden so weit es geht rückgängig zu machen (Response) sowie den wiederholten Auftritt dieser Incident-Art zu verhindern (Avoidance).

[1] aus [KOSSAKOWSKI00]
[2] [KOSSAKOWSKI00], Seite 13

Es gibt verschiedene Bezeichnungen für Gruppen die sich mit Incident Response beschäftigen:

- Computer Security Incident Response Capability (CSIRC),
- Computer Security Incident Response Center (CSRC),
- Computer Incident Response Team (CIRT) und
- Computer Emergency Response Team (CERT).
-

Während die ersten drei Begriffe synonym sind, bezieht sich der letzte, CERT, nicht auf reine IR-Arbeit, denn nicht jeder Incident muss gleich ein „Emergency" sein: „most of IR-Teams exist to assist users in a wide range of circumstances, some of which may be very routine, and a few of which may comprise genuine emergencies"[1]. Neben „soft-skills" wie Kommunikationsfähigkeiten und Belastbarkeit, müssen Mitarbeiter eines solchen Teams vor allem technisches Know-how haben[2]. Dieses sollte sowohl grundlegendes Wissen über Rechner- und Netzwerksicherheit, als auch Wissen über technische Details von Angriffen, z.B. Viren und Würmer, und häufigen System-Verletzlichkeiten, z.B. Betriebssystemschwächen, einschließen. Auf dieses technische Wissen bezieht sich der kommende Abschnitt.

2.2 Sicherheitsprobleme

2.2.1 Verwundbarkeiten: Sicherheitslücken von Betriebssystemen

Der Durschnittsbenutzer eines PC, der Windows 95/98 oder ME benutzt hat in der Standartinstallation seines Betriebssystems keine Netzwerkdienste installiert. Deswegen sollte man eigentlich annehmen, dass er sich nur in besonderen Fällen, z.B. falls er FTP- oder WWW-Server betreibt, Sicherheitslücken im Betriebssystem oder in den Netzwerkdiensten sorgen müsste. Aber auch beim normalen Surfen bietet Windows Einfallstore, die durch eine aufmerksame Konfiguration geschlossen werden müssen[3]. Andere, Unix-orientierte, Betriebssysteme wie Linux die hauptsächlich auf Servern eingesetzt werden, bieten in der Standartinstallation viele aktive Dienste, wie Telnet, FTP und httpd. Viele von diesen Diensten sind unsicher und können zu Incidents führen, weswegen sie geschlossen bzw. auf das notwendigste reduziert werden sollten. Zur Sicherheit dieser Betriebssysteme schreibt [NHackersGui, S.222] dass „ein durch adäquate Konfiguration sauber abgesichertes Open-Source-Betriebssystem eine extrem zuverlässige und sichere Alternative zu teuren kommerziellen Lösungen darstellen kann".

Die SANS-Gruppe gibt unter www.sans.org/topten.htm eine Liste der Sicherheitslücken heraus die am häufigsten in Computernetzwerken anzufinden sind. Zu dieser Liste gehören auch die CVE-Einträge[4], die die Sicherheitslücken detaillierter erklären. Um die Nutzung von Exploits zu verhindern, ist es nötig, sicherheitsrelevante Software stets mit den neuesten Patches zu aktualisieren.

[1] [SCHULTZ91]
[2] Zu den Kompetenzen die ein IR Team haben sollte, siehe auch [KOSSAKOWSKI00], Seite 279
[3] zur sicheren Konfiguration von Windows siehe auch [NTCSINET03]
[4] Common Vulnerabilities and Exposures, zu finden unter www.cve.mitre.org

2.2.2 Angriffe

Malware

Im Internet tummeln sich verschiedene Arten von Malware: Viren, Würmer und Trojaner können zu einem Vorfall führen der die Vertraulichkeit, Integrität, Authentizität, oder Erreichbarkeit von Computersystemen beeinträchtigt. Im folgenden werden diese Malware Typen kurz beschrieben.

Viren

F. Cohen definiert einen Virus als „ein Programm, das sich repliziert, indem es andere Programme infiziert, sodass diese eine (möglicherweise erweiterte) Kopie des Virus enthalten“[1]. Um Strafverfolgung zu entgehen versuchen die Autoren von Viren anonym zu bleiben, es wird aber angenommen dass sie aus der Gruppe der „männlichen, pubertierenden Jugendlichen zwischen 15 und 60 Jahren“ (Zitat K.Brunnstein) stammen. Dass heißt, sie versuchen Frust, begründet in mangelnder sozialer Anerkennung und Einbindung, durch aggressives, Aufmerksamkeit erregendes Verhalten zu kompensieren. Viren werden unter anderem nach der Häufigkeit ihres Auftretens klassifiziert: weit verbreitete Viren werden als „In-the-Wild“-Viren kategorisiert, bzw. „Ein Virus wird dann als In-the-Wild-Virus betrachtet, wenn er sich als Ergebnis normaler alltäglicher Vorgänge auf und zwischen den Computern nichts-ahnender Benutzer verbreitet. Das bedeutet, dass Viren, die einfach nur existieren, sich aber nicht ausbreiten, nicht als In-the-Wild-Viren betrachtet werden“[2]. Nach ihrem Verhalten werden Viren in weitere Kategorien unterteilt, die im Folgenden kurz beschrieben werden:

- **Bootsektor Viren**: infizieren den Master-Boot-Record und/oder das DOS-Boot-Record von Disketten. Früher war diese Art von Viren sehr verbreitet, aber weil Disketten heute nur noch wenig benutzt werden, ist sie heute kaum noch In-the-Wild anzutreffen
- **Dateiviren**: infizieren ausführbare Dateien. Zu beachten ist dass unter Windows nicht nur .com und .exe, sondern auch .dll-Dateien, bestimmte Treiber, einige Screensaver und sogar Font-Dateien zu den ausführbaren Dateien zählen.
- **Makroviren**: Es gibt umfangreichere Anwendungsprogramme mit denen nicht nur passive Daten bearbeitet werden können, sondern zusätzlich auch ausführbarer Code erstellt werden kann. Eine weitverbreitete Makrosprache ist Visual Basic (for Applications), das im MS-Office Paket verwendet wird. Ein Virus der in einer Makrosprache geschrieben ist, wird beim Öffnen des Dokuments zu dem das Makro gehört aktiviert. Die meisten Makrosprachen beinhalten nicht nur Befehle um Daten des zugehörigen Anwendungsprogramms zu manipulieren, sondern auch Befehle um Betriebssystem-Funktionen zu aktivieren, womit der Virus auch auf Systemkomponenten zugreifen kann.
- **Skriptviren**: Dieser Begriff bezieht sich normalerweise auf Viren die Code von Skriptsprachen wie VBScript und Jscript befallen. Diese Skripte sind in HTML eingebettet, und werden von E-Mail-Clients über den Windows Scripting Host ausgeführt.

Alle genannten Viren Kategorien können sich folgender Techniken bedienen:

- **Polymorphie**: um seine Erkennung durch einen Virenscanner zu erschweren verschlüsselt oder verändert der Virus sich bei neuen Infektionen. Dazu kann er

[1] aus [NHackersGui], Seite 401, das wiederum F.Cohen: „A short Course on Computer Viruses“ zitiert
[2] [NHackersGui], Seite 408, zitiert Paul Ducklin, „Counting Viruses“, Virus Bulletin 1999

entweder redundante Befehle in seinen Code einfügen, oder er ändert die Reihenfolge seiner Code Module.
- **Stealth**: Wenn ein vom Virus befallenes Programm ausgeführt oder der Virus auf andere Weise aktiviert wurde, entfernt der Virus seinen Code aus befallen Programmen. Er bleibt als nur im Speicher aktiv und verhindert so seine Entdeckung durch einen erfolgreichen Virenscan auf Dateien.

Würmer

Würmer hängen sich im Gegensatz zu Viren nicht an Wirtsdateien, sondern sie replizieren sich, d.h. sie erstellen Kopien von sich selbst. Diese Kopien werden dann, oft über ein Netzwerk, verteilt. In den neu infizierten Rechnern sind sie nun im Arbeitsspeicher vorhanden, und können mit anderen Instanzen des Wurms kommunizieren, für eine weitere Replikation oder die Ausführung eventueller Schadfunktionen sorgen. Man kann Würmer anhand ihres Transportmechanismus klassifizieren[1]:

- E-Mail-Würmer breiten sich über E-Mail Nachrichten aus. Aktiviert werden sie durch die Ausführung eines E-Mail-Attachments, oder durch Skripte (z.B. java-skript) die in einer E-Mail enthalten sind und ohne Zutun des Benutzers aktiv werden können.
- „Willkürliche Protokollwürmer“ breiten sich über Protokolle aus die nicht auf E-Mail basieren.

Nicht alle Würmer lassen sich genau einer Klasse zuordnen. Der Nimda-Wurm zum Beispiel bedient sich beider Techniken: Er verschickt sich sowohl per E-Mail als auch über das Netzwerk, und gehört somit zu beiden Klassen.
Eine andere Art der Klassifizierung unterscheidet Würmer nach ihrem Startmechanismus:

- Selbst ausführende Würmer nutzen Sicherheitslöcher in der Hostumgebung aus, und müssen nicht vom Anwender aktiviert werden.
- Vom Benutzer ausgeführte Würmer interagieren mit ihm, d.h. der Wurm muss den Benutzer irgendwie dazu bringen einen Anhang zu öffnen oder den Wurm durch andere Handlungen zu aktivieren.

Auch hier fällt Nimda in die Kategorie eines Hybrids zwischen den Klassen: der Wurm arbeitet sowohl selbst ausführend, in dem er sich über offene Netzwerkshares kopiert, als auch benutzerabhängig. Letzteres ist der Fall, wenn ein Benutzer erst eine infizierte Homepage besuchen muss um Nimda zu aktivieren, oder den Wurm durch das Öffnen eines Attachments (readme.exe) öffnet.

Um sich gegen Malware Angriffe zu schützen sind folgende Maßnahmen zu empfehlen:

- Software-Schwachstellen beseitigen: Patchen oder neue Versionen installieren
- Nur so viele Systemdienste wie nötig betreiben, unsichere Dienste deaktivieren
- Intrusion Detection Systeme installieren
- On-Access Virus Scanner für E-Mails installieren
- Virus-Warnungen dürfen nur von autorisierten Stellen gegeben werden, damit falsche Reaktionen auf übertriebene Meldungen oder auf Hoaxes verhindert werden
- E-Mail Anhänge nur öffnen, wenn der Absender bekannt ist und das Attachement erwartet wurde (einige Würmer benutzen die E-Mail Accounts von Opfern, so dass die Kenntnis des Absenders kein Garant für ein sicheres Attachement ist)

[1] [NHackersGui], Seite 408, führt hierzu Carey Nachenbergs Klassifikation an, enthalten in seinem Beitrag zur Virus Bulletin Conference 1999

- Häufiges erstellen von Backups, damit von Viren befallene Dateien gelöscht und durch nicht infizierte Originale ersetzt werden können

Trojaner bzw. trojanische Pferde sind Programme, die neben ihrer eigentlichen, dem Anwender bekannten, Funktion noch weitere Funktionen ausführen, von denen der Anwender nicht weiß. Ein bekanntes Beispiel für einen Trojaner ist „Back Orifice". Wenn dieses Programm auf einem Client installiert wurde, kann ein anderer Rechner (der Server) auf dem Client Rechner Funktionen ausführen ohne dass der Client davon Kenntnis hat. Unter anderem kann der Angreifer Dateien modifizieren.

Weitere Angriffsarten

Neben den automatisch ablaufenden Angriffen, wie durch Viren und Würmer, gibt es auch Angriffe die manuell ausgeführt werden. Ebenso wie automatische Angriffe können diese Vorfälle auslösen welche unter anderem die Integrität und Vertraulichkeit eines Systems beeinträchtigen. Hier einige Beispiele für Systembeeinträchtigungen durch nicht-automatische Angriffe:

- Angriff auf die **Verfügbarkeit**: wird mit einem Denial-of-Service Angriff ausgeführt. Ein bekannter Denial-of-Service Angriff ist der „Ping-of-Death", bei dem eine Schwäche von älteren IP-Implementationen ausgenutzt wird: Es werden Fragmente eine IP-Paketes generiert die die erlaubte Größe von 64 Kbyte überschreiten. Das führt zu einem Bufferoverflow bei dem empfangendem System, und anschließend zu einem Systemabsturz.
- Angriff auf die **Vertraulichkeit**: z.B. durch Sniffing, d.h. dem heimlichen Abhören von Daten die über ein Netzwerk gesendet werden.
- Angriff auf die **Authentizität**: z.B. durch Spoofing. Dabei gibt ein Angreifer eine falsche Identität an, um so Rechte zu erlangen die einer dritten Person gehören. Es gibt verschiedene Art Spoofing zu realisieren: Auf IP-Ebene kann eine falsche Absenderadresse in ein IP-Paket eingetragen werden, und auch auf TCP-Ebene ist Spoofing möglich (TCP-Sequenznummern-Angriff, siehe [SIVERSYS00], Seite 33).
- Angriff auf die **Integrität**: Kann durch einen Man-in-the-middle Angriff realisiert werden. Dabei klingt sich ein Angreifer in einen Kommunikationsvorgang ein, so dass die ursprünglichen Kommunikationspartner glauben sie würden direkt miteinander kommunizieren, die Daten aber in Wahrheit über einen dritten (den „man-in-the-middle") laufen. Dieser dritte Mann kann die abgefangenen Daten modifizieren bzw. ihre Integrität verletzen, und sie anschließend an den eigentlichen Empfänger weiterleiten[1] ohne dass die Kommunikationspartner dies bemerken.

Personen die solche Angriffe ausführen benutzen dabei oft sogenannte Cracker Tools[2], wie Scanner, Netzwerksniffer, Passwortcracker oder Exploits. Exploits sind Programme die von Crackern ausgeführt werden um Sicherheitslücken eines speziellen Programms ausnutzen[3]. Diese lassen sich unterteilen in lokale Exploits, die nur über einen direkten Zugang zur Shell zu nutzen sind, und in Remote Exploits, die auch über ein Netzwerk nutzbar sind.

[1] siehe [SIVERSYS00], Seite 33
[2] siehe auch [SIVERSYS00], Seite 178
[3] siehe auch [NHackersGui], Seite 122

2.3 Die Aufgaben eines CIRTs

In Abschnitt 2.1 wurden Detection, Analysis, Response und Avoidance als Funktionen eines Incident Response Labors genannt. Andere Organisationen beschreiben die Funktionen von Incident Response Gruppen ähnlich, so zählt Allaire in einem Artikel[1] über Incident Response die Grundfunktionen „Discovery and initial response", „Containement and elimination" und „proactive measures" auf. Des weiteren wird dort der Punkt „Initial Preparation and organization" aufgeführt, der im folgenden unter „Vorbereitung und Organisation" beschrieben wird.

2.3.1 Vorbereitung und Organisation

Es gibt ein Reihe von Maßnahmen welche die Schadensbehebung oder zumindest Schadensbegrenzung im Falle des Auftretens eines Incidents erheblich erleichtern. Dazu gehören technische Maßnahmen, wie das häufige Erstellen von Sicherheitskopien wichtiger Daten, und organisatorische Maßnahmen die eine effiziente Incident-Response Arbeit ermöglichen sollen. Zu den organisatorische Maßnahem zählt auch die Verteilung bzw. Zuweisung von Verantwortlichkeiten, damit im Falle eines Incidents eine klare Rollenverteilung gegeben ist und Entscheidungen schnell getroffen werden können. Allaire nennt folgende Bereiche („Areas of Definition") eines Betriebes, die für eine effektive Incident Response Arbeit voneinander abzugrenzen, bzw. zu evaluieren sind:

- **Technischer Bereich**. In diesen fallen Verantwortlichkeiten für das Netzwerk, Netzwerk-Software, Telefon-Infrastruktur etc..
- **Betriebswirtschaftlicher Bereich**. Es muss sichergestellt werden dass der CIO über die Wichtigkeit der IT-Sicherheit informiert ist. Das Management muss grundlegende Begriffe zur Unternehmens-(Daten-)Sicherheit kennen, und festlegen wer im Falle eines Incidents wichtige Entscheidungen zu Fällen hat.
- **Anwender**. Diese sollten über wichtige Elemente der Sicherheitspolitik informiert sein, und wissen dass bestimmte Handlungen (z.B. das Öffnen von Attachments unbekannter Absender) ein hohes Risiko für die IT-Sicherheit darstellen. Je besser sie informiert sind, desto eher können sie unnormales Verhalten registrieren und dafür sorgen dass Incidents schneller bemerkt werden.
- **Law enforcement.** Allaire empfiehlt Kontakte zu Behörden zu knüpfen („make a quick phone call to introduce yourself"), weil dadurch im Bedarfsfall die Zusammenarbeit mit diesen schneller und besser sei.

Ein Ansatz für die interne Organisation von Incident Response Unternehmen wird von den Autoren von [CSIRT][2] vorgeschlagen. Nach diesem muss ein solches Unternehmen aus mindestens drei Kernpunkten bestehen: dem Reporting Point, in dem eingehende Kunden-Meldungen entgegengenommen werden, einer Analyse Stelle zum Verifizieren von Meldungen und deren technischem Verständnis, und einer Instanz zur Publizierung der gewonnenen Erkenntnisse. Die Punkte werden in der folgenden Tabelle detaillierter beschrieben:

[1] [ALLAIRE01]
[2] [CSIRT], S.58

Reporting Point	- Deal with incoming reort that affect the constituency and pass them on (as appropriate) to the sites affected within the constituency - Deal with reports from the constituency that affect sites and CSIRTs external to the constituency, and pass them on accordingly - Both of the above
Analysis	- Examine log-files - Identify affected sites - Point to technical documents and advisories - Provide technical support - Provide workaround and fixes - Provide on-site assistance
Notification	- Point to resources that provide or can help establish appropiate points of contact - Provide a list of appropiate points of contact - Undertake contact of other parties affected in the incident - Undertake contact of other parties and law enforcement

Tabelle: Mögliche Aufgabentrennung eines IR Unternehmens[1]

Im Bereich der technischen Vorbereitung sind folgende Punkte wichtig:

- Netzwerkdiagramme sollten aktuell und gut auswertbar sein. Bereits bei kleinen Netzwerken ist es nötig Informationen über Netzwerkvorgänge zu visualisieren, um damit auf unnormales Verhalten schließen zu können.
- Es sind Informationen über die neuesten Sicherheits-Bedrohungen einzuholen. Als mögliche Quellen kommen BUGTRAQ (http://www.securityfocus.com/) oder der SANS Security Alert Consensus service (http://www.sans.org/nwcnews/) in Frage.
- Die Installation von Logging Mechanismen. Dabei muss sichergestellt sein dass die Log-Files sicher, d.h. nicht zu manipulieren sind, und regelmäßig analysiert werden.
- Die Einrichtung von Backup Mechanismen. Diese müssen dafür sorgen dass Backups häufig erstellt werden. Außerdem sollten die Log-Files auf Write-once Medien, wie WORM-drives, gesichert werden um im Falle eines Rechtstreits besser abgesichert zu sein.

2.3.2 Entdeckung und erste Reaktionen

Es gibt verschieden Möglichkeiten auf einen Incident aufmerksam zu werden: Intrusion detection Systeme können für eine automatische Meldung sorgen, System-Adminstratoren können auf Unnormalitäten in Log-Files oder beim Server-Betrieb stoßen, und Anwender können z.B. Mängel in der Funktionalität ihrer Systeme bemerken.

Nach der ersten Registrierung, muss der Schaden dann näher untersucht werden. Teile der Analyse wurde teilweise oben in der Tabelle genannt, nämlich die Log-File Analyse und das Identifizieren von betroffenen Stellen. Bei allen Schäden müssen die möglichen Folgen in Betracht gezogen werden. Wenn z.B. ein Angreifer auf Administrator-Ebene in einen NT Server eingedrungen ist, muss davon ausgegangen werden dass er die zugehörigen Passwörter eingesehen hat. Generell muss nach ungewöhnlichen (auch versteckten) und veränderten Dateien geschaut werden. Dabei helfen Programme wie Tripwire (http://www.tripwirde.com/)

[1] aus [CSIRTS], S.58

und AIDE (http://www.cs.tut.fi/~rammer/aide.html), die die Integrität von Daten überprüfen. Des weiteren sollten Prozesse die auf betroffenen Systemen laufen untersucht werden. Eine besondere Gefahr sind Hintertüren, die der Angreifer im System hinterlässt um so spätere Angriffe vorzubereiten. Schließlich sind die User accounts und Passwörter zu überprüfen, häufig legen Angreifer neue Nutzer an um später wieder in das System einbrechen zu können.

Um die Relevanz einer Incident-Meldung zu klären, kann man Incidents in Kategorien einordnen. Anhand der Schadenskategorie in die der Incident fällt, können dann entsprechend intensive Response-Maßnahmen getroffen werden. SANS beschreibt folgende Kategorien:

Criticality	Definition	Examples
High	Incidents that *have a monumental* impact on the organization's business or service to customers.	• Malicious code attacks, including Trojan horse programs and virus infestations
		• Unauthorized system access
Medium	Incidents that *has a significant* or *has the potential to have a monumental* impact on the organization's business or service to customers.	• Password cracking attempts
		• Password does not allow access to system, apparent change of password without user knowledge has occurred
Low	Incidents that *has the potential to have a significant or monumental* impact on the organization's business or service to customers.	• Probes and network mapping
		• Denial of access to the system due to unexpected lockout

Tabelle: Einordnung von Incidents[1]

Weil Organisationen verschiedene Prioritäten hinsichtlich der einzelnen Dimensionen der IT-Sicherheit haben, ist es schwierig allgemeingültige Kategorien aufzustellen. Einige Organisationen legen den größten Wert auf die Vertraulichkeit von Daten, andere finden Integrität am wichtigsten, und würden dementsprechend einen Incident, der die Integrität bedroht, als schwerwiegender Einstufen als die erstgenannten Organisationen.

Das CERT/CC bekommt sehr viele Meldungen über Incidents, und bearbeitet nur die seiner Meinung nach wichtigsten Meldungen: "Due to limited resources and the growing number of incident reports, we may not be able to respond to every incident reported to us. We must

[1] [OSBORNE01], veröffentlicht auf SANS Webseiten

prioritize our responses to have the greatest impact on the Internet community". Die folgende Liste[1] beschreibt Meldungen denen CERT die höchste Priorität gibt und die als "emergencies" anerkannt werden:

- possible life-threatening activity
- attacks on the Internet infrastructure, such as:
  - root name servers
  - domain name servers
  - major archive sites
  - network access points (NAPs)
- widespread automated attacks against Internet sites
- new types of attacks or new vulnerabilities

2.3.3 Schadensbegrenzung und Eliminierung

Das Whitepaper[2] von Allaire beschreibt zur Schadensbegrenzung und Eliminierung fünf Punkte, die im folgenden erläutert werden.

Betroffene Systeme vom Netzwerk nehmen

Nachdem ein Incident festgestellt wurde, gibt es zwei Möglichkeiten weiterzuarbeiten:

1. Das betroffene System vom Netzwerk zu nehmen bzw. außer Funktion zu setzen
2. Die Untersuchung und Bereinigung am aktiven System vorzunehmen

Die Entscheidung hängt von der Wichtigkeit des laufenden Systems ab, bzw. wie viel Schaden durch einen System-Stillstand entstehen würde, und von der akuten Gefahr die von dem Incident ausgeht. Im optimal Fall gibt es ein Backup System, so dass das betroffene System ersetzt werden kann bis der Schaden behoben ist. Obwohl in den meisten Fällen schnell festgestellt werden kann durch welche Sicherheitslücke ein Incident entstehen konnte, ist es doch meist ratsam das angegriffene System aus dem Betrieb zu ziehen: Auch wenn die verantwortliche Lücke schnell geschlossen und Passwörter ausgewechselt werden, können trotzdem noch Hintertüren existieren die eine Aufrechterhaltung des Betriebs zu einem unkalkulierbaren Risiko machen.

Sicherungskopie von betroffenen Daten erstellen

Ein Backup von dem betroffenem System ist aus drei Gründen wichtig:

1. Um Originaldaten zu retten, von denen sonst keine Kopie existiert.
2. Für die Analyse des Incidents. Es ist erforderlich den Systemzustand möglichst schnell nach bekannt werden des Incidents untersuchen zu können, was durch ein frühes Backup ermöglicht wird.
3. Um bei späteren rechtlichen Vorgängen Beweise vorlegen zu können.

Den Eintrittsort des Incidents finden und die Verwundbarkeit eliminieren

Die Wichtigkeit dieses Schrittes ist wohl offensichtlich: Wenn nicht bekannt ist über welche Schwachstelle der Incident eindringen konnte, kann diese Stelle nicht versiegelt werden, und das System kann immer wieder von der gleichen Incident Art angegriffen werden. Die meisten Angriffe im Internet greifen über bekannte Schwachstellen oder schlechte Konfiguration von weitverbreiteter Software an, „rare are the custom, site-specific attacks that

[1] aus [CERT]
[2] [ALLAIRE01]

target a proprietary application – although these certainly occur“[1]. Aus diesem Grund findet man bei den meisten Angriffen einen von den Software Herstellern angebotenen Patch, der die Verwundbarkeit beseitigt.

Durchführen von Vulnerability Scans

Abgesehen von der Eliminierung der Vulnerability, durch die der aktuelle Incident entstanden ist, sollte das System auf weitere Schwachstellen untersucht werden. Denn häufig bleibt es nicht bei einem Angriff: Wenn Hacker erst einmal auf ein angreifbares Opfer aufmerksam geworden sind, versuchen sie, oder andere aus dieser Szene die von dem Einbruch gehört haben, oft auf anderen Wegen in das gleiche System einzudringen. Diesen Folge-Incidents sollte man vorbeugen, indem man sowohl betroffene als auch benachbarte Systeme einer Sicherheitsprüfung unterzieht. Tools für Security Scans findet man zum Beispiel unter http://www.isss.net/ (ISS´ Internet Security Scanner) und http://www.nai.com/ (Network Associeates`CyberCop Scanner).

Das System wiederherstellen

Aus schon genannten Gründen (Hintertüren, Passwörter u.a.) ist es schwer ein erfolgreich angegriffenes System durch Löschen und Reparieren einiger Dateien wiederherzustellen. Selbst wenn man ein Programm wie Tripwire installiert hat, das hilft modifizierte Dateien zu finden, sind die anderen Probleme damit nicht gelöst. Eine bessere Möglichkeit ein System zu regenerieren, ist ein komplettes Backup zu verwenden. Dabei muss allerdings sichergestellt sein, dass das System zu dem Zeitpunkt der Erstellung des Backups wirklich noch nicht kontaminiert war. Und nach dem Aufspielen sollte man nicht vergessen, die Vulnerabilities zu schließen, die zu dem Incident geführt haben.

Das System wieder in Betrieb nehmen

Nachdem die vorigen Schritte ausgeführt worden sind, d.h. das dass System gespeichert, gesäubert, regeneriert, gepatcht, sowie auf ihm laufende Services sicher konfiguriert worden sind, kann das System wieder in Betrieb genommen werden.

3 Aufbau des Labors

Wie schon in der Einleitung erwähnt, teilt sich das IR-Team in zwei Gruppen auf, die zwei verschiedene Ansätze von Incident Response verfolgen. Die Windows-Gruppe bedient sich der dynamischen Analyse, und die Linux-Gruppe der statisch komparativen. Dynamische Analyse heißt dass Vorgänge im System fortlaufend analysiert werden, und so die Aktivität des Incidents Schritt für Schritt mitverfolgt werden kann. Dafür bieten sich verschiedene Software-Tools an: Monitore die Vorgänge im Dateisystem und Prozesse beobachten, sowie Sniffer zur Netzwerkanalyse. Die statische komparative Analyse hingegen zwei beobachtet ein System nicht ununterbrochen, sondern vergleicht nur zwei Systemzustände, nämlich den vor dem Eintritt eines Incidents, und die Situation des Systems im angegriffenen Zustand. Diese wird auch Prä-mortem/Post-mortem Analyse genannt. Für diese Analyse Art kann man zum Beispiel Tools benutzen die die Integrität von Dateien überprüfen, z.B. das Programm Tripwire.

[1] [ALLAIRE01] , Seite 11

Das Netzwerk

Für die Arbeit im Incident Response Labor wurde ein Netzwerk eingerichtet dass aus drei Rechnern besteht, die über ein Hub miteinander verbunden sind. In der unten stehenden Abbildung ist zusätzlich noch ein Server zu sehen, dieser wurde aber im Wintersemester 02/03 noch nicht an das Netz angeschlossen, weil für unsere Versuche kein Server benötigt wurde.

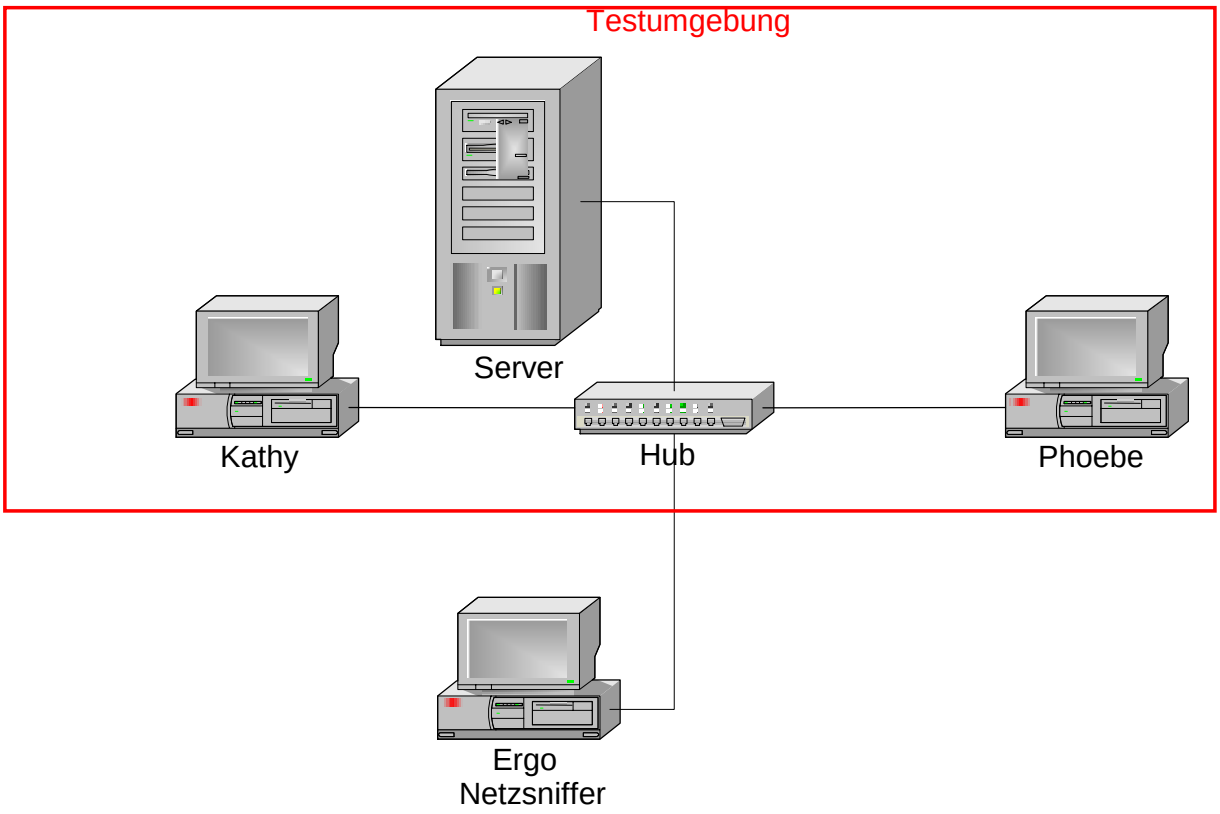


Abbildung: Die allgemeine Testumgebung im IR Labor

Mit diesem relativ kleinen Netzwerk lassen sich bereits viele in der Incident-Response Arbeit häufig vorkommende Konstellationen nachstellen. Dabei kann z.B. der Rechner Phoebe die Rolle eines Angreifers übernehmen, der entweder einen anderen Client, z.B. Kathy, oder auch den Server angreift. Der Rechner Ergo würde in diesem Szenario die Rolle des Beobachters übernehmen, indem er durch das Netz fließende Daten ausliest. Je nach Szenario wird passende Software ausgewählt, so können z.B. auf dem Opfer-PC ein Betriebssystem dass Schwachstellen aufweist sowie Monitor-Programme zur Auswertung des Angriffes laufen, während auf Ergo ein Sniffer-Programm läuft. Im Wintersemester 02/03 kamen als Sniffer auf Ergo das Programm Ethereal zum Einsatz, zusätzlich lief auf diesem Rechner ein Virenscanner (NAV) um zu gewährleisten dass dieser Rechner seiner Beobachterrolle gerecht werden konnte und nicht selber zum Opfer würde. Auf den beiden Clients, Phoebe und Kathy, liefen jeweils der Prozessmonitor PMON, der Filemonitor FMON sowie der Registrymonitor regMon.

4 Der Nimda-Wurm als Beispiel Incident

Der Nimda-Wurm wurde am 18.9.2001 das erste Mal registriert, und der Netzverkehr den er erzeuge war immens, so dass viele Seiten keine Dienste mehre Anbieten konnten. Er befällt die Betriebssysteme MS Windows 98/NT/2000/ME, die den Internet Information Server (IIS) 5.01 und 5.5 benutzen. Der Wurm selbst besteht aus einer ca. 57 Kbyte langen Datei, und wurde in Microsoft C++ geschrieben[1]. Bei seinen Angriffen nutzt er sowohl Softwareschwächen, z.B. in den genannten (ungepatchten) IIS Versionen, wie auch die Naivität von Anwendern aus, die sich z.B. durch das Öffnen von Anhänge E-Mails unbekannter Herkunft äußert. Die Wege die er zu seiner Verteilung nutzt sind vielfältig: es sind 19 verschiedene Mechanismen bekannt mit denen Nimda Systeme infiziert. Seine Fähigkeit von infizierten Web-Servern auf Clients überzuspringen, d.h. Rechner von Personen die im WWW surfen zu befallen, machen ihn besonders gefährlich. Infizierte Systeme sind schwer von dem Wurm zu befreien, weil der Wurm viele unterschiedliche System Modifikationen vornimmt. Der Name wurde dem Wurm in den Kaspersky-Labs gegeben, und führt auf den rückwärts gelesenen Begriff „Admin“ für Administrator zurück. Diese Wahl ist wiederum damit zu erklären dass der Wurm sich unter anderem über eine Datei namens ADMIN.DLL multipliziert.
Seine Komplexität, die gute Dokumentation die über diesen Wurm vorhanden ist sowie sein recht junges Alter, ließen den Nimda-Wurm als Testfall für das IR-Labor geeignet erscheinen.

Der Wurm ist unter folgenden Namen bzw. Aliasen[2] bekannt:
I-Worm.Nimda (AVP), I-Worm.Nimda.E (AVP), Nimda (F-Secure), Nimda.c (F-Secure), Nimda.d (F-Secure), Nimda.e (F-Secure), W32.Nimda.A@mm (NAV), W32.Nimda.C@mm (NAV), W32.Nimda.D@mm (NAV), W32.Nimda.E@mm (NAV), W32/Minda@MM, W32/Nimda-C (Sophos), W32/Nimda.a@MM, W32/Nimda.eml, W32/Nimda.htm, W32/Nimda@MM, Win32.Nimda.A@mm (AVX), Win32.Nimda.E (CA).

4.1 Spreading und Vulnerabilites

Der Nimda-Wurm hat vier unterschiedliche Verbreitungsmechanismen:

1. Über Webserver

   Der Wurm sucht das Internet nach den Webservern ab und versucht eine Reihe von bekannten Windows-Verwundbarkeitsstellen auszunutzen, um die Kontrolle über den Server zu erhalten. Er scannt dabei nach Vulnerabilites, wie z.B. der Unicode Directory Traversal Vulnerability, und nach Hintertüren die von dem Code Red II Wurm zurückgelassen wurden. Sobald der Wurm in einen Server eingedrungen ist, benutzt er TFTP um sich von dem angreifenden in den Opfer Rechner zu kopieren. Die Datei die kopiert wird trägt den Namen „Admin.dll“. Für diese Verbreitungsmechanismus sind IIS 3.0, 4.0, and 5.0 sowie der Personal Web Server (PWS) 1.0 und 3.0 anfällig.

2. Per E-Mail

   Der Wurm sammelt E-Mail-Adressen aus Windows Adressbuch, Eingangs- und Ausgangspostfach der Benutzer, lokalen HTML/HTM-Dateien und sendet sich an alle

[1] vergleiche [VIRUSLIST]
[2] aus [NAI]

gefundenen Adressen als Anhang readme.exe. Aufgrund der „Automatic Execution of Embedded MIME Types“ Vulnerability wird dieser Anhang von betroffenen Systemen selbst im „Preview“-Modus ausgeführt.

3. Über HTTP-Seiten

Wenn der Wurm einen Server erfolgreich infiziert hat, versucht er mittels HTTP-Service die Klienten, die auf den Serverseiten navigieren, anzustecken. Nach dem Infizieren eines Servers erzeugt Nimda Kopien von sich selbst im MIME-Format (README.EML) und durchsucht Verzeichnisse nach Web-Dateien, bzw. nach Dateien die auf .HTML, .HTM oder .ASP enden. An all diese Dateien hängt er JavaScript Code. Sobald ein Anwender so eine Datei über einen Browser öffnet (und eine anfällige Internet Explorer Version benutzt), bewirkt dieser Code die automatische Ausführung des README.EML-Files, und eine Aktivierung des Wurms im System des Anwenders.

4. Über freigegebene Laufwerke

Der Wurm legt eine Kopie von sich selbst auf alle Laufwerke, auch im Netzwerk, für die der Benutzer eine Schreibberechtigung hat. Dabei erzeugt er seine Kopien sowohl im .EML-Format (in 95% der Fälle) als auch im .NWS-Format (in 5% der Fälle). Des weiteren sucht er nach Web-Dateien (.HTML, .ASP etc.) und hängt an dessen Code ein Javascript Programm, das den Wurm bzw. die README.EML Datei ausführt, sobald die Web-Datei geöffnet wird. Außerdem durchsucht er alle Verzeichnisse nach exe-Files und hängt sich an diese an.

Aus den folgenden beiden Abbildungen geht hervor dass mit Abstand die meisten Nimda Angriffe von den USA ausgingen, während die meisten angegriffenen Rechner in England lagen. Dies könnte mit Nimda´s Funktion zur Adress-Auswahl von Opfer-Rechnern zusammenhängen.

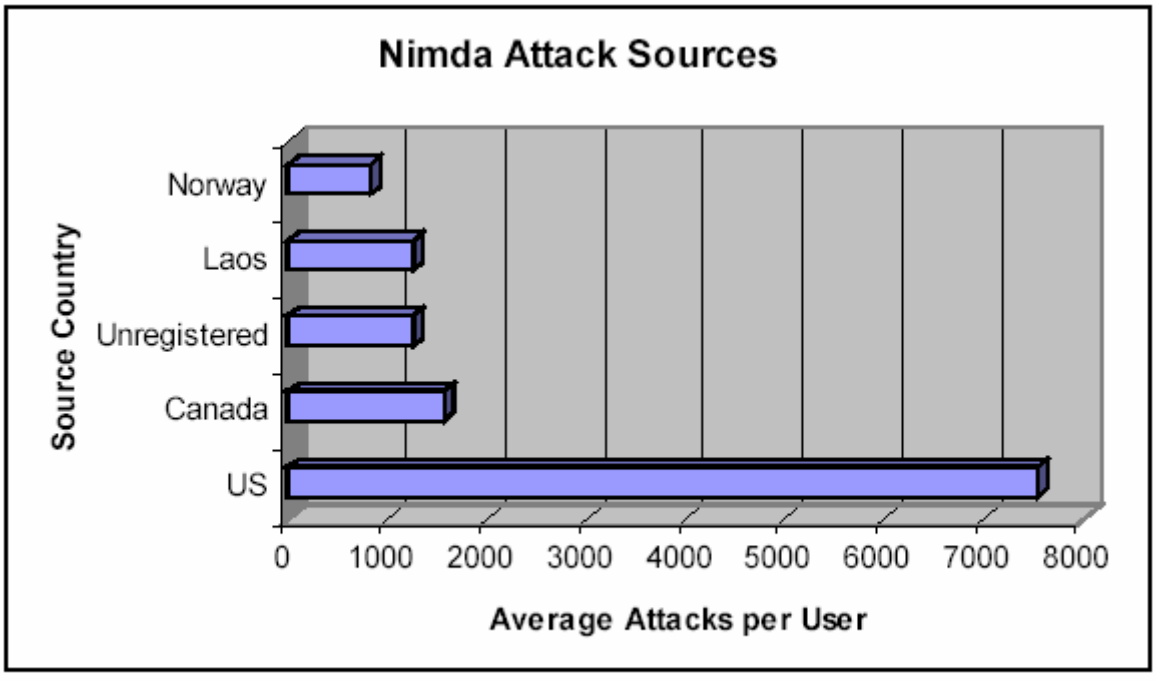


Figure 7— Top Five Nimda Source Countries on September 18

Abbildung: Nimda Quell-Länder am 18. September 2001. Quelle: [ARIS01]

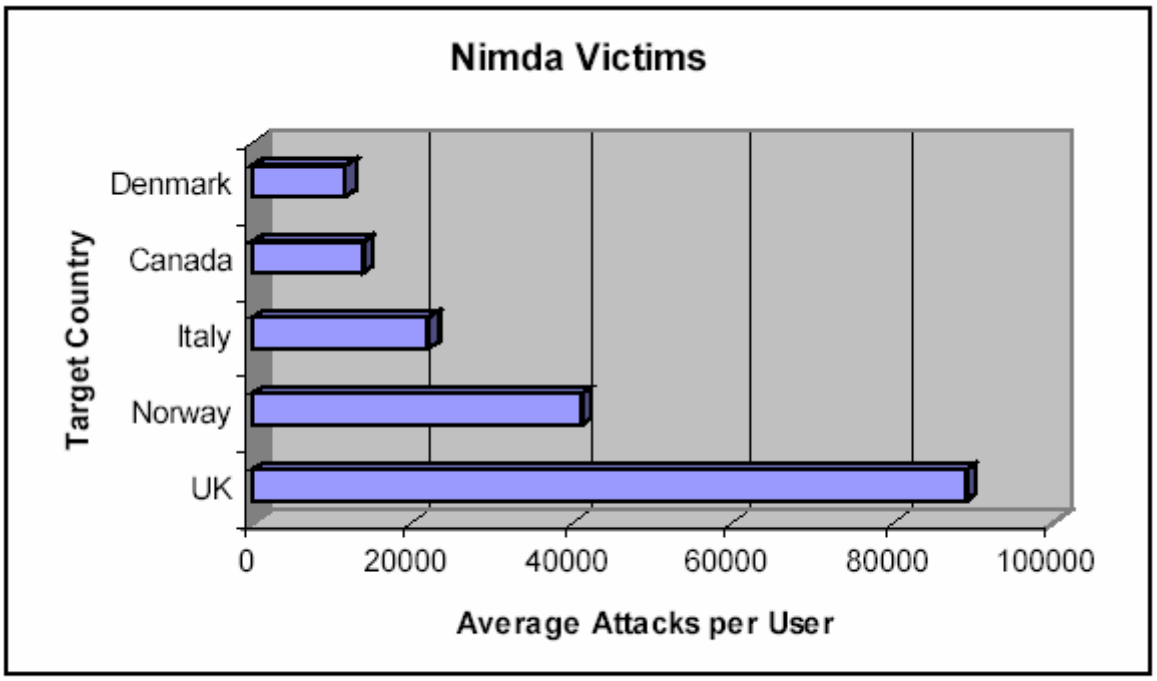


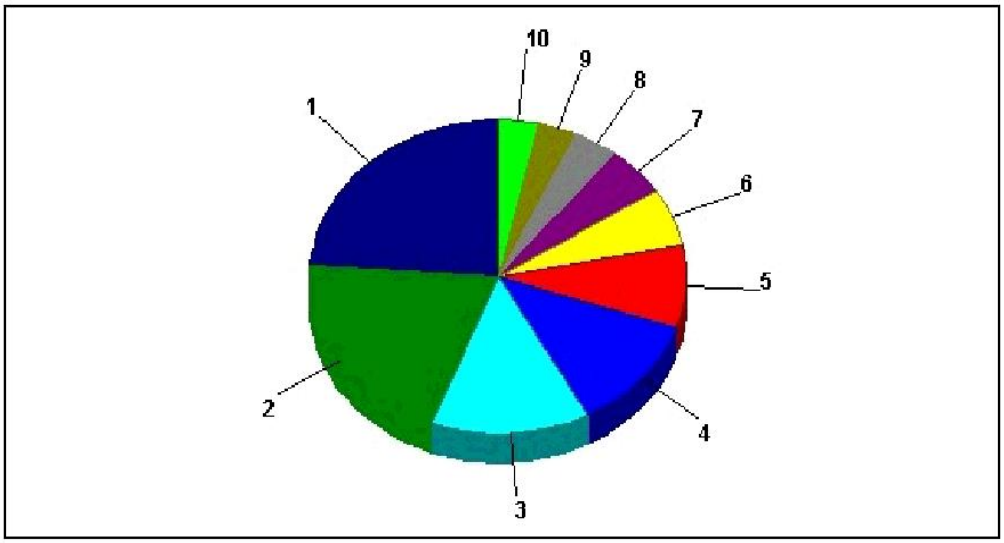
Figure 8 — Top Five Nimda Target Countries on September 18

Abbildung: Nimda Ziel-Länder am 18. September 2001. Quelle: [ARIS01]

Bei seiner Verbreitung benutzt der Nimda-Wurm 19 verschieden Systemschwächen um sich auszubreiten, die wichtigsten davon sind:

- Microsoft IIS/PWS Escaped Characters Decoding Command Execution Vulnerability
- Microsoft IE MIME Header Attachment Execution Vulnerability
- Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability
- Microsoft Office 2000 DLL Execution Vulnerability
- Ausserdem: CodeRed II-Hintertür

Weil diese Vulnerabilites an anderen Stellen, z.B. den Microsoft Security Bulletins, ausführlich beschrieben sind, wird hier auf eine Erläuterung dieser Systemspezifischen Schwachstellen verzichtet. Die folgende Abbildung zeigt aber die Relevanz dieser Vulnerabilities sowie ihren Zusammenhang mit der Ausbreitung des Nimda-Wurms: Aus der Abbildung geht hervor dass die genannten Vulnerabilites, gelistet von Position 3 bis 7, am Tag von Nimda´s erstem Auftreten sehr häufig ausgenutzt wurden. Die Vulnerability die am häufigsten genutzt wurden, IIS ISAPI Buffer Overflow, stammt jedoch nicht von der Nimda-Aktivität, sondern von dem Code Red II Wurm.



	Attacks	# Users	% Users	# Attacks
1	Microsoft Indexing Server/Indexing Services ISAPI Buffer Overflow Attack	56	48.28	314611
2	Generic HTTP 'cmd.exe' Request Attack	48	41.38	210428
3	Generic "../" Directory Traversal Attack	32	27.59	514585
4	Microsoft IIS/PWS Escaped Characters Decoding Command Execution Attack	28	24.14	202070
5	Microsoft IIS 4.0 / 5.0 Extended UNICODE Directory Traversal Attack	20	17.24	128816
6	Microsoft IIS 4.0/5.0 File Permission Canonicalization Attack	14	12.07	29190
7	Generic HTTP Directory Traversal Attack	12	10.34	22856
8	Matt Wright FormMail Attacks	9	7.76	136
9	NCSA ScriptAlias CGI Source Disclosure Attack	8	6.90	44
10	Matt Kruse Calendar Script 2.2 Attack	8	6.90	567

Figure 2 — Top Ten Attacks Affecting Users on September 18

Abbildung: die zehn häufigsten Angriffe am Tag von Nimda´s Auftreten[1]

4.2 Payload

Abgesehen von dem Schaden die der Nimda-Wurm durch seine Transfer-Aktivitäten und die damit verbundene Netzbelastung darstellt, erzeugt er folgende Schäden:

- Erzeugt oder aktiviert einen “Guest”-Account und gibt ihn die Rechte des Administrators
- Stellt den vollen Zugriff auf den C: Laufwerk für jeden Benutzer bereit, falls der IIS Server auf C: installiert ist
- Scannt freigegebene Ordner und Laufwerke nach EXE-Dateien und ersetzt diese unter der Beibehaltung der Dateinamen durch sich selbst
- Scannt lokale Festplatte nach HTM, HTML, und ASP-Dateien und fügt diesen ein Stück JavaScript hinzu, um Spreading über Webbrowser zu ermöglichen
- Erzeugt in dem selben Verzeichnis eine **readme.eml**-File, die eine MIME-enkodierte Version der Nimda enthält
- Modifiziert die Datei **system.ini**, so dass der Wurm automatisch mit jedem Systemneustart ausgeführt wird.
- Erzeugt multiple Instanzen von den ***.eml**-Dateien und **riched20.dll** auf den offenen Netzlaufwerken, sogar wenn keine HTML-Dateien im System gefunden werden.

[1] Quelle: [ARIS01]

- Nimda tritt alle zehn Tage in seine Email-Propagationsphase ein
- Verbreitung von folgenden Dateien: readme.exe, readme.eml, admin.dll, mmc.exe, load.exe, riched20.dll, .EML (häufig README.EML oder DESKTOP.EML)

4.3 Varianten

Es gibt einige Abwandlungen von der ursprünglichen W32/Nimda.a@MM Version. Diese beschränken sich aber auf kleinere Änderungen. Meist wurde nur der Text-String im Wurm-Code modifiziert. Die folgende Liste gibt einen Überblick über die verschiedenen Varianten:

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

Tabelle: Verschiedene Varianten des Nimda-Wurms, Quelle: [NAI]

4.4 Der Nimda-Wurm im IR Labor

Das Netzwerk in dem der Wurm getestet wurde hatte folgenden Aufbau:

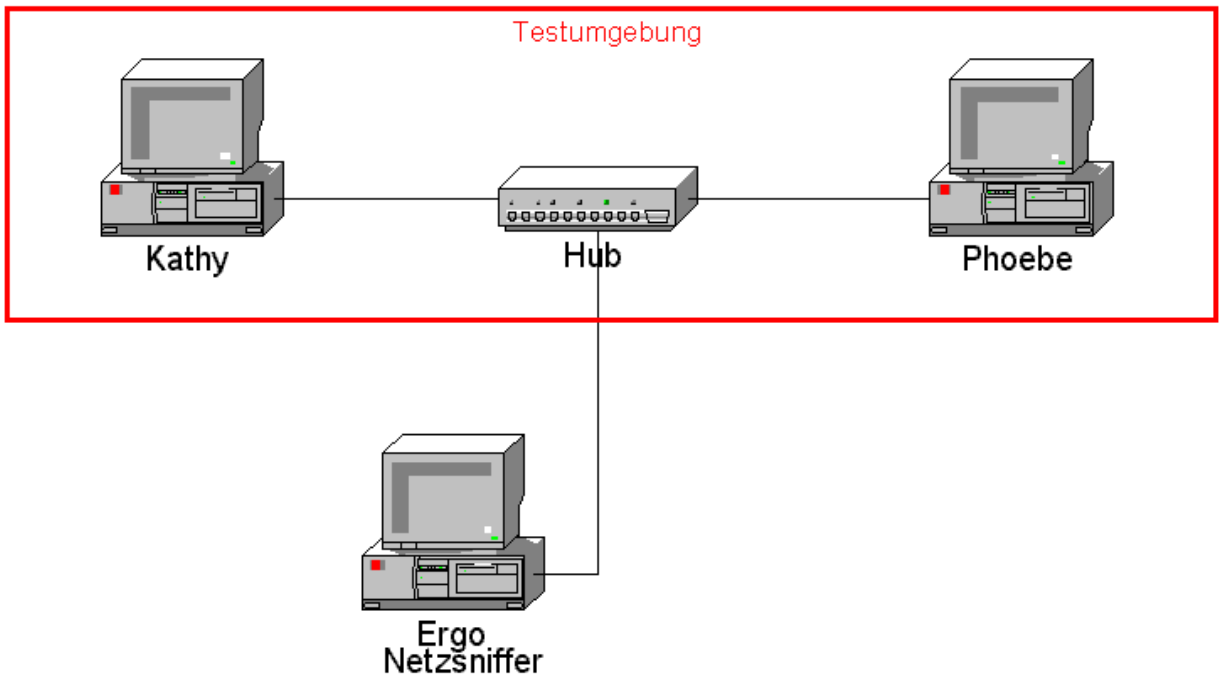


Abbildung: Das Netzwerk für die Analyse des Nimda-Wurms

Wie bereits im dritten Abschnitt beschrieben, diente Ergo dabei als Beobachter, der den Datenverkehr zwischen im befallenen Netz beobachtet. Kathy fungierte als angreifender Rechner, und Phoebe als Opfer.

Auf Ergo wurde folgende Software installiert:

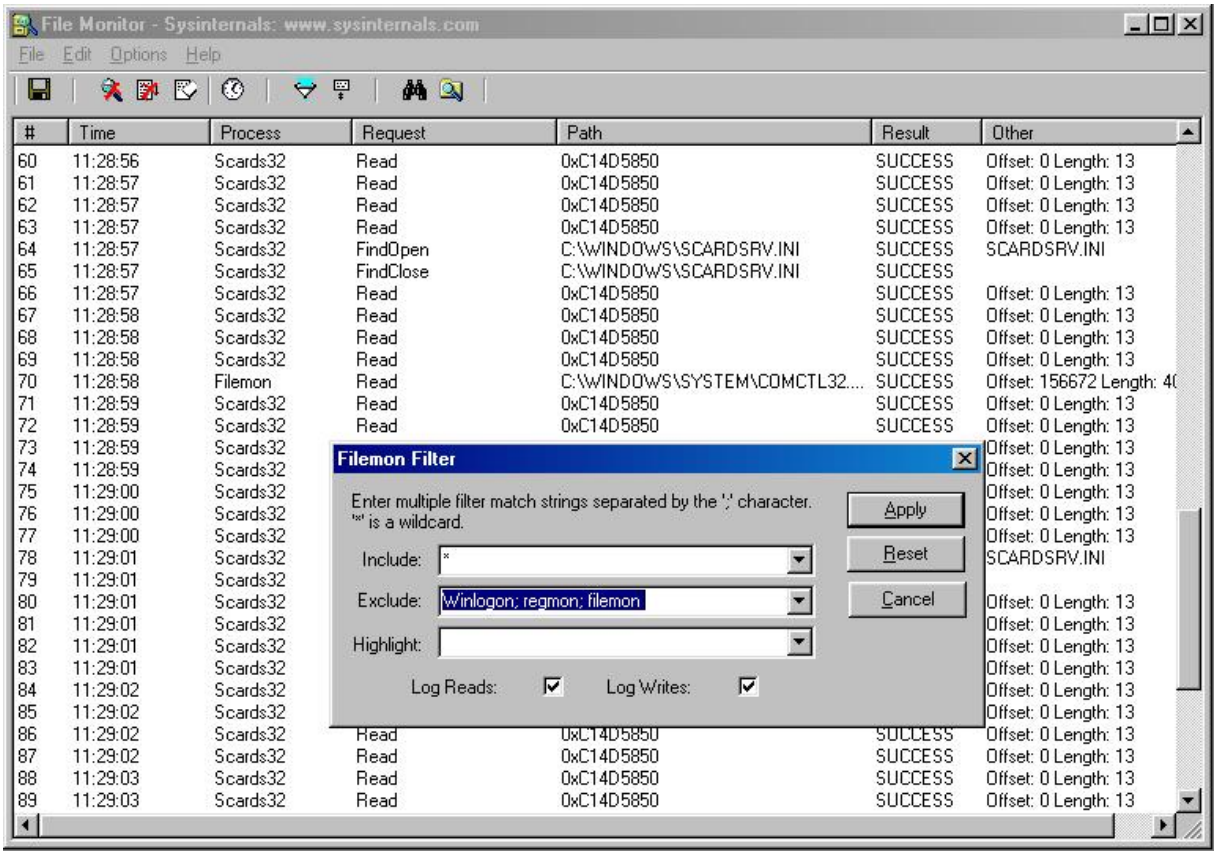
- Win 2000 SP3; d.h. eine gepatchte Windows Version die keine von Nimda angegriffenen Sicherheitslücken aufweist
- Ethereal; ein Sniffer um die über das Netwerk fliessenden Daten zu analysieren
- Norton Anti Virus; ein Antivirus Programm das zusätzlich die Immunität des Beobachter Rechners gegenüber Malware sicherstellen sollte.

Auf dem Opfer und Angriffsrechner wurden die gleichen Programme installiert, jeweils W2000, IIS 5.0, SP1, Internet Explorer 5.0; d.h. Windows und der IIS in einer Version die Sicherheitslücken aufweisen welche von Nimda angreifbar sind. Um die Vorgänge während des Angriffes analysieren zu können wurden außerdem ein Prozeß Monitor (PMON), ein File Monitor (File Mon), und ein Registry Monitor (Reg Mon) installiert.

Außerdem wurde weitere Maßnahmen getroffen um beide Rechner für den Wurm anfälliger zu machen:

- Shares: Ein Verzeichnis auf C freigegeben
- Auf Kathy und Phoebe wurde die Default.htm Seite installiert (c:\inetpub\webpub\default.htm), so dass sie vom jeweils anderen Rechner über den Internet Explorer einsichtig war

Um die Analyse zu vereinfachen, wurden die Monitore so eingestellt dass bestimmte Daten nicht von ihnen erfasst wurden. So sollten Prozesse bzw. Dateiveränderung ausgehend von Winlogon, regmon, filemon ignoriert werden. Die folgende Abbildung zeigt den Filemonitor, und die beschriebene Einstellung der Exclude-Funktion:



Nachdem die Monitor und der Sniffer konfiguriert und aktiviert waren, wurde Nimda auf dem Rechner Kathy von einer Diskette aus gestartet. Der Prozess und der Filemonitor dieses Rechners zeigten daraufhin eine länger anhaltende Aktivität an, die später auf Phoebe überging, was darauf hindeuten könnte dass Nimda sich, z.B. über die Netzwerkfreigaben, auf diesen Rechner kopiert hatte. Die Versuche sind im Anhang detaillierter aufgeführt.

4.5 Response Möglichkeiten

Es gibt sowohl Software Nimda Removal Tools, als Anleitungen für die manuelle Entfernung von Nimda, z.B. von Symantec unter http://www.cert.org/advisories/CA-2001-26.html. Aber selbst von den Herausgebern dieser Anleitung wird gewarnt, dass so eine Entfernung mit dem Risiko verbunden ist dass der Wurm nicht vollständig entfernt wurde. Aber vor allem ist mit so einer Entfernung nicht gewährleistet, dass Hintertüren entdeckt wurden, die von etwaigen Angreifern hinterlassen wurden während das System von dem Wurm verwundbar gemacht worden ist. Deswegen ist die sinnvollste Antwort auf eine Nimda-Infizierung, das gesamte System neu zu installieren.

5 Ausblick

In diesem Semester lag der Schwerpunkt unseres Projektes auf der Erarbeitung von IR relevanter Theorie und dem Aufbau des Netzwerkes. In folgenden Semestern könnte die Methodik der Analyse eines Incidents genauer erörtert werden: wie kann man systematisch Systemveränderungen verfolgen, wie Log-Files effizient analysieren, welche Software Tools gibt es noch für diese Art von Analyse?

6 Anhang

6.1 Protokolle

6.1.1 Protokoll vom 6.1.02: Test 1

Zu unterschiedlichen Rechnerzeiten: Phoebe und Ergo sind abgestimmt, während Kathy eine Minute vor den beiden ist.

Schritte für den Test:

1. Shares einrichten: Kathy/Phoebe: Verzeichnis „Watch-it“ freigegeben
2. Auf Kathy und Phoebe: die Default-Seite modifziert (c:\inetpub\webpub\default.htm)
3. Auf Kathy und Phoebe: File Mon, Reg Mon und Prozess Monitor gestartet. Dabei bei den beiden ersten probiert die anderen aus der Analyse zu nehmen, mit EDIT->Filter->Winlogon; regmon; filemon (bei Reg Mon über OPTIONS->Filter)
4. Ergo: NAV installiert, Ethereal gestartet (Capture Options: NDIS5.0 Driver)
5. Nimda auf Kathy gestartet
6. Viel Action auf Kathy, später ein bisschen auf Phoebe, noch später noch weniger auf Ergo

6.1.2 Protokoll über Test 2 vom 04.02.2003

Ziel des 04.02.2003: Wiederholung des letzten Versuches: die dynamische Analyse des Systems, nach der Aktivierung von Nimda. Der Versuchsaufbau bleibt derselbe. Die Systeme werden in ihren alten Zustand versetzt. Gleichzeitig werden einige Einstellungen geändert, die vor dem Versuch besprochen werden.

Phoebe (Rechnername Daphne) wird im Versuch als Opfer genutzt. Hierauf wird Outlook installiert und das Adressbuch mit Adressen gefüllt: Silvio@IR.de, Benjamin@IR.de usw. Im Postausgang wird eine und im Posteingang werden zwei E-Mails eingetragen. Diese Email-Adressen sind zwar nicht existent, aber die Gruppe hofft, dass Nimda versucht sich an diese Adressen zu versenden und dass es sichtbar wird.
Auf Kathy, dem angreifenden Rechner, wird eine standardisierte Website modifiziert und in den Dateiordner wwwroot gestellt. Diese Website ist von Phoebe über den IE sichtbar, wenn nach Kathy gesucht wird.
Auf Phoebe und Kathy werden einige shares für das Netzwerk freigegeben und Dateien vom Typ .exe und .dll in diesen shares gespeichert.
Auf Ergo wird Ethereal gestartet.
Auf Phoebe und Kathy werden die Audittools Filemon, Regmon und Pmon gestartet.
Nimda wird auf Kathy losgelassen.
Auf Phoebe wird Outlook gestartet und die oben genannte Website wird besucht.
Auf Phoebe und Kathy wird die Zeit vorgestellt.
Der Versuch wird gestoppt und die Logs gespeichert.

6.2 Literatur

[KOSSAKOWSKI00] Klaus-Peter Kossakowski, “Information Technology Incident Response Capabilities”, Dissertation, Universität Hamburg, 2000

[NHackersGui] Anonymous, „Der neue Hacker´s guide“, Markt&Technik Verlag 2001

[NTCSINET03] Network Test Center, „Sicherer ins Internet“, Universität Hamburg, Fachbereich Informatik

[SCHULTZ91] E.Eugene Schultz, “Computer Security Incident Response Teams”, Artikel für NASA 1st AIS Security Technology for Space Operations Conference, Houston, Texas, November 1991

[SIVERSYS00] “Sicherheit in Vernetzten Systemen”, Hans-Joachim Mück (Herausgeber), Bericht 224, Universität Hamburg, Fachbereich Informatik, Februar 2000

[WACK91] John P. Wack, „Establishing a Computer Security Incident Response Capability (CSIRC), NIST Special Publication 800-3, Gaithersburg, 1991

Links:

[ALLAIRE01] Allaire Corporation, „Incident Response – Allaire Security White Paper Series“, 2001, http://www.macromedia.com/v1/DocumentCenter/Partners/ASZ_ASWPS_Incident_Response.pdf

[ARIS01] ARIS Security Focus, Nimda Worm Analysis, 2001, http://aris.securityfocus.com/alerts/nimda/010919-Analysis-Nimda.pdf

[CERT] Incident Reporting, CERT, 2002, http://www.cert.org/tech_tips/incident_reporting.html#I

[CSIRTS] Handbook for Computer Security Incident Response Teams (CSIRTs), Moira J. West-Brown, Don Stikvoort, Klaus-Peter Kossakowski, 1998 http://www.sei.cmu.edu/publications/documents/98.reports/98hb001/98hb001abstract.html

[NAI] Nimda Varianten und Aliasse bei NAI, http://vil.nai.com/vil/content/v_99209.htm

[OSBORNE01] Tia R. Osborne, “Building an Incident Response Program To Suit Your Business”, 2001, http://www.sans.org/rr/incident/program.php

[VIRUSLIST] Viruslist, I-Worm.Nimda, http://www.viruslist.com/eng/viruslist.html?id=4261, Download am 9.4.2003