GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

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

Interested in learning more?

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

Forensics Analysis of an unknown System

(Part1 - Option 1)
&

Binary File Analysis

GCFA Practical
Version: 1.0

By
Stephen Pedersen

September 20, 2002

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Contents

Forensics Analysis of an unknown System 1
Binary File Analysis 1
Contents 2
Forensic Analysis of an unknown System 4
Synopsis of Case Facts 4
Description of the system 4
Hardware Description 4
Image the Media 5
Bart's modular boot disk 5
Virtual Linux 5
Connecting the notebooks hard disk to a forensic system 5
Media Analysis of System 8
My Forensic System 8
Mounting the Images for Analysis 8
System Commands Used 8
Search for modifications to operating system software or configuration 9
Searching setuid and setgid files 14
Check for hidden directories 16
User profile/Habits 16
Search for backdoors and RootKits 17
Examine file system for any sign of a sniffer program 17
Internet history file and other history files 18
System Registry or /proc examination 20
Show start up files and processes 20
MACTime Analysis 21
OS Version and Install Date 21
Major Updates 22
Last time the system was used 23
Recover Deleted Files 25
String Search 26
User Information/Conclusions 28
Chain of Custody 29
References 29
Appendix A 30
Appendix B 30
Appendix C 33
GCFA Binary Analysis (Part 2) 36
Executive Summary 36
Binary Details 36
Name of the program/file found on the system 36
File/MACTime. 36
File owner(s) – (user and/or group). 37
File size (in bytes). 37
MD5 hash of the file 37

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Other File Information 37
Key words found that are associated with the program/file. 38
Program Description 40
Program function 40
Confirmation of above 40
Forensic Details 41
What is happening? 43
Program Identification 43
Legal Implications 45
Prosecution by Law 45
Corporate Policy Violations 45
Interview Questions 45
Question One 45
Question Two 45
Question Three 45
Question Four (Assuming the individual is working for the company) 46
Question Five 46
Reference 46
Legal Issues of Incident Handling 47
Wiretap Statute 47
System Banners 48
Reference 49
Appendix A 50

© SANS Institute 2000 - 2002, Author retains full rights.

Forensic Analysis of an unknown System

Synopsis of Case Facts

A co-op student working for us was interested in computer security. He had installed Linux on his office notebook. Since his co-op term had ended, I decided to do a forensic analysis of his Linux system to see if he had installed any security, vulnerability assessment or "hacker" related tools. This would also be an excellent opportunity for me to gain forensic analysis skills.

Description of the system

The co-op student worked in the service development group and had unrestricted access to our corporate network and a DMZ LAN. This DMZ was configured with ingress filtering only, the IP addresses on the DMZ are statically assigned by dhcp.(I.E. the Ethernet Address of the network card was assigned a specific IP address in the dhcp server.) The system is a DELL Latitude Cpi D300XT notebook. It has a Pentium II 300Mhz with 128Mb of RAM, a 4Gb hard disk and a removable CDROM.
The corporate OS is Windows 2000, this notebook had an NTFS partition and two Linux partitions and a Linux swap partition. The system was used as a desktop, but was also taken to the LAB when testing was required.

Hardware Description

The following computer system and all of it components were located in the draw of the co-op desk; it was used as his desktop computer.

Tag # LGLY01's Description
The system in question is a DELL Latitude Cpi D300XT, Pentium II 300Mhz with 128MB of RAM, an IBM 4320MB internal IDE hard disk and DELL Latitude CP IDE cdrom drive. There was also a 3COM Megahertz 556 PCMCIA Card, it has a 10/100Mbit network interface and a 56k modem. System Serial # DP/N 0009321C-12800-92G-1593

Tag # LGLY01-01: IBM Travelstar 4GN DKLA-24320 IDE Hard Drive, Serial #: DP/N 0006212D-12567-92A-3J3S, Speed 4200RPM, Size: 4320MB

Tag # LGLY01-02: DELL Latitude CP removable IDE CDROM Drive, Serial #: DP/N 00089314

Tag # LGLY01-03: 3COM Megahertz Lan 10/100Mbit + 56K Modem PCMCIA card Model 3CCFEM556B, Serial #: 108T16Q91R5E

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Image the Media

The notebook was powered down and in a draw, so the was no opportunity to capture the more volatile system information, such as system memory, network connection and running processes.

My forensics system is a Linux notebook; unfortunately it can't take a second IDE hard drive, this would have made imaging the hard disk much easier. There was three ways that I attempted to get an image of this system. The first was to connect a diskette drive and boot from a special boot disk that would bring up the PCMCIA network card. The second was to boot from Virtual Linux on a cdrom and the third was to remove the disk and connect it to another system using a standard IDE interface with a special converter for the mini IDE connector on the notebooks hard drive.

Bart's modular boot disk

Barts modular boot disk is an excellent DOS booting system. It allows you to easy create a boot diskette with the required drives to enable the cdrom drive and the networking interface. It uses Microsoft's TCP/IP and LAN client. This allows you to logon to a Windows network and map a network drive. Perfect, I can boot from the diskette and map a network drive, were I can write the image too.
Unfortunately cygwin Unix tool such as dd, netcat (nc) and md5sum will not work under DOS and the SANS Forensics Track 8 cdrom does not have these tools in the Windows95/98 response kit. I resorted to searching the web and found two versions of dd for DOS, but I was not able to get them to read a hard disk partition as the "in file".

Virtual Linux

This was a great idea, after searching the web; I found a Virtual Linux 1.0 ISO image. I created a bootable cdrom and in my eagerness to get an image of the disk I did not test it on my own forensic notebook. This proves to be a big mistake as I found that the Virtual Linux automatically searches for hard disk partitions and mounts them in read/write mode. I noticed this right away and unmounted the partitions immediately. This was a great idea and given time I may attempt to create my own version of Virtual Linux as a forensic imaging OS.

Connecting the notebooks hard disk to a forensic system

As mentioned above, my forensic system is a notebook and it can't take a second hard drive. I phoned around to locate a computer cable accessory supplier and found an inexpensive converter for mini IDE to regular IDE cable (under C$5). Now I could connect the hard drive from the notebook to a system with a standard IDE controller. This

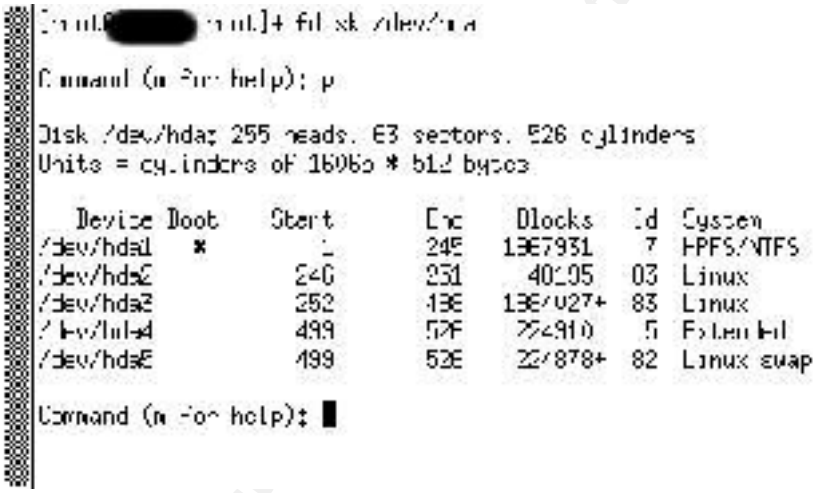
© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

would make life much easier. I used a Sun Ultra 10 system running Linux to create the images.

Procedure

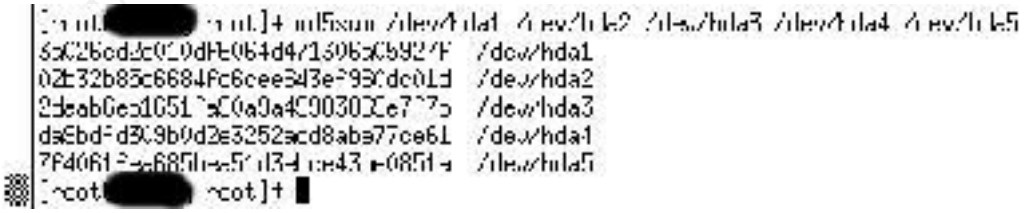
- Remove the hard drive from the notebooks. Remove hard drive from the mounting frame.
- Connect adapter to hard drive and connect to my forensic system with a standard IDE channel.
- Before I connected the hard drive I was going to image, I made sure that the system would not attempt to access or mount the drive I wanted to image.
- Powered up the forensic system and double-checked that the notebooks hard drive was not mounted.
- Print a copy of the partition table
    - From the solaris format command
    - Fdisk /dev/hda (Could have also used fdisk –l /dev/hda)

Print Screen of Partition table



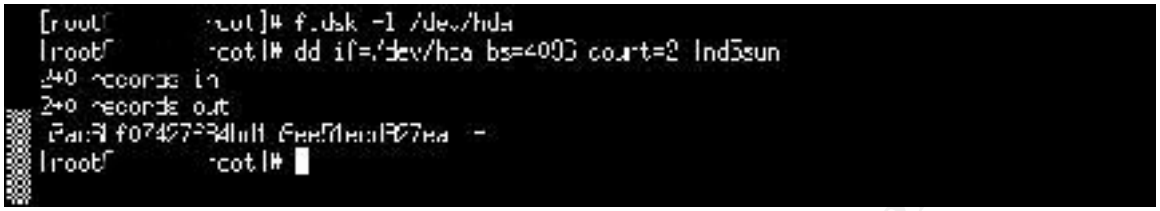
- Produced a md5 hash to so I can show that the data on this drive will not change from here on.
    - Md5sum /dev/hda1 /dev/hda2 /dev/hda3 /dev/hda4 /dev/hda5

Print Screen of above



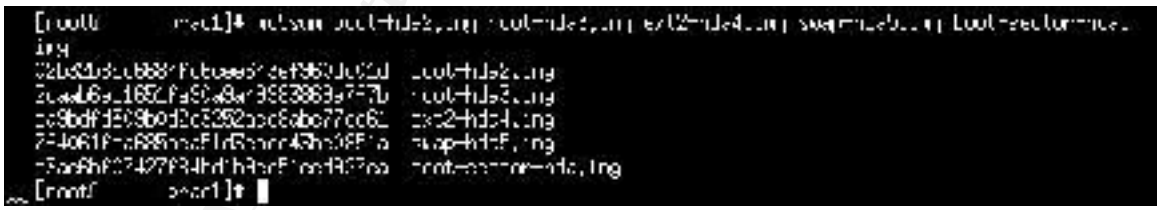
© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Print Screen of boot Sector md5sum

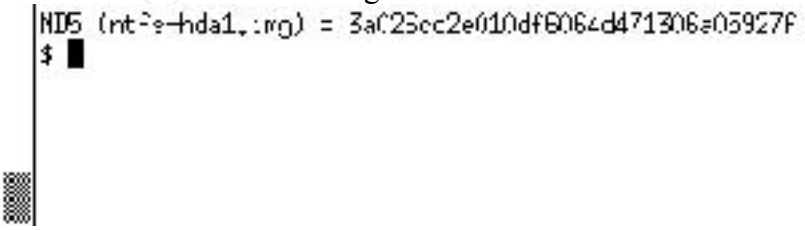


- Create images of the various partitions
  - dd if=/dev/hda1 | nc x.x.x.y 625
    - Remote system: nc –l –p 625 >ntfs-hda1.img
    - No space for all of the images on my Linux system
  - dd if=/dev/hda2 | nc x.x.x.x 625
    - Remote system: nc –l –p 625 >boot-hda2.img
  - dd if=/dev/hda3 | nc x.x.x.x 625
    - Remote system: nc –l –p 625 >root-hda3.img
  - dd if=/ dev/hda4 | nc x.x.x.x 625
    - Remote system: nc –l –p 625 >ext-part-hda4.img
  - dd if=/ dev/hda5 | nc x.x.x.x 625
    - Remote system: nc –l –p 625 >swap-hda4.img
  - dd if=/ dev/hda bs=4092 count=2 | nc x.x.x.x 625 (copy the boot sector)
    - Remote system: nc –l –p 625 >boot-sector-hda.img

Print Screen of above



The ntfs partition was placed on a different system as I had disk space issues. Below is the md5sum of ntfs-hda1.img



© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Media Analysis of System

My Forensic System

The forensic system I used was also a Dell Latitude Notebook with a 12GB internal hard drive. Redhat Linux 7.2 was installed as the base operating system. I am the only user of this system. I installed a few forensic tools such as TASK and Autopsy. I will explain these later.

Mounting the Images for Analysis

In order to examine the file systems, I mounted the images (on my Linux forensic system) using a loop device. The images were mounted as read-only with no execute permission, no set UID programs, no device drivers and no access time modification. I also set the time zone to match the system being analysed (America/Vancouver)

Commands used:
mount -t ext2 -o ro,loop,nosuid,noexec,nodev,noatime /home/userid/sans-gcfa/prac1/root-hda3.img /mnt/root-image

mount -t ext2 -o ro,loop,nosuid,noexec,nodev,noatime /home/userid/sans-gcfa/prac1/boot-hda4.img /mnt/user-image

export TZ=America/Vancouver

System Commands Used

Below is a brief explanation of some of the system commands used in the analysis. Why the command is used will be explained as it is used.

find
This command searches for files in a hierarchy file/directory system. There are many different arguments can be used to customize and narrow the search criteria. We also use the –printf subcommand to format the output of the search results.
Formatting used:

- %T@ - last modification time print is seconds since Jan. 1, 1970, 00:00 GMT
- %Tc - locale's date and time
- %k - print the size
- %h - print the directory
- %f - print the filename
- \n - new line

rpm
This is the **R**edhat **P**ackage **M**anagement application. It was used to verify the files installed on the system.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

file
This is used in an attempt to classify the file. There are three test performed and the first one to match will be returned. This helps identify whether the file is a file or a directory and the contents of the file.

grep
Search to a pattern and prints matches

egrep
Searches for pattern(s) using regular expressions and prints the matches.

Search for modifications to operating system software or configuration

The following command was used to search for files, where the owner is root, the file has execute permission and the file was newer than the date of the install.log file. (29 May 2002)

find /mnt/root-image -newer /mnt/root-image/root/install.log -type f -user root –perm\
+111 > -printf "%T@ %k %h/%f\n" |sort

Results:
1022747642 8 /mnt/root-image/usr/share/rhn/rhn_applet/rhn_utils.pyc
1022747643 32 /mnt/root-image/usr/lib/python1.5/site-packages/xmlrpclib.pyc
1022747645 24 /mnt/root-image/usr/lib/python1.5/site-packages/cgiwrap.pyc
1022747646 8 /mnt/root-image/usr/share/rhn/rhn_applet/rhn_applet_rpm.pyc
1022747647 12 /mnt/root-image/usr/share/rhn/up2date_client/config.pyc
1022747647 28 /mnt/root-image/usr/share/rhn/rhn_applet/rhn_applet.pyc
1022747649 12 /mnt/root-image/usr/share/rhn/rhn_applet/rhn_applet_model.pyc
1022747649 24 /mnt/root-image/usr/share/rhn/rhn_applet/rhn_applet_dialogs.pyc
1022747649 4 /mnt/root-image/usr/share/rhn/rhn_applet/rhn_applet_animation.pyc
1022747649 8 /mnt/root-image/usr/share/rhn/rhn_applet/rhn_applet_rpc.pyc
1022747658 8 /mnt/root-image/root/.gnome/metadata.db
1022747799 8 /mnt/root-image/root/.gconfd/saved_state
1022790623 24 /mnt/root-image/usr/share/rhn/register/rhnreg.pyc
1022790623 8 /mnt/root-image/usr/share/rhn/register/config.pyc
1022790624 12 /mnt/root-image/usr/share/rhn/register/translate.pyc
1022790625 8 /mnt/root-image/usr/share/rhn/register/configdlg.pyc
1022790630 12 /mnt/root-image/usr/share/rhn/register/hardware.pyc
1022790630 28 /mnt/root-image/usr/share/rhn/register/gui.pyc
1022790630 4 /mnt/root-image/usr/share/rhn/register/progress.pyc
1022790630 8 /mnt/root-image/usr/share/rhn/register/checklist.pyc

There is nothing out of the ordinary.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Doing the same for the boot file system showed no newer files.

Verifying the packages with RPM

This is a Redhat Linux distribution, which uses RPM package management system. RPM has a database, which maintains a record of all packages and files associated with each package. Among other things, RPM verify compares the size, MD5 sum, permissions, type, owner and group of each file. We can use this to verify whether files installed were changed. RPM has an option to change the root directory, effectively chrooting to a new root directory. This allows us to verify the files on the mounted image against the rpm database within the mounted image.

Command: rpm --root /mnt/root-image -Va >rpm-verify

Results where the md5 hash does not match:

```
S.5....T c /etc/sysconfig/rhn/rhn-applet
S.5....T   /usr/share/rhn/rhn_applet/rhn_applet.pyc
S.5....T   /usr/share/rhn/rhn_applet/rhn_applet_animation.pyc
S.5....T   /usr/share/rhn/rhn_applet/rhn_applet_dialogs.pyc
S.5....T   /usr/share/rhn/rhn_applet/rhn_applet_model.pyc
S.5....T   /usr/share/rhn/rhn_applet/rhn_applet_rpc.pyc
S.5....T   /usr/share/rhn/rhn_applet/rhn_applet_rpm.pyc
S.5....T   /usr/share/rhn/rhn_applet/rhn_utils.pyc
S.5....T   /usr/share/texmf/web2c/mf.base
S.5....T   /usr/share/texmf/web2c/mf.log
S.5....T   /usr/share/texmf/web2c/mpost.log
..5....T   /usr/share/texmf/web2c/mpost.mem
S.5....T c /etc/crontab
S.5....T   /usr/share/fonts/ja/TrueType/fonts.dir
S.5....T   /usr/share/fonts/ja/TrueType/fonts.scale
S.5....T c /etc/ldap.conf
S.5....T c /etc/pam.d/system-auth
S.5....T c /etc/mail/statistics
S.5....T c /usr/share/a2ps/afm/fonts.map
S.5....T   /usr/share/rhn/register/checklist.pyc
S.5....T   /usr/share/rhn/register/configdlg.pyc
S.5....T   /usr/share/rhn/register/gui.pyc
S.5....T   /usr/share/rhn/register/progress.pyc
S.5....T   /usr/share/rhn/up2date_client/config.pyc
S.5....T c /etc/xinetd.d/telnet
S.5....T c /etc/sysconfig/pcmcia
S.5..... c /etc/rndc.conf
S.5..... c /etc/rndc.key
..5....T c /etc/mime.types
S.5....T c /etc/openldap/ldap.conf
..5....T c /etc/inittab
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

S.5....T /usr/lib/python1.5/site-packages/cgiwrap.pyc
S.5....T /usr/lib/python1.5/site-packages/xmlrpclib.pyc
S.5....T c /etc/krb.conf
S.5....T c /etc/printcap
S.5....T c /etc/hotplug/usb.usermap
..5....T c /etc/sysconfig/redhat-config-users
SM5....T /usr/share/redhat-config-users/groupProperties.pyc
SM5....T /usr/share/redhat-config-users/groupWindow.pyc
SM5....T /usr/share/redhat-config-users/helpBrowser.pyc
SM5....T /usr/share/redhat-config-users/mainWindow.pyc
SM5....T /usr/share/redhat-config-users/userProperties.pyc
SM5....T /usr/share/redhat-config-users/userWindow.pyc
SM5....T c /etc/sysconfig/rhn/rhn_register
S.5....T /usr/share/rhn/register/config.pyc
S.5....T /usr/share/rhn/register/hardware.pyc
S.5....T /usr/share/rhn/register/rhnreg.pyc
S.5....T /usr/share/rhn/register/translate.pyc
S.5....T c /etc/xml/catalog
S.5....T c /usr/share/sgml/docbook/xmlcatalog

After listing all of the above files, most of them are configuration files and only a few are executable. I ran "file" against these executable files and found that they were python compiled files. The date stamp is within one day of the systems install date. I compared this list against a known clean system and confirmed that these same files also have the md5 flag set when doing a RPM verify on the clean system.

-rwxr-xr-x 1 root root 22777 May 30 01:34 /mnt/root-image/usr/lib/python1.5/site-packages/cgiwrap.pyc
-rwxr-xr-x 1 root root 30675 May 30 01:34 /mnt/root-image/usr/lib/python1.5/site-packages/xmlrpclib.pyc
-rwxr-xr-x 1 root root 5919 May 30 13:30 /mnt/root-image/usr/share/rhn/register/checklist.pyc
-rwxr-xr-x 1 root root 5692 May 30 13:30 /mnt/root-image/usr/share/rhn/register/configdlg.pyc
-rwxr-xr-x 1 root root 4869 May 30 13:30 /mnt/root-image/usr/share/rhn/register/config.pyc
-rwxr-xr-x 1 root root 24730 May 30 13:30 /mnt/root-image/usr/share/rhn/register/gui.pyc
-rwxr-xr-x 1 root root 8773 May 30 13:30 /mnt/root-image/usr/share/rhn/register/hardware.pyc
-rwxr-xr-x 1 root root 2205 May 30 13:30 /mnt/root-image/usr/share/rhn/register/progress.pyc
-rwxr-xr-x 1 root root 21293 May 30 13:30 /mnt/root-image/usr/share/rhn/register/rhnreg.pyc

© SANS Institute 2000 - 2002, Author retains full rights.

```
-rwxr-xr-x    1 root     root        11258 May 30 13:30 /mnt/root-
image/usr/share/rhn/register/translate.pyc
-rwxr-xr-x    1 root     root         4025 May 30 01:34 /mnt/root-
image/usr/share/rhn/rhn_applet/rhn_applet_animation.pyc
-rwxr-xr-x    1 root     root        21707 May 30 01:34 /mnt/root-
image/usr/share/rhn/rhn_applet/rhn_applet_dialogs.pyc
-rwxr-xr-x    1 root     root        10980 May 30 01:34 /mnt/root-
image/usr/share/rhn/rhn_applet/rhn_applet_model.pyc
-rwxr-xr-x    1 root     root        27910 May 30 01:34 /mnt/root-
image/usr/share/rhn/rhn_applet/rhn_applet.pyc
-rwxr-xr-x    1 root     root         5946 May 30 01:34 /mnt/root-
image/usr/share/rhn/rhn_applet/rhn_applet_rpc.pyc
-rwxr-xr-x    1 root     root         5553 May 30 01:34 /mnt/root-
image/usr/share/rhn/rhn_applet/rhn_applet_rpm.pyc
-rwxr-xr-x    1 root     root         5942 May 30 01:34 /mnt/root-
image/usr/share/rhn/rhn_applet/rhn_utils.pyc
-rwxr-xr-x    1 root     root         9018 May 30 01:34 /mnt/root-
image/usr/share/rhn/up2date_client/config.pyc
```

Create the list of configuration files with RPM

We can also use RPM to list all the files, which are tagged as configuration files by RPM.

Command: rpm --root /mnt/root-image -qac >rpm-config-files
head rpm-verify.md5.filesonly rpm-config-files

Below is a sample of the files that I compared.

```
==> rpm-verify.md5.filesonly <==
rhn-applet
rhn_applet.pyc
rhn_applet_animation.pyc
rhn_applet_dialogs.pyc
rhn_applet_model.pyc
rhn_applet_rpc.pyc
rhn_applet_rpm.pyc
rhn_utils.pyc
mf.base
mf.log

==> rpm-config-files <==
/etc/makedev.d/00macros
/etc/makedev.d/ataraid
/etc/makedev.d/cciss
/etc/makedev.d/cdrom
/etc/makedev.d/console
/etc/makedev.d/dac960
```

© SANS Institute 2000 - 2002 Author retains full rights.

/etc/makedev.d/ftape
/etc/makedev.d/generic
/etc/makedev.d/ia64
/etc/makedev.d/ibcs

I compared the list of configuration files against the list of files where the md5 sum had changed. I used the following script:

```
for file in `cat rpm-verify.md5.filesonly`; do
 grep -i $file rpm-config-files
 done
```

Results:
/etc/sysconfig/rhn/rhn-applet
/etc/crontab
/etc/anacrontab
/usr/share/fonts/default/Type1/fonts.dir
/usr/share/fonts/default/Type1/fonts.scale
/etc/ldap.conf
/etc/openldap/ldap.conf
/etc/pam.d/system-auth
/etc/mail/statistics
/usr/share/a2ps/afm/fonts.map
/etc/xinetd.d/telnet
/etc/pcmcia/config
/etc/pcmcia/config.opts
/etc/pcmcia/ftl.opts
/etc/pcmcia/ide.opts
/etc/pcmcia/memory.opts
/etc/pcmcia/parport.opts
/etc/pcmcia/scsi.opts
/etc/pcmcia/serial.opts
/etc/pcmcia/wireless.opts
/etc/sysconfig/pcmcia
/etc/rndc.conf
/etc/rndc.key
/etc/tux.mime.types
/etc/mime.types
/etc/ldap.conf
/etc/openldap/ldap.conf
/etc/inittab
/etc/krb.conf
/etc/printcap.local
/etc/printcap
/etc/hotplug/usb.usermap
/etc/X11/applnk/System/redhat-config-users.desktop
/etc/X11/sysconfig/redhat-config-users.desktop

© SANS Institute 2000 - 2002, Author retains full rights.

/etc/pam.d/redhat-config-users
/etc/security/console.apps/redhat-config-users
/etc/sysconfig/redhat-config-users
/usr/share/pixmaps/redhat-config-users.png
/etc/pam.d/rhn_register
/etc/security/console.apps/rhn_register
/etc/sysconfig/rhn/rhn_register
/etc/xml/catalog
/usr/share/sgml/docbook/xmlcatalog
/usr/share/sgml/docbook/xmlcatalog

The above shows some duplicates, but this not a concern as they are all still configuration files. After examining the configuration files (no execute permission) and are tagged by RPM as a configuration file, the only one that stands out is /etc/xinetd.d/telnet. This is a configuration file for the xinetd super-server. The xinetd daemon will listen on port 21, which is actually the ftp port, not the telnet port (tcp port 23).

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet_21
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        port            = 21
        disable         = no
}
```

Searching setuid and setgid files

I used the "find" command to search the mounted image for file that have the setuid and setgid bits set.

Command used:

- find /mnt/root-image \(-perm -004000 -o -perm -002000 \) -type f –ls
- find /mnt/boot-image \(-perm -004000 -o -perm -002000 \) -type f –ls

```
172133   36 -rwsr-xr-x   1 root     root       34296 Mar 27 17:40 /mnt/root-image/usr/bin/chage
172135   36 -rwsr-xr-x   1 root     root       36100 Mar 27 17:40 /mnt/root-image/usr/bin/gpasswd
172393   40 -rwsr-xr-x   1 root     root       37528 Jan 17  2002 /mnt/root-image/usr/bin/at
172497   20 -rwxr-sr-x   1 root     mail       17811 Mar 25 10:03 /mnt/root-image/usr/bin/lockfile
172610   28 -rwxr-sr-x   1 root     slocate    25020 Jun 24  2001 /mnt/root-image/usr/bin/slocate
172728   16 -r-s--x--x   1 root     root       15104 Mar 13 17:44 /mnt/root-image/usr/bin/passwd
```

© SANS Institute 2000 - 2002, Author retains full rights.

```
172830   8 -r-xr-sr-x   1 root     tty        6920 Mar 14 12:24 /mnt/root-image/usr/bin/wall
172863  12 -rws--x--x   1 root     root      12072 Apr  1 15:26 /mnt/root-image/usr/bin/chfn
172864  12 -rws--x--x   1 root     root      11496 Apr  1 15:26 /mnt/root-image/usr/bin/chsh
172882   8 -rws--x--x   1 root     root       4764 Apr  1 15:26 /mnt/root-image/usr/bin/newgrp
172893  12 -rwxr-sr-x   1 root     tty        8584 Apr  1 15:26 /mnt/root-image/usr/bin/write
173033  24 -rwsr-xr-x   1 root     root      21080 Apr 14 21:49 /mnt/root-image/usr/bin/crontab
173119  20 -rwsr-xr-x   1 root     root      19927 Apr 17 07:59 /mnt/root-image/usr/bin/lppasswd
173683 220 -rwsr-xr-x   1 root     root     219932 Apr  4 19:27 /mnt/root-image/usr/bin/ssh
174890  16 -rwsr-xr-x   1 root     root      14588 Jul 24  2001 /mnt/root-image/usr/bin/rcp
174892  12 -rwsr-xr-x   1 root     root      10940 Jul 24  2001 /mnt/root-image/usr/bin/rlogin
174893   8 -rwsr-xr-x   1 root     root       7932 Jul 24  2001 /mnt/root-image/usr/bin/rsh
175787  88 ---s--x--x   1 root     root      84680 Apr 18 09:35 /mnt/root-image/usr/bin/sudo
176129  72 -r-xr-s--x   1 root     games     69034 Apr  9 08:28 /mnt/root-image/usr/bin/gnome-stones
176125  56 -r-xr-s--x   1 root     games     50487 Apr  9 08:28 /mnt/root-image/usr/bin/gataxx
176126  36 -r-xr-s--x   1 root     games     35697 Apr  9 08:28 /mnt/root-image/usr/bin/glines
176127  84 -r-xr-s--x   1 root     games     80354 Apr  9 08:28 /mnt/root-image/usr/bin/gnibbles
176128  92 -r-xr-s--x   1 root     games     89644 Apr  9 08:28 /mnt/root-image/usr/bin/gnobots2
176131  36 -r-xr-s--x   1 root     games     35822 Apr  9 08:28 /mnt/root-image/usr/bin/gnotravex
176130  88 -r-xr-s--x   1 root     games     84726 Apr  9 08:28 /mnt/root-image/usr/bin/gnomine
176135  60 -r-xr-s--x   1 root     games     56608 Apr  9 08:28 /mnt/root-image/usr/bin/mahjongg
176132  32 -r-xr-s--x   1 root     games     30954 Apr  9 08:28 /mnt/root-image/usr/bin/gnotski
176133 248 -r-xr-s--x   1 root     games    248343 Apr  9 08:28 /mnt/root-image/usr/bin/gtali
176134  68 -r-xr-s--x   1 root     games     62770 Apr  9 08:28 /mnt/root-image/usr/bin/iagno
176136  32 -r-xr-s--x   1 root     games     31514 Apr  9 08:28 /mnt/root-image/usr/bin/same-gnome
177761 132 -r-sr-sr-x   1 uucp     uucp     129908 Feb  1  2002 /mnt/root-image/usr/bin/cu
177762  96 -r-sr-xr-x   1 uucp     uucp      91720 Feb  1  2002 /mnt/root-image/usr/bin/uucp
177764  40 -r-sr-sr-x   1 uucp     uucp      39108 Feb  1  2002 /mnt/root-image/usr/bin/uuname
177766 104 -r-sr-xr-x   1 uucp     uucp     101460 Feb  1  2002 /mnt/root-image/usr/bin/uustat
177768  96 -r-sr-xr-x   1 uucp     uucp      93512 Feb  1  2002 /mnt/root-image/usr/bin/uux
 1676   12 -rws--x--x   1 69       root       8291 Apr 12 14:27 /mnt/root-image/usr/lib/mc/bin/cons.saver
31436   32 -rwsr-xr-x   1 root     root      32673 Apr 18 14:40 /mnt/root-image/usr/sbin/ping6
31440   16 -rwsr-xr-x   1 root     root      13994 Apr 18 14:40 /mnt/root-image/usr/sbin/traceroute6
33809    8 -rwxr-sr-x   1 root     utmp       6604 Jun 24  2001 /mnt/root-image/usr/sbin/utempter
33841  448 -r-sr-xr-x   1 root     root     451280 Apr  8 03:55 /mnt/root-image/usr/sbin/sendmail.sendmail
33844   24 -rws--x--x   1 root     root      22388 Apr 15 15:15 /mnt/root-image/usr/sbin/userhelper
33857   20 -rwsr-xr-x   1 root     root      17461 Apr 19 09:35 /mnt/root-image/usr/sbin/usernetctl
33913   16 -rwxr-sr-x   1 root     utmp      13760 Apr 15 14:47 /mnt/root-image/usr/sbin/gnome-pty-helper
34970   12 -rwsr-xr-x   1 root     root      10492 Apr 17 09:10 /mnt/root-image/usr/sbin/userisdnctl
34975   20 -rwsr-xr-x   1 root     root      20140 Mar 14 16:22 /mnt/root-image/usr/sbin/traceroute
35021   16 -rwxr-sr-x   1 root     lock      13573 Feb 25  2002 /mnt/root-image/usr/sbin/lockdev
35254   24 -r-s--x---   1 root     48        22826 Apr  9 11:56 /mnt/root-image/usr/sbin/suexec
37820  228 -r-sr-sr-x   1 uucp     uucp     227712 Feb  1  2002 /mnt/root-image/usr/sbin/uucico
37823  108 -r-sr-sr-x   1 uucp     uucp     103776 Feb  1  2002 /mnt/root-image/usr/sbin/uuxqt
189035 1572 -rws--x--x   1 root     root    1602576 Apr 18 20:12 /mnt/root-image/usr/X11R6/bin/XFree86
186453  36 -rwsr-xr-x   1 root     root      35192 Apr 18 14:40 /mnt/root-image/bin/ping
186868  64 -rwsr-xr-x   1 root     root      60104 Apr  1 15:26 /mnt/root-image/bin/mount
186869  32 -rwsr-xr-x   1 root     root      30664 Apr  1 15:26 /mnt/root-image/bin/umount
187006  20 -rwsr-xr-x   1 root     root      19116 Apr  8 09:02 /mnt/root-image/bin/su
124524 124 -r-sr-xr-x   1 root     root     120264 Apr  9 20:24 /mnt/root-image/sbin/pwdb_chkpwd
124525  20 -r-sr-xr-x   1 root     root      16992 Apr  9 20:24 /mnt/root-image/sbin/unix_chkpwd
124622  16 -rwxr-sr-x   1 root     root      14657 Apr 19 09:35 /mnt/root-image/sbin/netreport
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Check for hidden directories

We will check for hidden directory, as this is a common technique used by hackers to avoid detection.
Command: find /mnt/root-image -name ".*" -type d -printf "%Tc %h/%f\n" >hidden-dir.root

```
Mon 10 Jun 2002 04:59:09 PM PDT /mnt/root-image/tmp/.sawfish-steve
Mon 10 Jun 2002 03:48:39 PM PDT /mnt/root-image/tmp/.font-unix
Mon 10 Jun 2002 04:59:22 PM PDT /mnt/root-image/tmp/.X11-unix
Mon 10 Jun 2002 04:59:09 PM PDT /mnt/root-image/tmp/.ICE-unix
Wed 29 May 2002 05:40:36 PM PDT /mnt/root-image/etc/skel/.kde
Thu 30 May 2002 01:36:14 AM PDT /mnt/root-image/root/.gnome
Thu 30 May 2002 01:33:23 AM PDT /mnt/root-image/root/.gnome_private
Thu 30 May 2002 01:36:39 AM PDT /mnt/root-image/root/.gconfd
Thu 30 May 2002 01:36:39 AM PDT /mnt/root-image/root/.gconf
Thu 30 May 2002 01:33:55 AM PDT /mnt/root-image/root/.sawfish
Thu 30 May 2002 01:34:05 AM PDT /mnt/root-image/root/.nautilus
Thu 30 May 2002 01:34:25 AM PDT /mnt/root-image/root/.gnome-desktop
Wed 29 May 2002 05:11:11 PM PDT /mnt/root-image/usr/share/control-center/.data
Wed 29 May 2002 06:24:55 PM PDT /mnt/root-image/home/steve/.kde
Mon 10 Jun 2002 04:59:09 PM PDT /mnt/root-image/home/steve/.gnome
Thu 30 May 2002 09:48:17 AM PDT /mnt/root-image/home/steve/.gnome_private
Mon 10 Jun 2002 04:59:32 PM PDT /mnt/root-image/home/steve/.gconfd
Mon 10 Jun 2002 04:59:32 PM PDT /mnt/root-image/home/steve/.gconf
Thu 30 May 2002 09:48:43 AM PDT /mnt/root-image/home/steve/.sawfish
Thu 30 May 2002 02:39:24 PM PDT /mnt/root-image/home/steve/.nautilus
Thu 30 May 2002 09:49:11 AM PDT /mnt/root-image/home/steve/.gnome-desktop
Thu 30 May 2002 01:57:28 PM PDT /mnt/root-image/home/steve/.mozilla
```

All of these directories are consistent with that of user who is using X windows.

To take a closer look, I did a recursive long listing of the above directories. I did not find anything out of the ordinary. There are only two active uses on this system root, UID=0 and steve, UID=500.
Command: ls -lR `cat hidden-dir.root |cut -d " " -f 8` |less

User profile/Habits

I looked closely at the following directories to understand the users habits:

/mnt/root-image/home/steve/.gnome/panel.d/default/launchers:

There were three files under this directory; this is the Gnome task bar config directory. This user has three quick launch icons. “Start Here”, “Gnome-Terminal” and “Mozilla”.

The “Start Here” icon execute “nautilus start-here:” and uses an icon named “gnome-starthere.png”
The “Gnome-Terminal” icon execute “gnome-terminal --use-factory --start-factory-server” and uses an icon named “gnome-terminal.png”
The “Mozilla” icon executes “/usr/bin/mozilla” and uses an icon named “mozilla-icon.png”.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

/mnt/root-image/home/steve/.gnome-desktop:

There are similar icons on the users desktop: “Start here”, “Trash” and “Steve’s home”.

/mnt/root-image/home/steve:

In the users home directory, there were a few interesting scripts. These scripts were used to bulk load the inetd and/or xinetd configuration file(s) with a telnet server listening on a range of ports. There was also an expect script which would open a connection on a range of tcp ports and log success and failures of a telnet server listening on the given range of ports.

This is the list of files:

- makeRecords-inetd.pl
- makeRecords-services.pl
- makeRecords-xinetd.d.pl
- telnetConnectScan.exp

Please See Appendix C for the content of the scripts.

Search for backdoors and RootKits

Chkrookit is a collection of C programs that are used to search your system for signs of trojan binaries, rootkits, worms, sniffer logs and other anomalies. It can be used on a running system as a quick check for rootkits. I used it to search through a mounted image. The –r switch tells chkrootkit where to start its search, the –q switch is used to only print if there is a possible hit on one of the searches. This software can be downloaded from http://www.chkrootkit.org/

Command: ./chkrootkit -q -r /mnt/root-image
./chkrootkit -q -r /mnt/boot-image

The mounted images came up clean

Examine file system for any sign of a sniffer program

I user three techniques to check for signs of a sniffer; First, I used RPM to see if any sniffer related packages were install. Secondly I grep’d through a strings of the image and thirdly, I grep’d through a recursive listing of the files on the image.

```
rpm --root /mnt/root-image -qa |egrep
'snort|ethereal|tcpdump|libpcap|linsniff|admsniff|sniff|analyse'
libpcap-0.6.2-12
ethereal-gnome-0.9.3-3
ethereal-0.9.3-3
tcpdump-3.6.2-12
```

© SANS Institute 2000 - 2002, Author retains full rights.

egrep 'snort|ethereal|tcpdump|libpcap|linsniff|admsniff|sniff|analyse' root-hda3.img.str

ls -R |egrep 'snort|ethereal|tcpdump|libpcap|linsniff|admsniff|sniff|analyse' >~userid/sans-gcfa/prac1/sniff.root-image

After examining the outputs from the above two searches, I came up with the following. All of these were installed during the system build.
results:

Libpcap	- Gnu packet capturing library.
Tcpdump	- tcpdump is a network sniffer.
Tcpslice	- Used to extract or combine pieces of tcpdump trace file. It is part of the tcpdump rpm package.
Ethereal	- An advanded network analyser gui.
Tethereal	- Part of Ethereal, text mode of ethereal
pppoe-sniff	- PPP over Ethernet sniffer used for trouble-shooting ADSL and PPPoE connections. It is part of the rp-pppoe-3.3-7 rpm package.

Internet history file and other history files

Browser history and cache files

/mnt/root-image/home/steve/.mozilla/default/ud3i7sik.slt:

This is where we find the users browser configuration and may be able to find the surfing habits of the user, "steve".

The bookmarks.html file is the base file with no user created additions.

There were two cookies in the cooktxt file:
.google.com TRUE / FALSE 2147368663 PREF ID=234dfaf6567d498f:TM=1022794586:LM=1022794586:S=sDRKvXDm1qE
.google.ca TRUE / FALSE 2147368662 PREF ID=31e588361a521dbf:LD=en:TM=1022794587:LM=1022794587:S=sZaHKbVoGYc

The "history.dat" file shows which URL's the user typed in at the URL dialogue window in the mozilla browser. He surfed to google and entered the following search: "linux telnet server"
This is from the history file
"(8C=http://www.google.ca/search?q=linux+telnet+server&hl=en&meta=)
(8D=1022794814802876)"

I looked in the users mozilla cache directory; most of the files were gif images from the www.google.com homepage. Two of the files were gzip compressed: 5A68EACFd01 and E55CDB0Cd01. After uncompressing the files, I ran "file" against them.

© SANS Institute 2000 - 2002, Author retains full rights.

gzip -cd 5A68EACFd01 >/tmp/5A68EACFd01.dell
gzip -cd E55CDB0Cd01 >/tmp/E55CDB0Cd01.dell
file /tmp/5A68EACFd01.dell /tmp/E55CDB0Cd01.dell
/tmp/5A68EACFd01.dell: HTML document text
/tmp/E55CDB0Cd01.dell: HTML document text

I viewed both with my browser. The first is googles homepage and second, E55CDB0Cd01, is the search results from the users google search for a Linux telnet server. Please Appendix B for this web page.

Users Command line history

There was nothing suspicious in the users command history

cat .bash_history
ls
ifconfig -a
[root@forensic01 steve]# pwd
/mnt/root-image/home/steve

Root Command line history

Examining roots command history reveals, that the user configured the network interface and then proceeded to look at the man pages for the Linux super server, inetd and/or xinetd. He also looked for the xinetd configuration files. Tested his connectivity to Google and ftp'd to one host.

cat .bash_history
shutdown -r 0
ifup eth0
tcsh
ifconfig -a
ifup eth0
ifconfig eth0 up
ifup eth0
ifup eth0
ping 198.x.x.x
ifdown eth0
ifup eth0
ifconfig -a
man inetd.cofn
man inetd
ls /etc/inetd.conf
cd /etc
ls
find
man find
find . -name inet
find . -name xinetd
find . -name xinetd*
ls /etc/xin*

© SANS Institute 2000 - 2002, Author retains full rights.

```
vi /etc/xinetd.conf
ps -ef | grep telnet
ps -ef | grep inet
cd /etc/xinetd.d
ls
ls -p
vi telnet
exit
traceroute www.google.com
tcsh
ifconfig
tcsh
exit
ifup eth0
/csh
/tcsh
tchs
tcsh
tcsh
ifdown eth0
tcsh
exit
shutdown -r 0
tcsh
ls
ifconfig -a
tcsh
tcsh
tcsh
ifconfig -a
tcsh
ftp fraser.XXX.XXX
ifconfig -a
tcsh
exit
tcsh
exit
tcsh
[root@forensic01 root]# pwd
/mnt/root-image/root
```

System Registry or /proc examination

This system was powered off when I started the forensic analysis. The /proc file system is actually memory and therefore was lost when the system was powered down. The /proc mount point in the image has nothing under it.

Show start up files and processes

If the system is brought up in single user mode only the following script will run:

S17keytable – Load the keyboard-mapping table (Contents check out Okay)

© SANS Institute 2000 - 2002, Author retains full rights.

S00single - Start single user mode and any other startup file in the /etc/rc1.d directory. (Contents check out Okay)

The system was configured to run at run-level 5, this is X Windows. We will check the startup files in /etc/rc5.d

S05kudzu, S06reconfig, S08ipchains S08iptables, S09isdn, S10network, S12syslog, S13portmap, S14nfslock, S17keytable, S20random, S24pcmcia, S25netfs, S26apmd, S28autofs, S56rawdevices, S56xinetd, S60lpd, S80sendmail, S85gpm, S90crond, S90xfs, S95anacron, S95atd, S97rhnsd, S98wine, S99local

All of the above files, except for S99local, are links to files in /etc/rc.d/init.d and these file check out fine using RPM verify. I also examined the /etc/rc.d/rc.sysinit file and found it to be okay.

The print screen below shows that the images I examined have not been altered in anyway.
Print Screen

MACTime Analysis

MAC Time analysis was generated with TASK V1.0 and Autopsy V1.5 from @Stake. The use of these tools makes generating a timeline based on MAC times very easy. Both programs compiled cleanly and were ready for use in minutes.

TASK is based on The Coroner's Toolkit (TCT) and TCTUTIL's. It is a suite of tools that can perform a low-level forensic analysis of a dd image of a partition.

Autopsy is a web based Gui fronted to the TASK tool set. Once you configure the fsmorgue file, you can use Autopsy to generate the timeline from the image file.

OS Version and Install Date

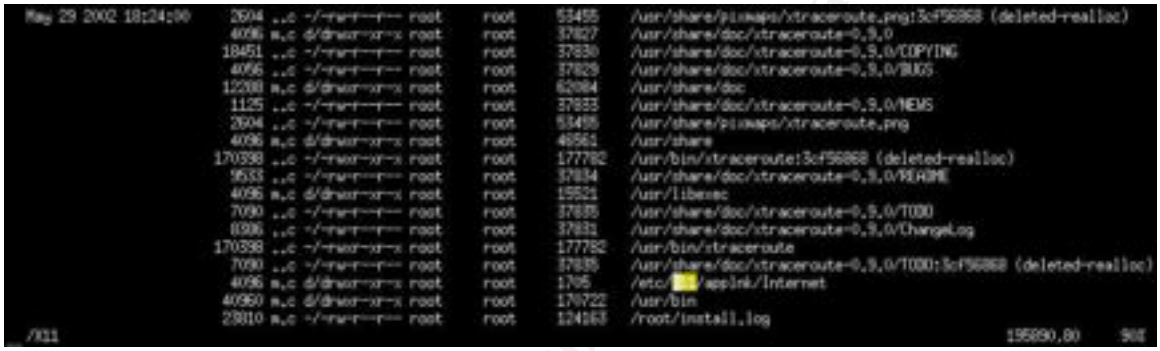
By looking at the /mnt/root-image/etc/redhat-release we can determine the release of Linux that was installed. "Red Hat Linux release 7.3 (Valhalla)". We can also determine the install date by looking at the date of the install.log and anaconda-ks.cfg files. This system was installed May 29 2002. By examining the anaconda-ks.cfg file we can determine that this system was configured to use America/Vancouver time zone when it was built and what packages were installed. This system had 782 RPM's installed. I confirmed this by using RPM query.
Command: rpm –root /mnt/root-image –qa |wc

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

pwd
/mnt/root-image/root
[root@forensic01 root]# cat ../etc/redhat-release
Red Hat Linux release 7.3 (Valhalla)

ls -l
total 28
-rw-r--r-- 1 root root 3976 May 29 18:26 anaconda-ks.cfg
-rw-r--r-- 1 root root 23810 May 29 18:24 install.log

mactime confirmation:
Print Screen



To confirm the time zone of this system, we can compare the /etc/localtime file with the time zone files in /usr/share/zoneinfo. I used md5sum of the two file to confirm this.

md5sum /mnt/root-image/etc/localtime /mnt/root-image/usr/share/zoneinfo/America/Vancouver
bd4333393bb17efb3d460d6441422405 /mnt/root-image/etc/localtime
bd4333393bb17efb3d460d6441422405 /mnt/root-image/usr/share/zoneinfo/America/Vancouver

Major Updates

This Redhat Linux 7.3 system was built on May 29th 2002 and no updates had been made to it. I used "rpm with the –last option to display all the package install dates, starting with the latest.

rpm --root /mnt/root-image -qa --last |head

xtraceroute-0.9.0-6	Wed 29 May 2002 06:24:01 PM PDT
xosview-1.7.3-10	Wed 29 May 2002 06:23:58 PM PDT
xdaliclock-2.18-8	Wed 29 May 2002 06:23:56 PM PDT
xcpustate-2.5-11	Wed 29 May 2002 06:23:55 PM PDT
wine-devel-20020327-1	Wed 29 May 2002 06:23:47 PM PDT
w3c-libwww-devel-5.3.2-5	Wed 29 May 2002 06:23:38 PM PDT
w3c-libwww-apps-5.3.2-5	Wed 29 May 2002 06:23:35 PM PDT
vsftpd-1.0.1-5	Wed 29 May 2002 06:23:32 PM PDT
vnc-doc-3.3.3r2-28	Wed 29 May 2002 06:23:28 PM PDT
vim-X11-6.1-2	Wed 29 May 2002 06:23:24 PM PDT

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Last time the system was used

The system was last used on June 10th 2002

```
Jun 10 2002 16:59:37    91464 .a. -/-rwxr-xr-x 0       0       124567   /sbin/insmod
                         8672 .a. -/-rwxr-xr-x 0       0       124585   /sbin/halt
                        30344 .a. -/-rwxr-xr-x 0       0       124604   /sbin/hwclock
                         8420 .a. -/-rwxr-xr-x 0       0       124587   /sbin/killall5
                           64 .a. -/-rw-r--r-- 0       0       224049   /etc/modules.conf
                            6 .a. l/lrwxrwxrwx 0       0       124575   /sbin/modprobe ->
insmod
                        85248 m.c -/-rw-rw-r-- 0       22      132063   /var/log/wtmp
```

To confirm the time this system was last used, we can look in the wtmp file. On Linux we can use the command "last –f " to examine a specific wtmp file.
Command: last -f /mnt/root-image/var/log/wtmp |less

```
shutdown ~~          2.4.18-3        Mon Jun 10 16:59 - 16:59  (00:00)
```

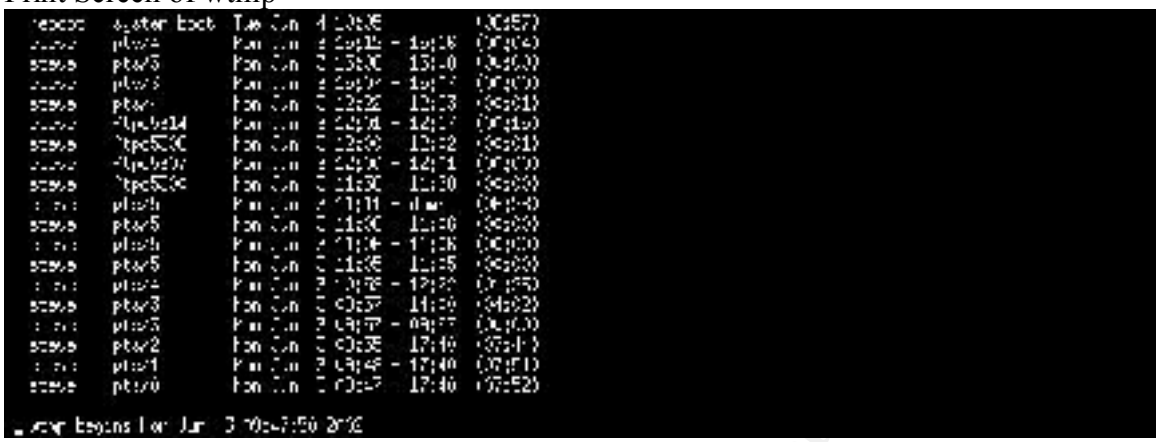
Print Screen

```
[root      root-image]# last -f var/log/wtmp -R -x |head
shutdown system down  Mon Jun 10 16:59 - 18:20 (91+01:21)
runlevel (to lvl 0)   Mon Jun 10 16:59 - 16:59  (00:00)
steve    pts/3        Mon Jun 10 16:06 - 16:06  (00:00)
steve    pts/2        Mon Jun 10 15:58 - 15:58  (00:00)
steve    pts/1        Mon Jun 10 15:57 - 15:57  (00:00)
steve    pts/0        Mon Jun 10 15:51 - 16:58  (01:07)
steve    :0           Mon Jun 10 15:49 - down   (01:09)
runlevel (to lvl 5)   Mon Jun 10 15:47 - 16:59  (01:11)
reboot   system boot  Mon Jun 10 15:47          (01:11)
shutdown system down  Mon Jun 10 13:07 - 16:59  (03:51)
[root      root-image]#
```

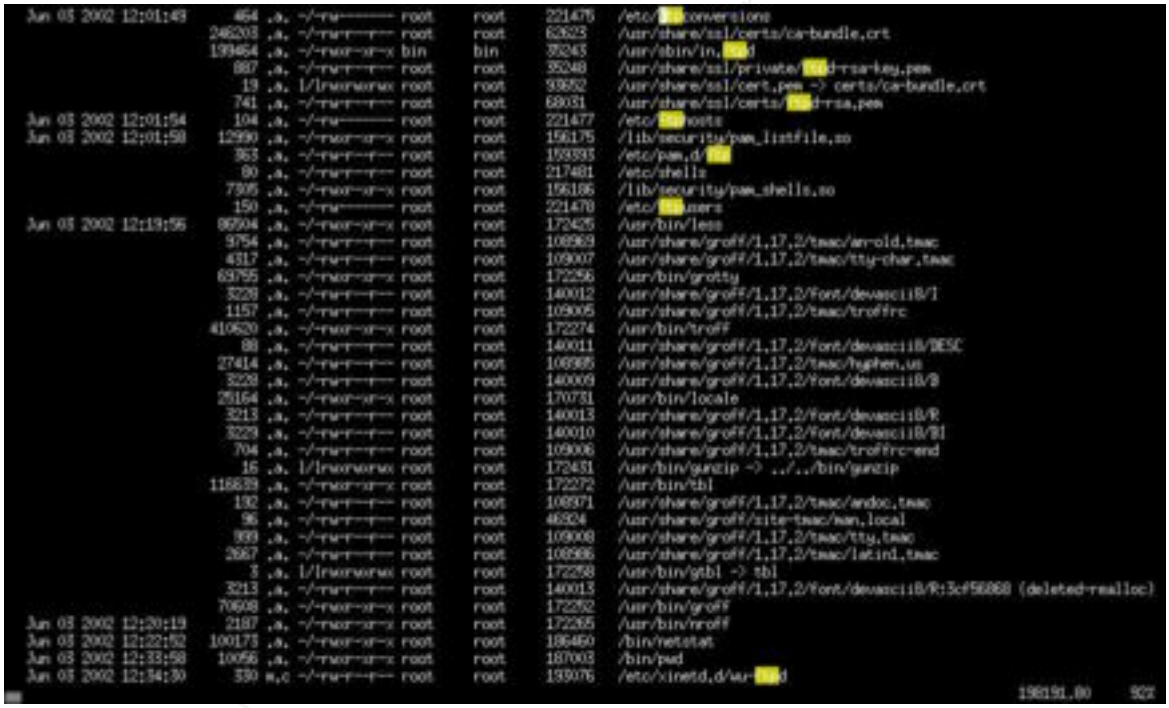
The next few screen shots show the user, Steve, ftp'd into this system. We can examine the mactimes and the wtmp file in an attempt piece together what took place.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Print Screen of wtmp



Print Screen of MAC times



FTPD Time Line

The wtmp file shows that steve login to this system via ftp four times. The process xinetd spawned in.ftpd with the process ID's of 5300, 5307, 5308 and 5314. We can use the mactime analysis to confirm the last ftp login. The mactimes show that the following files were access:

- June 03 2002 12:01:49
  - /etc/ftpconversions
  - /usr/sbin/in/ftpd
  - /usr/share/ssl/private/ftpd-rsa-key-pem
  - usr/share/ssl/certs/ftpd-rsa-pem
- June 03 2002 12:01:54

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

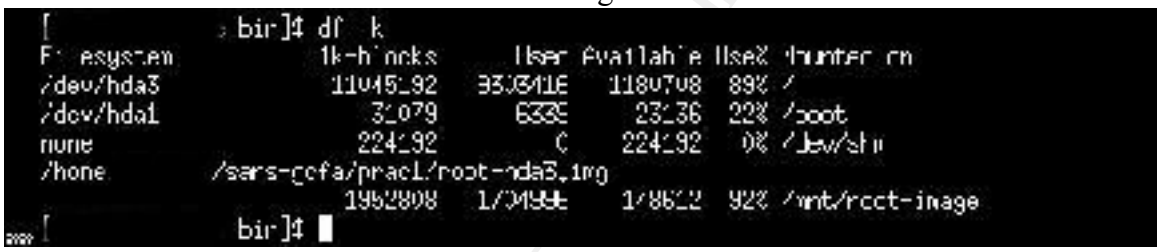
    - /etc/ftphosts
- June 03 2002 12:01:58
    - /etc/pam.d/ftp
    - /etc/ftpusers

Later, at June 03 2002 12:34:30, the xinetd configuration file (/etc/xinetd.dwu-ftpd) for ftp was modified. This file was not modified again and the ftp server spawned by xinetd was disabled. We can infer that he disabled the ftp server at this time.

Recover Deleted Files

The print screen below shows that this system is almost full. There is only 148MB of free space available out of 1.9GB. This is less than the minimum free. The last 10% is used by the system to keep the disk from becoming fragmented. This maybe the reason that none of the deleted files have any data and most of the deletes have been reallocated.

Print Screen of df –k to show how full the image disk was:



Below is a portion of the mactimes of the unallocated inodes. As you can see all of the unallocated space has a size of zero. The only inode that had any size is inode 62083.

May 30 2002 09:49:13 0 .a. -rw-r--r-- steve steve 208686 <root-hda3.img-dead-208686>
0 .a. -rw-r--r-- steve steve 208685 <root-hda3.img-dead-208685>
May 30 2002 13:44:20 0 .a. -rw-rw-r-- steve steve 37890 <root-hda3.img-dead-37890>
0 .a. -rw-rw-r-- steve steve 37883 <root-hda3.img-dead-37883>
May 30 2002 13:44:21 0 .a. -rw-r--r-- steve steve 37881 <root-hda3.img-dead-37881>
0 .a. -rw-r--r-- steve steve 37884 <root-hda3.img-dead-37884>
…..
…..
10 .ac lrwxrwxrwx root root 62083 <root-hda3.img-dead-62083>
0 .a. -rw-r--r-- root root 132039 <root-hda3.img-dead-132039>
0 .a. -rw-r--r-- root root 146492 <root-hda3.img-dead-146492>
0 m.c -rw-r--r-- root root 132039 <root-hda3.img-dead-132039>
0 m.c -rw-r--r-- root root 146492 <root-hda3.img-dead-146492>
0 ..c -rw-r--r-- root root 162037 <root-hda3.img-dead-162037>
0 ..c -rw-r--r-- root root 162047 <root-hda3.img-dead-162047>
0 m.c -rwx------ steve steve 115456 <root-hda3.img-dead-115456>
0 m.c -rwx------ steve steve 146510 <root-hda3.img-dead-146510>
0 ..c srwx------ root root 69012 <root-hda3.img-dead-69012>
0 m.c -rwx------ steve steve 146507 <root-hda3.img-dead-146507>
0 m.c drwx------ steve steve 146508 <root-hda3.img-dead-146508>
0 m.c -rwx------ steve steve 146509 <root-hda3.img-dead-146509>
0 ..c srwxr-xr-x steve steve 224140 <root-hda3.img-dead-224140>
0 m.c drwx------ steve steve 115453 <root-hda3.img-dead-115453>

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Jun 10 2002 16:59:41       0 mac -rw-r--r-- root     root     223751   <root-hda3.img-dead-223751>
                           0 mac ---------- root     root     217480   <root-hda3.img-dead-217480>
                           0 ..c -rw-r--r-- root     root     12       <root-hda3.img-dead-12>
```

The examination of the mactime from the combined allocated and unallocated inodes, revealed that most deleted inodes were reallocated and any deleted inode that were not reallocate, had a size of 0 bytes. It also shows the inode 62083 was a symbolic for /dev/modem to /dev/ttyS01.

```
10 m.. l/lrwxrwxrwx root     root     62083    /dev/modem -> /dev/ttyS1 (deleted)
```

Recovery of Inode 62083

There were only a few inode that had any size. All of these were links. The only files I could recover are symbolic links. I used istat to see the content of the inode and icat to confirm that there was no data in this inode.

```
Command: istat -f linux-ext2 ~userid/sans-gcfa/prac1/root-hda3.img 62083
inode: 62083
Not Allocated
Group: 4
symbolic link to: /dev/ttyS1
uid / gid: 0 / 0
mode: lrwxrwxrwx
size: 10
num of links: 0
Modified:       06.10.2002 15:48:27     (PDT)
Accessed:       06.10.2002 16:59:26     (PDT)
Changed:        06.10.2002 16:59:26     (PDT)
Deleted:        06.10.2002 16:59:26     (PDT)
Direct Blocks:
  0
```

```
Command: icat -f linux-ext2 ~userid/sans-gcfa/prac1/root-hda3.img 62083
Produced nothing.
```

String Search

I did a strings of the two Linux partitions and wrote each of them a strings file. I then used egrep to extract the all of the key words I was looking for to another file. The reason I broke the search down into multiple steps is to reduce the time for each search and to reduce the amount of data generated. I continued refining the search until the strings data was small enough to examine with a pager like less.

The following is the list of pattern I chose to search for: hax0r, Elite, 31337, warez, d00dz, lamer, cracker, irc and bot.

© SANS Institute 2000 - 2002, Author retains full rights.

I chose this list because these words are commonly associated with the most common type of hacker, the Script Kiddies and Internet Relay Chat. IRC is one of the ways the blackhat community uses to exchange information.

Procedure used on each of the following images: root, boot, boot sector and swap

- strings root-hda3.img > root-hda3.img.str
- egrep -i "hax0r|elite|31337|orifice|warez|d00d|lamer|cracker|irc|bot" boot-hda2.img.str >boot-hda2.img.str.search
- grep –i hax0r root-hda3.img.str.search > root-hda3.img.str.search.hax0r
- grep –i elite root-hda3.img.str.search > root-hda3.img.str.search.elite
- etc.

Results from 31337
1940763776 BackOrifice 31337/udp # cDc Back Orifice remote admin tool
1940763842 Elite 31337/tcp # Sometimes interesting stuff can be found here

Results from hax0r
NONE

Results from warez
444492252 To check the class 'C' network on which warez.com sits for popular
444492365 # nmap -fsp 21,22,23,25,80,110 warez.com/24
444492520 > nmap -F warez.com/24
853369445 - Handles those warez sites file names.
984038579 warezes
984499544 warez
1016454604 (warez traders). So we'll set the directory permissions for the incoming
1757215854 warezm

Results from d00d (selected results as there was a lot of hex returned)
177111970 - changed sound sample to linux d00d instead of gorby d00d
799786027 "die.net" => 1, # 'l33t h4ck3r d00dz.

Results from lamer (selected results as there was a lots)
1940747166 distrib-net-losers 2064/tcp # A group of lamers working on a silly closed-source client for solving the RSA cryptographic challenge. This is the keyblock proxy port.

Note: The integer of the left is the decimal offset into the image file.

If time permitted we could investigated each of these by searching the various images for the offsets. Once found, we could determine the file they came from. Then we could examine these files more closely.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

User Information/Conclusions

Based on my analysis, this Linux system was not used much. It was built on May 29th 2002. Since there was almost 1.7GB of space used and there are a total of 782 RPM packages installed, I suspect he did a full install of Redhat Linux 7.3. The first boot after the installation was on May 30th at 1:28AM. At 1:36AM it was shutdown. This system was configured to use DHCP to obtain its IP address.

On May 30th at 9:40AM, the system way booted up again and the user, steve, used Mozilla to surf to www.google.com. He proceeded to search for a Linux telnet server. I am not sure why as there was a telnet server installed during the system build. By examining the Mozilla cooktxt file we see two google cookies, one for google.com and a second for google.ca. This may indicate that the system was in Canadian address space when it was used to access google. Google automatically redirects uses from Canadian address space to www.google.ca. June 10th 2002 at 16:59:37 was the last time the system was used. It was also shutdown in an orderly fashion.

Examining roots .bash_history file, we can see want the user was interested in. He read the Linux man pages for inetd.conf, looked for the inetd.conf file. Since he could not find it. The user resorted to using the find command to help out, not before reading the man page for find. The user did not find inetd.conf, but instead found xinetd.d directory. We can determine this by him narrowing the search criteria in his find command.

After taking a close look at steve's home directory, we find a few interesting perl scripts. These scripts shed more light on why he was so interested in the Linux super server configuration files. There are three script to bulk load inetd.conf, xinetd.d and the services file. The scripts configured a telnet server on a range of ports. The fourth script was an expect script used to test for the presence of a telnet server on range of port.

Considering all the information, I can only conclude that this system was not cracked and was not used for malicious purposes. This user did not load any hacker tools or vulnerability assessment tools. This system was most likely built for specific tests in the LAB.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Chain of Custody

Date	Reason	Analyst
2002-08-15 4pm	System ceased and place in locked draw at the office. Address: 123-2020 XZY street, ABCville, ZA	Stephen Pedersen
2002-08-16-26	System drawn out for the weekend. It was locked up in a century fire safe when not in analyst presence	Stephen Pedersen
2002-08-26	Returned to locked draw at the office (above address)	Stephen Pedersen
2002-08-27 8am	Hard disk was removed and checkout	Stephen Pedersen
2002-08-27 4:30pm	Hard disk was returned to office locked draw (above address)	Stephen Pedersen
2002-09-10 8am	Hard disk was checkout (above address)	Stephen Pedersen
2002-09-10 3pm	Hard disk was returned to office locked draw (above address)	Stephen Pedersen

References

- SANS / GIAC forensics forum
  - URL:http://forum.sans.org/discus/messages/3099/3099.html?1029872597
- "Security Focus" Searching
  - URL:http://online.securityfocus.com/tools
- Spitzner, Lance. The Honeynet Project. "Know your Enemy: Revealing the security tools, tatics, and motives of the blackhat community". Addision Wesley. September 2001
- Honeynet Project. "Know Your Enemy: Honynets". 8 September 2002.
  - URL:http://www.honeynet.org/papers/honeynet/
- Dittrich, Dave. Honeynet Project. "The Forensic Challenge".
  - URL:http://www.honeynet.org/challenge/results/index.html
- "Bart's modular Boot Disk". 4 September 2002.
  - URL:http://www.nu2.nu
- "Virtual Linux" (Linux on a CDROM), 16 September 2001.
  - URL:http://sourceforge.net/projects/virtual-linux
- "CHKROOTKIT". 16 September 2002
  - URL:http://www.chkrootkit.org
- Linux man pages

- Currier, Brian. "Forensic Tools". "Task Version 1 and Autopsy Version 1.5".
  - URL:http://www.atstake.com/research/tools/index.html

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Appendix A

Not used

Appendix B

Search Results Web Page

Advanced Search Preferences Language Tools Search Tips

linux telnet server Google Search

Search: the web pages from Canada

Web Images Groups Directory

Sponsored Links

Telnet Terminal Emulator
VT100, VT220, VT420, TN3270, TN5250
Great Value - Download Free Trial
http://www.distinct.com/
Interest:

See your message here...

Searched the web for **linux telnet server**. Results **1** - **10** of about **506,000**. Search took **0.16** seconds.

Telnet/WAIS Servers - ServerWatch **Telnet**/WAIS Servers
... The Central Spot for News, Reviews, and **Server** Downloads. ServerWatch > Servers > **Telnet**/WAIS Servers. Instant Messaging Planet Fall SWatch News: Red Hat **Linux ...**
serverwatch.internet.com/telservers.html - 24k - 29 May 2002 - Cached - Similar pages

What's New at ServerWatch
... New for May 6, 2002 Updated Red Hat **Linux**, a 5 star **server** platform, to v7 **...** Anywhere,
a 4 star mail **server**, to v4.1. Updated VShell, a 4 star **telnet server ...**
serverwatch.internet.com/new.html - 37k - 29 May 2002 - Cached - Similar pages
[More results from serverwatch.internet.com]

redhat.com | Red Hat Support
... back, Security Advisory. Details: New **telnet**, **telnet-server** packages are available for Red Hat **Linux** 5.2, 6.2, 7.0 and 7.1. These **...**
www.redhat.com/support/errata/RHSA-2001-099.html - 17k - Cached - Similar pages

redhat.com | **linux**-security - [RHSA-2001:099-06] New **telnet ...**
... 1. Topic: New **telnet**, **telnet-server** packages are available for Red Hat **Linux** 5.2, 6.2, 7.0 and 7.1. These packages fix a problem **...**
www.redhat.com/mailing-lists/ linux-security/msg00029.html - 20k - Cached -

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Similar pages
[More results from www.redhat.com]

Software.**Linux**.Com
... **Linux**.com :: **Telnet** 15 projects found. **...** 6. MiaMUD MiaMUD is a MUD code base that features ELF SO-loadable commands, online OLC, a proper **telnet server ...**
software.linux.com/browse/422/?topic=422 - 39k - Cached - Similar pages

Sharing a **Linux server** under X in the classroom LG #45
... start MIX. **telnet** the **server**, with standard c:\windows**telnet**.exe. xterm **...** logout from **telnet**. ... **...** Copyright 1999, Alan Ward Published in Issue 45 of **Linux** Gazette **...**
Description: Sharing a **Linux server** under X in the classroom
Category: Computers > Software > Operating Systems > Linux > Projects > Education
www.linuxgazette.com/issue45/ward/ward.html - 7k - Cached - Similar pages

Re: [WKU-**Linux**] **Telnet server**
... Thread Index] Re: [WKU-**Linux**] **Telnet server**. **...** the message. Follow-Ups: Re: [WKU-**Linux**] **Telnet server**: From: Nathan York <craz@**linux**.wku.edu>. **...**
linux.wku.edu/archive/wku-linux/ wku-linux.200112/msg00044.html - 5k - Cached - Similar pages

RE: [WKU-**Linux**] **Telnet server**
... Thread Index] RE: [WKU-**Linux**] **Telnet server**. **...** the message. References: Re:
[WKU-**Linux**] **Telnet server**: From: Rob VanFleet <rvf@**linux**.wku.edu>. **...**
linux.wku.edu/archive/wku-linux/ wku-linux.200112/msg00054.html - 5k - Cached - Similar pages
[More results from linux.wku.edu]

?????????? | ????
The summary for this Japanese page contains characters that cannot be correctly displayed in this language/character set.
www.jp.redhat.com/support/errata/ RHSA/RHSA-2001-099J.html - 20k - Cached - Similar pages

Embedded **Linux** Starter Kit
... EMAC **Linux** 2.4: Custom configuration program; Boa high-performance web **server**; WU-FTPD
Internet-standard FTP **server**; **TELNET server**; CRON periodic task scheduler; DHCP **...**
www.emacinc.com/linux_starter_kit.htm - 12k - Cached - Similar pages

© SANS Institute 2000 - 2002, Author retains full rights.

linux telnet server Google Search Search within results

Dissatisfied with your results? Help us improve.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Try your query on: AltaVista Excite Lycos Yahoo!

Google Home - Advertise with Us - Search Solutions - News and Resources - Language Tools - Jobs, Press, Cool Stuff...

©2002 Google

© SANS Institute 2000 - 2002, Author retains full rights.

Appendix C

makeRecords-inetd.pl

```
#!/usr/local/bin/perl

$portStart      = $ARGV[0];
$portEnd        = $ARGV[1];

for( $port=$portStart; $port<=$portEnd; $port++) {
        print "telnet_${port}   stream  tcp6    nowait root    /usr/sbin/in.telnetd
in.telnetd\n";
}
```

makeRecords-xinetd.d.pl

```
#!/usr/bin/perl

$portStart      = $ARGV[0];
$portEnd        = $ARGV[1];

$service        = "telnet";
$protocol       = "tcp";


for( $port=$portStart; $port<=$portEnd; $port++) {
        print "service ${service}_${port}\n";
        print "\{\n";
        print " flags           = REUSE\n";
        print " socket_type     = stream\n";
        print " wait            = no\n";
        print " user            = root\n";
        print " server          = /usr/sbin/in.telnetd\n";
        print " log_on_failure  += USERID\n";
        print " port            = ${port}\n";
        print " disable         = no\n";
        print "\}\n\n";
}
```

makeRecords-services.pl

```
#!/usr/local/bin/perl

$portStart      = $ARGV[0];
$portEnd        = $ARGV[1];
```

© SANS Institute 2000 - 2002, Author retains full rights.

```
$service       = "telnet";
$protocol      = "tcp";


for( $port=$portStart; $port<=$portEnd; $port++) {
      print "${service}_${port}       ${port}/${protocol}\n";
}
```

telnetConnectScan.exp

```
#!/usr/local/bin/expect --

#############################################################
#
#  Usage: telnetConnectScan <host> <start port> <end port>
#
#  This script attempts to telnet into a host on every
#  port within the specified range.  The results are stored
#  into telnetResults.txt
#
#
#  Written by Steve XXX  <XXX@ABC.com>
#
#  Created:       May 9, 2002
#  Last Modified:  May 9, 2002
#           By:  Steve XXX
#
#############################################################

log_user 0

set <space> \x20
set loginPrompt       "login:"
set failureMessage    "Connection refused"

# Command line parameters
set host      [lindex $argv 0]
set startPort   [lindex $argv 1]
set endPort     [lindex $argv 2]

# Output file
set outputFile  [open "scanResults_${startPort}-${endPort}.txt" "w"]

# Begin telnet connection scan
puts $outputFile "Telnet connections attempted to:\n"
puts $outputFile "HOST:  ${host}"
```

© SANS Institute 2000 - 2002, Author retains full rights.

```
puts $outputFile "PORTS: ${startPort} to ${endPort}\n"
puts $outputFile "SUCCESS on all ports except the following:\n"

puts "Beginning scan ...\n"
set port $startPort

while {$port <= $endPort} {
      puts -nonewline "CONNECTING to: $host   PORT: $port ... "
      spawn telnet $host $port
      expect {
            "$loginPrompt"  {
                  puts "SUCCESS"
                  close
            }
            "$failureMessage" {
                  puts "FAILED"
                  puts $outputFile "${port}"
            }
            timeout {
                  puts "TIMEOUT"
                  puts $outputFile "${port} (TIMEOUT)"
            }
      }
      incr port 1
}

puts "\nScan complete.\n"
```

© SANS Institute 2000 - 2002 Author retains full rights.

GCFA Binary Analysis (Part 2)

Executive Summary

A binary file was found on one of your computer systems. The administrator discovered this file and invoked the Incident Response Procedures. The CIRT responded and captured the file for analysis. The forensic analysis revealed that this binary is a sniffer program called AMDsniff. It was installed on one of your company's computers. The program was last run on April, 11th 2002 at 9:29am. A hacker group who call them selves "The ADM Crew" wrote the program. This sniffer selectively logs network traffic for protocols such as telnet, ftp and email (popV2 &V3, imap), to name a few. Logging these protocols is most likely an attempt to harvest user ID and passwords.

Binary Details

Name of the program/file found on the system.

sn.dat

File/MACTime.

Apr 11 02 09:29:58 399124 ma. -rw-rw-rw- userid userid /home/userid/sans-gcfa/sn.dat

Jul 13 02 16:19:42 399124 ..c -rw-rw-rw- userid userid /home/userid/sans-gcfa/sn.dat

Note the date for the ctime, this because the file was created on my forensic system at that time. It does not reflect a time associate with this incident.

Screen shot to confirm the above mactime data.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

File owner(s) – (user and/or group).

The archive was provided in zip format, an attempt was made to extracted the archive with the original UID/GID information on a Linux system using root and a regular user account, the syntax we used was "unzip –X". Unfortunately the two files extracted assumed the UID/GID of the current user.

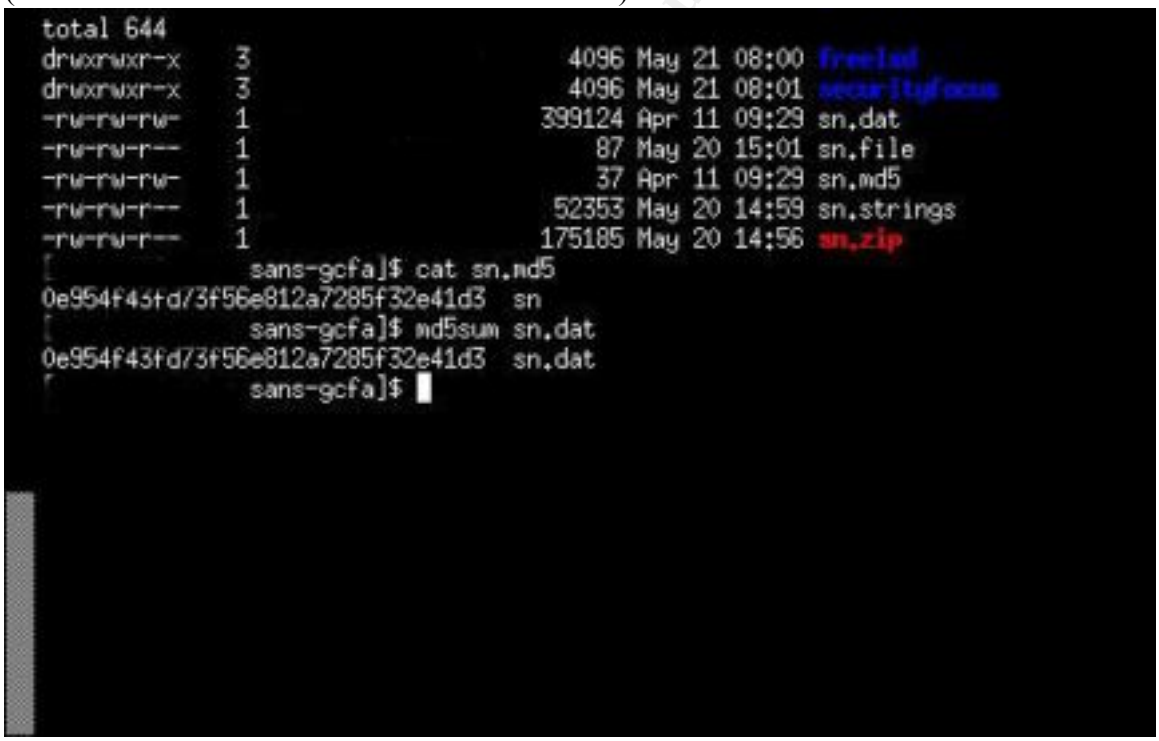
-rw-rw-rw- 1 userid userid 399124 Apr 11 09:29 sn.dat
Please see screen shot below under MD5 hash.

File size (in bytes).

Sn.dat was 399124 bytes in size
-rw-rw-rw- 1 userid userid 399124 Apr 11 09:29 sn.dat
Please see screen shot below under MD5 hash.

MD5 hash of the file

(include screen shots of the hash value obtained).



Other File Information

This binary is a 32 bit elf executable. It was compiled for the X86 architecture and is statically linked and stipped. This means it is a self contained binary and does not use external library functions when executing. The entry point to the program is normal for the X86 architecture, "0x80480e0". This information was obtained by using the "file" and "readelf" commands.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Command file sn.dat: ELF 32-bit LSB executable, Intel 80386, version 1, statically linked, stripped

Command elfread –h sn.dat
ELF Header:
```
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Intel 80386
  Version:                           0x1
  Entry point address:               0x80480e0
  Start of program headers:          52 (bytes into file)
  Start of section headers:          398364 (bytes into file)
  Flags:                             0x0
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         3
  Size of section headers:           40 (bytes)
  Number of section headers:         19
  Section header string table index: 18
```

Key words found that are associated with the program/file.

The information below was extracted by running strings on the binary (sn.dat) and selecting parts, which gave hints as to what the program was or could be used for.

Command strings sn.dat |less (and searched for pcap)
```
\*      The END      */
priv 1.0
ADMsniff %s <device> [HEADERSIZE] [DEBUG]
ex   : admsniff le0
 ..ooOO The ADM Crew OOoo..
cant open pcap device :<
init_pcap : Unknown device type!
ADMsniff %s  in libpcap we trust !
credits: ADM, mel , ^pretty^ for the mail she sent me
The_l0gz
@(#) $Header: pcap-linux.c,v 1.15 97/10/02 22:39:37 leres Exp $ (LBL)
@(#) $Header: pcap.c,v 1.29 98/07/12 13:15:39 leres Exp $ (LBL)
@(#) $Header: savefile.c,v 1.37 97/10/15 21:58:58 leres Exp $ (LBL)
```

© SANS Institute 2000 - 2002, Author retains full rights.

@(#) $Header: bpf_filter.c,v 1.33 97/04/26 13:37:18 leres Exp $ (LBL)

1997-12-20
+45 3325-6543
+45 3122-6543
keld@dkuug.dk
Keld Simonsen
ISO/IEC 14652 i18n FDCC-set
C/o Keld Simonsen, Skt. Jorgens Alle 8, DK-1615 Kobenhavn V

I downloaded a copy of ADMsniff from freelsd (http://adm.freelsd.net/ADM/ADMsniff.tar.gz) after statically linking and stripping the executable, the binary size was almost the same.
-rwxrwxr-x 1 userid userid 402168 Jun 19 20:50 ADMsniff-1
I also ran strings against this binary and you can see that I was able to fine the exact same text
* The END */
priv 1.0
ADMsniff %s <device> [HEADERSIZE] [DEBUG]
ex : admsniff le0
..ooOO The ADM Crew OOoo..
cant open pcap device :<
init_pcap : Unknown device type!
ADMsniff %s in libpcap we trust !
credits: ADM, mel , ^pretty^ for the mail she sent me
The_l0gz
@(#) $Header: pcap-linux.c,v 1.15 97/10/02 22:39:37 leres Exp $ (LBL)
@(#) $Header: pcap.c,v 1.29 98/07/12 13:15:39 leres Exp $ (LBL)
@(#) $Header: savefile.c,v 1.37 97/10/15 21:58:58 leres Exp $ (LBL)
@(#) $Header: bpf_filter.c,v 1.33 97/04/26 13:37:18 leres Exp $ (LBL)

1997-12-20
+45 3325-6543
+45 3122-6543
keld@dkuug.dk
Keld Simonsen
ISO/IEC 14652 i18n FDCC-set
C/o Keld Simonsen, Skt. Jorgens Alle 8, DK-1615 Kobenhavn V

© SANS Institute 2000 - 2002, Author retains full rights.

Program Description

Program function

This program is a sniffer and it is used to promiscuously listen on a network interface. It is coded to log traffic for certain ports to a log file called "The_l0gz". Depending of the compile options, the log file can be compressed using gzip format. The program also has some signal handling. If the program receives a SIGHUP, the program will flush the logs to the file. If the SIGTERM signal is received the logs are flushed to the file and the log file is closed and the program terminates.
The following table shows the network traffic that can be logged.

Port Number	Port Name	Decription
21	ftp	File Transfer Protocol
23	telnet	Telnet Remote Login
109	Pop version 2	Post Office Protocol - Version 2
110	Pop Version 3	Post Office Protocol - Version 3
143	Imap	
512	Execute	Remote Process Execution
513	Login	Remote Login
514	Command Shell	Remote Command shell (no password)
1521	Unknown	Unknown

Confirmation of above

I confirmed these ports by running the sn.dat. I then used nmap to scan all 65535 tcp and udp ports. After completion I check the sniffer log ("The_l0gz"), only the above ports were logged.
To validate my finding, I looked at the source code for the same version I downloaded from Freelsd (See Program Identification section for more details). The same ports were listed in the source code and one additional port:31337. UDP 31337 is typically associated with Back Orifice, but the source code is written to only log TCP ports. The source also explicitly excludes TCP port 31337.
Excert from the ADMsniff source code, downloaded from freelsd:

```
    for (i = 0; coolport[i] != 31337; i++)
      {
       if (coolport[i] == ntohs (tcp->th_sport) ||
           coolport[i] == ntohs (tcp->th_dport))
         LOG = 1;
      }
```

After compiling this version, I performed the same tests and I noticed that 31337 was still not logged.
Note: 31337 is a hackers spelling of 'elite', meaning 'elite hackers'.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

This program was last accessed on Apr 11 2002 09:29:58. I used the coroner's toolkit to do the mactime analysis. I unzipped the achieve into a working directory which produced sn.dat and sn.md5. I then ran grave-robber to produce the body file which is then used by mactime. Mactime gives you date and time stamps of all the files in the working directory, when they were last accessed, modified and changed. Mactime, requires a start time, I chose 1/1/2002. After capturing the mactimes, I then confirmed that the md5 sum matched that in sn.md5

- mkdir sans-gcfa
- unzip –X sn.zip
- ./grave-robber -c ~userid/sans-gcfa -o LINUX2 -m
- ./mactime 1/1/2002 >~userid/sans-gcfa/sn-mactime.report
- md5sum sn.dat

To validate the above, please see screen shots from the Binary Details section.

Forensic Details

The sn.dat binary is a statically linked and stripped version of the sniffer utility called ADMsniff. Since it is statically linked, it will not use any external or system libraries and therefore will not leave a very noisey footprint on a system when it is executed. But it is a sniffer and so you will be able to see that the promiscuous flag has been set when the sniffer binds to the network interface. In addition, this sniffer creates a log file called "The_l0gz", in the same directory as the binary

These are the sites I search for more information about ADMsniff:
www.securityfocus.com
www.incidents.org
http://www.xploiter.com/security/trojanport.html
www.google.com

Below is the stack trace of sn.dat. Please note the highlighted sections:

```
[root@forensic01 sans-gcfa]# strace -f –o sn.dat.strace ./sn.dat eth0
[userid@forensic01 sans-gcfa]$ cat sn.dat.strace
11139 execve("./sn.dat", ["./sn.dat", "eth0"], [/* 25 vars */]) = 0
11139 fcntl64(0, 0x1, 0, 0xbffff964)    = 0
11139 fcntl64(0x1, 0x1, 0, 0xbffff964)  = 0
11139 fcntl64(0x2, 0x1, 0, 0xbffff964)  = 0
11139 uname({sys="Linux", node="forensic01", ...}) = 0
11139 geteuid32()                       = 0
11139 getuid32()                        = 0
11139 getegid32()                       = 0
11139 getgid32()                        = 0
11139 brk(0)                          = 0x80ab488
11139 brk(0x80ab4a8)                        = 0x80ab4a8
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
11139 brk(0x80ac000)                    = 0x80ac000
11139 socket(PF_INET, SOCK_PACKET, 0x300 /* IPPROTO_??? */) = 3
11139 bind(3, {sin_family=AF_INET, sin_port=htons(25972),
sin_addr=inet_addr("104.48.0.0")}}, 16) = 0
11139 ioctl(3, 0x8927, 0xbffff780)      = 0
11139 ioctl(3, 0x8921, 0xbffff780)      = 0
11139 ioctl(3, 0x8913, 0xbffff780)      = 0
11139 ioctl(3, 0x8914, 0xbffff780)      = 0
11139 fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 1), ...}) = 0
11139 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x40000000
11139 write(1, "ADMsniff priv 1.0  in libpcap we"..., 41) = 41
11139 write(1, "credits: ADM, mel , ^pretty^ for"..., 54) = 54
11139 brk(0x80ad000)                    = 0x80ad000
11139 open("The_l0gz", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 4
11139 recvfrom(3, "\1\200\302\0\0\0\0\220\261\255Q\217\0&BB\3\0\0\0\0\0\200"...,
1564, 0, {sin_family=AF_UNIX, path="eth0"}, [18]) = 64
11139 ioctl(3, 0x8906, 0xbffff770)      = 0
11139 recvfrom(3, "\1\200\302\0\0\0\0\220\261\255Q\217\0&BB\3\0\0\0\0\0\200"...,
1564, 0, {sin_family=AF_UNIX, path="eth0"}, [18]) = 64
.
.
.
11139 time(NULL)                        = 1026710169
11139 fstat64(4, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0
11139 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x40001000
11139 write(4, "\n--=[ 199.185.220.248:110 --> 66"..., 201) = 201
11139 recvfrom(3, "\0\1\201\276\200E\0\20\244\2654-\10\0E\20\0004m\223@\0"...,
1564, 0, {sin_family=AF_UNIX, path="eth0"}, [18]) = 66
11139 ioctl(3, 0x8906, 0xbffff770)      = 0
11139 time(NULL)                        = 1026710169
11139 write(4, "\n--=[ 66.183.64.248:51473 --> 19"..., 143) = 143
11139 recvfrom(3, "\0\20\244\2654-\0\1\201\276\200E\10\0E(\0004\5\226@\0\370"...,
1564, 0, {sin_family=AF_UNIX, path="eth0"}, [18]) = 66
11139 ioctl(3, 0x8906, 0xbffff770)      = 0
11139 time(NULL)                        = 1026710169
11139 time(NULL)                        = 1026710169
11139 time(NULL)                        = 1026710169
```

© SANS Institute 2000 - 2002, Author retains full rights.

What is happening?

- The program executes itself (execve system call)
- It then binds to the network interface (bind system call)
- Writes a banner and credits to STDOUT (write system call)
- Opens its log file, called “The_l0gz”, for writing. (open system call)
- Compare the captured network traffic with it list of so-called “cool port”.
- When there is a tcp port match, the program logs this traffic to the log file.

Note that the execve system call is only for itself and there are no other processes spawned. This confirms that the code does not have a mechanism to self-replicate. The only open system call is to create the log file, “The_l0gz". The write system calls are to file handler 1 which is standout and file handle 4, which is the log file.

Program Identification

I downloaded a copy of ADMsniff from freelsd (http://adm.freelsd.net/ADM/ADMsniff.tar.gz). I confirm this by noting the version number that is in the banner when the sn.dat is executed. I proceeded to compile the source from freelsd with the option to statically link and strip the binary. Unfortunately, the size of the compiled program was slightly different. I did an md5 sum, but I knew it would not match.

- file sn.dat: ELF 32-bit LSB executable, Intel 80386, version 1, statically linked, stripped
- file ADMsniff-1: ELF 32-bit LSB executable, Intel 80386, version 1, statically linked, stripped
- ls –l sn.dat: -rwxrw-rw- 1 root root 399124 Apr 11 09:29 sn.dat
- ls –l ADMsniff-1: -rwxrwxr-x 1 userid userid 402168 Jun 19 20:50 ADMsniff-1
- md5sum sn.dat: 0e954f43fd73f56e812a7285f32e41d3 sn.dat
- md5sum ADMsniff-1: 80d0296d73ea399b7ab5f23f63fbb558 ADMsniff-1

I use objectdump to show the comments section of the binary. It reveals that sn.dat was compiled on Redhat Linux Version 7.1 with gcc version 2.96 20000731. When I did the same on the ADMsniff I compiled it showed Redhat Linux Version 7.2. This and the fact that 7.1 could have different versions of library functions that were included during the static linking, could be the cause of the compiled binary to be slightly different.

objdump -j .comment -s sn.dat |head

sn.dat: file format elf32-i386

Contents of section .comment:

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
 0000 00474343 3a202847 4e552920 322e3936  .GCC: (GNU) 2.96
 0010 20323030 30303733 31202852 65642048   20000731 (Red H
 0020 6174204c 696e7578 20372e31 20322e39  at Linux 7.1 2.9
 0030 362d3937 29000047 43433a20 28474e55  6-97)..GCC: (GNU
 0040 2920322e 39362032 30303030 37333120  ) 2.96 20000731
 0050 28526564 20486174 204c696e 75782037  (Red Hat Linux 7
```

objdump -j .comment -s ADMsniff-1 |head

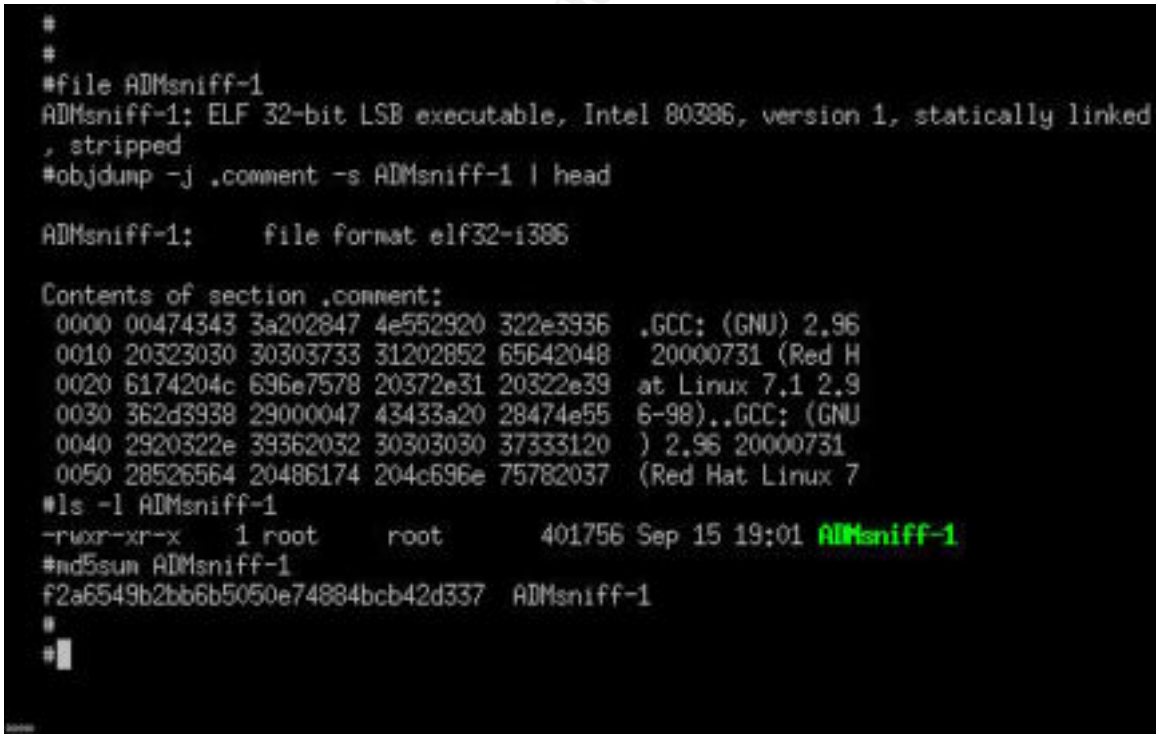
ADMsniff-1: file format elf32-i386

Contents of section .comment:

```
 0000 00474343 3a202847 4e552920 322e3936  .GCC: (GNU) 2.96
 0010 20323030 30303733 31202852 65642048   20000731 (Red H
 0020 6174204c 696e7578 20372e32 20322e39  at Linux 7.2 2.9
 0030 362d3130 382e3129 00004743 433a2028  6-108.1)..GCC: (
 0040 474e5529 20322e39 36203230 30303037  GNU) 2.96 200007
 0050 33312028 52656420 48617420 4c696e75  31 (Red Hat Linu
```

I then built a standard Redhat 7.1 Linux system and repeated the compilation steps with a similar result.

Screen shot of above



© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Legal Implications

Prosecution by Law

We can prove that "sn.dat" or more commonly known as ADMsniff, was run on the victim's system by referring to the last access time of this program (MACtime analysis). But it is not possible from the above evidence to prove who ran this program. If new evidence or a confession, lead to the identification of the person who executed this program. We may be able to prosecute under section 342.1(1b) of the Canadian Criminal Code. This section states," Every one who, fraudulently and without colour of right,
a. Obtains, directly or indirectly, any computer service,
(b) by means of an electro-magnetic, acoustic, mechanical or other device, intercepts or causes to be intercepted, directly or indirectly, any function of a computer system,
is guilty of an indictable offence and liable to imprisonment for a term not exceeding ten years, or is guilty of an offence punishable on summary conviction."

Corporate Policy Violations

Network Access

Only authorized company and non-company personnel can gain access to Network Elements, Systems or Applications. Personnel are responsible to protect against non-authorized access to network elements, systems or applications.

Monitoring of Telecommunications Transmissions

The monitoring of telecommunication transmission must only occur when performing company related duties. Information acquired from monitoring cannot be discussed with a third party or be used for any purpose other than company authorized.

Interview Questions

Question One

As a security analyst myself, I also perform security assessments. One question I have: Out of all of the network analysis tools available, why did you choose ADMsniff?

Question Two

When I worked in desktop support, ADMsniff would have been a great tools to use when trouble-shooting end user email access problems. Is this what you were using it for?

Question Three

Where you approached by someone in our company to perform password policy compliance testing. If so, what tool(s) would you use and who was that person?

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Question Four (Assuming the individual is working for the company)

We know you were on the system in question and management is all over my back to get this incident resolved quickly. If this incident escalates any farther, jobs could be on the line. Your co-operation will help bring this incident to a close quickly and we can all put this behind us. Will you detail what software you installed and ran? This will save a lot of time and money. I'm sure your co-operation will be look upon favourably.

Question Five

Someone of your skill and knowledge about computer security and vulnerability assessment tools could be useful in the CIRT centre. But a position like that requires trust, responsibility and integrity. I can appreciate individuals who are trying to better themselves and expand their knowledge. We all make mistake from time to time that is human. Perhaps you were only testing the software you compile in order to confirm your understanding of the source code. Is this what happened when you ran ADMsniff?

Reference

ADM. "ADMsniff 0.8". 22 Oct 2001. Security Focus

- URL:http://online.securityfocus.com/tools/215

The ADM Crews. "The ADM Crews official ftp site: Freedom vs Security"

- URL:http://adm.freelsd.net/ADM/ADMsniff.tar.gz

Mendel, Dion. "Honeynet Project: Reverse Challenge: Top 20 Submissions"

- URL:http://www.honeynet.org/reverse/results/sol/sol-06/

Clark, Michael. "Virtual Honeynets". 7 November 2001

- URL:http://online.securityfocus.com/infocus/1506

Honeynet Project. "Know your Enemy:Honeypots". 08 September, 2002.

- URL:http://project.honeynet.org/papers/honeynet/

Honeypots with Vmware

- URL:http://www.seifried.org/security/ids/20020107-honeypot-vmware-basics.html

Criminal Analysis Branch, Criminal Intelligence Directorate, Royal Canadian Mounted Police. "Criminal Intelligence Program". "Appendix A - Criminal code of Canada offences related to hacking" 14 March 2002.

- URL:http://www.rcmp-grc.gc.ca/crim_int/hackers_e-a.htm#appendixA

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Legal Issues of Incident Handling

Wiretap Statute

As an administrator of a system, I am expected to maintain a specific service level for this resource to its user community, as well as assist user who are having trouble accessing the resources on this system. In order to do this, I must have the authority to protect and defend this system from misuse or excessive user. Network monitoring is one of the tools used to facilitate trouble-shooting. It would also be used to detect malicious activity and the misuse of the systems resources. An example of malicious activity would be, DOS attacks, SPAM or an attempt to again unauthorized or privilege access to the system. All of these have the potential to affect the service level of this system. My job as the administrator is to maintain the availability and integrity of this system for authorized users.

Section 184.(2) of the Canadian Criminal Code allows for service providers to monitor traffic as long as it is related to maintaining service quality, service availability or related to facilitating the normal course of business.

Section 183 of the Canadian Criminal Code is related to wiretaps. This section of the code was originally written with telephones in mind, but the wording used was telecommunications. This allows it to be applied to network traffic. If authorities want to set-up a wiretap or monitor network traffic, they require a court order from the Supreme Court Judge. A wiretap is considered an invasion of privacy and therefore the hearing for such a request is done with complete confidentiality and the documentation is sealed until the evidence is presented in court for prosecution.

Wiretapping is considered the capturing of real time conversations and not to stored information. Telephone conversation would require a wiretap court order, but if you wanted to get the persons voicemail, a wiretap is not required. A search and seizure warrant is required.

Section 487 of the Canadian Criminal Code deals with search and seizure. Authorities can obtain a search warrant from the Justice of the Peace. The justice of the Peace is local to the province and is available 24 hour a day. It is much easier to obtain a search and seizure warrant, than a wiretap from the Supreme Court Judge.

If the administrator is monitoring network traffic with the intention of capturing information that is not directly related to trouble-shooting or the availability/integrity of the network, then that type of monitoring could be considered excessive use of their authority and they would most likely be breaking one of the company's policies. Lets consider the possibility of the administrator capturing information for personal again. For example; the administrator is capturing user ID's and password as they travel across the network. He/she, maybe doing so with the intention of using one of these credentials to

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

authorize the purchase of something for their personal benefit. Even worse they could be harvesting such information with the intention of sell it to a third party. In the above cases the administrator has exceeded their level of authority and has committed theft and therefore considered illegal. These actions could be prosecuted in a court of law.

System Banners

It is recommended to banner a system before the activities of the users are monitored. Service providers can also get consent to monitor, by including this in the service contract.

A banner is much like a no trespassing sign; it informs the users before they enter, that the resource is for authorized use only. A banner gives the system administrator consent to monitor his system for potential misuse or illegal activity and that evidence could be provided to law enforcement for prosecution.

A typical banner should inform the user or potential user that the system is for authorized use only. It should also state that unauthorized use or use in excess of their authority would be subject to monitoring. The banner should also state that by using the system, the user is consenting to the monitoring of their activities. If the monitoring should reveal evidence of illegal activity or the misuse of the system, this evidence may be provided to law enforcement officials.

A banner is only viable on user interactive applications and network connections such as: console ports, data terminals, telnet, ftp, secure shell, etc. If a non-interactive resource were used, such a client/server application (e.g. fileserver), the usefulness of a banner comes into question. A client application, on behalf of the user, makes a network connection of this nature and as such that users would never see the banner.

Under Section 184 of the Canadian Criminal Code, the provider may monitor the activities of the users in order to provide the service assurance. This in effect gives the service provider the right to monitor and defend their system even though user did not consent this monitoring.

© SANS Institute 2000 - 2002 Author retains full rights.

Reference

Criminal Analysis Branch, Criminal Intelligence Directorate, Royal Canadian Mounted Police. "Criminal Intelligence Program". "Appendix A - Criminal code of Canada offences related to hacking" 14 March 2002.

- URL:http://www.rcmp-grc.gc.ca/crim_int/hackers_e-a.htm#appendixA

Department of Justice of Canada. "Forgery and Offences Resembling Forgery". "Section 430 of the Canadian Criminal Code". 31 August 2001

- URL:http://laws.justice.gc.ca/en/C-46/39560.html

Department of Justice. "Invasion of Privacy Part". "Section 183 and 184 of the Canadian Criminal Code". 31 August 2001

- URL:http://laws.justice.gc.ca/en/C-46/38885.html

Smith P J. "R. v. Weir - 1998 Alberta case involving child pornography and privacy issues". 10 February 1998

- URL: http://aix1.uottawa.ca/~geist/Weir.html

Department of Justice, Industry Canada, Solicitor General Canada. "Lawful Access – Consultation Document" 25 August 2002

- URL:http://www.canada.justice.gc.ca/en/cons/la_al/

© SANS Institute 2000 - 2002, Author retains full rights.

Appendix A

Criminal Code of Canada offences related to hacking/cracking

Unauthorized Use of Computer

342.1 (1) Every one who, fraudulently and without colour of right,

a. obtains, directly or indirectly, any computer service,
(b) by means of an electro-magnetic, acoustic, mechanical or other device, intercepts or causes to be intercepted, directly or indirectly, any function of a computer system,
(c) uses or causes to be used, directly or indirectly, a computer system with intent to commit an offence under paragraph (a) or (b) or an offence under section 430 in relation to data or a computer system, or
(d) uses, possesses, traffics in or permits another person to have access to a computer password that would enable a person to commit an offence under paragraph (a), (b) or (c)

is guilty of an indictable offence and liable to imprisonment for a term not exceeding ten years, or is guilty of an offence punishable on summary conviction.

(2) In this section:
"computer password" means any data by which a computer service or computer system is capable of being obtained or used;

"computer program" means data representing instructions or statements that, when executed in a computer system, causes the computer system to perform a function;

"computer service" includes data processing and the storage or retrieval of data;

"computer system" means a device that, or a group of interconnected or related devices one or more of which,

(a) contains computer programs or other data, and pursuant to (b) computer programs,

performs logic and control, and
may perform any other function;
"data" means representations of information or of concepts that are being prepared or have been prepared in a form suitable for use in a computer system;

"electro-magnetic, acoustic, mechanical or other device" means any device or apparatus that is used or is capable of being used to intercept any function of a computer system, but does not include a hearing aid used to correct subnormal hearing of the user to not better than normal hearing;

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

"function" includes logic, control, arithmetic, deletion, storage and retrieval and communication or telecommunication to, from or within a computer system;

"intercept" includes listen to or record a function of a computer system, or acquire the substance, meaning or purport thereof.

"traffic" means, in respect of a computer password, to sell, export from or import into Canada, distribute or deal with in any other way.

Mischief in Relation to Data

430. (1) Everyone commits mischief who wilfully

destroys or damages property;
renders property dangerous, useless, inoperative or ineffective;
obstructs, interrupts or interferes with the lawful use, enjoyment or operation of property; or
obstructs, interrupts or interferes with any person in the lawful use, enjoyment or operation of property.
(1.1) Every one commits mischief who wilfully

destroys or alters data;
renders data meaningless, useless or ineffective;
obstructs, interrupts or interferes with the lawful use of data; or
obstructs, interrupts or interferes with any person in the lawful use of data or denies access to data to any person who is entitled to access thereto.
(2) Every one who commits mischief that causes actual danger to life is guilty of an indictable offence and liable to imprisonment for life.

(3) Every one who commits mischief in relation to property that is a testamentary instrument or the value of which exceeds five thousand dollars

is guilty of an indictable offence and liable to imprisonment for a term not exceeding ten years; or
is guilty of an offence punishable on summary conviction.
(4) Every one who commits mischief in relation to property, other than property described in subsection (3),

is guilty of an indictable offence and liable to imprisonment for a term not exceeding two years; or
is guilty of an offence punishable on summary conviction.
(5) Every one who commits mischief in relation to data

is guilty of an indictable offence and liable to imprisonment for a term not exceeding ten years; or

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

is guilty of an offence punishable on summary conviction.
(5.1) Every one who wilfully does an act or wilfully omits to do an act that it is his duty to do, if that act or omission is likely to constitute mischief causing actual danger to life, or to constitute mischief in relation to property or data,

is guilty of an indictable offence and liable to imprisonment for a term not exceeding five years; or
is guilty of an offence punishable on summary conviction.
(6) No person commits mischief within the meaning of this section by reason only that

he stops work as a result of the failure of his employer and himself to agree on any matter relating to his employment;
he stops work as a result of the failure of his employer and a bargaining agent acting on his behalf to agree on any matter relating to his employment; or
he stops work as a result of his taking part in a combination of workmen or employees for their own reasonable protection as workmen or employees.
(7) No person commits mischief within the meaning of this section by reason only that he attends at or near or approaches a dwelling-house or place for the purpose only of obtaining or communicating information.

(8) In this section, "data" has the same meaning as in section 342.1.

Theft of Telecommunication Service

326. (1) Every one commits theft who fraudulently, maliciously, or without colour of right,

abstracts, consumes or uses electricity or gas or causes it to be wasted or diverted; or
uses any telecommunication facility or obtains any telecommunication service.
Definition of "telecommunication"
(2) In this section and section 327,"telecommunication" means any transmission, emission or reception of signs, signals, writing, images or sounds or intelligence of any nature by wire, radio, visual or other electromagnetic system.

© SANS Institute 2000 - 2002 Author retains full rights.