Honeynet Challenge of the Month
September 2003

By
SpiderNick and DoubleR
spidernick_doubler at hotmail.com

HONEYNET CHALLENGE OF THE MONTH **1**

QUESTIONS: **3**

1. DESCRIBE THE PROCESS YOU USED TO CONFIRM THAT THE LIVE HOST WAS COMPROMISED WHILE REDUCING THE IMPACT TO THE RUNNING SYSTEM AND MINIMIZING YOUR TRUST IN THE SYSTEM. 3
Preparation *3*
First step: Gathering volatile evidence *3*
Second Step: Live Backup via dd and netcat. *4*
Third Step: MD5 Checksum Comparison Script. *5*
2. EXPLAIN THE IMPACT THAT YOUR ACTIONS HAD ON THE RUNNING SYSTEM. 5
3. LIST THE PID(S) OF THE PROCESS(ES) THAT HAD A SUSPECT PORT(S) OPEN (I.E. NON RED HAT 7.2 DEFAULT PORTS). 5
PID 3137 (smbd -D process) *6*
PID 15119 (initd process) *7*
PID 25241 (xopen process) *7*
4. WERE THERE ANY ACTIVE NETWORK CONNECTIONS? IF SO, WHAT ADDRESS(ES) WAS THE OTHER END AND WHAT SERVICE(S) WAS IT FOR? 8
5. HOW MANY INSTANCES OF AN SSH SERVER WERE INSTALLED AND AT WHAT TIMES? 8
/usr/sbin/sshd *8*
/lib/.x/s/xopen *9*
/tmp/sand/smbd –D *9*
6. WHICH INSTANCES OF THE SSH SERVERS FROM QUESTION 5 WERE RUN? 10
7. DID ANY OF THE SSH SERVERS IDENTIFIED IN QUESTION 5 APPEAR TO HAVE BEEN MODIFIED TO COLLECT UNIQUE INFORMATION? IF SO, WAS ANY INFORMATION COLLECTED? 10
8. WHICH SYSTEM EXECUTABLES (IF ANY) WERE TROJANED AND WHAT CONFIGURATION FILES DID THEY USE? 11
9. HOW AND FROM WHERE WAS THE SYSTEM LIKELY COMPROMISED? 11

BONUS QUESTION: **13**

WHAT NATIONALITY DO YOU BELIEVE THE ATTACKER(S) TO BE, AND WHY? 13

NOTES: **13**

APPENDIX A. **13**

1. "/MNT/CDROM/BIN/NETSTAT -NAP" COMMAND OUTPUT. 13
3. "CHECK.PL" SCRIPT. 14
4. "CHECK.PL" OUTPUT. 15
5. LSOF 3137 (SMBD –D) 16
6. RECOVER /VAR/LOG/MESSAGES FILE. 16

Questions:

1. Describe the process you used to confirm that the live host was compromised while reducing the impact to the running system and minimizing your trust in the system.

Preparation

Before starting to analyze the system, we built a trusted binary CD from a newly built Red Hat 7.2 system to mount on the live system. The CD contained statically linked binaries, tools such as dd and netcat, and the Coroner's Toolkit (also known as TCT).

This will help us obtain true results if the local binaries have been altered or replaced. If there are differences between the trusted and the local this first step will confirm a compromised system.

In addition, we gathered all the tools needed for the investigation:

EnCase:
Windows, GUI forensics investigation tool. (Commercial version)

Coroner's Toolkit:
Collection of forensics programs, Unix-based.
http://www.porcupine.org/forensics/tct.html

Netcat
Windows and Unix binaries available, reads and writes data across network connections.
http://www.atstake.com/research/tools/network_utilities/

dd
Used to perform a physical backup of the evidence.

First step: Gathering volatile evidence

We gathered the evidence according to the chart below:

Volatile Evidence (In order from most volatile to least)

Memory	Only Available if power is on	Coroner's toolkit
Swap Space or Page File	Only Available if power is on	Live hard drive imaging (dd, nc)
Network Status and Connections	Only Available if power is on	netstat and ifconfig
Processes Running	Only Available if power is on	ps and lsof
Hard Drive Media		Live hard drive imaging (dd, nc)
Removable Media		None

We first mounted the trusted CD and a blank, newly formatted floppy disk.

mount /mnt/cdrom
mount /mnt/floppy/

We then executed the netstat command to get the connections established and ports listening. To assist in covering our operation, we sent all results from the trusted binaries to a floppy drive instead of placing them on local system.

```
/mnt/cdrom/bin/netstat -nap > /mnt/floppy/netstat.good
netstat -nap > /mnt/floppy/netstat.bad
```

Afterwards, we executed the ifconfig command to gather information and statistics about the state of the network interfaces:

```
/mnt/cdrom/bin/ifconfig –a > /mnt/floppy/ifconfig.good
ifconfig –a > /mnt/floppy/ifconfig.bad
```

We then executed the ps and lsof command to identify the processes running in memory:

```
/mnt/cdrom/bin/ps -eaxf > /mnt/floppy/ps.good
ps -eaxf > /mnt/floppy/ps.bad

/mnt/cdrom/usrsbin/lsof > lsof.good
cp lsof.good /mnt/floppy/
```

As indicated above, we ran the four commands both from the trusted CD and the compromised machine. There was a definite gap between the output of the trusted and untrusted commands, with the untrusted commands not displaying as much information as the trusted commands. In addition, some of the information displayed by the trusted commands and not by the untrusted commands was definitely deemed as suspicious. The results indicating these differences were leads to further the investigation. From the four commands (netstat, ps, lsof and ifconfig) results and differences we decided that a live back up using a trusted dd was required for additional information collection and maintaining system integrity.

Second Step: Live Backup via dd and netcat.

We decided to get a live snapshot of the system, including its swap file. Among others, this will allow us to possibly get deleted files that may give crucial information that we cannot look at locally.

We imaged the compromised system using dd and netcat, to create a duplicate, bit-by-bit image of the original media, including all slack (free) space. In ddition, we imaged the swap file of the system. We used the dd and netcat commands from out forensics CD in order to bypass the OS and create bit-stream backups of all the evidence. The integrity of the compromised system is protected this way.

We connected the compromised machine to a hub, and we also connected our image collector machine to the same hub. This hub will make a stand-alone network for the live backup to a forensic system. Also using the hub the compromise system will not receive any network connectivity errors, which may leave processes that rely on network connectivity running (if any).

Commands executed:
1. On the image collector machine, set up netcat to listen on port 2020, and to log all input to victim_sda1.img file (which we will later view in EnCase):

```
nc -l -p 2020 > victim_sda1.img
```

```
nc -l -p 2021 > victim_sda2.img
```

2. On the compromised machine:

```
/mnt/cdrom/tools/dd if=/dev/sda1 | /mnt/cdrom/tools/nc -nv 192.168.1.110 2020
/mnt/cdrom/tools/dd if=/dev/sda2 | /mnt/cdrom/tools/nc -nv 192.168.1.110 2021
```

The forensics analysis was done on these images. To extract the server information, we analyzed the images of the compromised system and the swap file using EnCase. EnCase is a forensics investigation tool that we used for easy viewing of the imaged file system.

Third Step: MD5 Checksum Comparison Script.

We built a script to collect the MD5 checksums of the possible compromised system, and compare them with previous MD5 checksums. This step was to identify any changed or deleted files against the listing made available from the Scan 29 exercise.

NOTE: Results of the running of the script are attached in Appendix A.

2. Explain the impact that your actions had on the running system.

The decision to keep the compromised system on was made, at least until preliminary information could be retrieved. This was made in part to the fact that we were working with a VMware image, which has its limitations.

"Leaving the system on" decision allows the system to constantly change, which may compromise the integrity of the system. It will also allow some possible scripts/programs installed by intruder to access, monitor, change, install and delete files.

Our actions' impacts on the system are the following:

A) MAC Times changed. The initial commands above were executed locally and possibly leaving a trace to a start of an investigation. Since we wanted to compare the differences between the trusted binaries and the local binaries, the local binaries would have to be accessed. This would leave modified accessed created (MAC) times changed to the investigating date. The collection of this preliminary information was necessary to retrieve since it would be erased once the decision to shut off the power is made.

B) Root Account Access. Being logged on with this account, and executing the initial commands on the system will generate a .bash_history in the /root directory.

C) Live back-up performed. Since we performed a live back-up using a trusted dd binary over the network. This may have affected the page file system possibly overwriting information.

3. List the PID(s) of the process(es) that had a suspect port(s) open (i.e. non Red Hat 7.2 default ports).

To identify the processes that had ports open, we ran the netstat command from the forensics CD with the switches "n" "a" and "p". These switches are used as follows:

The unix netstat command prints information about the Linux subsystem. Basicaly, this command lists all open ports and connections to and from the machine.

These arguments are used as follows (from "man netstat" command output):

-n Show numerical addresses instead of trying to determine symbolic host, port or user names.
-a Show both listening and non-listening sockets. With the --interfaces option, show interfaces that are not up
-p Show the PID and name of the program to which each socket belongs.

The command executed was "/mnt/cdrom/bin/netstat -nap > /mnt/floppy/netstat.good.nap" The output of the "/mnt/cdrom/bin/netstat -nap" command can be viewed in at the end of this document in Appendix A, bullet 1.

The following processes open ports that are not default ports in Red Hat 7.2:

PID 3137 (smbd -D process)

This smbd –D process is different that the regular smbd process.

Samba default ports are not included in this list, since they are default ports if Samba was installed on the system (plus we verified that the Samba daemon binaries and configuration files except for secrets.tdb were not modified, their MD5 checksum was not chanced").

Some ports that are not usually associated to Samba show in the output:

```
tcp        0      0 0.0.0.0:80              0.0.0.0:*
LISTEN      3137/smbd -D
```

The line above indicates that the smbd daemon (PID 3137) controls TCP port 80. This port is usually associated with web daemons, so this might be considered as a non-default port.

```
tcp        0      0 0.0.0.0:2003            0.0.0.0:*
LISTEN      3137/smbd -D
```

The line above indicates that the Samba smbd daemon (PID 3137) controls TCP port 2003.

Crosscorrelating this process ID with the “ps -eaxf” output, we notice that there are two smbd –D processes started. One of the processes is valid (regular Samba process), while the PID 3137 smbd process start from the /tmp/sand directory, which makes it suspicious.

Later in the investigation, and also later described in this paper, we have gather additional information on the smbd –D process (PID 3137).

Note: The syslogd PID 3247 and the klogd PID 3252 processes also have the home directory of /tmp/sand. An investigation of the /tmp directory reveals no files existent in this directory.

Note: For lsof output of this processes please see Appendix A, bullet 5.

PID 15119 (initd process)

```
tcp        0      0 0.0.0.0:65336           0.0.0.0:*
LISTEN      15119/initd
```

The initd process (PID 15119) has TCP port 65536 opened. This is not a default Red Hat 7.2 port. By correlating this data with the output of the "/mnt/cdrom/bin/ps -eaxf" command (Appendix A, bullet 2), we see that this process runs an IRC proxy program called psyBNC (http://www.netknowledgebase.com/tutorials/psybnc.html):

```
15119 ?        S      0:00 initd PWD=/etc/opt/psybnc
HOSTNAME=sbm79.dtc.apu.edu

tcp        0      0 0.0.0.0:65436           0.0.0.0:*
LISTEN      15119/initd
```

The initd process is also listening on TCP port 65436, also a non-default port in this version of Red Hat.

The following three netstat output lines indicate established connections to the psyBNC IRC proxy and outgoing connection from the compromised machine to IRC servers. *(Since the question is a bit vague, we decided to include these in the answers to this question even though they are included in the answer to the next question)*

```
tcp        0      0 192.168.1.79:65336      213.154.118.200:1188
ESTABLISHED 15119/initd
```

Above, a connection is established from IP 213.154.118.200 to the compromised machine on port 65536, which as showed above, is the initd process.

```
tcp        0      0 192.168.1.79:1149       64.62.96.42:6667
ESTABLISHED 15119/initd
tcp        0      0 192.168.1.79:1146       199.184.165.133:6667
ESTABLISHED 15119/initd
```

Above, the compromised machine is establishing two outgoing IRC connections to IPs 64.62.96.42 and 199.184.165.133.

PID 25241 (xopen process)

```
tcp        0      0 0.0.0.0:3128            0.0.0.0:*
LISTEN      25241/xopen
```

The xopen process (PID 25241) has TCP port 3128 opened. The xopen process is not a default process and TCP 3128 is not a default port in Red Hat 7.2. Cross correlating with the output of the trusted ps command indicates that the xopen process starts in the /lib/.x/s/ directory:

```
25241 ?        S      0:00 /lib/.x/s/xopen -q -p 3128 PWD=/lib/.x/s
HOSTNAME=loc
```

```
udp        0      0 0.0.0.0:3049            0.0.0.0:*
25239/xopen
```

As seen above, the xopen process also listens on UDP port 3049.

4. Were there any active network connections? If so, what address(es) was the other end and what service(s) was it for?

Yes, according to the "/mnt/cdrom/bin/netstat -nap" command output (Appendix A, bullet 1), there were active network connections.
To identify these connections and the addresses on the other end, we again analyze the netstat output as printed in Appendix A, bullet 1:

```
tcp        0      0 192.168.1.79:65336      213.154.118.200:1188
ESTABLISHED 15119/initd
```

Above, a connection was established from IP **213.154.118.200** to the compromised machine on port 65536, which is managed by the initd process..

```
tcp        0      0 192.168.1.79:1149       64.62.96.42:6667
ESTABLISHED 15119/initd
tcp        0      0 192.168.1.79:1146       199.184.165.133:6667
ESTABLISHED 15119/initd
```

Above, the compromised machine is establishing two outgoing IRC connections to IPs **64.62.96.42** and **199.184.165.133**.

In conclusion, these connections were used by psyBNC, to proxy IP 213.154.118.200 through the compromised machine to IRC servers 64.62.96.42 and 199.184.165.133.

5. How many instances of an SSH server were installed and at what times?

From the ps command output and the investigation of suspicious processes we have gathered the following:

/usr/sbin/sshd

/usr/sbin/sshd is installed, which is a valid sshd binary and the MD5 checksum comparison shows it has not bee modified. The file has the following attributes, as reported by EnCase:

Encase reports the following MAC times (EST) from the image:

Last Accessed	08/09/03 05:34:44PM
Last Written	09/06/01 09:14:25AM
Entry Modified	08/10/03 04:33:57PM

Note: By default, EnCase displays the evidence times in the same zone as the investigator's computer. This is the way we have displayed the times when mentioning EnCase output.

/lib/.x/s/xopen

By analyzing this process which showed in the ps and netstat commands output, we realized it was also an sshd daemon. Our suspicious was confirmed when looking at the strings output of this file.

According to EnCase, the MAC times (EST) are as follows:

Last Accessed	08/10/03 06:32:16PM
Last Written	12/28/02 09:01:31PM
Entry Modified	08/10/03 06:32:16PM

This sshd daemon was installed at 06:32:16 PM EST on 08/10/03.

/usr/bin/smbd –D

We identified this as an SSH process, which was running on the compromised machine. We later identified the binary on the system using EnCase.

According to the netstat output, the process has the following ports open: 80, 443 and 2003.

EnCase report the following MAC times for the file:

Last Accessed	08/10/03 06:54:18PM
Last Written	09/04/02 02:54:10AM
Entry Modified	08/10/03 04:33:33PM

Configuration files in /usr/include/ iceconf.h icekey.h icepid.g iceseed.h

Last Accessed	08/10/03 04:33:33PM
Last Written	08/10/03 04:33:33PM
Entry Modified	08/10/03 04:33:33PM

It is usual for the configuration files and daemon to be installed at once, as a package. According to the file, 04:33:33 PM EST on 08/10/03 could be time when this process was installed.

/usr/lib/sp0

This SSH daemon was identified by using EnCase, and searching all the files within the image for a string we identified as possibly unique to the sshd server, “Received session key; encryption turned on”. That’s how we found this server:

According to EnCase, the following are the MAC times for this file, time zone is EST:

Last Accessed	08/10/03 06:30:21PM
Last Written	06/02/03 12:03:03AM
Entry Modified	08/10/03 06:30:54PM

6. Which instances of the SSH servers from question 5 were run?

The first **three** SSH daemons identified by us in question 5 were run, as indicated in the ps –eaxf output, Appendix A, bullet 2.

/usr/bin/smbd –D
/lib/.x/s/xopen
/usr/sbin/sshd

7. Did any of the SSH servers identified in question 5 appear to have been modified to collect unique information? If so, was any information collected?

The smbd –D (PID 3137) has been modified to collect certain information.

In order to be able to run strings on the smbd –D (PID 3137), we used the pcat tool to copy the process from memory. Pcat comes as part of the Coroner's Toolkit (TCT). The Coroners Toolkit (from here known as 'TCT') is a suite of tools written for the purpose of gathering and analyzing forensic data typically from Unix systems.

The following command was executed:

pcat 3137| strings | less

What caught our attention (excluding the /usr/include// configuration files), were the following strings:

```
e5e77c675e5ea20c0de4c36ac089b629    - probably the universal password,
                                    encrypted with md5sum.

2xvu2ole2olevkorj
+-[ User Login Incoming ]----------- --- --- - -
| username: %s password: %s%s hostname: %s
+---------------------------------- ----- --- -- -- -
```

The above lines indicate that process is set to capture the username, password and hostname of the incoming connections to this process.

The 2xvu2ole2olevkorj strings is interesting, since before writing to a file, the program would most likely need to open the file, so this might be the file that needs to be opened. It might be the path of the log file, but in a hidden form, using a substitution method:

```
2xvu2ole2olevkorj
2 could be /
/xvu/ole/olevkorj
/usr/lib/libshlog
```

The string is the result of a substitution cipher, a Caesar shift cipher, where the cipher text has been shifted by three place.

The file did not exist on the compromised system.

8. Which system executables (if any) were trojaned and what configuration files did they use?

To identify the files modified on the system, we wrote a Perl script to compare the original MD5 checksums with the current MD5 checksums. You can view the script and it's results in Appendix A, bullet 3.

In conclusion, the following system executables were modified:

```
Signatures do not match for /usr/bin/top
Signatures do not match for /bin/netstat
Signatures do not match for /bin/ls
Signatures do not match for /bin/ps
Signatures do not match for /sbin/ifconfig
```

Three configurations file were found for some of the above binaries:

1. /dev/ttyoa

The /dev/ttyoa file defines the IP addresses, TCP and UDP ports to be hidden when executing the netstat command. The file was identified by running "strings" command against the netstat binary and analyzing the output: "/mnt/cdrom/usrbin/strings netstat | less". We have seen this configuration file under the same name in other compromised machine. Within the configuration file, the lines that start with 1 hide the partially matched IP addresses, the lines that start with 3 hide the TCP ports and the lines that start with 4 hide the UDP ports from the netstat command output.

2. /dev/ttyof

The /dev/ttyof file lists the file names that the "/bin/ls" will hide from its output. Again, executing "strings" on the "/bin/ls" command allowed us to identify this file.

3. /dev/ttyop

The /dev/ttyop file is the configuration file for both "/usr/bin/top" and "/bin/ps" executables. Just as above, executing "strings" on the two files helped us identify the configuration file, but we already were suspect of what the file was.

9. How and from where was the system likely compromised?

From the deleted files, we recovered the following which looks like the httpd log file:

```
[Sun Aug 10 04:02:01 2003] [notice] Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4
OpenSSL/0.9.6b DAV/1.0.2 configured -- resuming normal operations
[Sun Aug 10 04:02:01 2003] [notice] suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Sun Aug 10 13:16:27 2003] [error] [client 213.154.118.219] client sent HTTP/1.1 request
without hostname (see RFC2616 section 14.23): /
[Sun Aug 10 13:16:37 2003] [error] [client 213.154.118.219] client sent HTTP/1.1 request
without hostname (see RFC2616 section 14.23): /
[Sun Aug 10 13:23:17 2003] [error] [client 213.154.118.219] File does not exist:
/var/www/html/sumthin
[Sun Aug 10 13:24:29 2003] [error] mod_ssl: SSL handshake failed (server
localhost.localdomain:443, client 213.154.118.219) (OpenSSL library error follows)
[Sun Aug 10 13:24:29 2003] [error] OpenSSL: error:1406908F:SSL
routines:GET_CLIENT_FINISHED:connection id is different
```

```
[Sun Aug 10 13:32:38 2003] [error] mod_ssl: SSL handshake failed (server
localhost.localdomain:443, client 213.154.118.219) (OpenSSL library error follows)
[Sun Aug 10 13:32:38 2003] [error] OpenSSL: error:1406908F:SSL
routines:GET_CLIENT_FINISHED:connection id is different
[Sun Aug 10 13:40:28 2003] [error] mod_ssl: Child could not open SSLMutex lockfile
/etc/httpd/logs/ssl_mutex.800 (System error follows)
[Sun Aug 10 13:40:28 2003] [error] System: No such file or directory (errno: 2)
```

There is an mod_ssl/OpenSSL vulnerability that allows for remote buffer overflow. According to the SecurityFocus website, there are exploits for this vulnerability in the wild:

http://www.securityfocus.com/bid/5363

When executed against a vulnerable target, these exploits will spawn a shell with root-level privileges.

Reading the web server banner (first line in the log file above), we confirmed that our compromised machine was running a vulnerable version of Apache/OpenSSL to this buffer overflow.

From the recovered /var/log/messages file (Appendix A, bullet 6):

```
Aug 10 14:14:41 localhost smbd -D[5505]: log: Connection from 213.154.118.218 port 2020
```

Also from a lost file, we identified an additional connection from 213.154.118.201:

```
bash.HOME=/root.USER=root.LOGNAME=root.PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:
/usr/local/sbin:/lib/.x:/lib/.x/s.MAIL=/var/spool/mail/root.SHELL=/bin/bash.SSH_CLIENT=21
3.154.118.201 2127 3128.SSH_TTY=/dev/pts/1.TERM=xterm./bin/bash
```

From the netstat outout, we also notice an established connection from the same subnet:

```
tcp        0      0 192.168.1.79:65336      213.154.118.200:1188
ESTABLISHED 15119/initd
```

From these four sources, we concluded that the machine was compromised through an mod_ssl/OpenSSL exploit described above, and the attackers were coming from 213.154.118.x subnet. In particular, the exploit was ran from 213.154.118.219 IP address, and the attacker(s) later connected from 213.154.118.218 and 213.154.118.201 also.

Note: We investigated a few other possibilities for how this machine was compromised:

A. SAMBA Buffer Overflow?
/var/cache/samba/connections.tdb -long strings, samba buffer overflow?
196.30.236.78

http://www.securiteam.com/exploits/5NP0J2A9PC.html

B. LPD Buffer Overflow? Auto-rooter.
Using EnCase, found the following strings in the free space (part of the deleted files):
"********* You g0t root ? ro0t ! r0ot ?! R00T !!! *********"

Doing a search on google, we fount the following exact string match in this exploit code:
http://www.blacksheepnetworks.com/security/hack/rdx/saxophone/rdx/mass-scan/lpd/lpd1.c

C. Ptrace local exploit.
Found strings in the /dev/shm/k file that match the exploit on this page:
http://www.securiteam.com/exploits/5CP0Q0U9FY.html

/dev/shm is used for POSIX shared memory in kernel 2.4.x

D. Doing a search for keyword “rootkit” on the evidence files turned out quite a large number of matches, most of which were in the swap file.

Bonus Question:

What nationality do you believe the attacker(s) to be, and why?

Based on the IRC channels (“aia buni” which means “those good one” in Romanian) and the connections to the compromised machine from .ro (assigned to Romania) addresses (extreme-service-10.is.pcnet.ro.), we believe the attacker(s) to be Romanian.

Notes:

1. /usr/bin/(swapd) looks like a sniffer which writes connections to: /usr/lib/libice.log

2. /usr/lib/libshtift is a directory that contains ifconfig, ls, netstat, ps and top executables. However, the md5sum hashes of these files do not correspond to the newly installed trojaned executables. Comparing the md5sum hashes with the original hashes show a match, so this directory has been used to store the back-up copies of the replaced binaries.

3. There were many other suspicious files and directories on the compromised system. For the purpose of keeping this report brief, we only identified them as they related to a question.

Appendix A.

1. "/mnt/cdrom/bin/netstat -nap" command output.

This command was one of the very first commands executed, once the forensics CD was mounted. The command's output printed below:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN      845/smbd
tcp        0      0 0.0.0.0:79              0.0.0.0:*               LISTEN      732/xinetd
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      3137/smbd -D
tcp        0      0 0.0.0.0:113             0.0.0.0:*               LISTEN      677/identd
tcp        0      0 0.0.0.0:2003            0.0.0.0:*               LISTEN      3137/smbd -D
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      732/xinetd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      699/sshd
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN      732/xinetd
tcp        0      0 0.0.0.0:65336           0.0.0.0:*               LISTEN      15119/initd
tcp        0      0 0.0.0.0:3128            0.0.0.0:*               LISTEN      25241/xopen
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      759/sendmail: accep
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      3137/smbd -D
tcp        0      0 0.0.0.0:65436           0.0.0.0:*               LISTEN      15119/initd
tcp        0      0 192.168.1.79:65336      213.154.118.200:1188    ESTABLISHED 15119/initd
tcp        0      0 192.168.1.79:1149       64.62.96.42:6667        ESTABLISHED 15119/initd
tcp        0      0 192.168.1.79:1146       199.184.165.133:6667    ESTABLISHED 15119/initd
udp        0      0 192.168.1.79:137        0.0.0.0:*                           850/nmbd
udp        0      0 0.0.0.0:137             0.0.0.0:*                           850/nmbd
udp        0      0 192.168.1.79:138        0.0.0.0:*                           850/nmbd
```

```
udp        0      0 0.0.0.0:138             0.0.0.0:*                           850/nmbd
udp        0      0 0.0.0.0:3049            0.0.0.0:*                           25239/xopen
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node PID/Program name    Path
unix  2      [ ACC ]     STREAM     LISTENING     943    778/gpm             /dev/gpmctl
unix  4      [ ]         DGRAM                    7984   3247/syslogd        /dev/log
unix  2      [ ]         DGRAM                    15679  732/xinetd
unix  2      [ ]         DGRAM                    7993   3252/klogd
unix  2      [ ]         DGRAM                    1078   893/login -- root
unix  2      [ ]         DGRAM                    990    820/crond
unix  2      [ ]         DGRAM                    924    759/sendmail: accep
unix  2      [ ]         DGRAM                    834    677/identd
unix  2      [ ]         DGRAM                    804    657/apmd
unix  2      [ ]         STREAM     CONNECTED     417    1/init
Active IPX sockets
Proto Recv-Q Send-Q Local Address              Foreign Address            State
```

2. "/mnt/cdrom/bin/ps -eaxf" command output.

```
  PID TTY      STAT   TIME COMMAND
    8 ?        SW     0:00 [kupdated]
    7 ?        SW     0:00 [bdflush]
    6 ?        SW     0:00 [kreclaimd]
    5 ?        SW     0:00 [kswapd]
    4 ?        SWN    0:00 [ksoftirqd_CPU0]
    1 ?        S      0:05 init HOME=/ TERM=linux
    2 ?        SW     0:00 [keventd]
    3 ?        SW     0:00 [kapm-idled]
    9 ?        SW<    0:00 [mdrecoveryd]
   17 ?        SW     0:04 [kjournald]
   92 ?        SW     0:00 [khubd]
  657 ?        S      0:00 /usr/sbin/apmd -p 10 -w 5 -W -P /etc/sysconfig/apm-sc
  677 ?        S      0:00 identd -e -o PWD=/ HOSTNAME=localhost.localdomain CON
  685 ?        S      0:00  \_ identd -e -o PWD=/ HOSTNAME=localhost.localdomain
  686 ?        S      0:00      \_ identd -e -o PWD=/ HOSTNAME=localhost.localdo
  695 ?        S      0:00      \_ identd -e -o PWD=/ HOSTNAME=localhost.localdo
  696 ?        S      0:00      \_ identd -e -o PWD=/ HOSTNAME=localhost.localdo
  699 ?        S      0:00 /usr/sbin/sshd PWD=/ HOSTNAME=localhost.localdomain C
  732 ?        S      0:00 xinetd -stayalive -reuse -pidfile /var/run/xinetd.pid
  759 ?        S      0:00 sendmail: accepting connections ons
  778 ?        S      0:00 gpm -t ps/2 -m /dev/mouse PWD=/ HOSTNAME=localhost.lo
  820 ?        S      0:00 crond PWD=/ HOSTNAME=localhost.localdomain CONSOLE=/d
15353 ?        S      0:00  \_ CROND PWD=/ HOSTNAME=localhost.localdomain CONSOL
15356 ?        S      0:00      \_ /usr/sbin/sendmail -FCronDaemon -i -odi -oem
  845 ?        S      0:00 smbd -D PWD=/ HOSTNAME=localhost.localdomain CONSOLE=
  850 ?        S      0:00 nmbd -D PWD=/ HOSTNAME=localhost.localdomain CONSOLE=
  893 tty1     S      0:00 login -- root
  901 tty1     S      0:00  \_ -bash HOME=/root PATH=/usr/local/sbin:/usr/local/
15361 tty1     R      0:00      \_ /mnt/cdrom/bin/ps -eaxf PWD=/root HOSTNAME=lo
  894 tty2     S      0:00 /sbin/mingetty tty2 HOME=/ TERM=linux PATH=/usr/local
  895 tty3     S      0:00 /sbin/mingetty tty3 HOME=/ TERM=linux PATH=/usr/local
  896 tty4     S      0:00 /sbin/mingetty tty4 HOME=/ TERM=linux PATH=/usr/local
  899 tty5     S      0:00 /sbin/mingetty tty5 HOME=/ TERM=linux PATH=/usr/local
  900 tty6     S      0:00 /sbin/mingetty tty6 HOME=/ TERM=linux PATH=/usr/local
 3137 ?        S      0:03 smbd -D PWD=/tmp/sand HOSTNAME=localhost.localdomain
 3153 ?        S      0:00 (swapd) PWD=/usr/bin HOSTNAME=localhost.localdomain M
 3247 ?        S      0:00 syslogd -m 0 PWD=/tmp/sand HOSTNAME=localhost.localdo
 3252 ?        S      0:00 klogd -2 PWD=/tmp/sand HOSTNAME=localhost.localdomain
25239 ?        S      0:00 /lib/.x/s/xopen -q -p 3128 PWD=/lib/.x/s HOSTNAME=loc
25241 ?        S      0:00 /lib/.x/s/xopen -q -p 3128 PWD=/lib/.x/s HOSTNAME=loc
25247 ?        S      0:00 /lib/.x/s/lsn PWD=/lib/.x/s HOSTNAME=localhost.locald
15119 ?        S      0:00 initd PWD=/etc/opt/psybnc HOSTNAME=sbm79.dtc.apu.edu
```

3. "check.pl" script.

```
#!/bin/perl

$file="/mnt/floppy/host79-2003-08-06.md5";
```

```
open (FILE,"$file");

while ($line=<FILE>) {
       # print $line;
            chomp $line;
            @newline = split(/ /,$line);
            $signature= @newline[0];
            $thefile= @newline[2];
            #print "$signature " ;
            #print "$thefile  ";

            if (-f $thefile) {
        open MD5, "/mnt/cdrom/usrbin/md5sum $thefile |";
        while ($line2=<MD5>){
            #print $line2;
            @newline2= split(/ /,$line2);
            $signature2= @newline2[0];
            #print "$signature2 \n";

            if ($signature ne $signature2) {
                  print "Signatures do not match for $thefile \n";
            }else{
                                    #print "Signatures match for $thefile \n";
                        }
        }
        close MD5;
            }else {
            print "File \"$thefile\" does not exist or is not a file...\n";
            }
    }

close FILE;
```

4. "check.pl" output.

```
Signatures do not match for /var/lib/slocate/slocate.db
Signatures do not match for /var/lib/random-seed
Signatures do not match for /var/lib/logrotate.status
File "/var/log/messages" does not exist or is not a file...
File "/var/log/lastlog" does not exist or is not a file...
Signatures do not match for /var/log/secure
Signatures do not match for /var/log/maillog
Signatures do not match for /var/log/wtmp
File "/var/log/sa/sa14" does not exist or is not a file...
File "/var/log/sa/sa15" does not exist or is not a file...
File "/var/log/sa/sar14" does not exist or is not a file...
File "/var/log/sa/sa16" does not exist or is not a file...
File "/var/log/sa/sar15" does not exist or is not a file...
File "/var/log/sa/sa06" does not exist or is not a file...
File "/var/log/samba/log.smbd" does not exist or is not a file...
File "/var/log/samba/smbd.log" does not exist or is not a file...
File "/var/log/samba/log.nmbd" does not exist or is not a file...
File "/var/log/samba/localhost.log" does not exist or is not a file...
File "/var/log/xferlog" does not exist or is not a file...
File "/var/log/httpd/error_log" does not exist or is not a file...
File "/var/log/httpd/ssl_engine_log" does not exist or is not a file...
File "/var/log/httpd/access_log" does not exist or is not a file...
File "/var/log/httpd/ssl_request_log" does not exist or is not a file...
File "/var/log/httpd/access_log.1" does not exist or is not a file...
File "/var/log/httpd/error_log.1" does not exist or is not a file...
File "/var/log/dmesg" does not exist or is not a file...
Signatures do not match for /var/log/cron
Signatures do not match for /var/log/boot.log
File "/var/log/rpmpkgs" does not exist or is not a file...
```

```
Signatures do not match for /var/cache/man/whatis
Signatures do not match for /var/cache/samba/smbd.pid
Signatures do not match for /var/cache/samba/connections.tdb
Signatures do not match for /var/cache/samba/nmbd.pid
Signatures do not match for /var/run/utmp
Signatures do not match for /var/run/runlevel.dir
Signatures do not match for /var/run/syslogd.pid
Signatures do not match for /var/run/klogd.pid
Signatures do not match for /var/run/apmd.pid
Signatures do not match for /var/run/sshd.pid
Signatures do not match for /var/run/sendmail.pid
Signatures do not match for /var/run/gpm.pid
Signatures do not match for /var/run/crond.pid
File "/var/run/ftp.rips-all" does not exist or is not a file...
Signatures do not match for /var/spool/anacron/cron.daily
Signatures do not match for /var/spool/anacron/cron.weekly
File "/tmp/root.md5" does not exist or is not a file...
Signatures do not match for /etc/mtab
Signatures do not match for /etc/rc.d/init.d/functions
Signatures do not match for /etc/rc.d/rc.sysinit
Signatures do not match for /etc/mail/statistics
Signatures do not match for /etc/aliases.db
Signatures do not match for /etc/adjtime
Signatures do not match for /etc/samba/secrets.tdb
Signatures do not match for /etc/httpd/conf/httpd.conf
Signatures do not match for /usr/bin/top
Signatures do not match for /bin/netstat
Signatures do not match for /bin/ls
Signatures do not match for /bin/ps
Signatures do not match for /sbin/ifconfig
```

5. lsof 3137 (smbd –D)

```
COMMAND  PID USER   FD   TYPE DEVICE     SIZE  NODE NAME
smbd    3137 root  cwd    DIR    8,1     4096     2 /
smbd    3137 root  rtd    DIR    8,1     4096     2 /
smbd    3137 root  txt    REG    8,1   672527 92030 /usr/bin/smbd -D
smbd    3137 root  mem    REG    8,1   485171 44656 /lib/ld-2.2.4.so
smbd    3137 root  mem    REG    8,1   436784 44674 /lib/libnsl-2.2.4.so
smbd    3137 root  mem    REG    8,1    85115 44667 /lib/libcrypt-2.2.4.so
smbd    3137 root  mem    REG    8,1    47872 44709 /lib/libutil-2.2.4.so
smbd    3137 root  mem    REG    8,1  5772268 44650 /lib/i686/libc-2.2.4.so
smbd    3137 root    0u   CHR    1,3          31876 /dev/null
smbd    3137 root    1u   CHR    1,3          31876 /dev/null
smbd    3137 root    2u   CHR    1,3          31876 /dev/null
smbd    3137 root    3u   REG    8,1        0  3187 /var/run/httpd.mm.800.sem (deleted)
smbd    3137 root    4u   REG    8,1        0 45309 /var/log/httpd/ssl_scache.sem (deleted)
smbd    3137 root    5u  sock    0,0           3626 can't identify protocol
smbd    3137 root    6u  IPv4   4571            TCP *:cfinger (LISTEN)
smbd    3137 root   15w   REG    8,1 23335716 46935 /var/log/httpd/error_log (deleted)
smbd    3137 root   16u  IPv4    976            TCP *:https (LISTEN)
smbd    3137 root   17u  IPv4    977            TCP *:http (LISTEN)
smbd    3137 root   18w   REG    8,1 22795530 46914 /var/log/httpd/ssl_engine_log (deleted)
smbd    3137 root   19w   REG    8,1        0 45308 /var/log/httpd/ssl_mutex.800 (deleted)
smbd    3137 root   20w   REG    8,1      253 46934 /var/log/httpd/access_log (deleted)
smbd    3137 root   21w   REG    8,1      253 46934 /var/log/httpd/access_log (deleted)
smbd    3137 root   22w   REG    8,1        0 46916 /var/log/httpd/ssl_request_log (deleted)
smbd    3137 root   23w   REG    8,1        0 45308 /var/log/httpd/ssl_mutex.800 (deleted)
```

6. Recovered /var/log/messages file.

```
Aug 10 13:33:57 localhost syslogd 1.4.1: restart.
Aug 10 13:33:57 localhost syslog: syslogd startup succeeded
Aug 10 13:33:57 localhost kernel: klogd 1.4.1, log source = /proc/kmsg started.
Aug 10 13:33:57 localhost kernel: Inspecting /boot/System.map-2.4.7-10
Aug 10 13:33:57 localhost syslog: klogd startup succeeded
Aug 10 13:33:57 localhost kernel: Loaded 15046 symbols from /boot/System.map-2.4.7-10.
Aug 10 13:33:57 localhost kernel: Symbols match kernel version 2.4.7.
Aug 10 13:33:57 localhost kernel: Loaded 371 symbols from 10 modules.
```

```
Aug 10 13:33:57 localhost kernel: (swapd) uses obsolete (PF_INET,SOCK_PACKET)
Aug 10 13:33:57 localhost kernel: eth0: Promiscuous mode enabled.
Aug 10 13:33:57 localhost kernel: device eth0 entered promiscuous mode
Aug 10 13:33:57 localhost kernel: NET4: Linux IPX 0.47 for NET4.0
Aug 10 13:33:57 localhost kernel: IPX Portions Copyright (c) 1995 Caldera, Inc.
Aug 10 13:33:57 localhost kernel: IPX Portions Copyright (c) 2000, 2001 Conectiva, Inc.
Aug 10 13:33:57 localhost kernel: NET4: AppleTalk 0.18a for Linux NET4.0
Aug 10 13:33:32 localhost syslog: syslogd shutdown succeeded
Aug 10 13:33:33 localhost smbd -D[3137]: log: Server listening on port 2003.
Aug 10 13:33:33 localhost smbd -D[3137]: log: Generating 768 bit RSA key.
Aug 10 13:33:34 localhost smbd -D[3137]: log: RSA key generation complete.
Aug 10 13:33:35 localhost smbd -D[3150]: error: bind: Address already in use
Aug 10 13:33:35 localhost smbd -D[3150]: fatal: Bind to port 2003 failed: Transport
endpoint is not connected.
Aug 10 13:33:56 localhost smbd -D[3225]: error: bind: Address already in use
Aug 10 13:33:56 localhost smbd -D[3225]: fatal: Bind to port 2003 failed: Transport
endpoint is not connected.
Aug 10 13:33:56 localhost syslog: klogd shutdown failed
Aug 10 13:33:57 localhost syslog: syslogd shutdown failed
Aug 10 14:13:47 localhost sshd: sshd -TERM failed
Aug 10 14:14:41 localhost smbd -D[5505]: log: Connection from 213.154.118.218 port 2020
Aug 10 14:14:42 localhost smbd -D[3137]: log: Generating new 768 bit RSA key.
Aug 10 14:14:44 localhost smbd -D[3137]: log: RSA key generation complete.
Aug 10 14:14:52 localhost smbd -D[5505]: log: Password authentication for root failed.
Aug 10 14:14:58 localhost smbd -D[5505]: log: Password authentication failed for user
root from extreme-service-10.is.pcnet.ro.
Aug 10 14:14:58 localhost smbd -D[5505]: log: Password authentication for root failed.
Aug 10 14:15:14 localhost smbd -D[5505]: log: Password authentication failed for user
root from extreme-service-10.is.pcnet.ro.
Aug 10 14:15:14 localhost smbd -D[5505]: log: Password authentication for root failed.
Aug 10 14:15:17 localhost smbd -D[5505]: fatal: Connection closed by remote host.
Aug 10 14:17:08 localhost smbd -D[8170]: log: Connection from 213.154.118.218 port 2021
Aug 10 14:17:09 localhost smbd -D[3137]: log: Generating new 768 bit RSA key.
Aug 10 14:17:10 localhost smbd -D[3137]: log: RSA key generation complete.
Aug 10 14:17:17 localhost smbd -D[8170]: log: Password authentication for root failed.
Aug 10 14:17:21 localhost smbd -D[8170]: log: Password authentication failed for user
root from extreme-service-10.is.pcnet.ro.
Aug 10 14:17:21 localhost smbd -D[8170]: log: Password authentication for root failed.
Aug 10 14:17:26 localhost smbd -D[8170]: log: Password authentication failed for user
root from extreme-service-10.is.pcnet.ro.
Aug 10 14:17:26 localhost smbd -D[8170]: log: Password authentication for root failed.
Aug 10 14:17:38 localhost smbd -D[8170]: log: Password authentication failed for user
root from extreme-service-10.is.pcnet.ro.
Aug 10 14:17:38 localhost smbd -D[8170]: log: Password authentication for root failed.
Aug 10 14:17:42 localhost smbd -D[8170]: log: Password authentication failed for user
root from extreme-service-10.is.pcnet.ro.
Aug 10 14:17:42 localhost smbd -D[8170]: log: Password authentication for root failed.
Aug 10 14:17:47 localhost smbd -D[8170]: fatal: Local: Too many password authentication
attempts from extreme-service-10.is.pcnet.ro for user root.
Aug 10 14:17:51 localhost smbd -D[8935]: log: Connection from 213.154.118.218 port 2022
Aug 10 14:17:52 localhost smbd -D[3137]: log: Generating new 768 bit RSA key.
Aug 10 14:17:53 localhost smbd -D[3137]: log: RSA key generation complete.
Aug 10 14:18:00 localhost smbd -D[8935]: log: Password authentication for root failed.
Aug 10 14:18:04 localhost smbd -D[8935]: log: Password authentication failed for user
root from extreme-service-10.is.pcnet.ro.
Aug 10 14:18:04 localhost smbd -D[8935]: log: Password authentication for root failed.
Aug 10 14:18:09 localhost smbd -D[8935]: log: Password authentication failed for user
root from extreme-service-10.is.pcnet.ro.
Aug 10 14:18:09 localhost smbd -D[8935]: log: Password authentication for root failed.
Aug 10 14:23:20 localhost smbd -D[8935]: log: Password authentication failed for user
root from extreme-service-10.is.pcnet.ro.
Aug 10 14:23:20 localhost smbd -D[8935]: log: Password authentication for root failed.
Aug 10 14:23:24 localhost smbd -D[8935]: fatal: Connection closed by remote host.
Aug 10 15:30:30 localhost kernel: eth0: Promiscuous mode enabled.
Aug 10 15:30:30 localhost modprobe: modprobe: Can't locate module ppp0
Aug 10 15:32:16 localhost kernel: eth0: Promiscuous mode enabled.
Aug 10 15:52:09 localhost smbd -D[14568]: error: bind: Address already in use
Aug 10 15:52:09 localhost smbd -D[14568]: fatal: Bind to port 2003 failed: Transport
endpoint is not connected.
Aug 10 15:52:10 localhost httpd: httpd shutdown succeeded
Aug 10 15:52:11 localhost smbd -D[14629]: error: bind: Address already in use
```

```
Aug 10 15:52:11 localhost smbd -D[14629]: fatal: Bind to port 2003 failed: Transport
endpoint is not connected.
Aug 10 15:52:12 localhost httpd: fopen: No such file or directory
Aug 10 15:52:12 localhost httpd: httpd: could not open error log file
/etc/httpd/logs/error_log.
Aug 10 15:52:12 localhost httpd: httpd startup failed
Aug 10 15:54:18 localhost smbd -D[14663]: error: bind: Address already in use
Aug 10 15:54:18 localhost smbd -D[14663]: fatal: Bind to port 2003 failed: Transport
endpoint is not connected.
Aug 10 15:54:18 localhost httpd: httpd shutdown failed
Aug 10 15:56:11 localhost su(pam_unix)[14689]: session opened for user root by (uid=0)
Aug 10 16:03:01 localhost su(pam_unix)[14689]: session closed for user root
Aug 10 16:04:38 localhost telnetd[15169]: ttloop: peer died: EOF
```