



Fight crime.
Unravel incidents... one byte at a time.

Copyright SANS Institute
Author Retains Full Rights

This paper is from the SANS Computer Forensics and e-Discovery site. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Advanced Digital Forensics, Incident Response, and Threat Hunting (FOR508)"
at <http://digital-forensics.sans.org><http://digital-forensics.sans.org/events/>

GIAC GCFA Practical Practical Assignment

Version 1.4 (July 21, 2003)

Hugh Pierce

29 December 2003

Abstract/Summary

The first section of this practical reviews a floppy disk image taken from a corporate setting, in which an employee John Price, was suspended for suspected distribution of copyrighted material. The conclusion based on the evidence recovered and analyzed, is that Mr. Price is guilty of certain contractual failures and potentially criminal infractions. No direct evidence was found to prove Mr. Price distributed specific copyright protected material.

The second section is a validation of Crocware's Mount Image Pro. The utility has the potential for an invaluable role in access to a variety of forensic image file formats, as well as conversion of image files between formats.

The third and final section is a discussion of the legal implications created by Mr. Price's activities in the first section.

© SANS Institute 2004, Author retains full rights.

Part 1: Analyze an Unknown Binary

SANS Case Detail	
Tag	fl-160703-jp1
Media	3.5 inch TDK floppy disk
MD5 Hash	4b680767a2aed974cec5fbcfb84cc97a
File	fl-160703-jp1.dd.gz

Forensic Platform	
Hardware	HP Pavilion N5000 laptop, 512MB RAM, 1GHz Athlon
Software	Windows XP Pro SP1 host ¹ VMWare Workstation 4.0.5 build-6030 ² with Red Hat Linux 9 guest ³ EnCase Forensic Edition 4.16a ⁴ Autopsy 1.75 ⁵ Other tools not used as a platform for analysis are mentioned in-line

Document Conventions

To improve readability, the following characteristics define the source and purpose of text within this document.

Commentary and analysis within this document is formatted in Arial 12-point font.

Section titles are **bolded**.

Output quoted from the analysis computer is formatted in Courier New 8-point font. Actual screen capture of commands and output from the analysis are presented in Courier New 8-point font with a box border.

Reference to specific file names are in *italics*.

All times are GMT unless otherwise noted.

¹ <http://www.microsoft.com/windowsxp>

² http://www.vmware.com/products/desktop/ws_features.html

³ <http://www.redhat.com>

⁴ <http://www.guidancesoftware.com>

⁵ <http://www.sleuthkit.org/autopsy/index.php>

Binary Details

Using *md5sum*, hashes were generated for the *gzip* compressed forensic image downloaded from SANS, the extracted *dd* image file, and the *prog* binary contained in the *dd* image. The binary *prog* was accessed by mounting the *dd* forensic image in a read-only state. As displayed in the following text box, the comparison hash matches the original SANS provided hash for the *gzip* file. The Linux command *stat* was run against the binary to gather the logical file size and the Modify, Access, and Change timestamps.

```
[root@localhost Default]# md5sum fl-160703-jp1.dd.gz
4b680767a2aed974cec5fbcfbf84cc97a  fl-160703-jp1.dd.gz

[root@localhost Default]# md5sum fl-160703-jp1.dd
20be7bc13a5cb8d77232659c52a3ba65  fl-160703-jp1.dd

[root@localhost tmp]# mount -ro,loop,nodev,noatime,noexec /tmp/fl-160703-jp1.dd /mnt/img

[root@localhost Default]# md5sum /mnt/img/prog
7b80d9aff486c6aa6aa3efa63cc56880  /mnt/img/prog

[root@localhost img]# stat prog
  File: `prog'
  Size: 487476          Blocks: 960          IO Block: 4096   Regular File
Device: 700h/1792d    Inode: 18           Links: 1
Access: (0755/-rwxr-xr-x)  Uid: ( 502/ UNKNOWN)   Gid: ( 502/ UNKNOWN)
Access: 2003-07-16 02:12:45.000000000 -0400
Modify: 2003-07-14 10:24:00.000000000 -0400
Change: 2003-07-16 02:05:33.000000000 -0400

[root@localhost Default]# ls -alis /mnt/img/prog
 18 480 -rwxr-xr-x  1 502  502  487476 Jul 14 10:24 /mnt/img/prog
```

Stat provided the inode information for the binary, and Autopsy was used to gather information about inode 18.

Autopsy Inode Report (ver 1.75)

```
-----
Inode: 18
Pointed to by file:
a:\prog
MD5 of istat output: 898f620ab5ef734c00c83ba366fe55d9
Image: /misc/gcfa/fl-160703-jp1.dd/images/fl-160703-jp1.dd
Image Type: linux-ext2
Date Generated: Mon Dec  8 12:37:12 2003
Investigator: hpierce
-----
```

```
inode: 18
Allocated
Group: 0
uid / gid: 502 / 502
mode: -rwxr-xr-x
size: 487476
num of links: 1
```

```
Inode Times:
Accessed:      Wed Jul 16 06:12:45 2003
File Modified: Mon Jul 14 14:24:00 2003
Inode Modified: Wed Jul 16 06:05:33 2003
```

Binary Details	
Name	prog (derived from the binary bmap)
Logical file size	487476
Physical file size	488448
Modify time	Mon Jul 14 14:24:00 2003
Access time	Wed Jul 16 06:12:45 2003
Change time	Wed Jul 16 06:05:33 2003
MD5 hash	7b80d9aff486c6aa6aa3efa63cc56880
Group	502 (unable to resolve gid)
User	502 (unable to resolve uid)
File permissions	0755 Write: owner only. Read:all. Execute: all.
Interesting strings within binary (complete output of <i>strings</i> against <i>prog</i> contained in Appendix A)	lengthy listing of devices reference to version "1.0.20 (07/15/03)" newt use block-list knowledge to perform special operations on files test for fragmentation wipe place data display data generate SGML invocation info %s has slack %s does not have slack bogowipe Keld Simonsen Remote I/O error Socket type not supported Destination address required Too many users

Program Description

Several tools were setup for this section. Cases were created in both Autopsy and Encase to track and correlate information about the forensic image.

Output from Autopsy and the utility *file* show the binary *prog* to be a statically linked, stripped, Executable and Linking Format (ELF) file. When a binary is statically linked, it means it does not depend on specific programming library versions.⁶ Stripping an executable removes debugging information, making the file smaller in size.⁷ The ELF binary format provides increased programming flexibility compared to previous binary formats.⁸

```
[root@localhost img]# file prog
prog: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.2.5,
statically linked, stripped
```

Encase shows a lost file at Inode 27, originally allocated as *prog*. This file in unallocated space at offset 414720, was exported file for further analysis. The similarity

⁶ <http://project.honeynet.org/reverse/results/sol/sol-10/answers.html>

⁷ http://www.cerias.purdue.edu/homes/forensics/old_projects/reverse/answers.html

⁸ <http://ww.telent.net/linux/ELF-HOWTO-1.htm>

in the *strings* output between *prog* and the unallocated file supports that the partially overwritten lost file at inode 27, represents the binary before it was moved or recompiled (last accessed time of 7/14/03 14:47:10). Encase and Autopsy show the last accessed timestamp for *prog* as 7/16/03 06:12:45, which indicates the last time it was run from this media.

Several seemingly unique strings were pulled from the larger strings listing of the binary and entered into the Google⁹ search engine. The following strings from *prog* suggest the function and identity of the binary. The version of 1.0.20 corresponds to, among others, *bonobo*, *libxslt*, *binutils*, and *bmap*. By doing more research with Google, the source for the binary *prog* becomes apparent. Google found the other three strings listed below in descriptions for the binary *bmap*, displayed on various web pages¹⁰:

```
1.0.20 (07/15/03)
generate SGML invocation info
newt
use block-list knowledge to perform special operations on files
```

The source code¹¹ for *bmap* version 1.0.20, contains the comments:

```
/* bmap.c userlevel blockmap utility for Linux.
 *
 * Maintained 2000 by Daniel Ridge in support of:
 * Scyld Computing Corporation.
 *
 * The maintainer may be reached as newt@scyld.com or C/O
 * Scyld Computing Corporation
 * 10227 Wincopin Circle, Suite 212
 * Columbia, MD 21044
 *
 * Written 1998,1999 by Daniel Ridge in support of:
 * Computer Crime Division, Office of Inspector General,
 * National Aeronautics and Space Administration.
 */
```

The binary *prog* was copied to the root folder of the VMWare Red Hat 9 (2.4.20-6 kernel) installation, where it was executed. Prior to execution, ethereal was started capturing data on the only network connection active (lo). Execution of the binary produced the command line options listing, as no arguments were provided:

⁹ <http://www.google.com>

¹⁰ One of the biggest clues that *prog* originated from *bmap* was finding a match for a descriptive string contained in the binary, at <http://old.lwn.net/2000/0413/announce.php3>

¹¹ The source code was downloaded from ftp://ftp.scyld.com/pub/forensic_computing/bmap/

```

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

```

Since the code within *prog* was based on the code for *bmap*, so must the function of the forensic binary be based on the function of the original tool.¹² Google reveals the purpose of *bmap* as a file system block mapping and data hiding utility¹³. The utility can be used to analyze a file for slack space available, and then place and retrieve data from the slack space.¹⁴ The utility will also wipe data contained in the slack space of a file.¹⁵

Given the nature of the *bmap* binary, a closer review of slack space was conducted on the files within the image. Encase, *bmap*, and the *prog* binary itself were used to analyze the files in the Price image. No other files contained obviously notable data in their slack, and all allocated files contained null data in their slack, except as follows:

File	Slack Size	Slack Notes
Sound-HOWTO-html.tar.gz	805 bytes	170 bytes are non-null and include the string "downloads"

The volume unallocated space was 646144 bytes, and contained notable strings such as:

¹² Anton Chuvakin discusses the intent of *bmap* at <http://mail.gnu.org/archive/html/bug-fileutils/2002-03/msg00017.html>

¹³ <http://www.google.com/search?hl=en&ie=UTF-8&oe=UTF-8&q=bmap+hiding>

¹⁴ http://www.linuxsecurity.com/feature_stories/data-hiding-forensics.html

¹⁵ <http://www.madchat.org/crypto/stegano/unix/covert/>

String	Comment
cannot stat file:	String contained in <i>prog</i> binary
GCC: (GNU) 3.2.2 20030222 (Red Hat Linux 3.2.2-5)	GCC compiler header information
vmware-config.pl	Suggests likelihood of VMWare for Linux running on a system in which the Price floppy was used
LOGNAME=root	Suggests that someone using the Price floppy, was operating as the Linux superuser.

The following is the approach taken to establish the binary's relationships with other files on a system as well as its operating behavior. The framework consists of analyzing a subset of the strings for the binary, reviewing the source code for *bmap*, running *strace*¹⁶ with the binary, and looking for unusual processes and network activity when the binary is run on a sandboxed Linux host.

The review of the strings output for the binary *prog* was conducted, this time focusing on the presence of "/" to draw out file references and commands embedded in the binary. Notable strings:

¹⁶ <http://www.liacs.nl/~wichert/strace>


```

syslog: unknown facility/priority: %x      /proc/self/cwd
/dev/console                               /proc
/dev/log                                   /etc/mtab
/etc/suid-debug                            /etc/fstab
/proc/sys/kernel/osrelease                 /cpuinfo
/usr/lib/gconv                              /meminfo
/usr/lib/gconv/gconv-modules.cache         /lib/
/usr/share/locale                           /usr/lib/
/usr/share/locale                           /etc/ld.so.cache
/locale.alias                              /proc/self/exe
/etc/localtime                             /usr/lib/locale
/usr/share/zoneinfo                        /SYS_

```

Earlier in the analysis it was noted that the binary is stripped, as indicated by the *file* command output. Further reverse analysis on the binary could be conducted by using the utility *dress*, which is part of the *fenris* package.¹⁷

The first *strace* output for *prog*, with the command switches for the binary set to display file block information for a copy of the *dd* binary:

```

[root@localhost tmp]# strace -c ./prog -s dd.tmp
execve("./prog", ["/prog", "-s", "dd.tmp"], [/* 33 vars */]) = 0
getting from block 437672
file size was: 30772
slack size: 1996
block size: 4096
% time      seconds  usecs/call   calls   errors syscall
-----
 28.41     0.007109      711        10      0      ioctl
 21.54     0.005392     2696         2      0      open
 19.84     0.004965     1655         3      0      fcntl64
 14.83     0.003711     3711         1      0      read
  9.17     0.002294      459         5      0      write
  2.63     0.000659      330         2      0      close
  1.43     0.000358      119         3      0      lstat64
  1.06     0.000265       53         5      0      brk
  0.27     0.000068       68         1      0      _llseek
  0.24     0.000059       59         1      0      uname
  0.16     0.000040       40         1      0      geteuid32
  0.14     0.000036       36         1      0      getuid32
  0.14     0.000036       36         1      0      getegid32
  0.14     0.000035       35         1      0      getgid32
-----
100.00     0.025027      37          37      0      total

```

The utility *strace* was run again on *prog* to capture process forking. The output is as follows:

¹⁷ <http://lcamtuf.coredump.cx/phenris>


```

[root@localhost tmp]# strace -ff ./prog -p textfile
execve("./prog", ["/prog", "-p", "textfile"], [/* 33 vars */]) = 0
fcntl64(0, F_GETFD) = 0
fcntl64(1, F_GETFD) = 0
fcntl64(2, F_GETFD) = 0
uname({sys="Linux", node="localhost.localdomain", ...}) = 0
geteuid32() = 0
getuid32() = 0
getegid32() = 0
getgid32() = 0
brk(0) = 0x80bedec
brk(0x80bee0c) = 0x80bee0c
brk(0x80bf000) = 0x80bf000
brk(0x80c0000) = 0x80c0000
lstat64("textfile", {st_mode=S_IFREG|0700, st_size=1722, ...}) = 0
open("textfile", O_RDONLY|O_LARGEFILE) = 3
ioctl(3, FIGETBSZ, 0xbffff244) = 0
lstat64("textfile", {st_mode=S_IFREG|0700, st_size=1722, ...}) = 0
lstat64("/dev/sda2", {st_mode=S_IFBLK|0660, st_rdev=makedev(8, 2), ...}) = 0
open("/dev/sda2", O_WRONLY|O_LARGEFILE) = 4
ioctl(3, FIGETBSZ, 0xbffff1b4) = 0
brk(0x80c2000) = 0x80c2000
ioctl(3, FIBMAP, 0xbffff244) = 0
write(2, "stuffing block 439632\n", 22stuffing block 439632
) = 22
write(2, "file size was: 1722\n", 20file size was: 1722
) = 20
write(2, "slack size: 2374\n", 17slack size: 2374
) = 17
write(2, "block size: 4096\n", 17block size: 4096
) = 17
_llseek(4, 1800734394, [1800734394], SEEK_SET) = 0
read(0, hi
"hi\n", 2374) = 3
write(4, "hi\n", 3) = 3
close(3) = 0
close(4) = 0
_exit(0) = ?

```

In all three cases, the *prog* binary performs expected operations:

strace Summary for Unknown Binary	
Function	Comment
uname({sys="Linux"	Check of system type
geteuid32()	Check to see if running as root
lstat64("textfile	Get information about logical file
lstat64("/dev/sda2	Get information about raw block device
open("/dev/sda2", O_WRONLY	Open raw block device for write only
write(4, "hi\n",	Write "hi" to the slack space of <i>textfile</i>

prog checks the system type and whether the binary is being run as root, then gets the logical size and slack size of the target file. It also checks the file system block size. The binary opens the device */dev/sda2* for writing, and writes the data input to the slack space of the target file. A display of the activity and file statistics is printed, and the program exits.

The *strace* output indicates that no additional processes begin and no network activity is initiated. However the binary does check to see if it is being run as root, as indicated by the *getuid* requests.

2. *prog* makes calls to `fcntl64`
3. *prog* makes calls to `geteuid32`, `getuid32`, `getegid32`, and `getgid32`

Prior to execution of the binary *prog*, *netstat* was run within the Linux host as shown below.

```
[root@localhost tmp]# netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 *:1024                  :::*                    LISTEN
tcp      0      0 localhost.localdom:1025 :::*                    LISTEN
tcp      0      0 localhost.localdoma:783 :::*                    LISTEN
tcp      0      0 *:sunrpc                :::*                    LISTEN
tcp      0      0 *:x11                   :::*                    LISTEN
tcp      0      0 *:ssh                   :::*                    LISTEN
tcp      0      0 localhost.localdoma:ipp :::*                    LISTEN
tcp      0      0 localhost.localdom:sntp :::*                    LISTEN
udp      0      0 *:1024                  :::*                    LISTEN
udp      0      0 *:sunrpc                :::*                    LISTEN
udp      0      0 *:631                   :::*                    LISTEN
udp      0      0 *:892                   :::*                    LISTEN
```

A port scan of all ports between 1 and 65535 using SuperScan3.00 on a WindowsXP host, verifies the *netstat* output. The TCP port scan against the IP address assigned to the VMWare Linux host showed ports 22, 111, 1024, and 6000 open, corresponding with the *netstat* listing of SSH, SunRPC, unknown on 1024, and x11 (respectively).

Also before the binary was executed, the current process list was captured:

```
[root@localhost tmp]# ps -x > /mnt/hgfs/Default/before.wri
```

Execution of the binary *prog*:

```
[root@localhost tmp]# ./prog -s dd.tmp
getting from block 437672
file size was: 30772
slack size: 1996
block size: 4096
```

A capture of the post-execution process list was performed:

```
[root@localhost tmp]# ps -x > /mnt/hgfs/Default/after.wri
```

The utility *diff* created a side-by-side comparison of the two process lists:

```
[root@localhost tmp]# diff -y /mnt/hgfs/Default/before.wri /mnt/hgfs/Default/after.wri
```

The comparison shows only changes in processor time for existing processes, and does not show any new processes or any apparent changes to existing processes (no new PIDs).

A repeat of the *netstat* and remote SuperScan analysis of network ports returns identical results, showing no changes in network port presence as a result of execution of the binary *prog*.

Running the binary with command line switches to place data allowed for capture of the files it opened during its execution, using *lsuf*:

```
[root@localhost tmp]# ps -x | grep prog
2408 pts/0    S      0:00 ./prog -p dd.tmp this is a test
2420 pts/1    S      0:00 grep prog
[root@localhost tmp]# lsuf -p 2408
COMMAND  PID USER  FD  TYPE DEVICE  SIZE  NODE NAME
prog     2408 root   cwd  DIR   8,2   4096  207585 /tmp
prog     2408 root   rtd  DIR   8,2   4096    2 /
prog     2408 root   txt  REG   8,2  487476  210403 /tmp/prog
prog     2408 root    0u  CHR  136,0    2 /dev/pts/0
prog     2408 root    1u  CHR  136,0    2 /dev/pts/0
prog     2408 root    2u  CHR  136,0    2 /dev/pts/0
prog     2408 root    3r  REG   8,2   30772  210404 /tmp/dd.tmp
prog     2408 root    4w  BLK   8,2    66190 /dev/sda2
```

While the strings output for the binary *prog* displays a number of phrases that suggest the binary is network aware, the execution of the binary so far has not resulted in any detected network activity. Network activity must be triggered by a switch on the command line that has not yet been included. The binary *bmap* does not contain any similar strings, suggesting that *bmap* in its original form is not network aware and that network awareness was added to the feature set when *prog* was compiled.

Two major questions remain regarding this binary: how does the binary become network-active? And, does the binary modify otherwise unrelated files on the host system?

To answer the second question, a hash set was created of system files.

Files contained in the `bin`, `boot`, `sbin`, and `tmp` directories were hashed using:

```
find /path/to/md5/here -type f -exec md5sum {} \; > outputfile
```

The binary *prog* was run with the display data and place data switches set. A post-execution hash set was made of the same directories as listed above. A *diff* comparison shows no changes to the files as a result of execution of the binary. Evidently the binary does not modify other files on the system, unless specified by the user (i.e. data placement in slack space).

Turning attention to the binary with more obvious network awareness, the RPM of *netcat*, some notable details emerge. The *netcat* RPM located in the Price image neither contains an internal digital signature, nor matches the same version RPM distribution from <http://at.rpmsfind.net/opsys/linux/RPM/redhat.com/dist/linux/8.0/en/os/i386/nc-1.10-16.i386.html>. The RPM md5 hash at the distribution web site for *netcat1.10-16* is `28a74be2ef8415af4486770f66ebfffc`. While the *md5sum* output for the Price image *netcat1.10-16* RPM is `535003964e861aad97ed28b56fe67720`. Screen output detailing these findings is below.

```
[root@localhost tmp]# rpm -qpl nc-1.10-16.i386.rpm..rpm
warning: nc-1.10-16.i386.rpm..rpm: V3 DSA signature: NOKEY, key ID db42a60e
/usr/bin/nc
/usr/share/doc/nc-1.10
/usr/share/doc/nc-1.10/Changelog
/usr/share/doc/nc-1.10/README
/usr/share/doc/nc-1.10/scripts
/usr/share/doc/nc-1.10/scripts/README
/usr/share/doc/nc-1.10/scripts/alta
/usr/share/doc/nc-1.10/scripts/bsh
/usr/share/doc/nc-1.10/scripts/dist.sh
/usr/share/doc/nc-1.10/scripts/irc
/usr/share/doc/nc-1.10/scripts/iscan
/usr/share/doc/nc-1.10/scripts/ncp
/usr/share/doc/nc-1.10/scripts/probe
/usr/share/doc/nc-1.10/scripts/web
/usr/share/doc/nc-1.10/scripts/webproxy
/usr/share/doc/nc-1.10/scripts/webrelay
/usr/share/doc/nc-1.10/scripts/websearch
/usr/share/man/man1/nc.1.gz
[root@localhost tmp]# md5sum nc-1.10-16.i386.rpm..rpm
535003964e861aad97ed28b56fe67720 nc-1.10-16.i386.rpm..rpm
```

Next, the *netcat* RPM was installed on the Linux VMWare guest.

```
[root@localhost tmp]# rpm -ivh nc-1.10-16.i386.rpm..rpm
warning: nc-1.10-16.i386.rpm..rpm: V3 DSA signature: NOKEY, key ID db42a60e
Preparing...      ##### [100%]
 1:nc              ##### [100%]
```

Data placement in slack space is possible over a network by employing *netcat* and piping the data through *prog*. First, the *netcat* binary is placed in listening mode on the Linux VMWare guest, with data piped to the *prog* binary in slack placement mode. Data will be stuffed in the slack space of the test file *textfile*. The binary *prog* shows the available slack space for *textfile* as 2374 bytes.

```
[root@localhost tmp]# nc -l -p 5000 | ./prog -p textfile
stuffing block 439632
file size was: 1722
slack size: 2374
block size: 4096
```

The second step occurs on the Windows host computer, where a file is chosen to send via the Windows port of *netcat*. In this case a file containing known data (the character “f”) is created with a file size of 3118 bytes. The Windows command *type* (similar in function to Linux’s *cat*) was used to feed the contents of *text.txt* to the Windows port of *netcat*.

```

C:\DOS\tools>dir text.txt
Volume in drive C is HPNOTEBOOK
Volume Serial Number is E001-1316

Directory of C:\DOS\tools

10/26/2003  04:41 PM                3,118 text.txt
             1 File(s)                3,118 bytes
             0 Dir(s)  68,442,088,448 bytes free

C:\DOS\tools>type text.txt | nc 192.168.1.250 5000

26/10/2003  21:41:50 (UTC)
26/10/2003  16:41:50 (local time)

```

The result of this exercise was the stuffing of the contents of *text.txt* on the Windows host, into the slack space of the file *textfile* on the Linux guest, accomplished over the network using the *netcat* installation found in the Price forensic image.

It should be noted that the amount of data to be stuffed in slack space was greater than the amount of slack space remaining in *textfile* (3118 bytes sent for stuffing vs. 2374 bytes of slack space available). The binary *prog* handled this by silently truncating the stuffed data at 2374 bytes.

The file system of the floppy contained in the Price forensic image is ext2, which has a block size of 4096 bytes. This means the maximum amount of space usable to *prog* per allocated file, is 4096 bytes. This amount of data would be sufficient for hiding short text messages or very small binaries. However, if Mr. Price was hiding large amounts of data, he could accomplish the task by spreading the hidden data across many files within a file system. In a typical file system containing a Linux or Windows operating system (the Price evidence shows that Mr. Price was using both operating systems to produce the evidence¹⁸), there may be 2000 to 3000 files available for data hiding. If each file only contains an average of 500 bytes, Mr. Price could hide between 1 and 1.5 megabytes of data. This volume of data could consist of a large binary, a short compressed video clip, a compressed medium bit-rate audio file, or a large spreadsheet or text document. It is plausible that with the intention of hiding data in slack space, the number of files contained in a file system could be intentionally increased to make more aggregate slack space available. The creation of many files with small amounts of data (as few as only a few bytes), could provide as much as 4000 bytes of slack space per file in an ext2 file system.

Placement and retrieval of data spread across the slack space of many files in a file system could be automated by a shell or perl script.¹⁹

Forensic Details

¹⁸ The file system of the floppy image is Linux, and the Word documents contained within the image have references to the Windows version of Microsoft Word in the meta-data. See the Case Information section.

¹⁹ <http://indigo.ie/~enda/listarchives/0202/1687.html>

Much of the foundational information on which the following conclusions are based can be found in the preceding section.

No evidence was found that the binary *prog* causes changes to other files as a result of installing the program. In fact, installation of the binary consisted of copying the binary from the forensic image to a location on the Linux analysis system. The only activity that did cause detectable changes to files, was execution of the binary using the command switch to place or wipe data from file slack space. Upon execution, the binary makes system calls to get information about the operating system, the context in which it is being run²⁰, information about the target file, and makes system calls to write data to the target file if specified in the execution switch.

Conclusions on the binary's dependence on other files on the system can be drawn from analysis of *bmap* and of the strings pulled from *prog*. Since the binary is statically linked, there are no dynamic dependencies. The *strings* output for *prog* in the previous section returned a number of relevant hits, when using *grep* for the character "/".

```
/etc/suid-debug
/proc/sys/kernel/osrelease
/usr/lib/gconv
/usr/lib/gconv/gconv-modules.cache
/usr/share/locale
/locale.alias
/etc/localtime
/usr/share/zoneinfo
/proc/self/cwd
/proc
/etc/mstab
/etc/fstab
/cpuinfo
/meminfo
/lib/
/usr/lib/
/etc/ld.so.cache
/proc/self/exe
/usr/lib/locale
/SYS_
```

Encase performed keyword searches on a number of strings, without any notable hits. Those strings providing no enlightenment include:

```
pass
User
ftp
jprice
e2compr
ebay
May03
BurnEye
ELF
John
```

And, a search was done for all strings that look like email addresses and IP addresses. There were a number of hits for email addresses, most of which occur in the *netcat* RPM, the DVD How-to document, and in the binary *prog*. The first two sources are not

²⁰ The fact that the binary checks to see if it is being run in the root context is likely due to the need for root privileges when manipulating raw devices in Linux. See <http://ftp3.mplayerhq.hu/MPlayer/DOCS.bak/cd-dvd.html> for discussion on raw sector access.

notable since they are references to their respective authors. The email address appearing in the binary was that of Keld Simonsen (keld@dkuug.dk), who works on character sets and locale formats.²¹

The grep search for IP addresses across the entire forensic image returned no matches.

Several leads may be useful if this investigation continued to include other computers to which Mr. Price had access:

Lead	Interpretation
The file <i>Mikemsg.doc</i> , within the forensic image, contains the following string: 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	Suggests the files received are music files, and that the intent exists to sell them.
The file in the forensic image: DVD-Playing-HOWTO-html.tar	Provides information on enabling DVD playback on Linux systems
The file in the forensic image: Kernel-HOWTO-html.tar.gz	Instruction on recompiling the Linux kernel to support multimedia playback
The file in the forensic image: MP3-HOWTO-html.tar.gz	Provides information on MP3 playback on Linux systems
The file in the forensic image: Sound-HOWTO-html.tar.gz	Information about configuring and troubleshooting sound on Linux systems

Program Identification

One conclusion of this analysis is that the binary *prog* originated from the binary *bmap*. The source code for *bmap* was located on the Internet at:
ftp.scyld.com/pub/forensic_computing/bmap/

Upon being compiled on the RedHat 9 analysis machine, *bmap* generates an md5 hash of `fdf425e23ff71cbbf507f1f507fcd445` which differs from the *prog* md5sum output of `7b80d9aff486c6aa6aa3efa63cc56880`.

At least three factors contribute to the hash discrepancy. First, network functions were added to *prog* that are not present in *bmap*. Second, internal filename and date references are dissimilar. Third, the *bmap* binary was compiled on a RedHat 9 system using GCC 3.2.2. The binary *prog* was compiled on a Red Hat Linux 7.3 system using GCC 2.96 (objdump output shown below for *prog*).

²¹ <http://lists.w3.org/Archives/Public/www-international/2000AprJun/0087.html>

```

[root@localhost tmp]# objdump -j .comment -s prog

prog:      file format elf32-i386

Contents of section .comment:
0000 00474343 3a202847 4e552920 322e3936 .GCC: (GNU) 2.96
0010 20323030 30303733 31202852 65642048 20000731 (Red H
0020 6174204c 696e7578 20372e33 20322e39 at Linux 7.3 2.9
0030 362d3131 32290000 4743433a 2028474e 6-112)..GCC: (GN
0040 55292032 2e393620 32303030 30373331 U) 2.96 20000731
0050 20285265 64204861 74204c69 6e757820 (Red Hat Linux
0060 372e3320 322e3936 2d313132 29000047 7.3 2.96-112)..G
0070 43433a20 28474e55 2920322e 39362032 CC: (GNU) 2.96 2
0080 30303030 37333120 28526564 20486174 0000731 (Red Hat
0090 204c696e 75782037 2e332032 2e39362d Linux 7.3 2.96-
00a0 31313329 00004743 433a2028 474e5529 113)..GCC: (GNU)
00b0 20322e39 3e332032 30303037 33312028 2.96 20000731 (
00c0 52656420 48617420 4c696e75 7820372e Red Hat Linux 7.
00d0 3320322e 39362d31 31332900 00474343 3 2.96-113)..GCC
00e0 3a202847 4e552920 322e3936 20323030 : (GNU) 2.96 200
00f0 30303733 31202852 65642048 6174204c 00731 (Red Hat L
0100 696e7578 20372e33 20322e39 362d3131 inux 7.3 2.96-11
0110 33290000 4743433a 2028474e 55292032 3)..GCC: (GNU) 2
0120 2e393620 32303030 30373331 20285265 .96 20000731 (Re
0130 64204861 74204c69 6e757820 372e3320 d Hat Linux 7.3
0140 322e3936 2d313133 29000047 43433a20 2.96-113)..GCC:
0150 28474e55 2920322e 39362032 30303030 (GNU) 2.96 20000
0160 37333120 28526564 20486174 204c696e 731 (Red Hat Lin
0170 75782037 2e332032 2e39362d 31313329 ux 7.3 2.96-113)
0180 00004743 433a2028 474e5529 20322e39 ..GCC: (GNU) 2.9
0190 36203230 30303037 33312028 52656420 6 20000731 (Red
01a0 48617420 4c696e75 7820372e 3320322e Hat Linux 7.3 2.
01b0 39362d31 31332900 00474343 3a202847 96-113)..GCC: (G
01c0 4e552920 322e3936 20323030 30303733 NU) 2.96 2000073
01d0 31202852 65642048 6174204c 696e7578 1 (Red Hat Linux
01e0 20372e33 20322e39 362d3131 33290000 7.3 2.96-113)..
01f0 4743433a 2028474e 55292032 2e393620 GCC: (GNU) 2.96
0200 32303030 30373331 20285265 64204861 20000731 (Red Ha
0210 74204c69 6e757820 372e3320 322e3936 t Linux 7.3 2.96
0220 2d313133 29000047 43433a20 28474e55 -113)..GCC: (GNU)
0230 2920322e 39362032 30303030 37333120 ) 2.96 20000731
0240 28526564 20486174 204c696e 75782037 (Red Hat Linux 7
0250 2e332032 2e39362d 31313329 00004743 .3 2.96-113)..GC
0260 433a2028 474e5529 20322e39 36203230 C: (GNU) 2.96 20
0270 30303037 33312028 52656420 48617420 000731 (Red Hat
0280 4c696e75 7820372e 3320322e 39362d31 Linux 7.3 2.96-1
0290 31332900 00474343 3a202847 4e552920 13)..GCC: (GNU)
02a0 322e3936 20323030 30303733 31202852 2.96 20000731 (R
02b0 65642048 6174204c 696e7578 20372e33 ed Hat Linux 7.3
02c0 20322e39 362d3131 33290000 4743433a 2.96-113)..GCC:
02d0 2028474e 55292032 2e393620 32303030 (GNU) 2.96 2000
02e0 30373331 20285265 64204861 74204c69 0731 (Red Hat Li
02f0 6e757820 372e3320 322e3936 2d313133 nux 7.3 2.96-113
0300 29000047 43433a20 28474e55 2920322e )..GCC: (GNU) 2.
0310 39362032 30303030 37333120 28526564 96 20000731 (Red
0320 20486174 204c696e 75782037 2e332032 Hat Linux 7.3 2
0330 2e39362d 31313229 00 .96-112)..

```

Arriving at a compiled *bmap* binary that matches the *md5sum* output from *prog* would be a significant challenge, considering the differences in the operating system environment, compiler used, and the addition of known and unknown features in the source code. Some features known only by their existence in the strings output from *prog* include network awareness. Unknown changes may include removal of sections of code, and changes to the commands that are passed to the binary at execution. Additionally, the binary is statically linked and stripped, making it infeasible to reverse

engineer *prog* to the point where it is possible to collect enough information to compile *bmap*, so that it results in an identical hash.²²

Legal Implications

Proof that Mr. Price had distributed copyrighted material was not found in the investigation. A portion of circumstantial evidence was found suggesting he was distributing audiovisual material.

The evidence ties Mr. Price's user account to access of the *prog* binary, and access of a Linux system as the root user, as well as access to a Windows system as the Administrator.

If Mr. Price had not been granted authority to access company computer systems as this privilege level, the Computer Fraud and Abuse Act ("CFAA") (18 USC 1030) would apply. The CFAA may have been violated if the computer/computer system was accessed without authorization or if authorization was exceeded.

Depending on the terms of Mr. Price's contract with the company, and the company's Acceptable Use Policy, there may be civil recourse using contract law. One potential catch-all clause in an AUP, is that company computer resources are for authorized company business only. Clearly Mr. Price violated this requirement in several ways.

A breach of contract suit may result in punitive damages as well as recovery of costs associated with investigation of the matter and legal fees, especially if the company was involved in a criminal charged based on Mr. Price's actions alone.

Interview Questions

1. A Microsoft Word document was found in the evidence we collected. The letter is written to Mike, signed with your initials, and saved to a folder bearing your name. Can you explain how this letter came to bear your name?
2. Regarding the same letter, what role has Mike been playing?
3. When we talk to Mike, will he admit to using this binary to distribute copyrighted material, or will he blame you? Who should we believe? Why?
4. The text of the letter states that you are "ready to rock-n-roll". What did you mean by this? How does that statement relate to the statement earlier in the letter where you say that you "received the latest batch of files last night"?
5. Why did you wipe your hard drive before it was seized?
6. The binary was last modified on Monday, 14 July 2003 at 2:24pm GMT. That was a work day; were you at work that day using your computer?
7. The binary was last accessed on Wednesday, 16 July 2003 at 6:12am GMT. That was also a work day; were you at work then or elsewhere?
8. What other computers do you have access to? A Linux 7.3 system? A Windows XP or 2000 system? A system with VMWare software installed?

²² <http://project.honeynet.org/reverse/results/sol/sol-08/answers.html>

Case Information

In an effort to reveal potentially hidden content, the GIF and JPG pictures in the Price forensic image were analyzed steganography content. The file *sect-num.gif* was located via Google at a University of Melbourne web site²³, but now returns a 404 – page not found error when attempting to load the image. A comparison of *sect-num.gif* from the Google cache and that from the Price image follows:

<i>Sect-num.gif</i> from Google cache md5sum:	fd8cb5272589dc32a4d88a22fe60d0b0
<i>Sect-num.gif</i> from Price forensic image md5sum:	636be3f63d098684b23965390cea0705

Given that these two files are not identical, it opens the possibility that should the two files have originated from the same source, then the file contained in the Price forensic image may have been altered. The alteration could be in the form of embedded and hidden data.

Attempts to detect steganography embedded data in each of the GIF and JPG files failed using: JPHS for Windows Beta rev0.5, S-Tools for Windows v4, and Stegdetect v0.4.

None of the GIF or JPG files appears to contain data appended to the logical file in the slack space, by the binary *prog*. This conclusion was attained after review of the slack space for each file, with Encase 4.16. The slack space for all three picture files was null (hex “00”).

Meta data contained within the two Microsoft Word document shows strings indicating both documents were edited on a computer with the Windows operating system.

```
C:\Documents and Settings\Administrator\Desktop\Mikemsg.doc
John Price::C:\Documents and Settings\Administrator\Desktop\Letter.doc
```

This meta data, extracted using Encase, also shows John Price was using the Administrator account on the Windows computer by virtue of accessing the Administrator Desktop folder. The folder hierarchy displayed above contains the “C:\Documents and Settings\” folder, which was not introduced to Windows computers until Windows 2000. From this it can be concluded that the operating system of the computer on which the documents were edited, was either Windows 2000 or Windows XP.

The Microsoft Word file contained in the floppy image, called *Mikemsg.doc*, is useful because it shows that in addition to the Linux system that created the floppy, there was also a Windows computer being used by Mr. Price. The content of the Word document opens the possibility that Mr. Price emailed the document to “Mike”. If this is the case,

²³ http://www.dis.unimelb.edu.au/mm/hwtute/peripheral_devices/disk.htm

an examination of Mr. Price's company email account should be conducted²⁴ at least to establish that he did send the Word document using his email. This would be compelling evidence that the floppy disk did belong to Mr. Price, refuting his statement that it did not.

The existence of two machines in Mr. Price's control, and the fact that the case summary only discusses the wiping of one of those machines, leaves to possibilities. One, that the one machine contained VMWare, and the second machine was a guest on the host machine. Or second, that a completely separate computer exists that may contain evidence in this matter. If a second machine exists, it may contain significant amounts of information related to this case.

By using the MAC times of the files (most importantly *prog*) contained in the forensic image, some correlation may be possible with other technical or human information. The times associated with the binary *prog* are limited to 14 and 16 of July 2003. If there are other sources of time information that can be linked to Mr. Price, it could establish his presence at the computer when the binary was being accessed or modified. Potential sources might be corporate network login, print, email, firewall, or web proxy logs that would contain both personally identifiable information and a time stamp. For example, if a log exists of a print job that was submitted at 06:13 GMT on 16 July 2003 using Mr. Price's network credentials. That log entry would provide strong support for Mr. Price's control of the system during the time when the binary was last accessed, a minute earlier.

The Systems Administrators could watch several aspects of the company network to detect whether this binary is in use elsewhere in the company. A port scan would reveal unexpected listening ports, such as *netcat* or if *prog* were otherwise configured for network activity. The presence of *netcat* listening on a system may only be differentiated by the fact that it would not display a connect banner, such as would be expected from most other legitimate services (mail, ftp, web).

Another indirect way of determining whether the activity of the binary remains, is by looking for the byproduct of the binary: the copyrighted material. Based on the evidence, the copyrighted material Mr. Price has been accused of distributing, appears to be multimedia files. A review of the storage locations on the company network for audio and video files may reveal a repository of Mr. Price's activity, and any continuing activity as a result of still-active binary installations. Considering that no encryption function was found in any of the available Price evidence, the Systems Administrators may also configure their Intrusion Detection Systems to alert on MP3 and MPEG headers from files in transit across the network. For example:

```
25alert tcp $HOME_NET any <> $EXTERNAL_NET any (msg:"MP3 File Found"; flow:to_server,established; content:".mp3"; nocase; classtype:policy-violation; sid:562; rev:5;)
```

²⁴ Assuming there is an acceptable use policy in-place at the company, and Mr. Price was duly aware of that policy. And, that the policy makes clear that the company has the right to access employee email, as a condition of employment.

²⁵ <http://www.cc.gatech.edu/~andre/cs6265/19>

MP3 without embedded data in track:

\xFF\xFB\xD0\x04\x00\x00\x00\x00

MP3 with track data²⁶:

\x49\x44\x33\x03\x00\x00\x00\x00

Additional Information

Information sources where specific details were used in the analysis are footnoted on each page. However, background information and sources for additional research are as follows:

Source	Relevance
http://www.esj.com/columns/print.asp?editorialsId=88	High-level background information on data hiding, file system structures, and tools.
http://www.giac.org/practical/GCFA/Alexander_Kotkov.pdf	Analysis of a different Linux binary, but the framework is similar to this analysis.
http://slashdot.org/articles/02/03/13/2053246.shtml?tid=106	Linux data hiding and recovery web forum, mentions <i>bmap</i>
http://project.honeynet.org/reverse/index.html	The Honeynet Project Reverse Challenge contains analyses of unknown binaries and reverse engineering methods.

²⁶ Headers courtesy of a posting to the Encase Users' Forum by Craig Wilson, Kent Police, UK

Part 2 – Option 2: Perform Forensic Tool Validation

Scope

Guidance Software of the United States, and Crocware of Australia, are vendors of software products that assist in the forensic investigation of computers. Guidance Software (GSI) produces the EnCase multipurpose forensics tool, and Crocware produces file recovery and forensic image access utilities.

The EnCase tool is widely used among investigators using Microsoft Windows as their forensics platform.²⁷ GSI currently produces forensics software that runs on the Windows platform only.²⁸ The sizable market penetration of EnCase has resulted in many investigations with EnCase proprietary-format forensic images. The EnCase image file format is not currently native to any other forensic utility²⁹. However, EnCase itself can read a number of forensic image file formats, including images created with Linux *dd*.

Mount Image Pro is a product from Crocware that enables Windows systems to mount EnCase and Linux *dd* forensic images as a logical drive. It provides this function without the presence of EnCase or *dd*. Mount Image Pro also mounts SMART images as logical drives.³⁰

This paper attempts to validate the reliability and suitability of Crocware's Mount Image Pro, as a tool for manipulation of the forensic image files produced by EnCase and Linux *dd*.

Tool Description

Computer forensics practitioners may use the Mount Image Pro tool (and subsequently find this validation useful) for several reasons:

1. There are currently only two methods of accessing an EnCase image: by using EnCase (\$2495 or \$1995 for Gov/Edu)³¹ or by mounting the image file with Mount Image Pro (\$299)³². The price disparity is noteworthy.
2. The ability to mount an EnCase image as a logical or physical drive may allow re-acquisition of the image using another imaging tool (*dd*, Safeback, etc.)
3. Mounting an EnCase image as a drive from Windows, allows for use of third-party tools on the image. Two likely examples of these tools are anti-virus and file recovery (when image mounted as physical drive).

²⁷ <http://www.guidancesoftware.com/corporate/index.shtml>

²⁸ Guidance Software does currently produce what it terms a servlet that runs on a Linux system, and provides access to that host for investigation done via EnCase Enterprise Edition. Both EnCase Forensic Edition, produced primarily for law enforcement and consultants, and EnCase Enterprise Edition, marketed to Corporations, run only on Windows operating systems. See <http://www.guidancesoftware.com/products/EnCaseEnterprise/features.shtml>

²⁹ Except Crocware's Mount Image Pro. Read on.

³⁰ <http://www.mountimage.com/>

³¹ <http://www.guidancesoftware.com/products/EnCaseForensic/purchase.shtml>

³² <http://www.crocware.com/>

4. GSI intends to release an add-on module for EnCase that will enable mounting of EnCase images. However, as of this writing it has not been released. Additionally, it is not clear yet what level of compatibility the module will provide. The advertised pricing for the GSI module is \$355.³³

If Crocware's Mount Image Pro delivers on the technology it promises, it opens a significant amount of flexibility in the requirements for accessing EnCase proprietary image files, and would foster conversion from an EnCase image to another format without the need to reacquire the original media. Reducing the need to touch original media furthers one best practice in computer forensics: protect the original media. As well, Mount Image Pro could significantly reduce the burden of using third-party tools on EnCase images, and *dd* images from within Windows.

The subject of validation is Mount Image Pro, version 1.05, released 28 November 2003. This is a young product, as the release of version 1.00 was 10 November 2003.

Crocware's web site makes specific claims about the capabilities of Mount Image Pro. Some highlights include:³⁴

...enables you to mount ENCASE®, Unix DD, or SMART® forensic images as a drive letter on your file system.

It fully maintains the MD5 HASH integrity which can be tested by a reacquisition of the mounted drive...

Map images as a single drive letter to explore "Unused/Non partitioned" disk space...

You now have a Read Only forensic image as a drive letter.

Contact information for the vendor follows:

CrocWare Pty Ltd

PO Box 71

Engadine NSW

Australia 2233

+61 2 95457788

<http://www.mountimage.com>

Mount Image Pro is produced for Microsoft Windows only. The documentation also notes that while the product has the ability to mount images containing ext2/ext3 file systems, it does not contain drivers to interpret the file system. Third party file system drivers are required for ext2 and ext3 interpretation.

Crocware offers a 21-day fully-functional evaluation download of Mount Image Pro at its web site. This is the software to be used in the validation. The software must be installed on a Windows system for use. In most respects, use of this tool assumes that evidence has been collected in a forensically sound manner. The tool's use is limited to a laboratory setting, rather than *in-situ* or analysis of a system that is still powered on.

³³ Pricing per a post to the Guidance Software User's Board by Jonathan Bair, Senior Director of Product Development, on 21 November 2003.

³⁴ <http://www.mountimage.com/>

This validation will review the accuracy of the claims regarding the tool's operation as read-only, and handling image files during reacquisition so the hash is maintained.

Test Apparatus

Three data sources will be used: a Microsoft Windows 2000 Professional Service Pack 2 CD, the floppy disk image provided for the SANS GCFA practical version 1.4, and an hda8 partition from the HoneyNet Project Forensic Challenge³⁵ of January 15, 2001.

The Windows 2000 Professional Service Pack 2 CD will be referred to in testing as "Win2kProCD".

The SANS GCFA v1.4 floppy disk image will be referred to in testing as "fl-160703-jp1". The HoneyNet Project hda8 partition image will be referred to in testing as "hn-hda8"

The test design involves establishing a baseline hash for six images of the three data sources. A Linux *dd* and EnCase image file will be created and hashed for each data source, totaling six image files and six baseline MD5 hashes.

Testing Apparatus	Version	Function
Windows XP Professional	5.1.2600	Test platform operating system
GSI EnCase Forensic Edition	4.16a	EnCase image file generation and hashing
VMWare Workstation	4.0.5 build-6030	Contains Linux guest operating system
Red Hat Linux 9	2.4.20-6	Virtual machine guest operating system for <i>dd</i> generation and hashing
<i>dd</i> on Linux	4.5.3	Image file generation
<i>md5sum</i> on Linux	4.5.3	Hash generation
<i>dd</i> for Windows (cygwin)	3,16,2,1029	Image file generation for mounted image files
<i>Md5sum</i> on Windows (cygwin)	2,0,1,1029	Hash generation on Windows system

A complete list of patches installed on the Windows XP Professional testing platform are included in Appendix B. For the purpose of brevity, it should be noted that the operating system was fully patched with all Microsoft released critical updates as of 10 December 2003.

Environmental Conditions

All testing was completed using an Hewlett-Packard Pavilion N5000 laptop, with an AMD Athlon4 1GHz processor, 512MB of RAM, a 7200rpm hard disk drive, and a 24x

³⁵ <http://honeynet.linuxsecurity.com/download/honeypot.hda8.dd.gz> -- the md5 hash provided by the HoneyNet project was verified upon download of the gzipped *dd* image (original and verified hash was b4ff10d5fd1b889a6237fa9c2979ce77)

CD-RW drive. The laptop was connected to an active LAN during the testing to allow for Internet research. None of the testing directly required or used network functions. The VMWare Red Hat 9 guest operating system was configured for direct access to the CD drive, and was allocated 128MB of system RAM.

Description of Procedures

In creating the *dd* image for Win2kProCD, among the commands given to *dd* were the specification that data should be read and written in 2048 byte units, with a block count of 189500. This specification was used because the documentation for Mount Image Pro states that *dd* images must be multiples of 512 bytes. If the image is not a multiple of that unit, the remainder will be excluded when the image is read.³⁶ It remains to be seen whether this limitation in Mount Image Pro will have a substantive effect on reacquisition of images. Information regarding the block size and count on the CD was determined by using *isozsize*.³⁷

```
[root@localhost tmp]# isozsize -x /dev/cdrom
sector count: 189500, sector size: 2048
[root@localhost tmp]# dd bs=2048 count=189500 if=/dev/cdrom of=/tmp/Win2kProCD.dd
189498+0 records in
189498+0 records out
[root@localhost tmp]# md5sum Win2kProCD.dd
45f06a9b19f26b5964f13faf67d9b670 Win2kProCD.dd
```

In the acquisition of the CDROM image, *dd* imaged two sectors less than *isozsize* reports.

A verification hash of the SANS *dd* floppy image.

```
[root@localhost tmp]# md5sum fl-160703-jp1.dd
20be7bc13a5cb8d77232659c52a3ba65 fl-160703-jp1.dd
```

A verification hash of the HoneyNet Project *dd* hard disk partition image.

```
[root@localhost tmp]# gunzip -d honeypot.hda8.dd.gz
[root@localhost tmp]# md5sum honeypot.hda8.dd
8f244a87b8d38d06603396810a91c43b honeypot.hda8.dd
```

EnCase acquisitions were performed without using compression or password protection of the resulting image files.

The acquisition of the floppy image fl-160703-jp1 was done by restoring the original SANS image to a physical floppy disk, using *dd* to perform the restore. EnCase was used to prepare the floppy disk by wiping the volume with hex 00.

The command sequence to restore the floppy image follows:

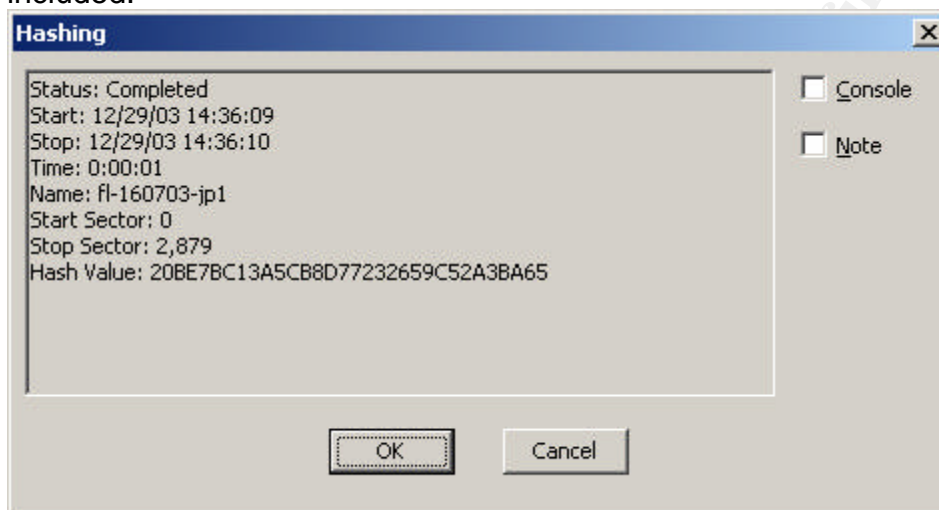
³⁶ <http://www.mountimage.com/downloads/mountimagepro-manual.pdf>

³⁷ <http://www.usssg.iu.edu/hypermail/linux/kernel/0112.3/0602.html>

```
[root@localhost tmp]# dd if=f1-160703-jp1.dd of=/dev/fd0
2880+0 records in
2880+0 records out
[root@localhost tmp]# md5sum f1-160703-jp1.dd
20be7bc13a5cb8d77232659c52a3ba65  f1-160703-jp1.dd
[root@localhost tmp]# md5sum /dev/fd0
20be7bc13a5cb8d77232659c52a3ba65  /dev/fd0
```

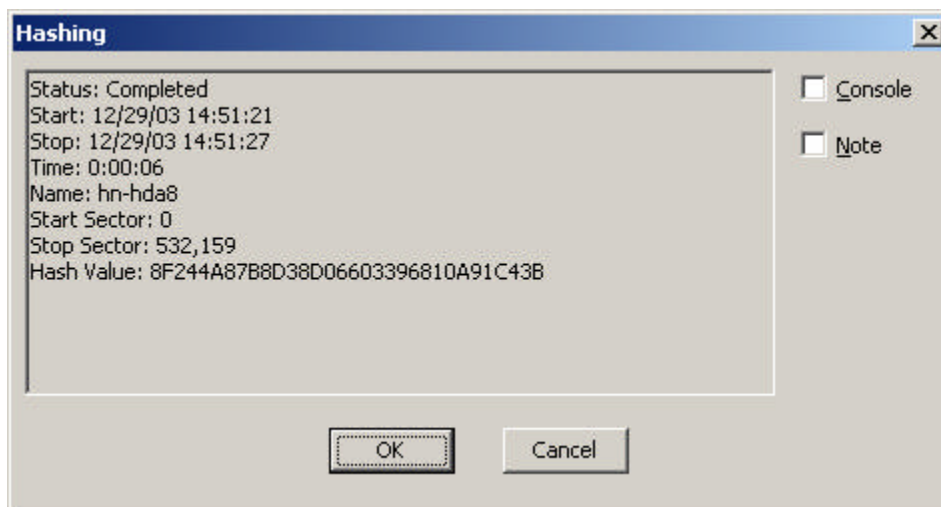
The matching *md5sum* hashes indicates a successful restore of the *dd* image to the floppy disk. The floppy diskette write-protect window was opened to enable write-protection, as soon as the restore was completed (before the hash was done).

Below shows a screen capture of the acquisition of the fl-160703-jp1 floppy disk by EnCase into EnCase image file format. The simultaneous hash of the new image file is included.



The EnCase image hash matches both the original *dd* image file hash and the restored floppy image hashes.

The hn-hda8 *dd* image was opened via EnCase, and a reacquisition of the image was performed into the EnCase image file format. The result and verified hash is in the next screen capture.



The last EnCase image file format acquisition was for the Win2kProCD. The results of the acquisition and verification hash are included as follows:

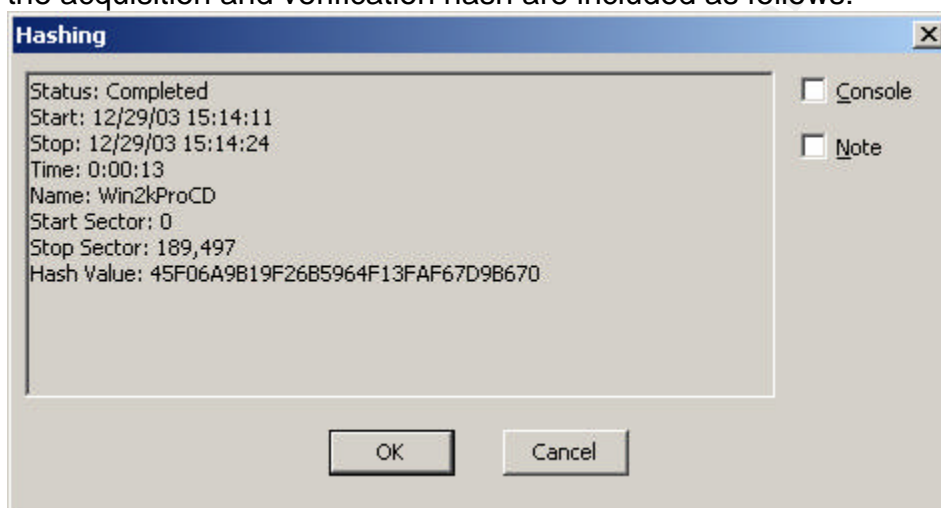


Image Preparation Summary		
Data Source	Image Type	MD5 Hash
Win2kProCD	dd	45f06a9b19f26b5964f13faf67d9b670
Win2kProCD	EnCase	45f06a9b19f26b5964f13faf67d9b670
fl-160703-jp1	dd	20be7bc13a5cb8d77232659c52a3ba65
fl-160703-jp1	EnCase	20be7bc13a5cb8d77232659c52a3ba65
hn-hda8	dd	8f244a87b8d38d06603396810a91c43b
hn-hda8	EnCase	8f244a87b8d38d06603396810a91c43b

Now that the preparation of test data is concluded, all image files have been saved in one directory, and have been given file names corresponding to their names listed in the Image Preparation Summary table.

Installation of Mount Image Pro on the Windows XP Professional test system, was conducted at this point. Installation involved running the executable file downloaded

from the vendor web site, and answering prompts concerning the installation point and license agreement.

The next step is the reacquisition of the data using Mount Image Pro as an intermediary for access to each image format.

Criteria for Approval

Mount Image Pro's success in this test will be derived from how closely the reacquired images match the original hashes. Rehashing of the original image files (all six of them) will be done to ensure Mount Image Pro does not alter the content of the image files. For the purposes of this testing, alteration of the image files' MAC times is an acceptable by-product of using the tool. However, absolutely no modifications should be made to the image file content as a result of using Mount Image Pro. To measure the latter test, the re-hashing of the image files after the testing will indicate any modification to the integrity of the image files.

The suitability criteria will be established by whether it is indeed possible to create a forensic image of an image mounted by Mount Image Pro. As previously mentioned, forensic investigators may find it useful to mount an EnCase forensic image, and reacquire the image using either the Linux or Windows port of *dd*. Such an investigator would not necessarily need to purchase the EnCase package, but merely the Mount Image Pro software at a potential savings of \$2200. Following the logic of this example, once the image is reacquired into *dd*, Linux or other third-party forensics tools may be used to perform the investigation.

Data and Results

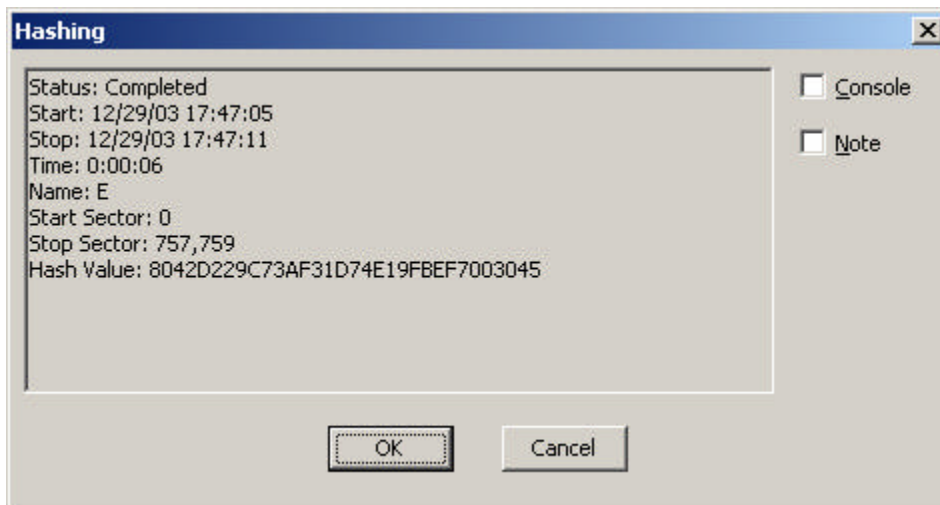
All mounted drives were unmounted, via Mount Image Pro, before the next image file was mounted.

Win2kProCD: dd to EnCase

The extension ".raw" needed to be added to the *dd* image files so that Mount Image Pro would provide the option to mount the file via a context menu. Additionally, Mount Image Pro was unable to mount *dd* format forensic images as logical drives, even when run from the command line. No explanation was given for this apparent limitation in the documentation.

The image was mounted as a physical ("single") drive, assigned drive letter E: by Windows.

Despite repeated attempts to attain an appropriate disk configuration with EnCase, it was only possible to view the image in its raw state, which appeared to contain several hundred bytes of padding at the beginning of the image. The md5 hash of the image using EnCase did not match the original hash. Further analysis was not taken.



It should be noted that EnCase is displaying a different sector size than the original file utilized (total sectors= 757,759 vs. 189,497)

Win2kProCD: EnCase to dd

Mount Image Pro also failed to mount the Win2kProCD EnCase image as a logical drive. Testing continued with the drive mounted as a single or physical drive.

dd for Windows was used to image the mounted EnCase image file.

```
C:\VirtualMachines\Shared>c:\dos\tools\dd bs=2048 count=189500 if=\\.e: of=Win2kProCD-EnC.dd
Forensic Acquisition Utilities, 3, 16, 2, 1029
dd, 3, 16, 2, 1029
Copyright (C) 2002 George M. Garner Jr.

Command Line: c:\dos\tools\dd bs=2048 count=189500 if=\\.e: of=Win2kProCD-EnC.dd
Based on original version developed by Paul Rubin, David MacKenzie, and Stuart Kemp
Microsoft Windows: Version 5.1 (Build 2600.Professional Service Pack 1)

29/12/2003 23:06:10 (UTC)
29/12/2003 18:06:10 (local time)

Current User: DEFAULT\hpierce

unable to display device infoCopying \\.e: to C:\VirtualMachines\Shared\Win2kProCD-EnC.dd...

Output C:\VirtualMachines\Shared\Win2kProCD-EnC.dd 388096000/386695168 bytes (compressed/uncompressed)
189500+0 records in
189500+0 records out
```

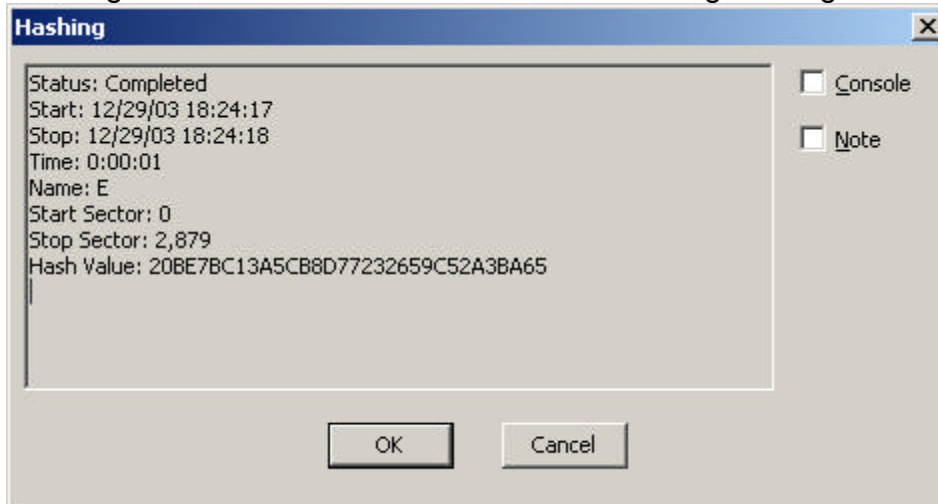
The new *dd* image file, created from the mounting of the EnCase image, results in a mismatched hash.

```
C:\VirtualMachines\Shared>c:\dos\tools\md5sum Win2kProCD-EnC.dd
\ee4f40926ad23adc6dd25e1afb0c0ddc *C:\\VirtualMachines\\Shared\\Win2kProCD-EnC.dd
```

fl-160703-jp1: dd to EnCase

When the *dd* image for the GCFA floppy was mounted as a drive, EnCase opened the logical drive properly in preview mode, showing the contents of the image as expected.

Hashing the new case file returns a hash matching the original *dd* image hash.



This result provides evidence that mounting *dd* images of an ext2 file system using Mount Image Pro, and reacquiring the image will result in a new forensically sound image. While this type of conversion in image file formats may be useful to many, the conversion in the opposite direction is bound to be more useful. EnCase already has the intrinsic ability to open *dd* image files, and reacquire them without the intervention of Mount Image Pro.

fl-160703-jp1: EnCase to dd

Once again Mount Image Pro failed to mount the EnCase file image as a logical drive. This failure is understandable when dealing with images of Linux file systems, as Windows can not natively read ext2 formatted partitions.

Mount Image Pro did mount the fl-160703-jp1 EnCase image file as a single drive.

The Windows *dd* utility displayed several errors in its imaging of the mounted drive, but created an image file nonetheless.


```

C:\VirtualMachines\Shared>c:\dos\tools\dd if=\\.\e: of=f1-160703-jp1-EnC.dd
Forensic Acquisition Utilities, 3, 16, 2, 1029
dd, 3, 16, 2, 1029
Copyright (C) 2002 George M. Garner Jr.

Command Line: c:\dos\tools\dd if=\\.\e: of=f1-160703-jp1-EnC.dd
Based on original version developed by Paul Rubin, David MacKenzie, and Stuart Kemp
Microsoft Windows: Version 5.1 (Build 2600.Professional Service Pack 1)

29/12/2003 23:35:07 (UTC)
29/12/2003 18:35:07 (local time)

Current User: DEFAULT\hpierce

CopyFileEx \\.\e: to C:\VirtualMachines\Shared\f1-160703-jp1-EnC.dd...Failed!
Incorrect function.
unable to display device infoCopying \\.\e: to C:\VirtualMachines\Shared\f1-1607
03-jp1-EnC.dd...
c:\dos\tools\dd.exe:
    \\.\e:: Invalid argument

Output C:\VirtualMachines\Shared\f1-160703-jp1-EnC.dd (0 bytes)
360+0 records in
360+0 records out

```

Despite the complaints from Windows *dd*, the hash of the new EnCase-based *dd* image matches the original hash.

```

C:\VirtualMachines\Shared>c:\dos\tools\md5sum f1-160703-jp1-EnC.dd
\20be7bc13a5cb8d77232659c52a3ba65 *C:\\VirtualMachines\\Shared\\f1-160703-jp1-EnC.dd

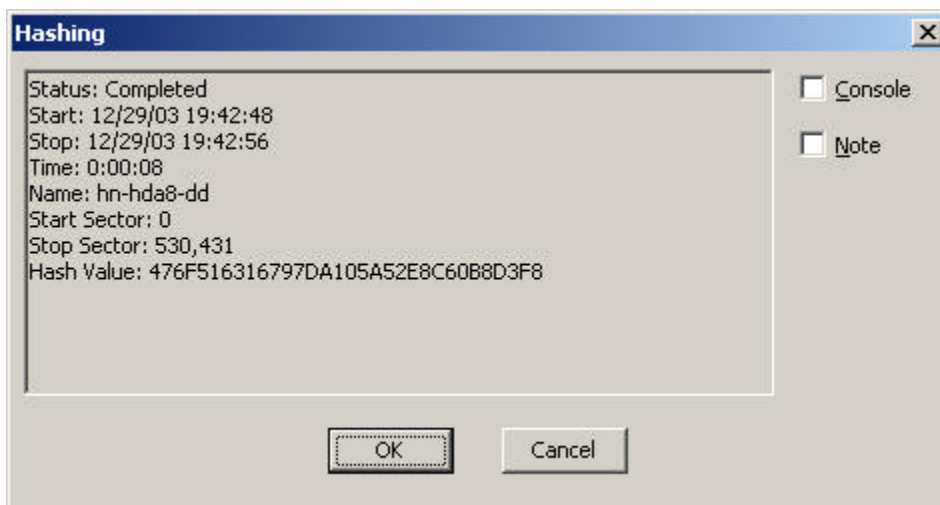
```

This result is good news to those who do not own EnCase, but are faced with processing an EnCase image as part of a case. With Mount Image Pro, the EnCase image file can be mounted as a local drive, and then reacquired using *dd*. And, this can be accomplished while maintaining the forensic integrity of the case.

hn-hda8: dd to EnCase

Next, mounting of the Honeynet Linux image (*dd* format, ext2 filesystem) was done. The mounted drive was added as a device in EnCase, and displayed the contents of the image as expected. The mounted drive was successfully acquired within EnCase.

The EnCase generated md5 hash did not match the original hash for this image. The reacquisition of the mounted drive resulted in an EnCase image file with 530431 sectors, rather than the original image file sector count of 532159. It is possible that this discrepancy is the result of the documented truncating of image files that are not a multiple of 512 bytes. The documentation regarding this limitation does not indicate a work around, but does include a command-line method of determining how the Mount Image Pro sees the image.



```
C:\Program Files\Mount Image Pro>mip view c:\VirtualMachines\Shared\hn-hda8.dd.raw
Mount Image Pro 1.05

Disk Capacity   : 532160 sectors (259 MB)
Number Of Files : 1

  Type      Size      Path
  ----      -
  RAW      532160  c:\VirtualMachines\Shared\hn-hda8.dd.raw

Partitions      :
  #  Start Sector  Length in sectors  Type
  --  -
  0  0              532160 ( 259 MB)  ext2fs
```

As can be seen by the command output, Mount Image Pro does interpret this image as having the correct number of sectors (532160). It is unknown whether the absence of the remaining sectors is a deficiency on the part of EnCase or Mount Image Pro, or perhaps a configuration deficiency on the part of the examiner.

hn-hda8: EnCase to dd

The EnCase image file of the HoneyNet forensic challenge was mounted by Mount Image Pro. As with the previous attempts, it also is an ext2 file system, and would not mount in as a logical drive.

The Windows port of *dd* imaged the mounted drive and produced an image that generated a matching hash.

```
C:\VirtualMachines\Shared>c:\dos\tools\dd if=\\.\e: of=hn-hda8-EnC.dd
Forensic Acquisition Utilities, 3, 16, 2, 1029
dd, 3, 16, 2, 1029
Copyright (C) 2002 George M. Garner Jr.

Command Line: c:\dos\tools\dd if=\\.\e: of=hn-hda8-EnC.dd
Based on original version developed by Paul Rubin, David MacKenzie, and Stuart Kemp
Microsoft Windows: Version 5.1 (Build 2600.Professional Service Pack 1)

30/12/2003 00:46:52 (UTC)
29/12/2003 19:46:52 (local time)

Current User: DEFAULT\hpierce

CopyFileEx \\.\e: to C:\VirtualMachines\Shared\hn-hda8-EnC.dd...Failed!
Incorrect function.
unable to display device infoCopying \\.\e: to C:\VirtualMachines\Shared\hn-hda8-EnC.dd...
c:\dos\tools\dd.exe:
    \\.\e:: Invalid argument

Output C:\VirtualMachines\Shared\hn-hda8-EnC.dd 272465920/268337152 bytes (compressed/uncompressed)
66520+0 records in
66520+0 records out

C:\VirtualMachines\Shared>c:\dos\tools\md5sum hn-hda8-EnC.dd
\8f244a87b8d38d06603396810a91c43b *C:\\VirtualMachines\\Shared\\hn-hda8-EnC.dd
```

Bonus Round Using FAT

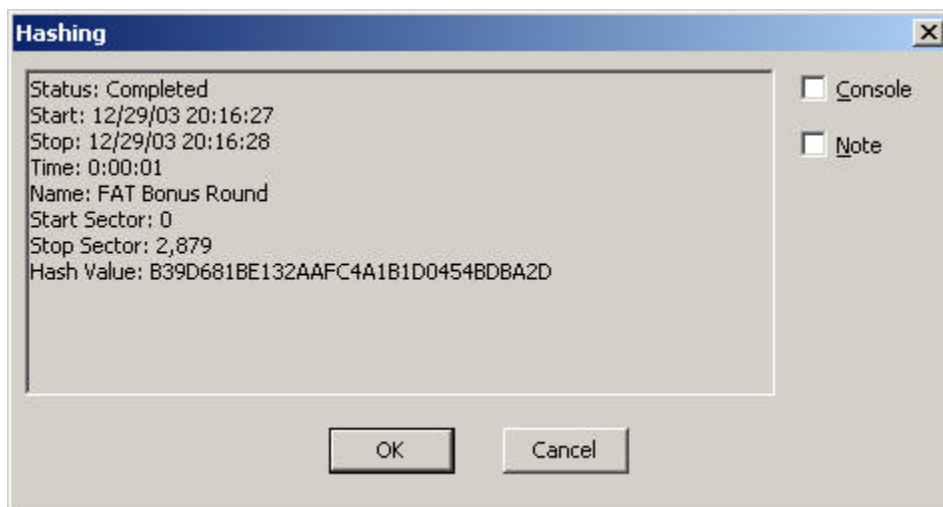
A floppy disk was wiped with EnCase (using hex 00 for the overwriting character) and then quick-formatted with the FAT file system using Windows XP Professional. The binary *Fport*, version 1.33 from Foundstone, was copied to the newly formatted diskette, along with the EICAR antivirus test file.³⁸ Once the files were copied to the floppy disk, the write protect window on the diskette was opened.

The Windows version of *md5sum* was used to generate a baseline hash value for the drive.

```
C:\VirtualMachines\Shared>c:\dos\tools\md5sum \\.\a:
\b39d681be132aafc4a1b1d0454bdba2d *\\.\a:
```

Acquisition of the floppy disk was carried out with EnCase, which returned a hash matching the baseline.

³⁸ <http://www.eicar.org/download/eicar.com>



Mount Image Pro mounted this EnCase image file as both a logical and single drive, due to the fact that it is one of the supported file system types.³⁹

³⁹ Supported file system mount targets are: NTFS, FAT, FAT16, FAT32. See <http://www.mountimage.com/>

Imaging and hashing the resulting *dd* file, created a matching hash.

```
C:\VirtualMachines\Shared>c:\dos\tools\dd if=\\.\e: of=Bonus-Logical.dd
Forensic Acquisition Utilities, 3, 16, 2, 1029
dd, 3, 16, 2, 1029
Copyright (C) 2002 George M. Garner Jr.

Command Line: c:\dos\tools\dd if=\\.\e: of=Bonus-Logical.dd
Based on original version developed by Paul Rubin, David MacKenzie, and Stuart Kemp
Microsoft Windows: Version 5.1 (Build 2600.Professional Service Pack 1)

30/12/2003 01:21:24 (UTC)
29/12/2003 20:21:24 (local time)

Current User: DEFAULT\hpierce

CopyFileEx \\.\e: to C:\VirtualMachines\Shared\Bonus-Logical.dd...Failed!
The parameter is incorrect.
Statistics for logical volume \\.\e:
          1330176 bytes available
          1330176 bytes free
          1457664 bytes total

Volume Name:          \\.\e:
Volume Label:
Drive Type:          fixed
Volume Serial Number: 2280-80DE
Maximum Component Length: 255
Volume Characteristics:

                          File system preserves case
                          File system supports Unicode file names
File System:          FAT
Clustered:           No

Copying \\.\e: to C:\VirtualMachines\Shared\Bonus-Logical.dd...

Output C:\VirtualMachines\Shared\Bonus-Logical.dd (0 bytes)
360+0 records in
360+0 records out

C:\VirtualMachines\Shared>c:\dos\tools\md5sum Bonus-Logical.dd
\b39d681be132aafc4a1b1d0454bdba2d *C:\\VirtualMachines\\Shared\\Bonus-Logical.dd
```

© SANS Institute

If the preceding steps are repeated, mounting the floppy image as a single drive, identical results are attained.

```
C:\VirtualMachines\Shared>c:\dos\tools\dd if=\\.\e: of=Bonus-Single.dd
Forensic Acquisition Utilities, 3, 16, 2, 1029
dd, 3, 16, 2, 1029
Copyright (C) 2002 George M. Garner Jr.

Command Line: c:\dos\tools\dd if=\\.\e: of=Bonus-Single.dd
Based on original version developed by Paul Rubin, David MacKenzie, and Stuart Kemp
Microsoft Windows: Version 5.1 (Build 2600.Professional Service Pack 1)

30/12/2003 01:25:47 (UTC)
29/12/2003 20:25:47 (local time)

Current User: PORTADIG\hpierce

CopyFileEx \\.\e: to C:\VirtualMachines\Shared\Bonus-Single.dd...Failed!
The parameter is incorrect.
Statistics for logical volume \\.\e:
    1330176 bytes available
    1330176 bytes free
    1457664 bytes total

Volume Name:          \\.\e:
Volume Label:
Drive Type:           fixed
Volume Serial Number: 2280-80DE
Maximum Component Length: 255
Volume Characteristics:

                        File system preserves case
                        File system supports Unicode file names
File System:          FAT
Clustered:            No

Copying \\.\e: to C:\VirtualMachines\Shared\Bonus-Single.dd...

Output C:\VirtualMachines\Shared\Bonus-Single.dd (0 bytes)
360+0 records in
360+0 records out

C:\VirtualMachines\Shared>c:\dos\tools\md5sum Bonus-Single.dd
\b39d681be132aafc4a1b1d0454bdba2d *C:\\VirtualMachines\\Shared\\Bonus-Single.dd
```

The conclusion based on these results, is that Mount Image Pro currently offers advantages to those who only deal with the Crocware supported file systems, which unfortunately do not include native Linux or Mac formats.

Mount Image Pro Acquisition Hashes			
Data Source	Image Type	Mounted as Logical Drive	Mounted as Physical Drive
Win2kProCD	dd	Failed to mount	8042d229c73af31d74e19fbef7003045
Win2kProCD	EnCase	Failed to mount	ee4f40926ad23adc6dd25e1afb0c0ddc
fl-160703-jp1	dd	Failed to mount	20be7bc13a5cb8d77232659c52a3ba65
fl-160703-jp1	EnCase	Failed to mount	20be7bc13a5cb8d77232659c52a3ba65
hn-hda8	dd	Failed to mount	476f516316797da105a52e8c60b8d3f8
hn-hda8	EnCase	Failed to mount	8f244a87b8d38d06603396810a91c43b

Image File Format Conversion Using Mount Image Pro		
Image File System	Conversion	Consistent Hashes?

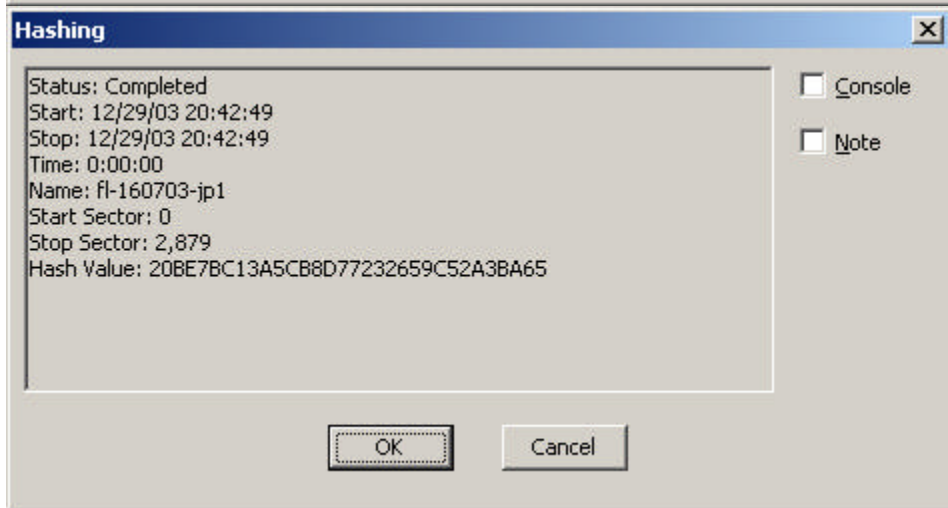
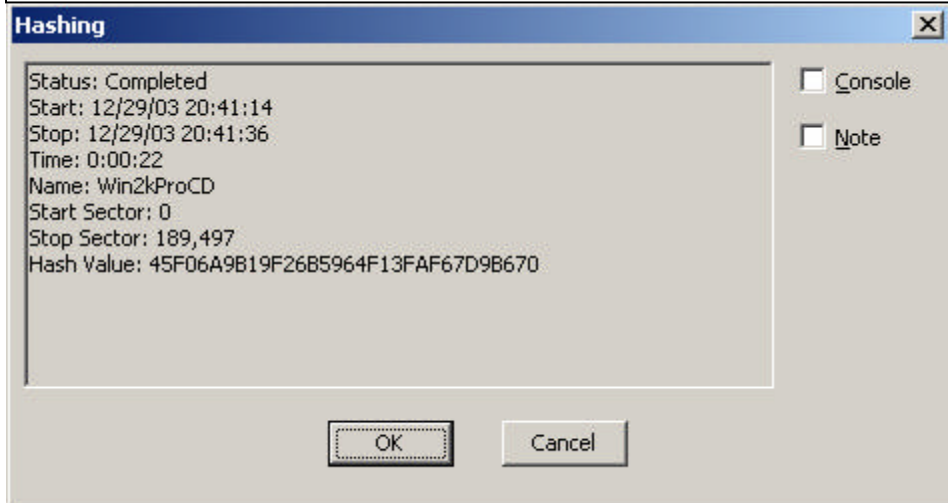
ISO9660	dd to EnCase	No
ISO9660	EnCase to dd	No
Ext2	dd to EnCase	Yes
Ext2	EnCase to dd	Yes
Ext2	dd to EnCase	No
Ext2	EnCase to dd	Yes
FAT	EnCase to dd	Yes

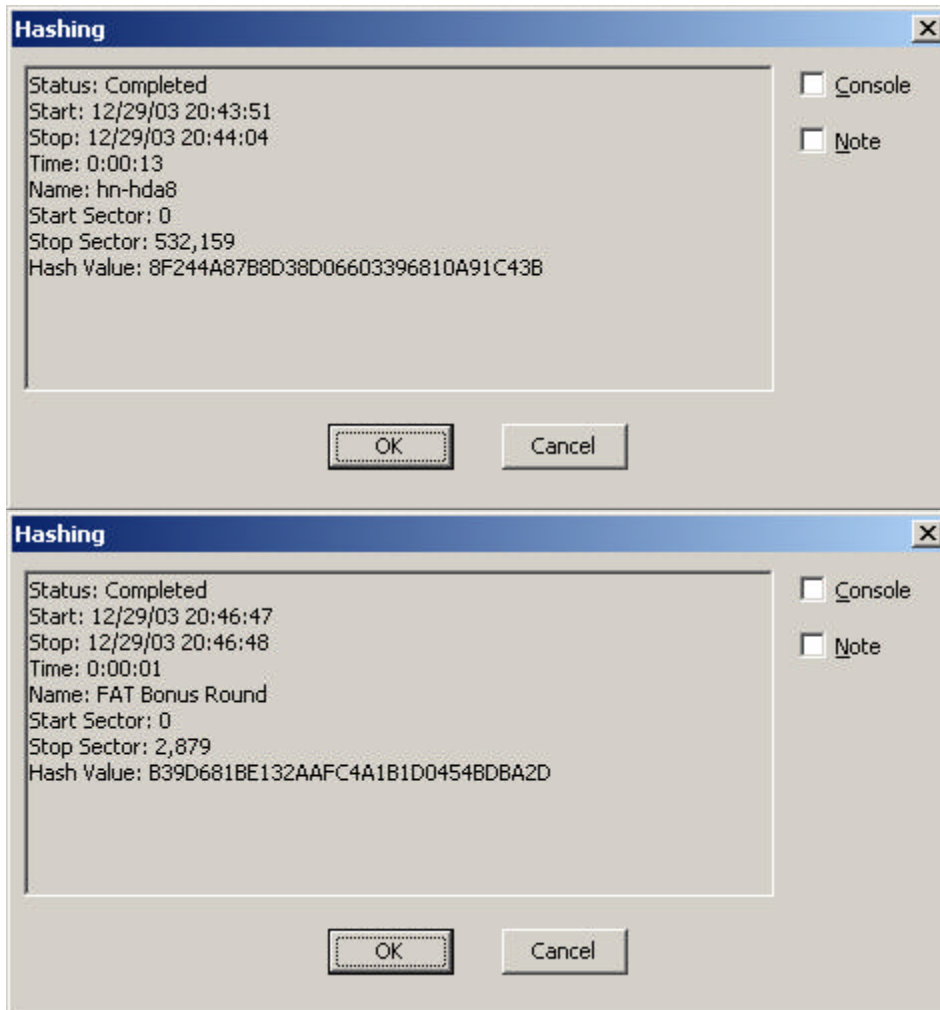
The following is screen output from the image file re-hashing:

```
C:\VirtualMachines\Shared\original>c:\dos\tools\md5sum fl-160703-jp1.dd.raw
\20be7bc13a5cb8d77232659c52a3ba65 *C:\\VirtualMachines\\Shared\\original\\fl-160
703-jp1.dd.raw

C:\VirtualMachines\Shared\original>c:\dos\tools\md5sum hn-hda8.dd.raw
\8f244a87b8d38d06603396810a91c43b *C:\\VirtualMachines\\Shared\\original\\hn-hda
8.dd.raw

C:\VirtualMachines\Shared\original>c:\dos\tools\md5sum Win2kProCD.dd.raw
\45f06a9b19f26b5964f13faf67d9b670 *C:\\VirtualMachines\\Shared\\original\\Win2kP
roCD.dd.raw
```





Re-hashing of the Original Image Files
 (matched hashes validate integrity preservation during Mount Image Pro use)

Data Source	Image Type	Original Hash	Post-Testing Hash
Win2kProCD	dd	45f06a9b19f26b5964f13faf67d9b670	45f06a9b19f26b5964f13faf67d9b670
Win2kProCD	EnCase	45f06a9b19f26b5964f13faf67d9b670	45f06a9b19f26b5964f13faf67d9b670
fl-160703-jp1	dd	20be7bc13a5cb8d77232659c52a3ba65	20be7bc13a5cb8d77232659c52a3ba65
fl-160703-jp1	EnCase	20be7bc13a5cb8d77232659c52a3ba65	20be7bc13a5cb8d77232659c52a3ba65
hn-hda8	dd	8f244a87b8d38d06603396810a91c43b	8f244a87b8d38d06603396810a91c43b
hn-hda8	EnCase	8f244a87b8d38d06603396810a91c43b	8f244a87b8d38d06603396810a91c43b
FAT floppy	EnCase	b39d681be132aafc4a1b1d0454bdba2d	b39d681be132aafc4a1b1d0454bdba2d

The post-testing hash comparison demonstrates that Mount Image Pro does deliver on its claim that it mounts image files as read-only. The documentation does contain references to command line switches that allow for mounting images containing supported file systems, as read-write. This is possible only for images other than EnCase. The documentation states that EnCase image files are always mounted as read-only.

Analysis

Of the test images, three of seven failed to produce a forensically identical image when accessed through Mount Image Pro. Two of those three were the CDROM images. The third mismatched image file was the HoneyNet Project ext2 partition image, which failed a consistent hash when going from *dd* to EnCase, but did result in an identical image hash when acquiring from EnCase to *dd*.

The four cases where an image reacquired through Mount Image Pro did produce a consistent hash, included both ext2 floppy images, one of the HoneyNet ext2 partition images, and the FAT floppy image done as an add-on.

Anecdotal evidence was noted regarding performance of acquisitions through Mount Image Pro. There was no noticeable decline in performance in either EnCase or *dd* when acquiring through Mount Image Pro, as compared to acquiring directly from media.

Mount Image Pro's failure to forensically represent ext2 image files may be a drawback in appealing to investigators who want to convert EnCase image files to *dd*. The newest release EnCase supports many of the Linux file system formats, so it is possible for an acquisition and investigation to be conducted on a Unix/Linux/Mac file system completely from a Windows platform using EnCase. These investigations would result in EnCase image files, that if required to be analyzed in a Linux environment would either have to be restored to a drive or reacquired from the original media using a Linux tool.

Additionally, given the young age of this product, feature additions such as support of various non-Windows file systems (notably ISO9660 and ext2/ext3), allowing read-only access to the file structure and data through the Windows Explorer interface, would be seemingly natural progressions.

Presentation

Should the Mount Image Pro tool be used in a proceeding where the results of an investigation could be challenged, extra effort would need to be expended ensuring that the involvement of this tool was well documented not to affect the evidential integrity.

If Mount Image Pro was used as a tool to convert forensic image file formats, a process similar to this test should be documented showing an identical comparison of the original hash and the converted image file hash.

Until additional documentation becomes available, and this product has gained additional industry awareness (and perhaps the awareness of the Courts), all other uses of this tool where the forensic image is concerned, should be well documented as to the file integrity before and after Mount Image Pro access.

Presenting this information in a condensed table format, as above, may be the best way to represent a comparison of before-and-after hash values.

Conclusion

Mount Image Pro achieved a qualified success in this validation.

The significant utility value of this tool, combined with its early stages of development, make it possible to largely overlook the failings. The three failures documented in the testing can be discounted since the documentation states there is no support of ISO9660 or ext2 file systems. However, by the same token, the documentation lacks any information on whether a lack of support for these file systems may lead to reacquisition problems. Given that the image files that resulted in a hash discrepancy were all mounted as “single” or “physical” drives, support for any particular file system should not be a consideration if imaging is the only goal.

For the computer forensic investigator, Mount Image Pro would easily be worth its price in labor hours if it saved the reacquisition of original media. Consider a large storage source with no fast method of acquisition. If the original acquisition in EnCase took 8 hours (for example), then a reacquisition to produce a *dd* image will likely take a similar amount of time. Using Mount Image Pro to convert the EnCase image to *dd* would only depend on the hardware endowment of acquisition machine as a limitation of performance. How many hours of your time are worth \$299?

As Crocware develops a larger user-base for this tool, more feedback regarding bugs and incompatibilities will emerge, leading to a more reliable product. Use of this product in situations with similar objectives to the testing criteria, will result in documentation of configuration settings for Windows, *dd*, and EnCase that will foster interoperability.

Once this support is in place, Mount Image Pro will be an important toolkit component for an investigator handling cases acquired in both EnCase and *dd*. This tool may be especially useful for those investigators that receive case image files from segments of the computer forensics professional population, that largely use EnCase (such as law enforcement).

Several other features of Mount Image Pro are worthy of mention, but were not covered in the testing. First, if the mounted image file is one of the supported file systems, then the tool allows direct access to the image file system for third party forensic tools (antivirus, deleted file recovery, hex editor, file export, etc.). If the file system within the image is not one of the supported types, it may be possible to use a third-party driver to access the file system as if it were native to Windows. The Mount Image Pro web site specifically mentions this capability. Second, once the image file is mounted as a local drive, that drive can be shared on a network for various network enabled tools.

Part 3: Legal Issues of Incident Handling

A.

John Price may have violated Federal and/or State laws pertaining to Intellectual Property Rights Protection due to the type of material he was distributing.

John Price may have violated Federal Copyright law 17 USC 506 (which was amended by the No Electronic Theft Act ("NET Act") in 1997 to proscribe copyright violations even where there is no financial gain, such as may occur with software and entertainment reproduction -- i.e. piracy). This is based on the assumption that the copyrighted material Mr. Price was distributing was either audio or video computer files. The Act prohibits willful copyright infringement for purposes of either commercial advantage or private financial gain or "by the reproduction or distribution, including by electronic means, during any 180-day period, of 1 or more copies or phono records of 1 or more copyrighted works, which have a total retail value of more than \$1,000." (17 USC 506).⁴⁰ The volume of material he was distributing would determine whether the \$1000 requirement was met. Regardless however, he still may face charges under the Net Act even if he received no financial benefit. The quote captured in Mr. Price's Word document suggests that he had received "orders", from which one can logically conclude there was a financial interest in the filling of these orders.

Additionally, under this section, he may have broken laws which proscribe against fraudulent removal or placement of copyright notice. Section 18 USC 2319 (Criminal Infringement of a Copyright) outlines punishment of criminal infringement of copyright.⁴¹ By definition, Mr. Price infringed on copyright by distributing copyrighted material.

Mr. Price may also be deemed to have broken 18 USC 2318, which covers "trafficking in counterfeit labels for phono records, copies of computer programs or computer program documentation or packaging, and copies of motion pictures or other audio visual works, and trafficking in counterfeit computer program documentation or packaging."⁴² Under this federal law, he could be punished if he knowingly traffics in a counterfeit label affixed to a phono record, a copy of a computer program or documentation or packaging for a computer program, or a copy of a motion picture or other audiovisual work. The efficacy of this statute would depend on how he presented the material to his "customers". For example, should he present the material as originating from a legitimate source such as BMG Records, a prosecutor may cite the section of the code that prohibits trafficking of a counterfeit label affixed to an audiovisual work. In this case the label would be virtual or digital in nature.

Another violation comes as a result of the Digital Millennium Copyright Act ("DMCA")(17 USC 1201 et seq.). Mr. Price may have violated the DMCA (17 USC 1201)⁴³ if he was circumventing copyright protection systems (i.e. technological measures), or offering to

⁴⁰ <http://www4.law.cornell.edu/uscode/17/506.html>

⁴¹ <http://www.usdoj.gov/criminal/cybercrime/18usc2319.htm>

⁴² <http://www.usdoj.gov/criminal/cybercrime/18usc2318.htm>

⁴³ <http://www.usdoj.gov/criminal/cybercrime/17usc1201.htm>

the public, providing or otherwise trafficking in any technology designed primarily to circumvent protection afforded to a copyright owner. In addition, he may be found to have violated 17 USC 1202 of the DMCA if he knowingly falsified copyright management information, such as the name of the author of the work.⁴⁴ Under the DMCA (17 USC 1204 criminal offenses and penalties), he could be subject to criminal penalties if he willfully and for purposes of commercial advantage or private financial gain violated the Act. Mr. Price's actions of hiding the copyrighted material could be one form of circumvention. Another possibility is if he was "ripping" audio or video content from protected original media. Several legitimate audiovisual distributors are protecting content on consumer media. Ripping the data from the protected media would constitute circumvention of these protections.

John Price could have broken federal 18 USC 2319A if it could be shown that he was trafficking in sound recordings of live musical events (i.e. bootlegging).⁴⁵ The evidence recovered was not clear as to whether the material he was distributing was either audiovisual, or live audiovisual.

The evidence showed that Mr. Price was accessing the Linux system as "root" and accessing the Windows system as "Administrator". Since these are the two highest levels of system access for Linux and Windows, respectively, it could also be argued that Mr. Price exceeded his authorized access level. This action violates the Computer Fraud and Abuse Act (18 USC 1030) of 1986, which prohibits the use of computers to access data without or in excess of authorization.⁴⁶

In Vermont, protection of copyright is provided for by the Federal Copyright Act of 1976. Vermont currently has no copyright protection statutes.

As a stretch, it could be argued that John Price violated the Uniform Computer Information Transaction Act (UCITA). However, the state of Vermont implicitly rejected the UCITA when it adopted (in 2003) the UETA--Uniform Electronic Transaction Act. Vermont was one of only a few states to do so. UETA is actually an anti-UCITA measure.⁴⁷ The UETA may be implicated here since it pertains to software licensing and electronic contract issues. If Mr. Price sourced his material from media that either posted or inherently included (CD shrink wrap license) license or contractual obligations, not only could he be in violation of UETA, but also open to a variety of civil liabilities.

Finally, John Price may have breached the terms of his contract with his employer, particularly if the company has a use policy that governs acceptable conduct.

B.

The main consideration in my mind if evidence of a crime was discovered on my systems, is limiting the liability to me and the company. Acting in good faith to report

⁴⁴ <http://www.usdoj.gov/criminal/cybercrime/17usc1202.htm>

⁴⁵ <http://www.usdoj.gov/criminal/cybercrime/18usc2319A.htm>

⁴⁶ http://www.usdoj.gov/criminal/cybercrime/1030_new.html

⁴⁷ http://www.thinkvermont.com/publications/pdf/legguide_09.pdf

criminal activity on company systems, may reduce the likelihood that the company itself will become the target of the investigation. As mentioned above 17 USC 1201 makes the statement that “No person shall ... offer to the public, provide, or otherwise traffic in any technology, ... that ... has only limited commercially significant purpose or use other than to circumvent a technological measure that effectively controls access to a work protected under this title ...”⁴⁸

Possession of the *prog* data hiding utility would be classified as a technology that serves to circumvent a control for protecting copyrighted work. For example, *prog* may evade network content filtering devices and network storage scanning software that looks specifically for unauthorized audiovisual files.

C.

If the Corporate Attorney refrains from pursuing the matter, the evidence must be saved while considering both computer forensic best practices and legal rules of evidence.

Pursuant to the Rules of Evidence (both Federal and State versions), evidence must be relevant, reliable, and have probative value to be admissible in proceedings. (Rules of Evidence 401, 404).⁴⁹ Accordingly, it is necessary that the evidence be preserved in a manner in which it can be later authenticated in court (that is, shown to be what it purports to be), including proven to be reliable, unaltered (i.e. no tampering), undamaged. Essentially, you must be able to document chain of custody with respect to the evidence. That from the time it was first seized to when it was introduced in legal proceedings, there has been full documentation of what has been done to the evidence, how it has been stored, who had access to it.

The technical implications of these requirements are significant, and touch on the technical methods of evidence collection, investigation, and storage. Generally, the evidence must be collected without the collection act changing the evidence. A chain of custody for the evidence must begin immediately. The investigation stage must involve authentication of the evidence work-copy, compared to the original evidence, and all actions should be documented so conclusions can be scrutinized and methodology repeated if necessary (by opposing Counsel). Preservation requires that storage media be used that will maintain the integrity of the data for longer than the required storage duration. The location of storage is important as well, as the media must be protected from human and environmental threats.

D.

Ultimately, it would be necessary to notify law enforcement, as possession of child pornography by ANYONE not authorized by the court, constitutes punishable offenses on both Federal and State law levels. Pursuant to Title 18 of the United States Code which governs child pornography (18 USC 2256)⁵⁰, producing, possessing or distributing child pornography is illegal. Some porn web sites are citing this statute in an

⁴⁸ <http://www.usdoj.gov/criminal/cybercrime/17usc1201.htm>

⁴⁹ <http://www.law.cornell.edu/rules/fre/overview.html>

⁵⁰ Case Law: <http://www.usdoj.gov/osg/briefs/2001/0responses/2000-1936.resp.html>

attempt to appear more conforming to the law.⁵¹ In addition, 18 USC 2252 prohibits the production, transportation or knowing receipt or distribution of any visual depiction "of a minor engaging in sexually explicit conduct."⁵² Accordingly, if the investigation disclosed that Mr. Price was distributing child pornography via the corporate network or property, then it could be argued that the corporation is liable and accountable for its possession (vicarious liability). As such, it could be argued that the company is actually in possession of child pornography.

There are no exceptions for the corporation which would allow it to continue possession of the child pornography. In *New York v. Ferber*, 458 US 747 (1982), the Supreme Court held that "content that depicts children engaged in sexual conduct is a category of material outside the protection of the First Amendment", accordingly, a discovery of child pornography on a corporate system does not enjoy any First Amendment protections.⁵³

In order to ensure that the company is not prosecuted for possessing or sharing a role with Mr. Price's actions in distributing the child pornography, it is necessary that the company promptly and in good faith contact law enforcement. Pursuant to 18 USC 2252, it is an affirmative defense to a charge of possession to "promptly and in good faith, and without retaining or allowing any person, other than a law enforcement agency to access any visual depiction or copy thereof"⁵⁴ contact law enforcement to report the incident.

On the State level, Vermont law Title 13, Chapter 64 section 2821 prohibits the promotion of child pornography, including mailing, delivery, distribution, dissemination, circulation, presentation and exhibition, thus possession of child pornography on the company system could implicate this statute and hence criminal liability.⁵⁵

⁵¹ <http://www.badtushy.com/2257.html>

⁵² <http://www4.law.cornell.edu/uscode/18/2252.html>

⁵³ <http://www.adultweblaw.com/laws/childporn.htm>

⁵⁴ <http://www4.law.cornell.edu/uscode/18/2252.html>

⁵⁵ <http://www.vtspecialcrimes.org/VermontStatutes.htm>

Appendix A. Complete strings output for prog

PTRh	hbK	AFJy	CX9C
QVhx	huK	[^_]	<[^_]
h@=	PhHK	[^_]	[^_]
hK=	hHK	[^_]	gfff
h@=	h L	[^_]	[^_]
0hT=	h M	C t	t/Qj
h@=	h:M	XZSh	[^_]
0he=	hKM	[^_]	WVSQ
8-tx	h`M	[^_]	0<
h@=	RPSQ	F\xX	Z[^_]
h@=	RPSQ	[^_]	C<
h!>	[^_]	P0+H	0<
h=>	h@N	[^_]	[^_]
hP>	I, RPSQ	L	Rj@WS
0hm>	h_N	C +C	<
h@=	p8hxN	RPWV	0<
h\$?	h@N	2PVS	tHRV
h-?	h`M	[^_]	[^_]
h7?	h@O	[^_]	[^_]
h7?	heO	[^_]	[^_]
hH?	H^_]	[^_]	[^_]
hH?	X^_]	@t:	Ph b
PhI?	[^_]	[^_]	[^_]
hI?	[^_]	Ph"@	,;u
hU?	tY9u	[^_]	t(;u
h[?	}.;	[^_]	t`Qh`b
h`?	L[^_]	[^_]	tLRh`b
hg?	<	w&;=	[^_]
hn?	ug;]	[^_]	[^_]
hs?	s=;]	[^_]	Qh`b
hy?	PSH	[^_]	Rh`b
Ph @	[^_]	PWVS	[^_]
hg@	XZSV	[^_]	[^_]
0h A	VPVS	tB;u	[^_]
0h@A	[^_]	\$v?V	t
0h B	[^_]	[^_]	B<:u
0h B	[^_]	[^_]	AF<:tel
0h&C	<bT!<b	[^_]	t
0h5C	RPSW	[^_]	B<:u
0hDC	/FBH~	[^_]	VQRP
0hXC	[^_]	@t-	Vh`b
0h_C	;C_tU	[^_]	SQRP
0hbC	t\$QVPS	[^_]	[^_]
hhC	PVRS	AELD	[^_]
0hnC	;S t<	;AELD	[^_]
hzC	[^_]	[^_]	SRVW
h,D	G +G	[^_]	[^_]
0h7D	G +G	@t5=	[^_]
0h`D	t';]	[^_]	[^_]
0h_C	V +V	[^_]	@F;E
0hbC	FDHP	[^_]	[^_]
hzC	[^_]	[^_]	@F;E
h,D	[^_]	[^_]	@G;E
0h-E	t PS	[^_]	[^_]
h@I	VPRQ	RSVP	@G;E
hDI	[^_]	[^_]	[^_]
hXI	VPRQ	[^_]	FB;u
h)J	~"PSV	SPWQ	FB;u
hCJ	CDHP	[^_]	BG;U
h`J	[^_]	[^_]	tS;}
h K	F0+V	[^_]	AC;M
PhAK	[^_]	Gu~1	[^_]
PhHK	V\$PQ)	GuL1	AC;M
hLK	~VQSV	[^_]	@F;E
hbK	[^_]	[^_]	[^_]
huK	FAJy	\[^_]	@F;E
hbK	~>PS	CX9C	AC;M
huK	[^_]	\[^_]	[^_]
			AC;M

@F;E		[^_]	[^_]	[^_]	[^_]
[^_]		t SVj	Gu[1	tLPj	
@F;E		QVj	Gu#1	[^_]	
[^_]		t*QVj	t0@Nt	tJVPj	
[^_]		RVj	t(@Nt	[^_]	
[^_]		t);E	[^_]	[^_]	
[^_]		gfff	[^_]	Qj	h_
[^_]		@PSR	[^_]	_Xhs	
[^_]		RWVS	[^_]	Qj	hh
?/tt		PWVS	[^_]	Wj	hr
[^_]		v A)	RSj	ZYhs	
[^_]		[^_]	QSj0	[^_]	
[^_]		QWVS	0t'QSj	0j\$P	
[^_]		[^_]	PSj0	[^_]	
WQRV		t PVj	RSj	t8<:u4	
B</t		WVj	QSj0	<\$t0	
[^_]		t&QVj	RSj	[^_]	
[^_]		RVj	QSj0	t8<:u4	
[^_]		WVj0	RSj	RSVP	
Xhs		SVj0	QSj0	[^]	
[^_]		RWVS	QSj	t0QV	
u @P		QVh	PSj0	[^_]	
u\$@P		t";5	RSj	[^_]	
u,@P		VSh	QSj0	F<:t	
u4@P		C,+C\$)	0tsQSj	VQSP	
u8@P		[^_]	PSj0	[^_]	
[^_]		[^_]	RSj	[^_]	
AC;]		QVWP	QSj0	@bQs	
tQOt		[^]	RSj	[^_]	
[^_]		[^_]	QSj0	[^_]	
t=Ky		QSh	RSj	v)WRj	
t=Ky		[^_]	QSj0	[^_]	
t=Ky		tQRS	<GtZ<gt.	[^_]	
t=Ky		[^_]	RSj	WVSQ	
t=Ky		~pRSV	QSj0	0<	v
t=Ky		[^_]	0t'RSj	Z[^_]	
t=Ky		[^_]	QSj0	C<	w+
0<	w	x~QS	gfff	0<	v
0<	v	[^_]	0t+PSj	[^_]	
9=u>A		[^_]	RSj0	C09U	
<	w1j	[^_]	0tsRVj	9T00w	
[^_]		[^_]	QVj0	tcB9	
[^_]		~ERS	0t(RVj	[^_]	
[^_]		[^_]	QVj0	w%;]	
[^_]		tANt:	[^_]	[^_]	
[^_]		[^_]	[^_]	[^_]	
[^_]		[^_]	Wj@j	[^_]	
[^_]		~!Q+B	VQSP	[^_]	
t,RVWP		[^_]	</t\$	t	;
[^_]		[^_]	;:t7G	w!;	
tY9u		@t5	;:toG	[^_]	
}.;]		G +G	t	t	;
L[^_]		G +G	\$[^_]	w%;u	
<	v61	t-;	gfff	[^_]	
ug;]		vwWSQ	tCVS	[^_]	
s=;]		[^_]	C\$+E	t'WS	
<	w[[^_]	[^_]	QPSV	
[^_]		[^_]	[^_]	[^_]	
[^_]		CTPV	WQRV	[^_]	
tkWQ		[^_]	[^_]	st	C
WQj0		CTPV	tnF;5	[^_]	
WQj0		[^_]	[^_]	<0/t	
F(Pj		C(PV	[^_]	t\$PS	
;\$t,		[^_]	\[^_]	tj/P	
9\$t.		PSVj	CX9C	VQSP	
[^_]		[^_]	[^_]	@j/P	
[^_]		[^_]	WVS1	[^_]	
SWh		t0@t	[^_]	VRSP	
[^_]		GuP1	[^_]	WRSP	
[^_]		[^_]	[^_]	4\$tK	
QWVS		t(Nu	[^_]	[^_]	

© SANS Institute 2004, Author retains full rights.

P ;U	[^_]	0<	v	<table
[^_]	/j	t	N;	bgcolor=%s><tr><td></td
[^_]	hs	[^_]		></tr></table>
Ph4n	@t0R	t%Pj		Brazil
[^_]	XZh0	0<	v	.TH %s "%d" "%s" "%s"
Wh4n	[^_]	0<	w	"%s"
Sj:P	[^_]	<	v\$.SH NAME
[^_]	[^_]	VUUU		%s \- %s
tQ9u	PSRW	t0Wj		.SH SYNOPSIS
=S0P	QPRW	0<	v	.B %s
=S0P	[^_]	0<	v	[\fIOPTION\fR]...
[^_]	R Iu	t	N;	.SH DESCRIPTION
SQRP	WVUS	[^_]		\fB\-\-%s\fR %s
/GBH~	[^_]	(^_)		\fB\-\-%s\fR \fIARG\fR
/GBH~	[^_]	u^9u		%s
[^_]	<*tm<'ti<Ite	8^_]		\fB\-\-%s\fR \fIINT\fR
[^_]	<*t><*	mft_getopt		%s
/SYS	*<'t	no index		\fB\-\-%s\fR
[^_]	PVh	invalid index %d		\fIFILENAME\fR %s
[^_]	[^_]	argv[%d] is NULL		\fB\-\-%s\fR
M9u	QQQQQQQQQQQQQQQQQQQQQQQQ	argv[%d] (%s) is not an		\fIVALUE\fR %s
<[^_]	QQQQQQQQQQQQQQQQQQQQQQQQ	option		\fIVALUE\fR can be one
[^_]	QQQQQQQQQQQQQj	examining a filename or		of:
w_ ;M	t4Qj	url!		\fB%s\fR
[^_]	t>Qj	%s is a well-formed		\fB%s\fR
wA;U	x Rj	argument		\fBSHORTAND
Jt.P	[^_]	checking against %s		INVOKATION:\fR
X[^_]	RPh`	flag-		Any of the valid values
Bt (P	VQSW	flagized option		for \fB--%s\fR can be
X[^_]	RPh`	invokation		supplied directly as
Jt}l	QSVW	examining an enum!		options. For instance,
[^_]	@bQs	matched against an enum		\fB--%s\fR can be used
WVUS	[^_]	val		in place of \fB--
[^_]	[^_]	examining a venum!		%s=%s\fR.
JtTG	[^_]	matched against an		\fB%s\fR %s
[^_]	[^_]	venum val		--%s %s
JtPG	[^_]	arg matches against %s		.SH REPORTING BUGS
[^_]	tmPh	process_match		Report bugs to %s.
SWV	tiPh	true		Usage: %s [OPTION]...
RSWV	[^_]	matches against %s		[\< %s-filename>]
RSWV	[^_]	invalid value for enum		--%s %s
Jte1	}+;M	mft_log_init		--%s <arg> %s
[^_]	[^_]	mbd-server		--%s <int> %s
Jt1l	t(;U	MFT_LOG_THRESH		--%s <filename> %s
Jtd1	<	none		--%s <
[^_]	uc;u	fatal		%s
Jtd1	s=;u	error		> %s
[^_]	}+;M	info		--%s VALUE
R Iu	[^_]	branch		where VALUE is one
WVUS	t(;U	progress		of:
[^_]	<	entryexit		%s %s
L[^_]	uc;u	mft_log_shutdown		<tt>%s</tt> invocation
\[^_]	s=;u	unspecified		<tt>%s
[^_]	[^_]	enter		[\<OPTIONS>]
[^_]	t%Pj	exit		[\< %s-filename>]
tC;E	0<	%s: %s		</tt>
[^_]	0<	violet		Where <bf>OPTIONS</bf>
[^_]	<	blue		may include any of:
[^_]	VUUU	green		<descrip>
[^_]	~3SVRR	yellow		<tag>--%s</tag>
[^_]	t0Wj	orange		%s
t&PS	0<	white		<tag>--%s
RSVP	t	%s: %s		<arg></tag> %s
[^_]	[^_]	<table		<tag>--%s
[^_]	t%Pj	bgcolor=%s><tr><td>%s:		<int></tag> %s
[^_]	0<	%s</td></tr></table><br		<tag>--%s
H%T=	0<	>		<filename></tag>
<\t~<\tn	<	<table		%s
[^_]	VUUU	bgcolor=%s><tr><td>%s</		<tag>--%s <
[^_]	t0Wj	td></tr></table> 		></tag> %s
< tZ< tB<\t2	0<	v		<tag>--%s VALUE</tag>

<tag>%s</tag> %s	no filename. try '--	NULL filename supplied	/dev/xdb17
</descrip>	help' for help.	Unable to stat file: %s	/dev/xdb16
<tag>--%s</tag> %s	target filename: %s	%s is not a regular	/dev/xdb15
%s:%s %s	Unable to stat file: %s	file.	/dev/xdb14
operate on ...	%s is not a regular	unable to determine raw	/dev/xdb13
target	file.	device of %s	/dev/xdb12
entryexit	%s has multiple links.	unable to stat raw	/dev/xdb11
progress	Unable to open file: %s	device %s	/dev/xdb10
branch	Unable to determine	device mismatch 0x%x !=	/dev/xdb1
info	blocksize	0x%x	/dev/xdb
error	target file block size:	unable to open raw	/dev/xda9
fatal	%d	device %s	/dev/xda8
none	unable to raw open %s	raw fd is %d	/dev/xda7
logging threshold ...	Unable to determine	bmap_raw_close	/dev/xda63
log-thresh	count	/...7image	/dev/xda62
be verbose	Unable to allocate	bogowipe	/dev/xda61
verbose	buffer	write error	/dev/xda60
name	%s has holes in excess	/dev/xdb9	/dev/xda6
useless bogus option	of %ld bytes...	/dev/xdb8	/dev/xda59
label	error mapping block %d	/dev/xdb7	/dev/xda58
write output to ...	(%s)	/dev/xdb63	/dev/xda57
outfile	nul block while mapping	/dev/xdb62	/dev/xda56
test for fragmentation	block %d.	/dev/xdb61	/dev/xda55
(returns 0 if file is	seek failure	/dev/xdb60	/dev/xda54
fragmented)	read error	/dev/xdb6	/dev/xda53
checkfrag	write error	/dev/xdb59	/dev/xda52
display fragmentation	%s fragmented between	/dev/xdb58	/dev/xda51
information for the	%d and %d	/dev/xdb57	/dev/xda50
file	%d %s	/dev/xdb56	/dev/xda5
frag	getting from block %d	/dev/xdb55	/dev/xda49
wipe the file from the	file size was: %ld	/dev/xdb54	/dev/xda48
raw device	slack size: %d	/dev/xdb53	/dev/xda47
print number of bytes	block size: %d	/dev/xdb52	/dev/xda46
available	seek error	/dev/xdb51	/dev/xda45
test (returns 0 if	# File: %s Location:	/dev/xdb50	/dev/xda44
exist)	%ld size: %d	/dev/xdb49	/dev/xda43
wipe	stuffing block %d	/dev/xdb48	/dev/xda42
place data	%s has slack	/dev/xdb47	/dev/xda41
display data	%s does not have slack	/dev/xdb46	/dev/xda40
extract a copy from the	%s has fragmentation	/dev/xdb45	/dev/xda4
raw device	%s does not have	/dev/xdb44	/dev/xda39
list sector numbers	fragmentation	/dev/xdb43	/dev/xda38
operation to perform on	bmap_get_slack_block	/dev/xdb42	/dev/xda37
files	NULL value for	/dev/xdb41	/dev/xda36
mode	slack_block	/dev/xdb40	/dev/xda35
generate SGML	Unable to stat fd	/dev/xdb5	/dev/xda34
invocation info	Unable to determine	/dev/xdb4	/dev/xda33
sgml	blocksize	/dev/xdb39	/dev/xda32
generate man page and	error getting block	/dev/xdb38	/dev/xda31
exit	count	/dev/xdb37	/dev/xda30
display options and	fd has no blocks	/dev/xdb36	/dev/xda3
exit	mapping block %lu	/dev/xdb35	/dev/xda29
help	error mapping block %d.	/dev/xdb34	/dev/xda28
display version and	ioctl failed with %s	/dev/xdb33	/dev/xda27
exit	error mapping block %d.	/dev/xdb32	/dev/xda26
version	block returned 0	/dev/xdb31	/dev/xda25
autogenerate document	bmap_get_block_count	/dev/xdb30	/dev/xda24
...	unable to stat fd	/dev/xdb3	/dev/xda23
1.0.20 (07/15/03)	unable to determine	/dev/xdb29	/dev/xda22
newt	filesystem blocksize	/dev/xdb28	/dev/xda21
use block-list	filesystem reports 0	/dev/xdb27	/dev/xda20
knowledge to perform	blocksize	/dev/xdb26	/dev/xda2
special operations on	computed block count:	/dev/xdb25	/dev/xda19
files	%d	/dev/xdb24	/dev/xda18
prog	stat reports %d blocks:	/dev/xdb23	/dev/xda17
main	%d	/dev/xdb22	/dev/xda16
off t too small!	bmap_get_block_size	/dev/xdb21	/dev/xda15
07/15/03	bmap_map_block	/dev/xdb20	/dev/xda14
invalid option: %s	nul block while mapping	/dev/xdb2	/dev/xda13
try '--help' for help.	block %d.	/dev/xdb19	/dev/xda12
how did we get here?	bmap_raw_open	/dev/xdb18	/dev/xda11

/dev/xda10	/dev/sdv7	/dev/sdr14	/dev/sdm9
/dev/xda1	/dev/sdv6	/dev/sdr13	/dev/sdm8
/dev/xda	/dev/sdv5	/dev/sdr12	/dev/sdm7
/dev/sonycd	/dev/sdv4	/dev/sdr11	/dev/sdm6
/dev/sjcd	/dev/sdv3	/dev/sdr10	/dev/sdm5
/dev/sdz9	/dev/sdv2	/dev/sdr1	/dev/sdm4
/dev/sdz8	/dev/sdv15	/dev/sdr	/dev/sdm3
/dev/sdz7	/dev/sdv14	/dev/sdq9	/dev/sdm2
/dev/sdz6	/dev/sdv13	/dev/sdq8	/dev/sdm15
/dev/sdz5	/dev/sdv12	/dev/sdq7	/dev/sdm14
/dev/sdz4	/dev/sdv11	/dev/sdq6	/dev/sdm13
/dev/sdz3	/dev/sdv10	/dev/sdq5	/dev/sdm12
/dev/sdz2	/dev/sdv1	/dev/sdq4	/dev/sdm11
/dev/sdz15	/dev/sdv	/dev/sdq3	/dev/sdm10
/dev/sdz14	/dev/sdu9	/dev/sdq2	/dev/sdm1
/dev/sdz13	/dev/sdu8	/dev/sdq15	/dev/sdm
/dev/sdz12	/dev/sdu7	/dev/sdq14	/dev/sdl9
/dev/sdz11	/dev/sdu6	/dev/sdq13	/dev/sdl8
/dev/sdz10	/dev/sdu5	/dev/sdq12	/dev/sdl7
/dev/sdz1	/dev/sdu4	/dev/sdq11	/dev/sdl6
/dev/sdz	/dev/sdu3	/dev/sdq10	/dev/sdl5
/dev/sdy9	/dev/sdu2	/dev/sdq1	/dev/sdl4
/dev/sdy8	/dev/sdu15	/dev/sdq	/dev/sdl3
/dev/sdy7	/dev/sdu14	/dev/sdp9	/dev/sdl2
/dev/sdy6	/dev/sdu13	/dev/sdp8	/dev/sdl15
/dev/sdy5	/dev/sdu12	/dev/sdp7	/dev/sdl14
/dev/sdy4	/dev/sdu11	/dev/sdp6	/dev/sdl13
/dev/sdy3	/dev/sdu10	/dev/sdp5	/dev/sdl12
/dev/sdy2	/dev/sdu1	/dev/sdp4	/dev/sdl11
/dev/sdy15	/dev/sdu	/dev/sdp3	/dev/sdl10
/dev/sdy14	/dev/sdt9	/dev/sdp2	/dev/sdl1
/dev/sdy13	/dev/sdt8	/dev/sdp15	/dev/sdl
/dev/sdy12	/dev/sdt7	/dev/sdp14	/dev/sdk9
/dev/sdy11	/dev/sdt6	/dev/sdp13	/dev/sdk8
/dev/sdy10	/dev/sdt5	/dev/sdp12	/dev/sdk7
/dev/sdy1	/dev/sdt4	/dev/sdp11	/dev/sdk6
/dev/sdy	/dev/sdt3	/dev/sdp10	/dev/sdk5
/dev/sdx9	/dev/sdt2	/dev/sdp1	/dev/sdk4
/dev/sdx8	/dev/sdt15	/dev/sdp	/dev/sdk3
/dev/sdx7	/dev/sdt14	/dev/sdo9	/dev/sdk2
/dev/sdx6	/dev/sdt13	/dev/sdo8	/dev/sdk15
/dev/sdx5	/dev/sdt12	/dev/sdo7	/dev/sdk14
/dev/sdx4	/dev/sdt11	/dev/sdo6	/dev/sdk13
/dev/sdx3	/dev/sdt10	/dev/sdo5	/dev/sdk12
/dev/sdx2	/dev/sdt1	/dev/sdo4	/dev/sdk11
/dev/sdx15	/dev/sdt	/dev/sdo3	/dev/sdk10
/dev/sdx14	/dev/sds9	/dev/sdo2	/dev/sdk1
/dev/sdx13	/dev/sds8	/dev/sdo15	/dev/sdk
/dev/sdx12	/dev/sds7	/dev/sdo14	/dev/sdj9
/dev/sdx11	/dev/sds6	/dev/sdo13	/dev/sdj8
/dev/sdx10	/dev/sds5	/dev/sdo12	/dev/sdj7
/dev/sdx1	/dev/sds4	/dev/sdo11	/dev/sdj6
/dev/sdx	/dev/sds3	/dev/sdo10	/dev/sdj5
/dev/sdw9	/dev/sds2	/dev/sdo1	/dev/sdj4
/dev/sdw8	/dev/sds15	/dev/sdo	/dev/sdj3
/dev/sdw7	/dev/sds14	/dev/sdn9	/dev/sdj2
/dev/sdw6	/dev/sds13	/dev/sdn8	/dev/sdj15
/dev/sdw5	/dev/sds12	/dev/sdn7	/dev/sdj14
/dev/sdw4	/dev/sds11	/dev/sdn6	/dev/sdj13
/dev/sdw3	/dev/sds10	/dev/sdn5	/dev/sdj12
/dev/sdw2	/dev/sds1	/dev/sdn4	/dev/sdj11
/dev/sdw15	/dev/sds	/dev/sdn3	/dev/sdj10
/dev/sdw14	/dev/sdr9	/dev/sdn2	/dev/sdj1
/dev/sdw13	/dev/sdr8	/dev/sdn15	/dev/sdj
/dev/sdw12	/dev/sdr7	/dev/sdn14	/dev/sdi9
/dev/sdw11	/dev/sdr6	/dev/sdn13	/dev/sdi8
/dev/sdw10	/dev/sdr5	/dev/sdn12	/dev/sdi7
/dev/sdw1	/dev/sdr4	/dev/sdn11	/dev/sdi6
/dev/sdw	/dev/sdr3	/dev/sdn10	/dev/sdi5
/dev/sdv9	/dev/sdr2	/dev/sdn1	/dev/sdi4
/dev/sdv8	/dev/sdr15	/dev/sdn	/dev/sdi3

/dev/sdi2	/dev/sde1	/dev/sdtt4	/dev/sddp11
/dev/sdi15	/dev/sde	/dev/sdtt3	/dev/sddp10
/dev/sdi14	/dev/sddx9	/dev/sdtt2	/dev/sddp1
/dev/sdi13	/dev/sddx8	/dev/sdtt15	/dev/sddp
/dev/sdi12	/dev/sddx7	/dev/sdtt14	/dev/sddo9
/dev/sdi11	/dev/sddx6	/dev/sdtt13	/dev/sddo8
/dev/sdi10	/dev/sddx5	/dev/sdtt12	/dev/sddo7
/dev/sdi1	/dev/sddx4	/dev/sdtt11	/dev/sddo6
/dev/sdi	/dev/sddx3	/dev/sdtt10	/dev/sddo5
/dev/sdh9	/dev/sddx2	/dev/sdtt1	/dev/sddo4
/dev/sdh8	/dev/sddx15	/dev/sdtt	/dev/sddo3
/dev/sdh7	/dev/sddx14	/dev/sdds9	/dev/sddo2
/dev/sdh6	/dev/sddx13	/dev/sdds8	/dev/sddo15
/dev/sdh5	/dev/sddx12	/dev/sdds7	/dev/sddo14
/dev/sdh4	/dev/sddx11	/dev/sdds6	/dev/sddo13
/dev/sdh3	/dev/sddx10	/dev/sdds5	/dev/sddo12
/dev/sdh2	/dev/sddx1	/dev/sdds4	/dev/sddo11
/dev/sdh15	/dev/sddx	/dev/sdds3	/dev/sddo10
/dev/sdh14	/dev/sddw9	/dev/sdds2	/dev/sddo1
/dev/sdh13	/dev/sddw8	/dev/sdds15	/dev/sddo
/dev/sdh12	/dev/sddw7	/dev/sdds14	/dev/sddn9
/dev/sdh11	/dev/sddw6	/dev/sdds13	/dev/sddn8
/dev/sdh10	/dev/sddw5	/dev/sdds12	/dev/sddn7
/dev/sdh1	/dev/sddw4	/dev/sdds11	/dev/sddn6
/dev/sdh	/dev/sddw3	/dev/sdds10	/dev/sddn5
/dev/sdg9	/dev/sddw2	/dev/sdds1	/dev/sddn4
/dev/sdg8	/dev/sddw15	/dev/sdds	/dev/sddn3
/dev/sdg7	/dev/sddw14	/dev/sddr9	/dev/sddn2
/dev/sdg6	/dev/sddw13	/dev/sddr8	/dev/sddn15
/dev/sdg5	/dev/sddw12	/dev/sddr7	/dev/sddn14
/dev/sdg4	/dev/sddw11	/dev/sddr6	/dev/sddn13
/dev/sdg3	/dev/sddw10	/dev/sddr5	/dev/sddn12
/dev/sdg2	/dev/sddw1	/dev/sddr4	/dev/sddn11
/dev/sdg15	/dev/sddw	/dev/sddr3	/dev/sddn10
/dev/sdg14	/dev/sddv9	/dev/sddr2	/dev/sddn1
/dev/sdg13	/dev/sddv8	/dev/sddr15	/dev/sddn
/dev/sdg12	/dev/sddv7	/dev/sddr14	/dev/sddm9
/dev/sdg11	/dev/sddv6	/dev/sddr13	/dev/sddm8
/dev/sdg10	/dev/sddv5	/dev/sddr12	/dev/sddm7
/dev/sdg1	/dev/sddv4	/dev/sddr11	/dev/sddm6
/dev/sdg	/dev/sddv3	/dev/sddr10	/dev/sddm5
/dev/sdf9	/dev/sddv2	/dev/sddr1	/dev/sddm4
/dev/sdf8	/dev/sddv15	/dev/sddr	/dev/sddm3
/dev/sdf7	/dev/sddv14	/dev/sddq9	/dev/sddm2
/dev/sdf6	/dev/sddv13	/dev/sddq8	/dev/sddm15
/dev/sdf5	/dev/sddv12	/dev/sddq7	/dev/sddm14
/dev/sdf4	/dev/sddv11	/dev/sddq6	/dev/sddm13
/dev/sdf3	/dev/sddv10	/dev/sddq5	/dev/sddm12
/dev/sdf2	/dev/sddv1	/dev/sddq4	/dev/sddm11
/dev/sdf15	/dev/sddv	/dev/sddq3	/dev/sddm10
/dev/sdf14	/dev/sddu9	/dev/sddq2	/dev/sddm1
/dev/sdf13	/dev/sddu8	/dev/sddq15	/dev/sddm
/dev/sdf12	/dev/sddu7	/dev/sddq14	/dev/sdd19
/dev/sdf11	/dev/sddu6	/dev/sddq13	/dev/sdd18
/dev/sdf10	/dev/sddu5	/dev/sddq12	/dev/sdd17
/dev/sdf1	/dev/sddu4	/dev/sddq11	/dev/sdd16
/dev/sdf	/dev/sddu3	/dev/sddq10	/dev/sdd15
/dev/sde9	/dev/sddu2	/dev/sddq1	/dev/sdd14
/dev/sde8	/dev/sddu15	/dev/sddq	/dev/sdd13
/dev/sde7	/dev/sddu14	/dev/sddp9	/dev/sdd12
/dev/sde6	/dev/sddu13	/dev/sddp8	/dev/sdd115
/dev/sde5	/dev/sddu12	/dev/sddp7	/dev/sdd114
/dev/sde4	/dev/sddu11	/dev/sddp6	/dev/sdd113
/dev/sde3	/dev/sddu10	/dev/sddp5	/dev/sdd112
/dev/sde2	/dev/sddu1	/dev/sddp4	/dev/sdd111
/dev/sde15	/dev/sddu	/dev/sddp3	/dev/sdd110
/dev/sde14	/dev/sdtt9	/dev/sddp2	/dev/sdd11
/dev/sde13	/dev/sdtt8	/dev/sddp15	/dev/sdd1
/dev/sde12	/dev/sdtt7	/dev/sddp14	/dev/sddk9
/dev/sde11	/dev/sdtt6	/dev/sddp13	/dev/sddk8
/dev/sde10	/dev/sdtt5	/dev/sddp12	/dev/sddk7

/dev/sddk6	/dev/sddg13	/dev/sddb8	/dev/sdcy15
/dev/sddk5	/dev/sddg12	/dev/sddb7	/dev/sdcy14
/dev/sddk4	/dev/sddg11	/dev/sddb6	/dev/sdcy13
/dev/sddk3	/dev/sddg10	/dev/sddb5	/dev/sdcy12
/dev/sddk2	/dev/sddg1	/dev/sddb4	/dev/sdcy11
/dev/sddk15	/dev/sddg	/dev/sddb3	/dev/sdcy10
/dev/sddk14	/dev/sddf9	/dev/sddb2	/dev/sdcy1
/dev/sddk13	/dev/sddf8	/dev/sddb15	/dev/sdcy
/dev/sddk12	/dev/sddf7	/dev/sddb14	/dev/sdcx9
/dev/sddk11	/dev/sddf6	/dev/sddb13	/dev/sdcx8
/dev/sddk10	/dev/sddf5	/dev/sddb12	/dev/sdcx7
/dev/sddk1	/dev/sddf4	/dev/sddb11	/dev/sdcx6
/dev/sddk	/dev/sddf3	/dev/sddb10	/dev/sdcx5
/dev/sddj9	/dev/sddf2	/dev/sddb1	/dev/sdcx4
/dev/sddj8	/dev/sddf15	/dev/sddb	/dev/sdcx3
/dev/sddj7	/dev/sddf14	/dev/sdda9	/dev/sdcx2
/dev/sddj6	/dev/sddf13	/dev/sdda8	/dev/sdcx15
/dev/sddj5	/dev/sddf12	/dev/sdda7	/dev/sdcx14
/dev/sddj4	/dev/sddf11	/dev/sdda6	/dev/sdcx13
/dev/sddj3	/dev/sddf10	/dev/sdda5	/dev/sdcx12
/dev/sddj2	/dev/sddf1	/dev/sdda4	/dev/sdcx11
/dev/sddj15	/dev/sddf	/dev/sdda3	/dev/sdcx10
/dev/sddj14	/dev/sdde9	/dev/sdda2	/dev/sdcx1
/dev/sddj13	/dev/sdde8	/dev/sdda15	/dev/sdcx
/dev/sddj12	/dev/sdde7	/dev/sdda14	/dev/sdcw9
/dev/sddj11	/dev/sdde6	/dev/sdda13	/dev/sdcw8
/dev/sddj10	/dev/sdde5	/dev/sdda12	/dev/sdcw7
/dev/sddj1	/dev/sdde4	/dev/sdda11	/dev/sdcw6
/dev/sddj	/dev/sdde3	/dev/sdda10	/dev/sdcw5
/dev/sddi9	/dev/sdde2	/dev/sdda1	/dev/sdcw4
/dev/sddi8	/dev/sdde15	/dev/sdda	/dev/sdcw3
/dev/sddi7	/dev/sdde14	/dev/sdd9	/dev/sdcw2
/dev/sddi6	/dev/sdde13	/dev/sdd8	/dev/sdcw15
/dev/sddi5	/dev/sdde12	/dev/sdd7	/dev/sdcw14
/dev/sddi4	/dev/sdde11	/dev/sdd6	/dev/sdcw13
/dev/sddi3	/dev/sdde10	/dev/sdd5	/dev/sdcw12
/dev/sddi2	/dev/sdde1	/dev/sdd4	/dev/sdcw11
/dev/sddi15	/dev/sdde	/dev/sdd3	/dev/sdcw10
/dev/sddi14	/dev/sddd9	/dev/sdd2	/dev/sdcw1
/dev/sddi13	/dev/sddd8	/dev/sdd15	/dev/sdcw
/dev/sddi12	/dev/sddd7	/dev/sdd14	/dev/sdcv9
/dev/sddi11	/dev/sddd6	/dev/sdd13	/dev/sdcv8
/dev/sddi10	/dev/sddd5	/dev/sdd12	/dev/sdcv7
/dev/sddi1	/dev/sddd4	/dev/sdd11	/dev/sdcv6
/dev/sddi	/dev/sddd3	/dev/sdd10	/dev/sdcv5
/dev/sddh9	/dev/sddd2	/dev/sdd1	/dev/sdcv4
/dev/sddh8	/dev/sddd15	/dev/sdd	/dev/sdcv3
/dev/sddh7	/dev/sddd14	/dev/sdcz9	/dev/sdcv2
/dev/sddh6	/dev/sddd13	/dev/sdcz8	/dev/sdcv15
/dev/sddh5	/dev/sddd12	/dev/sdcz7	/dev/sdcv14
/dev/sddh4	/dev/sddd11	/dev/sdcz6	/dev/sdcv13
/dev/sddh3	/dev/sddd10	/dev/sdcz5	/dev/sdcv12
/dev/sddh2	/dev/sddd1	/dev/sdcz4	/dev/sdcv11
/dev/sddh15	/dev/sddd	/dev/sdcz3	/dev/sdcv10
/dev/sddh14	/dev/sddc9	/dev/sdcz2	/dev/sdcv1
/dev/sddh13	/dev/sddc8	/dev/sdcz15	/dev/sdcv
/dev/sddh12	/dev/sddc7	/dev/sdcz14	/dev/sdcu9
/dev/sddh11	/dev/sddc6	/dev/sdcz13	/dev/sdcu8
/dev/sddh10	/dev/sddc5	/dev/sdcz12	/dev/sdcu7
/dev/sddh1	/dev/sddc4	/dev/sdcz11	/dev/sdcu6
/dev/sddh	/dev/sddc3	/dev/sdcz10	/dev/sdcu5
/dev/sddg9	/dev/sddc2	/dev/sdcz1	/dev/sdcu4
/dev/sddg8	/dev/sddc15	/dev/sdcz	/dev/sdcu3
/dev/sddg7	/dev/sddc14	/dev/sdcy9	/dev/sdcu2
/dev/sddg6	/dev/sddc13	/dev/sdcy8	/dev/sdcu15
/dev/sddg5	/dev/sddc12	/dev/sdcy7	/dev/sdcu14
/dev/sddg4	/dev/sddc11	/dev/sdcy6	/dev/sdcu13
/dev/sddg3	/dev/sddc10	/dev/sdcy5	/dev/sdcu12
/dev/sddg2	/dev/sddc1	/dev/sdcy4	/dev/sdcu11
/dev/sddg15	/dev/sddc	/dev/sdcy3	/dev/sdcu10
/dev/sddg14	/dev/sddb9	/dev/sdcy2	/dev/sdcu1

/dev/sdcu	/dev/sdcp3	/dev/sdcl10	/dev/sdcg5
/dev/sdct9	/dev/sdcp2	/dev/sdcl1	/dev/sdcg4
/dev/sdct8	/dev/sdcp15	/dev/sdcl	/dev/sdcg3
/dev/sdct7	/dev/sdcp14	/dev/sdck9	/dev/sdcg2
/dev/sdct6	/dev/sdcp13	/dev/sdck8	/dev/sdcg15
/dev/sdct5	/dev/sdcp12	/dev/sdck7	/dev/sdcg14
/dev/sdct4	/dev/sdcp11	/dev/sdck6	/dev/sdcg13
/dev/sdct3	/dev/sdcp10	/dev/sdck5	/dev/sdcg12
/dev/sdct2	/dev/sdcp1	/dev/sdck4	/dev/sdcg11
/dev/sdct15	/dev/sdcp	/dev/sdck3	/dev/sdcg10
/dev/sdct14	/dev/sdco9	/dev/sdck2	/dev/sdcg1
/dev/sdct13	/dev/sdco8	/dev/sdck15	/dev/sdcg
/dev/sdct12	/dev/sdco7	/dev/sdck14	/dev/sdcf9
/dev/sdct11	/dev/sdco6	/dev/sdck13	/dev/sdcf8
/dev/sdct10	/dev/sdco5	/dev/sdck12	/dev/sdcf7
/dev/sdct1	/dev/sdco4	/dev/sdck11	/dev/sdcf6
/dev/sdct	/dev/sdco3	/dev/sdck10	/dev/sdcf5
/dev/sdcs9	/dev/sdco2	/dev/sdck1	/dev/sdcf4
/dev/sdcs8	/dev/sdco15	/dev/sdck	/dev/sdcf3
/dev/sdcs7	/dev/sdco14	/dev/sdcj9	/dev/sdcf2
/dev/sdcs6	/dev/sdco13	/dev/sdcj8	/dev/sdcf15
/dev/sdcs5	/dev/sdco12	/dev/sdcj7	/dev/sdcf14
/dev/sdcs4	/dev/sdco11	/dev/sdcj6	/dev/sdcf13
/dev/sdcs3	/dev/sdco10	/dev/sdcj5	/dev/sdcf12
/dev/sdcs2	/dev/sdco1	/dev/sdcj4	/dev/sdcf11
/dev/sdcs15	/dev/sdco	/dev/sdcj3	/dev/sdcf10
/dev/sdcs14	/dev/sdcn9	/dev/sdcj2	/dev/sdcf1
/dev/sdcs13	/dev/sdcn8	/dev/sdcj15	/dev/sdcf
/dev/sdcs12	/dev/sdcn7	/dev/sdcj14	/dev/sdce9
/dev/sdcs11	/dev/sdcn6	/dev/sdcj13	/dev/sdce8
/dev/sdcs10	/dev/sdcn5	/dev/sdcj12	/dev/sdce7
/dev/sdcs1	/dev/sdcn4	/dev/sdcj11	/dev/sdce6
/dev/sdcs	/dev/sdcn3	/dev/sdcj10	/dev/sdce5
/dev/sdcr9	/dev/sdcn2	/dev/sdcj1	/dev/sdce4
/dev/sdcr8	/dev/sdcn15	/dev/sdcj	/dev/sdce3
/dev/sdcr7	/dev/sdcn14	/dev/sdci9	/dev/sdce2
/dev/sdcr6	/dev/sdcn13	/dev/sdci8	/dev/sdce15
/dev/sdcr5	/dev/sdcn12	/dev/sdci7	/dev/sdce14
/dev/sdcr4	/dev/sdcn11	/dev/sdci6	/dev/sdce13
/dev/sdcr3	/dev/sdcn10	/dev/sdci5	/dev/sdce12
/dev/sdcr2	/dev/sdcn1	/dev/sdci4	/dev/sdce11
/dev/sdcr15	/dev/sdcn	/dev/sdci3	/dev/sdce10
/dev/sdcr14	/dev/sdcn9	/dev/sdci2	/dev/sdce1
/dev/sdcr13	/dev/sdcn8	/dev/sdci15	/dev/sdce
/dev/sdcr12	/dev/sdcn7	/dev/sdci14	/dev/sdcd9
/dev/sdcr11	/dev/sdcn6	/dev/sdci13	/dev/sdcd8
/dev/sdcr10	/dev/sdcn5	/dev/sdci12	/dev/sdcd7
/dev/sdcr1	/dev/sdcn4	/dev/sdci11	/dev/sdcd6
/dev/sdcq9	/dev/sdcn3	/dev/sdci10	/dev/sdcd5
/dev/sdcq8	/dev/sdcn2	/dev/sdci1	/dev/sdcd4
/dev/sdcq7	/dev/sdcn15	/dev/sdci	/dev/sdcd3
/dev/sdcq6	/dev/sdcn14	/dev/sdch9	/dev/sdcd2
/dev/sdcq5	/dev/sdcn13	/dev/sdch8	/dev/sdcd15
/dev/sdcq4	/dev/sdcn12	/dev/sdch7	/dev/sdcd14
/dev/sdcq3	/dev/sdcn11	/dev/sdch6	/dev/sdcd13
/dev/sdcq2	/dev/sdcn10	/dev/sdch5	/dev/sdcd12
/dev/sdcq15	/dev/sdcn1	/dev/sdch4	/dev/sdcd11
/dev/sdcq14	/dev/sdcn	/dev/sdch3	/dev/sdcd10
/dev/sdcq13	/dev/sdc19	/dev/sdch2	/dev/sdcd1
/dev/sdcq12	/dev/sdc18	/dev/sdch15	/dev/sdcd
/dev/sdcq11	/dev/sdc17	/dev/sdch14	/dev/sdcc9
/dev/sdcq10	/dev/sdc16	/dev/sdch13	/dev/sdcc8
/dev/sdcq1	/dev/sdc15	/dev/sdch12	/dev/sdcc7
/dev/sdcq	/dev/sdc14	/dev/sdch11	/dev/sdcc6
/dev/sdcp9	/dev/sdc13	/dev/sdch10	/dev/sdcc5
/dev/sdcp8	/dev/sdc12	/dev/sdch1	/dev/sdcc4
/dev/sdcp7	/dev/sdc115	/dev/sdch	/dev/sdcc3
/dev/sdcp6	/dev/sdc114	/dev/sdcg9	/dev/sdcc2
/dev/sdcp5	/dev/sdc113	/dev/sdcg8	/dev/sdcc15
/dev/sdcp4	/dev/sdc112	/dev/sdcg7	/dev/sdcc14
	/dev/sdc111	/dev/sdcg6	/dev/sdcc13

/dev/sdcc12	/dev/sdby7	/dev/sdbu14	/dev/sdbp9
/dev/sdcc11	/dev/sdby6	/dev/sdbu13	/dev/sdbp8
/dev/sdcc10	/dev/sdby5	/dev/sdbu12	/dev/sdbp7
/dev/sdcc1	/dev/sdby4	/dev/sdbu11	/dev/sdbp6
/dev/sdcc	/dev/sdby3	/dev/sdbu10	/dev/sdbp5
/dev/sdcb9	/dev/sdby2	/dev/sdbu1	/dev/sdbp4
/dev/sdcb8	/dev/sdby15	/dev/sdbu	/dev/sdbp3
/dev/sdcb7	/dev/sdby14	/dev/sdbt9	/dev/sdbp2
/dev/sdcb6	/dev/sdby13	/dev/sdbt8	/dev/sdbp15
/dev/sdcb5	/dev/sdby12	/dev/sdbt7	/dev/sdbp14
/dev/sdcb4	/dev/sdby11	/dev/sdbt6	/dev/sdbp13
/dev/sdcb3	/dev/sdby10	/dev/sdbt5	/dev/sdbp12
/dev/sdcb2	/dev/sdby1	/dev/sdbt4	/dev/sdbp11
/dev/sdcb15	/dev/sdby	/dev/sdbt3	/dev/sdbp10
/dev/sdcb14	/dev/sdbx9	/dev/sdbt2	/dev/sdbp1
/dev/sdcb13	/dev/sdbx8	/dev/sdbt15	/dev/sdbp
/dev/sdcb12	/dev/sdbx7	/dev/sdbt14	/dev/sdbo9
/dev/sdcb11	/dev/sdbx6	/dev/sdbt13	/dev/sdbo8
/dev/sdcb10	/dev/sdbx5	/dev/sdbt12	/dev/sdbo7
/dev/sdcb1	/dev/sdbx4	/dev/sdbt11	/dev/sdbo6
/dev/sdcb	/dev/sdbx3	/dev/sdbt10	/dev/sdbo5
/dev/sdca9	/dev/sdbx2	/dev/sdbt1	/dev/sdbo4
/dev/sdca8	/dev/sdbx15	/dev/sdbt	/dev/sdbo3
/dev/sdca7	/dev/sdbx14	/dev/sdbs9	/dev/sdbo2
/dev/sdca6	/dev/sdbx13	/dev/sdbs8	/dev/sdbo15
/dev/sdca5	/dev/sdbx12	/dev/sdbs7	/dev/sdbo14
/dev/sdca4	/dev/sdbx11	/dev/sdbs6	/dev/sdbo13
/dev/sdca3	/dev/sdbx10	/dev/sdbs5	/dev/sdbo12
/dev/sdca2	/dev/sdbx1	/dev/sdbs4	/dev/sdbo11
/dev/sdca15	/dev/sdbx	/dev/sdbs3	/dev/sdbo10
/dev/sdca14	/dev/sdbw9	/dev/sdbs2	/dev/sdbo1
/dev/sdca13	/dev/sdbw8	/dev/sdbs15	/dev/sdbo
/dev/sdca12	/dev/sdbw7	/dev/sdbs14	/dev/sdbn9
/dev/sdca11	/dev/sdbw6	/dev/sdbs13	/dev/sdbn8
/dev/sdca10	/dev/sdbw5	/dev/sdbs12	/dev/sdbn7
/dev/sdca1	/dev/sdbw4	/dev/sdbs11	/dev/sdbn6
/dev/sdca	/dev/sdbw3	/dev/sdbs10	/dev/sdbn5
/dev/sdc9	/dev/sdbw2	/dev/sdbs1	/dev/sdbn4
/dev/sdc8	/dev/sdbw15	/dev/sdbs	/dev/sdbn3
/dev/sdc7	/dev/sdbw14	/dev/sdbr9	/dev/sdbn2
/dev/sdc6	/dev/sdbw13	/dev/sdbr8	/dev/sdbn15
/dev/sdc5	/dev/sdbw12	/dev/sdbr7	/dev/sdbn14
/dev/sdc4	/dev/sdbw11	/dev/sdbr6	/dev/sdbn13
/dev/sdc3	/dev/sdbw10	/dev/sdbr5	/dev/sdbn12
/dev/sdc2	/dev/sdbw1	/dev/sdbr4	/dev/sdbn11
/dev/sdc15	/dev/sdbw	/dev/sdbr3	/dev/sdbn10
/dev/sdc14	/dev/sdbv9	/dev/sdbr2	/dev/sdbn1
/dev/sdc13	/dev/sdbv8	/dev/sdbr15	/dev/sdbn
/dev/sdc12	/dev/sdbv7	/dev/sdbr14	/dev/sdbm9
/dev/sdc11	/dev/sdbv6	/dev/sdbr13	/dev/sdbm8
/dev/sdc10	/dev/sdbv5	/dev/sdbr12	/dev/sdbm7
/dev/sdc1	/dev/sdbv4	/dev/sdbr11	/dev/sdbm6
/dev/sdbz9	/dev/sdbv3	/dev/sdbr10	/dev/sdbm5
/dev/sdbz8	/dev/sdbv2	/dev/sdbr1	/dev/sdbm4
/dev/sdbz7	/dev/sdbv15	/dev/sdbr	/dev/sdbm3
/dev/sdbz6	/dev/sdbv14	/dev/sdbq9	/dev/sdbm2
/dev/sdbz5	/dev/sdbv13	/dev/sdbq8	/dev/sdbm15
/dev/sdbz4	/dev/sdbv12	/dev/sdbq7	/dev/sdbm14
/dev/sdbz3	/dev/sdbv11	/dev/sdbq6	/dev/sdbm13
/dev/sdbz2	/dev/sdbv10	/dev/sdbq5	/dev/sdbm12
/dev/sdbz15	/dev/sdbv1	/dev/sdbq4	/dev/sdbm11
/dev/sdbz14	/dev/sdbv	/dev/sdbq3	/dev/sdbm10
/dev/sdbz13	/dev/sdbu9	/dev/sdbq2	/dev/sdbm1
/dev/sdbz12	/dev/sdbu8	/dev/sdbq15	/dev/sdbm
/dev/sdbz11	/dev/sdbu7	/dev/sdbq14	/dev/sdb19
/dev/sdbz10	/dev/sdbu6	/dev/sdbq13	/dev/sdb18
/dev/sdbz1	/dev/sdbu5	/dev/sdbq12	/dev/sdb17
/dev/sdbz	/dev/sdbu4	/dev/sdbq11	/dev/sdb16
/dev/sdby9	/dev/sdbu3	/dev/sdbq10	/dev/sdb15
/dev/sdby8	/dev/sdbu2	/dev/sdbq1	/dev/sdb14
	/dev/sdbu15	/dev/sdbq	/dev/sdb13

/dev/sdb12	/dev/sdbh1	/dev/sdbc4	/dev/sdaz11
/dev/sdb115	/dev/sdbh	/dev/sdbc3	/dev/sdaz10
/dev/sdb114	/dev/sdbg9	/dev/sdbc2	/dev/sdaz1
/dev/sdb113	/dev/sdbg8	/dev/sdbc15	/dev/sdaz
/dev/sdb112	/dev/sdbg7	/dev/sdbc14	/dev/sday9
/dev/sdb111	/dev/sdbg6	/dev/sdbc13	/dev/sday8
/dev/sdb110	/dev/sdbg5	/dev/sdbc12	/dev/sday7
/dev/sdb11	/dev/sdbg4	/dev/sdbc11	/dev/sday6
/dev/sdb1	/dev/sdbg3	/dev/sdbc10	/dev/sday5
/dev/sdbk9	/dev/sdbg2	/dev/sdbc1	/dev/sday4
/dev/sdbk8	/dev/sdbg15	/dev/sdbc	/dev/sday3
/dev/sdbk7	/dev/sdbg14	/dev/sdbb9	/dev/sday2
/dev/sdbk6	/dev/sdbg13	/dev/sdbb8	/dev/sday15
/dev/sdbk5	/dev/sdbg12	/dev/sdbb7	/dev/sday14
/dev/sdbk4	/dev/sdbg11	/dev/sdbb6	/dev/sday13
/dev/sdbk3	/dev/sdbg10	/dev/sdbb5	/dev/sday12
/dev/sdbk2	/dev/sdbg1	/dev/sdbb4	/dev/sday11
/dev/sdbk15	/dev/sdbg	/dev/sdbb3	/dev/sday10
/dev/sdbk14	/dev/sdbf9	/dev/sdbb2	/dev/sday1
/dev/sdbk13	/dev/sdbf8	/dev/sdbb15	/dev/sday
/dev/sdbk12	/dev/sdbf7	/dev/sdbb14	/dev/sdax9
/dev/sdbk11	/dev/sdbf6	/dev/sdbb13	/dev/sdax8
/dev/sdbk10	/dev/sdbf5	/dev/sdbb12	/dev/sdax7
/dev/sdbk1	/dev/sdbf4	/dev/sdbb11	/dev/sdax6
/dev/sdbk	/dev/sdbf3	/dev/sdbb10	/dev/sdax5
/dev/sdbj9	/dev/sdbf2	/dev/sdbb1	/dev/sdax4
/dev/sdbj8	/dev/sdbf15	/dev/sdbb	/dev/sdax3
/dev/sdbj7	/dev/sdbf14	/dev/sdba9	/dev/sdax2
/dev/sdbj6	/dev/sdbf13	/dev/sdba8	/dev/sdax15
/dev/sdbj5	/dev/sdbf12	/dev/sdba7	/dev/sdax14
/dev/sdbj4	/dev/sdbf11	/dev/sdba6	/dev/sdax13
/dev/sdbj3	/dev/sdbf10	/dev/sdba5	/dev/sdax12
/dev/sdbj2	/dev/sdbf1	/dev/sdba4	/dev/sdax11
/dev/sdbj15	/dev/sdbf	/dev/sdba3	/dev/sdax10
/dev/sdbj14	/dev/sdbe9	/dev/sdba2	/dev/sdax1
/dev/sdbj13	/dev/sdbe8	/dev/sdba15	/dev/sdax
/dev/sdbj12	/dev/sdbe7	/dev/sdba14	/dev/sdaw9
/dev/sdbj11	/dev/sdbe6	/dev/sdba13	/dev/sdaw8
/dev/sdbj10	/dev/sdbe5	/dev/sdba12	/dev/sdaw7
/dev/sdbj1	/dev/sdbe4	/dev/sdba11	/dev/sdaw6
/dev/sdbj	/dev/sdbe3	/dev/sdba10	/dev/sdaw5
/dev/sdbi9	/dev/sdbe2	/dev/sdba1	/dev/sdaw4
/dev/sdbi8	/dev/sdbe15	/dev/sdba	/dev/sdaw3
/dev/sdbi7	/dev/sdbe14	/dev/sdb9	/dev/sdaw2
/dev/sdbi6	/dev/sdbe13	/dev/sdb8	/dev/sdaw15
/dev/sdbi5	/dev/sdbe12	/dev/sdb7	/dev/sdaw14
/dev/sdbi4	/dev/sdbe11	/dev/sdb6	/dev/sdaw13
/dev/sdbi3	/dev/sdbe10	/dev/sdb5	/dev/sdaw12
/dev/sdbi2	/dev/sdbe1	/dev/sdb4	/dev/sdaw11
/dev/sdbi15	/dev/sdbe	/dev/sdb3	/dev/sdaw10
/dev/sdbi14	/dev/sdbd9	/dev/sdb2	/dev/sdaw1
/dev/sdbi13	/dev/sdbd8	/dev/sdb15	/dev/sdaw
/dev/sdbi12	/dev/sdbd7	/dev/sdb14	/dev/sdav9
/dev/sdbi11	/dev/sdbd6	/dev/sdb13	/dev/sdav8
/dev/sdbi10	/dev/sdbd5	/dev/sdb12	/dev/sdav7
/dev/sdbi1	/dev/sdbd4	/dev/sdb11	/dev/sdav6
/dev/sdbi	/dev/sdbd3	/dev/sdb10	/dev/sdav5
/dev/sdbh9	/dev/sdbd2	/dev/sdb1	/dev/sdav4
/dev/sdbh8	/dev/sdbd15	/dev/sdb	/dev/sdav3
/dev/sdbh7	/dev/sdbd14	/dev/sdaz9	/dev/sdav2
/dev/sdbh6	/dev/sdbd13	/dev/sdaz8	/dev/sdav15
/dev/sdbh5	/dev/sdbd12	/dev/sdaz7	/dev/sdav14
/dev/sdbh4	/dev/sdbd11	/dev/sdaz6	/dev/sdav13
/dev/sdbh3	/dev/sdbd10	/dev/sdaz5	/dev/sdav12
/dev/sdbh2	/dev/sdbd1	/dev/sdaz4	/dev/sdav11
/dev/sdbh15	/dev/sdbd	/dev/sdaz3	/dev/sdav10
/dev/sdbh14	/dev/sdbc9	/dev/sdaz2	/dev/sdav1
/dev/sdbh13	/dev/sdbc8	/dev/sdaz15	/dev/sdav
/dev/sdbh12	/dev/sdbc7	/dev/sdaz14	/dev/sdau9
/dev/sdbh11	/dev/sdbc6	/dev/sdaz13	/dev/sdau8
/dev/sdbh10	/dev/sdbc5	/dev/sdaz12	/dev/sdau7

/dev/sdau6	/dev/sdaq13	/dev/sdal8	/dev/sdah15
/dev/sdau5	/dev/sdaq12	/dev/sdal7	/dev/sdah14
/dev/sdau4	/dev/sdaq11	/dev/sdal6	/dev/sdah13
/dev/sdau3	/dev/sdaq10	/dev/sdal5	/dev/sdah12
/dev/sdau2	/dev/sdaq1	/dev/sdal4	/dev/sdah11
/dev/sdau15	/dev/sdaq	/dev/sdal3	/dev/sdah10
/dev/sdau14	/dev/sdap9	/dev/sdal2	/dev/sdah1
/dev/sdau13	/dev/sdap8	/dev/sdal15	/dev/sdah
/dev/sdau12	/dev/sdap7	/dev/sdal14	/dev/sdag9
/dev/sdau11	/dev/sdap6	/dev/sdal13	/dev/sdag8
/dev/sdau10	/dev/sdap5	/dev/sdal12	/dev/sdag7
/dev/sdau1	/dev/sdap4	/dev/sdal11	/dev/sdag6
/dev/sdau	/dev/sdap3	/dev/sdal10	/dev/sdag5
/dev/sdat9	/dev/sdap2	/dev/sdal1	/dev/sdag4
/dev/sdat8	/dev/sdap15	/dev/sdal	/dev/sdag3
/dev/sdat7	/dev/sdap14	/dev/sdak9	/dev/sdag2
/dev/sdat6	/dev/sdap13	/dev/sdak8	/dev/sdag15
/dev/sdat5	/dev/sdap12	/dev/sdak7	/dev/sdag14
/dev/sdat4	/dev/sdap11	/dev/sdak6	/dev/sdag13
/dev/sdat3	/dev/sdap10	/dev/sdak5	/dev/sdag12
/dev/sdat2	/dev/sdap1	/dev/sdak4	/dev/sdag11
/dev/sdat15	/dev/sdap	/dev/sdak3	/dev/sdag10
/dev/sdat14	/dev/sdao9	/dev/sdak2	/dev/sdag1
/dev/sdat13	/dev/sdao8	/dev/sdak15	/dev/sdag
/dev/sdat12	/dev/sdao7	/dev/sdak14	/dev/sdaf9
/dev/sdat11	/dev/sdao6	/dev/sdak13	/dev/sdaf8
/dev/sdat10	/dev/sdao5	/dev/sdak12	/dev/sdaf7
/dev/sdat1	/dev/sdao4	/dev/sdak11	/dev/sdaf6
/dev/sdat	/dev/sdao3	/dev/sdak10	/dev/sdaf5
/dev/sdas9	/dev/sdao2	/dev/sdak1	/dev/sdaf4
/dev/sdas8	/dev/sdao15	/dev/sdak	/dev/sdaf3
/dev/sdas7	/dev/sdao14	/dev/sdaj9	/dev/sdaf2
/dev/sdas6	/dev/sdao13	/dev/sdaj8	/dev/sdaf15
/dev/sdas5	/dev/sdao12	/dev/sdaj7	/dev/sdaf14
/dev/sdas4	/dev/sdao11	/dev/sdaj6	/dev/sdaf13
/dev/sdas3	/dev/sdao10	/dev/sdaj5	/dev/sdaf12
/dev/sdas2	/dev/sdao1	/dev/sdaj4	/dev/sdaf11
/dev/sdas15	/dev/sdao	/dev/sdaj3	/dev/sdaf10
/dev/sdas14	/dev/sdan9	/dev/sdaj2	/dev/sdaf1
/dev/sdas13	/dev/sdan8	/dev/sdaj15	/dev/sdaf
/dev/sdas12	/dev/sdan7	/dev/sdaj14	/dev/sdae9
/dev/sdas11	/dev/sdan6	/dev/sdaj13	/dev/sdae8
/dev/sdas10	/dev/sdan5	/dev/sdaj12	/dev/sdae7
/dev/sdas1	/dev/sdan4	/dev/sdaj11	/dev/sdae6
/dev/sdas	/dev/sdan3	/dev/sdaj10	/dev/sdae5
/dev/sdar9	/dev/sdan2	/dev/sdaj1	/dev/sdae4
/dev/sdar8	/dev/sdan15	/dev/sdaj	/dev/sdae3
/dev/sdar7	/dev/sdan14	/dev/sdai9	/dev/sdae2
/dev/sdar6	/dev/sdan13	/dev/sdai8	/dev/sdae15
/dev/sdar5	/dev/sdan12	/dev/sdai7	/dev/sdae14
/dev/sdar4	/dev/sdan11	/dev/sdai6	/dev/sdae13
/dev/sdar3	/dev/sdan10	/dev/sdai5	/dev/sdae12
/dev/sdar2	/dev/sdan1	/dev/sdai4	/dev/sdae11
/dev/sdar15	/dev/sdan	/dev/sdai3	/dev/sdae10
/dev/sdar14	/dev/sdam9	/dev/sdai2	/dev/sdae1
/dev/sdar13	/dev/sdam8	/dev/sdai15	/dev/sdae
/dev/sdar12	/dev/sdam7	/dev/sdai14	/dev/sdad9
/dev/sdar11	/dev/sdam6	/dev/sdai13	/dev/sdad8
/dev/sdar10	/dev/sdam5	/dev/sdai12	/dev/sdad7
/dev/sdar1	/dev/sdam4	/dev/sdai11	/dev/sdad6
/dev/sdar	/dev/sdam3	/dev/sdai10	/dev/sdad5
/dev/sdaq9	/dev/sdam2	/dev/sdai1	/dev/sdad4
/dev/sdaq8	/dev/sdam15	/dev/sdai	/dev/sdad3
/dev/sdaq7	/dev/sdam14	/dev/sdah9	/dev/sdad2
/dev/sdaq6	/dev/sdam13	/dev/sdah8	/dev/sdad15
/dev/sdaq5	/dev/sdam12	/dev/sdah7	/dev/sdad14
/dev/sdaq4	/dev/sdam11	/dev/sdah6	/dev/sdad13
/dev/sdaq3	/dev/sdam10	/dev/sdah5	/dev/sdad12
/dev/sdaq2	/dev/sdam1	/dev/sdah4	/dev/sdad11
/dev/sdaq15	/dev/sdam	/dev/sdah3	/dev/sdad10
/dev/sdaq14	/dev/sdal9	/dev/sdah2	/dev/sdad1

/dev/sdad	/dev/scd1	/dev/pdc10	/dev/nb0
/dev/sdac9	/dev/scd0	/dev/pdc1	/dev/md31
/dev/sdac8	/dev/sbpcd9	/dev/pdc	/dev/md30
/dev/sdac7	/dev/sbpcd8	/dev/pdb9	/dev/md29
/dev/sdac6	/dev/sbpcd7	/dev/pdb8	/dev/md28
/dev/sdac5	/dev/sbpcd6	/dev/pdb7	/dev/md27
/dev/sdac4	/dev/sbpcd5	/dev/pdb6	/dev/md26
/dev/sdac3	/dev/sbpcd4	/dev/pdb5	/dev/md25
/dev/sdac2	/dev/sbpcd3	/dev/pdb4	/dev/md24
/dev/sdac15	/dev/sbpcd2	/dev/pdb3	/dev/md23
/dev/sdac14	/dev/sbpcd15	/dev/pdb2	/dev/md22
/dev/sdac13	/dev/sbpcd14	/dev/pdb15	/dev/md21
/dev/sdac12	/dev/sbpcd13	/dev/pdb14	/dev/md20
/dev/sdac11	/dev/sbpcd12	/dev/pdb13	/dev/md19
/dev/sdac10	/dev/sbpcd11	/dev/pdb12	/dev/md18
/dev/sdac1	/dev/sbpcd10	/dev/pdb11	/dev/md17
/dev/sdac	/dev/sbpcd1	/dev/pdb10	/dev/md16
/dev/sdab9	/dev/sbpcd0	/dev/pdb1	/dev/mcdx
/dev/sdab8	/dev/ram9	/dev/pdb	/dev/mcd
/dev/sdab7	/dev/ram8	/dev/pda9	/dev/loop9
/dev/sdab6	/dev/ram7	/dev/pda8	/dev/loop8
/dev/sdab5	/dev/ram6	/dev/pda7	/dev/loop7
/dev/sdab4	/dev/ram5	/dev/pda6	/dev/loop6
/dev/sdab3	/dev/ram4	/dev/pda5	/dev/loop5
/dev/sdab2	/dev/ram3	/dev/pda4	/dev/loop4
/dev/sdab15	/dev/ram2	/dev/pda3	/dev/loop3
/dev/sdab14	/dev/ram19	/dev/pda2	/dev/loop2
/dev/sdab13	/dev/ram18	/dev/pda15	/dev/loop15
/dev/sdab12	/dev/ram17	/dev/pda14	/dev/loop14
/dev/sdab11	/dev/ram16	/dev/pda13	/dev/loop13
/dev/sdab10	/dev/ram15	/dev/pda12	/dev/loop12
/dev/sdab1	/dev/ram14	/dev/pda11	/dev/loop11
/dev/sdab	/dev/ram13	/dev/pda10	/dev/loop10
/dev/sdaa9	/dev/ram12	/dev/pda1	/dev/loop1
/dev/sdaa8	/dev/ram11	/dev/pda	/dev/loop0
/dev/sdaa7	/dev/ram10	/dev/pcd3	/dev/jsfd
/dev/sdaa6	/dev/ram1	/dev/pcd2	/dev/initrd
/dev/sdaa5	/dev/ram0	/dev/pcd1	/dev/hitcd
/dev/sdaa4	/dev/pf3	/dev/pcd0	/dev/hdt9
/dev/sdaa3	/dev/pf2	/dev/optcd	/dev/hdt8
/dev/sdaa2	/dev/pf1	/dev/nb9	/dev/hdt7
/dev/sdaa15	/dev/pf0	/dev/nb8	/dev/hdt6
/dev/sdaa14	/dev/pdd9	/dev/nb7	/dev/hdt5
/dev/sdaa13	/dev/pdd8	/dev/nb6	/dev/hdt4
/dev/sdaa12	/dev/pdd7	/dev/nb5	/dev/hdt32
/dev/sdaa11	/dev/pdd6	/dev/nb4	/dev/hdt31
/dev/sdaa10	/dev/pdd5	/dev/nb31	/dev/hdt30
/dev/sdaa1	/dev/pdd4	/dev/nb30	/dev/hdt3
/dev/sdaa	/dev/pdd3	/dev/nb3	/dev/hdt29
/dev/sda9	/dev/pdd2	/dev/nb29	/dev/hdt28
/dev/sda8	/dev/pdd15	/dev/nb28	/dev/hdt27
/dev/sda7	/dev/pdd14	/dev/nb27	/dev/hdt26
/dev/sda6	/dev/pdd13	/dev/nb26	/dev/hdt25
/dev/sda5	/dev/pdd12	/dev/nb25	/dev/hdt24
/dev/sda4	/dev/pdd11	/dev/nb24	/dev/hdt23
/dev/sda3	/dev/pdd10	/dev/nb23	/dev/hdt22
/dev/sda2	/dev/pdd1	/dev/nb22	/dev/hdt21
/dev/sda15	/dev/pdd	/dev/nb21	/dev/hdt20
/dev/sda14	/dev/pdc9	/dev/nb20	/dev/hdt2
/dev/sda13	/dev/pdc8	/dev/nb2	/dev/hdt19
/dev/sda12	/dev/pdc7	/dev/nb19	/dev/hdt18
/dev/sda11	/dev/pdc6	/dev/nb18	/dev/hdt17
/dev/sda10	/dev/pdc5	/dev/nb17	/dev/hdt16
/dev/sda1	/dev/pdc4	/dev/nb16	/dev/hdt15
/dev/sda	/dev/pdc3	/dev/nb15	/dev/hdt14
/dev/scd7	/dev/pdc2	/dev/nb14	/dev/hdt13
/dev/scd6	/dev/pdc15	/dev/nb13	/dev/hdt12
/dev/scd5	/dev/pdc14	/dev/nb12	/dev/hdt11
/dev/scd4	/dev/pdc13	/dev/nb11	/dev/hdt10
/dev/scd3	/dev/pdc12	/dev/nb10	/dev/hdt1
/dev/scd2	/dev/pdc11	/dev/nb1	/dev/hdt

/dev/hds9	/dev/hdq4	/dev/hdo29	/dev/hdm24
/dev/hds8	/dev/hdq32	/dev/hdo28	/dev/hdm23
/dev/hds7	/dev/hdq31	/dev/hdo27	/dev/hdm22
/dev/hds6	/dev/hdq30	/dev/hdo26	/dev/hdm21
/dev/hds5	/dev/hdq3	/dev/hdo25	/dev/hdm20
/dev/hds4	/dev/hdq29	/dev/hdo24	/dev/hdm2
/dev/hds32	/dev/hdq28	/dev/hdo23	/dev/hdm19
/dev/hds31	/dev/hdq27	/dev/hdo22	/dev/hdm18
/dev/hds30	/dev/hdq26	/dev/hdo21	/dev/hdm17
/dev/hds3	/dev/hdq25	/dev/hdo20	/dev/hdm16
/dev/hds29	/dev/hdq24	/dev/hdo2	/dev/hdm15
/dev/hds28	/dev/hdq23	/dev/hdo19	/dev/hdm14
/dev/hds27	/dev/hdq22	/dev/hdo18	/dev/hdm13
/dev/hds26	/dev/hdq21	/dev/hdo17	/dev/hdm12
/dev/hds25	/dev/hdq20	/dev/hdo16	/dev/hdm11
/dev/hds24	/dev/hdq2	/dev/hdo15	/dev/hdm10
/dev/hds23	/dev/hdq19	/dev/hdo14	/dev/hdm1
/dev/hds22	/dev/hdq18	/dev/hdo13	/dev/hdm
/dev/hds21	/dev/hdq17	/dev/hdo12	/dev/hdl9
/dev/hds20	/dev/hdq16	/dev/hdo11	/dev/hdl8
/dev/hds2	/dev/hdq15	/dev/hdo10	/dev/hdl7
/dev/hds19	/dev/hdq14	/dev/hdo1	/dev/hdl6
/dev/hds18	/dev/hdq13	/dev/hdo	/dev/hdl5
/dev/hds17	/dev/hdq12	/dev/hdn9	/dev/hdl4
/dev/hds16	/dev/hdq11	/dev/hdn8	/dev/hdl32
/dev/hds15	/dev/hdq10	/dev/hdn7	/dev/hdl31
/dev/hds14	/dev/hdq1	/dev/hdn6	/dev/hdl30
/dev/hds13	/dev/hdq	/dev/hdn5	/dev/hdl3
/dev/hds12	/dev/hdp9	/dev/hdn4	/dev/hdl29
/dev/hds11	/dev/hdp8	/dev/hdn32	/dev/hdl28
/dev/hds10	/dev/hdp7	/dev/hdn31	/dev/hdl27
/dev/hds1	/dev/hdp6	/dev/hdn30	/dev/hdl26
/dev/hds	/dev/hdp5	/dev/hdn3	/dev/hdl25
/dev/hdr9	/dev/hdp4	/dev/hdn29	/dev/hdl24
/dev/hdr8	/dev/hdp32	/dev/hdn28	/dev/hdl23
/dev/hdr7	/dev/hdp31	/dev/hdn27	/dev/hdl22
/dev/hdr6	/dev/hdp30	/dev/hdn26	/dev/hdl21
/dev/hdr5	/dev/hdp3	/dev/hdn25	/dev/hdl20
/dev/hdr4	/dev/hdp29	/dev/hdn24	/dev/hdl2
/dev/hdr32	/dev/hdp28	/dev/hdn23	/dev/hdl19
/dev/hdr31	/dev/hdp27	/dev/hdn22	/dev/hdl18
/dev/hdr30	/dev/hdp26	/dev/hdn21	/dev/hdl17
/dev/hdr3	/dev/hdp25	/dev/hdn20	/dev/hdl16
/dev/hdr29	/dev/hdp24	/dev/hdn2	/dev/hdl15
/dev/hdr28	/dev/hdp23	/dev/hdn19	/dev/hdl14
/dev/hdr27	/dev/hdp22	/dev/hdn18	/dev/hdl13
/dev/hdr26	/dev/hdp21	/dev/hdn17	/dev/hdl12
/dev/hdr25	/dev/hdp20	/dev/hdn16	/dev/hdl11
/dev/hdr24	/dev/hdp2	/dev/hdn15	/dev/hdl10
/dev/hdr23	/dev/hdp19	/dev/hdn14	/dev/hdl1
/dev/hdr22	/dev/hdp18	/dev/hdn13	/dev/hdl
/dev/hdr21	/dev/hdp17	/dev/hdn12	/dev/hdk9
/dev/hdr20	/dev/hdp16	/dev/hdn11	/dev/hdk8
/dev/hdr2	/dev/hdp15	/dev/hdn10	/dev/hdk7
/dev/hdr19	/dev/hdp14	/dev/hdn1	/dev/hdk6
/dev/hdr18	/dev/hdp13	/dev/hdn	/dev/hdk5
/dev/hdr17	/dev/hdp12	/dev/hdm9	/dev/hdk4
/dev/hdr16	/dev/hdp11	/dev/hdm8	/dev/hdk32
/dev/hdr15	/dev/hdp10	/dev/hdm7	/dev/hdk31
/dev/hdr14	/dev/hdp1	/dev/hdm6	/dev/hdk30
/dev/hdr13	/dev/hdp	/dev/hdm5	/dev/hdk3
/dev/hdr12	/dev/hdo9	/dev/hdm4	/dev/hdk29
/dev/hdr11	/dev/hdo8	/dev/hdm32	/dev/hdk28
/dev/hdr10	/dev/hdo7	/dev/hdm31	/dev/hdk27
/dev/hdr1	/dev/hdo6	/dev/hdm30	/dev/hdk26
/dev/hdr	/dev/hdo5	/dev/hdm3	/dev/hdk25
/dev/hdq9	/dev/hdo4	/dev/hdm29	/dev/hdk24
/dev/hdq8	/dev/hdo32	/dev/hdm28	/dev/hdk23
/dev/hdq7	/dev/hdo31	/dev/hdm27	/dev/hdk22
/dev/hdq6	/dev/hdo30	/dev/hdm26	/dev/hdk21
/dev/hdq5	/dev/hdo3	/dev/hdm25	/dev/hdk20

/dev/hdk2	/dev/hdi15	/dev/hdg10	/dev/hdd7
/dev/hdk19	/dev/hdi14	/dev/hdg1	/dev/hdd6
/dev/hdk18	/dev/hdi13	/dev/hdg	/dev/hdd5
/dev/hdk17	/dev/hdi12	/dev/hdf9	/dev/hdd4
/dev/hdk16	/dev/hdi11	/dev/hdf8	/dev/hdd32
/dev/hdk15	/dev/hdi10	/dev/hdf7	/dev/hdd31
/dev/hdk14	/dev/hdi1	/dev/hdf6	/dev/hdd30
/dev/hdk13	/dev/hdi	/dev/hdf5	/dev/hdd3
/dev/hdk12	/dev/hdh9	/dev/hdf4	/dev/hdd29
/dev/hdk11	/dev/hdh8	/dev/hdf32	/dev/hdd28
/dev/hdk10	/dev/hdh7	/dev/hdf31	/dev/hdd27
/dev/hdk1	/dev/hdh6	/dev/hdf30	/dev/hdd26
/dev/hdk	/dev/hdh5	/dev/hdf3	/dev/hdd25
/dev/hdj9	/dev/hdh4	/dev/hdf29	/dev/hdd24
/dev/hdj8	/dev/hdh32	/dev/hdf28	/dev/hdd23
/dev/hdj7	/dev/hdh31	/dev/hdf27	/dev/hdd22
/dev/hdj6	/dev/hdh30	/dev/hdf26	/dev/hdd21
/dev/hdj5	/dev/hdh3	/dev/hdf25	/dev/hdd20
/dev/hdj4	/dev/hdh29	/dev/hdf24	/dev/hdd2
/dev/hdj32	/dev/hdh28	/dev/hdf23	/dev/hdd19
/dev/hdj31	/dev/hdh27	/dev/hdf22	/dev/hdd18
/dev/hdj30	/dev/hdh26	/dev/hdf21	/dev/hdd17
/dev/hdj3	/dev/hdh25	/dev/hdf20	/dev/hdd16
/dev/hdj29	/dev/hdh24	/dev/hdf2	/dev/hdd15
/dev/hdj28	/dev/hdh23	/dev/hdf19	/dev/hdd14
/dev/hdj27	/dev/hdh22	/dev/hdf18	/dev/hdd13
/dev/hdj26	/dev/hdh21	/dev/hdf17	/dev/hdd12
/dev/hdj25	/dev/hdh20	/dev/hdf16	/dev/hdd11
/dev/hdj24	/dev/hdh2	/dev/hdf15	/dev/hdd10
/dev/hdj23	/dev/hdh19	/dev/hdf14	/dev/hdd1
/dev/hdj22	/dev/hdh18	/dev/hdf13	/dev/hdd
/dev/hdj21	/dev/hdh17	/dev/hdf12	/dev/hdc9
/dev/hdj20	/dev/hdh16	/dev/hdf11	/dev/hdc8
/dev/hdj2	/dev/hdh15	/dev/hdf10	/dev/hdc7
/dev/hdj19	/dev/hdh14	/dev/hdf1	/dev/hdc6
/dev/hdj18	/dev/hdh13	/dev/hdf	/dev/hdc5
/dev/hdj17	/dev/hdh12	/dev/hde9	/dev/hdc4
/dev/hdj16	/dev/hdh11	/dev/hde8	/dev/hdc32
/dev/hdj15	/dev/hdh10	/dev/hde7	/dev/hdc31
/dev/hdj14	/dev/hdh1	/dev/hde6	/dev/hdc30
/dev/hdj13	/dev/hdh	/dev/hde5	/dev/hdc3
/dev/hdj12	/dev/hdg9	/dev/hde4	/dev/hdc29
/dev/hdj11	/dev/hdg8	/dev/hde32	/dev/hdc28
/dev/hdj10	/dev/hdg7	/dev/hde31	/dev/hdc27
/dev/hdj1	/dev/hdg6	/dev/hde30	/dev/hdc26
/dev/hdj	/dev/hdg5	/dev/hde3	/dev/hdc25
/dev/hdi9	/dev/hdg4	/dev/hde29	/dev/hdc24
/dev/hdi8	/dev/hdg32	/dev/hde28	/dev/hdc23
/dev/hdi7	/dev/hdg31	/dev/hde27	/dev/hdc22
/dev/hdi6	/dev/hdg30	/dev/hde26	/dev/hdc21
/dev/hdi5	/dev/hdg3	/dev/hde25	/dev/hdc20
/dev/hdi4	/dev/hdg29	/dev/hde24	/dev/hdc2
/dev/hdi32	/dev/hdg28	/dev/hde23	/dev/hdc19
/dev/hdi31	/dev/hdg27	/dev/hde22	/dev/hdc18
/dev/hdi30	/dev/hdg26	/dev/hde21	/dev/hdc17
/dev/hdi3	/dev/hdg25	/dev/hde20	/dev/hdc16
/dev/hdi29	/dev/hdg24	/dev/hde2	/dev/hdc15
/dev/hdi28	/dev/hdg23	/dev/hde19	/dev/hdc14
/dev/hdi27	/dev/hdg22	/dev/hde18	/dev/hdc13
/dev/hdi26	/dev/hdg21	/dev/hde17	/dev/hdc12
/dev/hdi25	/dev/hdg20	/dev/hde16	/dev/hdc11
/dev/hdi24	/dev/hdg2	/dev/hde15	/dev/hdc10
/dev/hdi23	/dev/hdg19	/dev/hde14	/dev/hdc1
/dev/hdi22	/dev/hdg18	/dev/hde13	/dev/hdc
/dev/hdi21	/dev/hdg17	/dev/hde12	/dev/hdb9
/dev/hdi20	/dev/hdg16	/dev/hde11	/dev/hdb8
/dev/hdi2	/dev/hdg15	/dev/hde10	/dev/hdb7
/dev/hdi19	/dev/hdg14	/dev/hde1	/dev/hdb6
/dev/hdi18	/dev/hdg13	/dev/hde	/dev/hdb5
/dev/hdi17	/dev/hdg12	/dev/hdd9	/dev/hdb4
/dev/hdi16	/dev/hdg11	/dev/hdd8	/dev/hdb32

/dev/hdb31	/dev/fd7u1760	/dev/fd5u1040	/dev/fd3h1494
/dev/hdb30	/dev/fd7u1743	/dev/fd5h880	/dev/fd3h1476
/dev/hdb3	/dev/fd7u1722	/dev/fd5h720	/dev/fd3h1440
/dev/hdb29	/dev/fd7u1680	/dev/fd5h420	/dev/fd3h1200
/dev/hdb28	/dev/fd7u1660	/dev/fd5h410	/dev/fd3d360
/dev/hdb27	/dev/fd7u1440	/dev/fd5h360	/dev/fd3H720
/dev/hdb26	/dev/fd7u1120	/dev/fd5h1660	/dev/fd3H360
/dev/hdb25	/dev/fd7u1040	/dev/fd5h1494	/dev/fd3H1440
/dev/hdb24	/dev/fd7h880	/dev/fd5h1476	/dev/fd3D720
/dev/hdb23	/dev/fd7h720	/dev/fd5h1440	/dev/fd3D360
/dev/hdb22	/dev/fd7h420	/dev/fd5h1200	/dev/fd3CompaQ
/dev/hdb21	/dev/fd7h410	/dev/fd5d360	/dev/fd3
/dev/hdb20	/dev/fd7h360	/dev/fd5CompaQ	/dev/fd2u830
/dev/hdb2	/dev/fd7h1660	/dev/fd5	/dev/fd2u820
/dev/hdb19	/dev/fd7h1494	/dev/fd4u830	/dev/fd2u800
/dev/hdb18	/dev/fd7h1476	/dev/fd4u820	/dev/fd2u720
/dev/hdb17	/dev/fd7h1440	/dev/fd4u800	/dev/fd2u3840
/dev/hdb16	/dev/fd7h1200	/dev/fd4u720	/dev/fd2u360
/dev/hdb15	/dev/fd7d360	/dev/fd4u3840	/dev/fd2u3520
/dev/hdb14	/dev/fd7CompaQ	/dev/fd4u360	/dev/fd2u3200
/dev/hdb13	/dev/fd7	/dev/fd4u3520	/dev/fd2u2880
/dev/hdb12	/dev/fd6u830	/dev/fd4u3200	/dev/fd2u1920
/dev/hdb11	/dev/fd6u820	/dev/fd4u2880	/dev/fd2u1840
/dev/hdb10	/dev/fd6u800	/dev/fd4u1920	/dev/fd2u1760
/dev/hdb1	/dev/fd6u720	/dev/fd4u1840	/dev/fd2u1743
/dev/hdb	/dev/fd6u3840	/dev/fd4u1760	/dev/fd2u1722
/dev/hda9	/dev/fd6u360	/dev/fd4u1743	/dev/fd2u1680
/dev/hda8	/dev/fd6u3520	/dev/fd4u1722	/dev/fd2u1660
/dev/hda7	/dev/fd6u3200	/dev/fd4u1680	/dev/fd2u1440
/dev/hda6	/dev/fd6u2880	/dev/fd4u1660	/dev/fd2u1120
/dev/hda5	/dev/fd6u1920	/dev/fd4u1440	/dev/fd2u1040
/dev/hda4	/dev/fd6u1840	/dev/fd4u1120	/dev/fd2h880
/dev/hda32	/dev/fd6u1760	/dev/fd4u1040	/dev/fd2h720
/dev/hda31	/dev/fd6u1743	/dev/fd4h880	/dev/fd2h420
/dev/hda30	/dev/fd6u1722	/dev/fd4h720	/dev/fd2h410
/dev/hda3	/dev/fd6u1680	/dev/fd4h420	/dev/fd2h360
/dev/hda29	/dev/fd6u1660	/dev/fd4h410	/dev/fd2h1660
/dev/hda28	/dev/fd6u1440	/dev/fd4h360	/dev/fd2h1494
/dev/hda27	/dev/fd6u1120	/dev/fd4h1660	/dev/fd2h1476
/dev/hda26	/dev/fd6u1040	/dev/fd4h1494	/dev/fd2h1440
/dev/hda25	/dev/fd6h880	/dev/fd4h1476	/dev/fd2h1200
/dev/hda24	/dev/fd6h720	/dev/fd4h1440	/dev/fd2d360
/dev/hda23	/dev/fd6h420	/dev/fd4h1200	/dev/fd2H720
/dev/hda22	/dev/fd6h410	/dev/fd4d360	/dev/fd2H360
/dev/hda21	/dev/fd6h360	/dev/fd4CompaQ	/dev/fd2H1440
/dev/hda20	/dev/fd6h1660	/dev/fd4	/dev/fd2D720
/dev/hda2	/dev/fd6h1494	/dev/fd3u830	/dev/fd2D360
/dev/hda19	/dev/fd6h1476	/dev/fd3u820	/dev/fd2CompaQ
/dev/hda18	/dev/fd6h1440	/dev/fd3u800	/dev/fd2
/dev/hda17	/dev/fd6h1200	/dev/fd3u720	/dev/fdlu830
/dev/hda16	/dev/fd6d360	/dev/fd3u3840	/dev/fdlu820
/dev/hda15	/dev/fd6CompaQ	/dev/fd3u360	/dev/fdlu800
/dev/hda14	/dev/fd6	/dev/fd3u3520	/dev/fdlu720
/dev/hda13	/dev/fd5u830	/dev/fd3u3200	/dev/fdlu3840
/dev/hda12	/dev/fd5u820	/dev/fd3u2880	/dev/fdlu360
/dev/hda11	/dev/fd5u800	/dev/fd3u1920	/dev/fdlu3520
/dev/hda10	/dev/fd5u720	/dev/fd3u1840	/dev/fdlu3200
/dev/hda1	/dev/fd5u3840	/dev/fd3u1760	/dev/fdlu2880
/dev/hda	/dev/fd5u360	/dev/fd3u1743	/dev/fdlu1920
/dev/gscd	/dev/fd5u3520	/dev/fd3u1722	/dev/fdlu1840
/dev/fd7u830	/dev/fd5u3200	/dev/fd3u1680	/dev/fdlu1760
/dev/fd7u820	/dev/fd5u2880	/dev/fd3u1660	/dev/fdlu1743
/dev/fd7u800	/dev/fd5u1920	/dev/fd3u1440	/dev/fdlu1722
/dev/fd7u720	/dev/fd5u1840	/dev/fd3u1120	/dev/fdlu1680
/dev/fd7u3840	/dev/fd5u1760	/dev/fd3u1040	/dev/fdlu1660
/dev/fd7u360	/dev/fd5u1743	/dev/fd3h880	/dev/fdlu1440
/dev/fd7u3520	/dev/fd5u1722	/dev/fd3h720	/dev/fdlu1120
/dev/fd7u3200	/dev/fd5u1680	/dev/fd3h420	/dev/fdlu1040
/dev/fd7u2880	/dev/fd5u1660	/dev/fd3h410	/dev/fdlh880
/dev/fd7u1920	/dev/fd5u1440	/dev/fd3h360	/dev/fdlh720
/dev/fd7u1840	/dev/fd5u1120	/dev/fd3h1660	/dev/fdlh420

/dev/fdlh410	/dev/md1	Invalid request	Transport endpoint is
/dev/fdlh360	/dev/md0	descriptor	already connected
/dev/fdlh1660	/dev/md10	Invalid exchange	Software caused
/dev/fdlh1494	/dev/null	Level 2 halted	connection abort
/dev/fdlh1476	Wrong medium type	No CSI structure	Network dropped
/dev/fdlh1440	No medium found	available	connection on reset
/dev/fdlh1200	Disk quota exceeded	Protocol driver not	Cannot assign requested
/dev/fdlD360	Remote I/O error	attached	address
/dev/fdlH720	Is a named type file	Link number out of	Address family not
/dev/fdlH360	No XENIX semaphores	range	supported by protocol
/dev/fdlH1440	available	Level 3 reset	Protocol wrong type for
/dev/fdlD720	Not a XENIX named type	Level 3 halted	socket
/dev/fdlD360	file	Level 2 not	Socket operation on
/dev/fdlCompaQ	Structure needs	synchronized	non-socket
/dev/fdl	cleaning	Channel number out of	Interrupted system call
/dev/fd0u830	Stale NFS file handle	range	should be restarted
/dev/fd0u820	Operation now in	Identifier removed	Invalid or incomplete
/dev/fd0u800	progress	No message of desired	multibyte or wide
/dev/fd0u720	Operation already in	type	character
/dev/fd0u3840	progress	Directory not empty	Cannot exec a shared
/dev/fd0u360	No route to host	Function not	library directly
/dev/fd0u3520	Host is down	implemented	Attempting to link in
/dev/fd0u3200	Connection refused	No locks available	too many shared
/dev/fd0u2880	Connection timed out	File name too long	libraries
/dev/fd0u1920	No buffer space	Resource deadlock	.lib section in a.out
/dev/fd0u1840	available	avoided	corrupted
/dev/fd0u1760	Connection reset by	Numerical result out of	Accessing a corrupted
/dev/fd0u1743	peer	range	shared library
/dev/fd0u1722	Network is unreachable	Broken pipe	Can not access a needed
/dev/fd0u1680	Network is down	Too many links	shared library
/dev/fd0u1660	Address already in use	Read-only file system	Value too large for
/dev/fd0u1440	Protocol family not	Illegal seek	defined data type
/dev/fd0u1120	supported	No space left on device	Too many levels of
/dev/fd0u1040	Operation not supported	File too large	symbolic links
/dev/fd0h880	Socket type not	Text file busy	Numerical argument out
/dev/fd0h720	supported	Too many open files	of domain
/dev/fd0h420	Protocol not supported	Too many open files in	Inappropriate ioctl for
/dev/fd0h410	Protocol not available	system	device
/dev/fd0h360	Message too long	Invalid argument	Resource temporarily
/dev/fd0h1660	Destination address	Is a directory	unavailable
/dev/fd0h1494	required	Not a directory	,ccs=
/dev/fd0h1476	Too many users	No such device	TOP_PAD_
/dev/fd0h1440	Streams pipe error	Invalid cross-device	MMAP_MAX_
/dev/fd0D360	Remote address changed	link	TRIM_THRESHOLD_
/dev/fd0H720	File descriptor in bad	File exists	MMAP_THRESHOLD_
/dev/fd0H360	state	Device or resource busy	Arena %d:
/dev/fd0h1200	Name not unique on	Block device required	system bytes = %10u
/dev/fd0D720	network	Bad address	in use bytes = %10u
/dev/fd0D360	Bad message	Permission denied	Total (incl. mmap):
/dev/fd0H1440	RFS specific error	Cannot allocate memory	max mmap regions = %10u
/dev/fd0	Multihop attempted	No child processes	max mmap bytes =
/dev/fd0CompaQ	Protocol error	Bad file descriptor	%10lu
/dev/cm206cd	Communication error on	Exec format error	malloc: top chunk is
/dev/cm205cd	send	Argument list too long	corrupt
/dev/cdu535	Srmount error	No such device or	free(): invalid pointer
/dev/cdu31a	Advertise error	address	%p!
/dev/bpcd	Link has been severed	Input/output error	malloc: using debugging
/dev/aztcd	Object is remote	Interrupted system call	hooks
/dev/md15	Package not installed	No such process	realloc(): invalid
/dev/md14	Machine is not on the	No such file or	pointer %p!
/dev/md13	network	directory	Unknown error
/dev/md12	Out of streams	Operation not permitted	ANSI_X3.4-
/dev/md11	resources	Success	1968//TRANSLIT
/dev/md9	Timer expired	Too many references:	syslog: unknown
/dev/md8	No data available	cannot splice	facility/priority: %x
/dev/md7	Device not a stream	Cannot send after	out of memory [
/dev/md6	Bad font file format	transport endpoint	<%d>
/dev/md5	Invalid slot	shutdown	%h %e %T
/dev/md4	Invalid request code	Transport endpoint is	[%d]
/dev/md3	No anode	not connected	/dev/console
/dev/md2	Exchange full		/dev/log

apic	ISO-IR-6// ANSI_X3.4-	messages	ELF file ABI version
mtrr	1968//	/usr/share/locale	invalid
cmov	ANSI_X3.4// ANSI_X3.4-	POSIX	internal error
pse36	1968//	LC_COLLATE	trying file=%s
clflush	OSF00010102// ISO-	LC_CTYPE	file=%s; needed by %s
acpi	10646/UCS2//	LC_MONETARY	find library=%s;
fxsr	OSF00010101// ISO-	LC_NUMERIC	searching
sse2	10646/UCS2//	LC_TIME	RPATH
ia64	OSF00010100// ISO-	LC_MESSAGES	RUNPATH
amd3d	10646/UCS2//	LC_ALL	cannot create cache for
i386	UCS-2// ISO-10646/UCS2//	LC_XXX	search path
i486	UCS2// ISO-10646/UCS2//	LANGUAGE	cannot create
i586	OSF05010001// ISO-	charset=	RUNPATH/RPATH copy
i686	10646/UTF8//	OUTPUT_CHARSET	cannot create search
LD_AOUT_LIBRARY_PATH	ISO-IR-193// ISO-	/usr/share/locale	path array
LD_AOUT_PRELOAD	10646/UTF8//	/locale.alias	file=%s; generating
LD_PRELOAD	UTF-8// ISO-10646/UTF8//	parse error	link map
LD_LIBRARY_PATH	UTF8// ISO-10646/UTF8//	parser stack overflow	cannot create shared
LD_ORIGIN_PATH	WCHAR_T// INTERNAL	plural=	object descriptor
LD_DEBUG_OUTPUT	OSF00010106// ISO-	plurals=	ELF load command
LD_PROFILE	10646/UCS4//	0123456789abcdefghijklm	alignment not page-
GCONV_PATH	OSF00010105// ISO-	nopqrstuvwxyz	aligned
HOSTALIASES	10646/UCS4//	(null)	ELF load command
LOCALDOMAIN	OSF00010104// ISO-	(nil)	address/offset not
LOCPATH	10646/UCS4//		properly aligned
MALLOC_TRACE	ISO-10646// ISO-	0000000000000000	failed to map segment
NLSPATH	10646/UCS4//	%m/%d/%y	from shared object
RESOLV_HOST_CONF	CSUCS4// ISO-	%Y-%m-%d	cannot dynamically load
RES_OPTIONS	10646/UCS4//	%H:%M	executable
TMPDIR	UCS-4BE// ISO-	%I:%M:%S %p	cannot change memory
TZDIR	10646/UCS4//	%H:%M:%S	protections
LD_WARN	UCS-4// ISO-10646/UCS4//	/etc/localtime	cannot allocate memory
LD_LIBRARY_PATH	alias	Universal	for program header
LD_BIND_NOW	module	%[^0-9,+]	object file has no
LD_BIND_NOT	UNICODELITTLE// ISO-	%hu:%hu:%hu	dynamic section
LD_DYNAMIC_WEAK	10646/UCS2//	M%hu.%hu.%hu%n	dynamic: 0x%0*lx
/etc/suid-debug	OSF00010020//	/usr/share/zoneinfo	base: 0x%0*lx size:
MALLOC_CHECK_	ANSI_X3.4-1968//	TZDIR	0x%0*Zx
/proc/sys/kernel/osrele	ISO_646.IRV:1991//	posixrules	entry: 0x%0*lx
ase	ANSI_X3.4-1968//	/proc/self/cwd	phdr: 0x%0*lx phnum:
FATAL: kernel too old	ANSI_X3.4-1986//	/proc	%*u
FATAL: cannot determine	ANSI_X3.4-1968//	/etc/mtab	shared object cannot be
library version	ISO-10646/UTF-8/ ISO-	/etc/fstab	dlopen()ed
/usr/lib/gconv	10646/UTF8//	proc	ELF file data encoding
gconv-modules	10646-1:1993/UCS4/ ISO-	/cpuinfo	not big-endian
=INTERNAL->ucs2reverse	10646/UCS4//	processor	ELF file data encoding
=ucs2reverse->INTERNAL	10646-1:1993// ISO-	/meminfo	not little-endian
=INTERNAL->ascii	10646/UCS4//	MemTotal: %ld kB	ELF file version ident
=ascii->INTERNAL	GCONV_PATH	MemFree: %ld kB	does not match current
=INTERNAL->ucs2	/usr/lib/gconv/gconv-	/lib/	one
=ucs2->INTERNAL	modules.cache	/usr/lib/	ELF file version does
=utf8->INTERNAL	gconv	ORIGIN	not match current one
=INTERNAL->utf8	gconv_init	PLATFORM	ELF file's phentsize
=ucs4le->INTERNAL	gconv_end	cannot allocate name	not the expected size
=INTERNAL->ucs4le	toupper	record	only ET_DYN and ET_EXEC
UCS-4LE//	tolower	system search path	can be loaded
=ucs4->INTERNAL	upper	cannot stat shared	cannot open shared
=INTERNAL->ucs4	lower	object	object file
UCS-2BE// UNICODEBIG//	alpha	cannot read file data	AT_HWCAP:
UCS-2LE// ISO-	digit	cannot map zero-fill	/etc/ld.so.cache
10646/UCS2//	xdigit	pages	search cache=%s
CSASCII// ANSI_X3.4-	space	cannot create	ld.so-1.7.0
1968//	print	searchlist	glibc-ld.so.cache1.1
CP367// ANSI_X3.4-	graph	search path=	undefined symbol:
1968//	blank	(%s from	symbol=%s; lookup in
IBM367// ANSI_X3.4-	cntrl	file %s)	file=%s
1968//	punct	(%s)	file=%s; needed by %s
US-ASCII// ANSI_X3.4-	alnum	file too short	(relocation dependency)
1968//	libc	invalid ELF header	binding file %s to %s:
ISO646-US// ANSI_X3.4-	POSIX	ELF file OS ABI invalid	%s symbol '%s'
1968//	ANSI_X3.4-1968		relocation error

```

<main program>          i18n:1999          invalid mode for
symbol                  i18n:1999          dlopen()
, version               i18n:1999          DST not allowed in
  not defined in file   1997-12-20        SUID/SGID programs
  with link time        +45 3325-6543      empty dynamic string
reference               +45 3122-6543      token substitution
  (no version symbols) keld@dkuug.dk      opening file=%s;
protected              Keld Simonsen      opencount == %u
normal                  ISO/IEC 14652 i18n  shared object not open
 [%s]                   FDCC-set           calling fini: %s
out of memory           C/o Keld Simonsen, Skt. closing file=%s;
DYNAMIC LINKER BUG!!!  Jorgens Alle 8, DK-1615 opencount == %u
<program name unknown> Kobenhavn V        (lazy)
%s: %s: %s%s%s%s      ISO/IEC JTC1/SC22/WG20 relocation processing:
error while loading    - internationalization %s%s
shared libraries       !"#%&'()*+,-      cannot make segment
/proc/self/exe         ./0123456789:;<=>?@ABCD writable for relocation
IGNORE                 EFGHIJKLMNOPQRSTUVWXYZ [%s: Symbol '%s' has
gconv_trans_context   \]^_`abcdefghijklmnopqr different size in
gconv_trans            stuvwxyz{|}~        shared object, consider
gconv_trans_init       [Am-                re-linking
gconv_trans_end        kpnJ                %s: profiler found no
LC_IDENTIFICATION     uD;s                PLTREL in object %s
LC_MEASUREMENT         )r+[                %s: profiler out of
LC_TELEPHONE           [!|n                memory shadowing PLTREL
LC_ADDRESS              uYD?e                of %s
LC_NAME                 I9C-                cannot restore segment
LC_PAPER                I!G.                prot after reloc
LOCPATH                 U^h6LU3            unexpected reloc type
/usr/lib/locale         U.y`                0x
LANG                    3?Cy                unexpected PLT reloc
/SYS_                   ' Djz                type 0x
^[nN_                  $po?b                empty dynamics string
^[yY]                   w);u                token substitution
%a %b %e %H:%M:%S %Z %Y =t%j                cannot load auxiliary
%a %b %e %H:%M:%S %Y   MP0!                '%s' because of empty
December               t0tv                dynamic string token
November               =u8Q)+              substitution
October                *~xx                load auxiliary
September              ~j2=                object=%s requested by
August                 |;#o                file=%s
July                   Ac+;                load filtered object=%s
June                   ^2XX%               requested by file=%s
April                  !{>;b              cannot allocate
March                  dI@B                dependency list
February               2I%%                cannot allocate symbol
January                {fG5                search list
Saturday               0123456789ABCDEFHGHIJKLM Filters not supported
Friday                 NOPQRSTUVWXYZ        with
Thursday               %d %d                LD_TRACE_PRELINKING
Wednesday              %s %s %s %s %d %d  calling init: %s
Tuesday                gmon                calling preinit: %s
Monday                 seconds              checking for version
Sunday                 .profile            '%s' in file %s
%p%t%g%t%t%t%f       %s: cannot open file: required by file %s
%a%N%f%N%d%N%b%N%s %h %s                no version information
%e %r%N%C-%z %T%N%c%N %s: cannot stat file: available (required by
+c %a %l              %s                cannot allocate version
i18n:1999              %s: cannot create file: reference table
i18n:1999              %s                unsupported version
i18n:1999              %s: cannot map file: %s of Verdef record
i18n:1999              %s: file is no correct weak version `
i18n:1999              profile data file for ' not found (required
i18n:1999              `s'                by
i18n:1999              Out of memory while of Verneed record
i18n:1999              initializing profiler inity
i18n:1999              cannot extend global
i18n:1999              scope
i18n:1999              dlopen
i18n:1999              cannot create scope
i18n:1999              list

```


Appendix B. Microsoft Windows Update listing of installed patches on test host.

Successful	Wednesday, December 10, 2003	Security Update for Windows XP (KB810217)
Successful	Monday, November 17, 2003	Security Update for Microsoft Windows XP (KB828035)
Successful	Wednesday, November 12, 2003	Cumulative Security Update for Internet Explorer 6 SP1 (KB824145)
Successful	Thursday, October 16, 2003	814078: Security Update (Microsoft Jscript version 5.6, Windows 2000, Windows XP)
Successful	Thursday, October 16, 2003	Security Update for Microsoft Windows (KB824141)
Successful	Thursday, October 16, 2003	Security Update for Microsoft Windows (KB823182)
Successful	Thursday, October 16, 2003	Security Update for Microsoft Windows XP (KB825119)
Successful	Thursday, October 16, 2003	Security Update for Microsoft Windows XP (KB828035)
Successful	Monday, October 06, 2003	Security Update for Windows Media Player (KB828026)
Successful	Monday, October 06, 2003	October 2003, Cumulative Patch for Internet Explorer 6 Service Pack 1 (KB828750)
Successful	Wednesday, September 24, 2003	Microsoft Windows Journal Viewer (Windows XP) Read more...
Successful	Wednesday, September 24, 2003	Q282010: Recommended Update for Microsoft Jet 4.0 Service Pack 7 (SP7) - Windows XP
Successful	Wednesday, September 24, 2003	Q810243 Update: Watch television shows recorded by Media Center PCs on other Microsoft Windows XP PCs
Successful	Wednesday, September 24, 2003	Q322011: Recommended Update Read more...
Successful	Wednesday, September 24, 2003	814995: Recommended Update
Successful	Wednesday, September 24, 2003	Recommended Update for Windows XP SP1 (817778)
Successful	Wednesday, September 24, 2003	820291: Recommended Update (Windows XP)
Successful	Wednesday, September 24, 2003	Recommended Update for Windows XP SP1 (KB822603)
Successful	Thursday, September 11, 2003	Security Update for Windows XP (KB824146)
Successful	Thursday, September 04, 2003	Security Update for Microsoft Windows (KB824105)

Successful	Thursday, August 21, 2003	Security Update for Microsoft Data Access Components (823718)
Successful	Thursday, August 21, 2003	August 2003, Cumulative Patch for Internet Explorer 6 Service Pack 1 (822925)
Successful	Wednesday, August 06, 2003	Flaw In Windows Media Player May Allow Media Library Access (819639)
Successful	Thursday, July 24, 2003	Security Update for Windows XP (819696)
Successful	Friday, July 18, 2003	Windows Error Reporting: Recommended Update (Windows XP)
Successful	Friday, July 18, 2003	815485: Recommended Update
Successful	Wednesday, July 16, 2003	821557: Security Update (Windows XP)
Successful	Wednesday, July 16, 2003	Security Update for Windows XP (823980)
Successful	Thursday, July 10, 2003	817606: Security Update (Windows XP)
Successful	Thursday, July 10, 2003	823559: Security Update for Microsoft Windows
Successful	Monday, June 09, 2003	818529: June 2003, Cumulative Patch for Internet Explorer 6 Service Pack 1
Successful	Friday, May 30, 2003	811493: Security Update (Windows XP)
Successful	Wednesday, May 28, 2003	Q815021 XP: Security Update
Successful	Tuesday, May 13, 2003	817787: Security Update Windows Media Player for XP
Successful	Thursday, April 24, 2003	330994: April 2003, Security Update for Outlook Express 6 SP1
Successful	Thursday, April 24, 2003	813489: April 2003, Cumulative Patch for Internet Explorer 6 Service Pack 1
Successful	Thursday, April 17, 2003	811493: Security Update (Windows XP)
Successful	Thursday, April 17, 2003	Q817287: Critical Update (Catalog Database Corruption in Microsoft Windows)
Successful	Thursday, April 10, 2003	816093: Security Update Microsoft Virtual Machine (Microsoft VM)
Successful	Tuesday, April 01, 2003	813951: Update for Internet Explorer 6 SP1
Successful	Monday, March 31, 2003	329170: Security Update
Successful	Saturday, March 29, 2003	811630: Critical Update (Windows XP) Read more...
Successful	Saturday, March 29, 2003	Windows MovieMaker 2 Read more...

Successful	Saturday, March 29, 2003	331953: Security Update (Windows XP)
Successful	Saturday, March 29, 2003	329170: Security Update
Successful	Saturday, March 29, 2003	814078: Security Update (Microsoft Jscript version 5.6, Windows 2000, Windows XP)
Successful	Saturday, March 29, 2003	810577: Security Update
Successful	Saturday, March 29, 2003	810833: Security Update (Windows XP)
Successful	Saturday, March 29, 2003	814033: Critical Update
Successful	Saturday, March 29, 2003	Q329441: Critical Update
Successful	Saturday, March 29, 2003	810847: February 2003, Cumulative Patch for Internet Explorer 6 Service Pack 1
Successful	Thursday, January 02, 2003	Q327405: Recommended Update (Windows XP Professional) Read more...
Successful	Thursday, January 02, 2003	327979: Recommended Update Read more...
Successful	Thursday, January 02, 2003	Q329048: Security Update Read more...
Successful	Thursday, January 02, 2003	Q323255: Security Update (Windows XP) Read more...
Successful	Thursday, January 02, 2003	Q329115: Security Update (Windows XP) Read more...
Successful	Thursday, January 02, 2003	Q329834: Security Update (Windows XP) Read more...
Successful	Thursday, January 02, 2003	810565: Critical Update Read more...
Successful	Thursday, January 02, 2003	Q329390: Security Update Read more...
Successful	Thursday, January 02, 2003	Q328310: Security Update (Windows XP) Read more...
Successful	Thursday, January 02, 2003	810030: Microsoft VM Security Update Read more...
Successful	Thursday, January 02, 2003	Q324929: December 2002, Cumulative Patch for Internet Explorer 6 Service Pack 1 Read more...

Successful	Monday, October 28, 2002	Windows XP Service Pack 1 (Express) Read more...
Successful	Monday, October 14, 2002	Largan Image Driver Version 1.0.0.0
Successful	Monday, October 14, 2002	Largan Image Driver Version 1.0.0.0
Successful	Monday, October 14, 2002	Q328676: Security Update (Outlook Express 6) Read more...
Successful	Thursday, October 03, 2002	Q329048: Security Update Read more...
Successful	Thursday, October 03, 2002	Q323255: Security Update (Windows XP) Read more...
Successful	Thursday, October 03, 2002	Q324380: Security Update (Windows XP) Read more...
Successful	Thursday, October 03, 2002	Q323172: Security Update (Windows XP) Read more...
Successful	Thursday, October 03, 2002	Q324096: Security Update (Windows XP) Read more...
Successful	Thursday, October 03, 2002	Q329077: Security Update Read more...
Successful	Thursday, August 29, 2002	Computer Fails When Connected To A UPS Read more...
Successful	Thursday, August 29, 2002	Windows Messenger Audio Update Read more...
Successful	Thursday, August 29, 2002	Resuming From Standby Update Read more...
Successful	Thursday, August 29, 2002	Q318966: Recommended Update Read more...
Successful	Thursday, August 29, 2002	Windows Automatic Updating, June 2002 Read more...
Successful	Thursday, August 29, 2002	Q320552: Recommended Update Read more...
Successful	Thursday, August 29, 2002	Q320678: Recommended Update Read more...
Successful	Thursday, August 29, 2002	Windows Movie Maker 1.2.1 Read more...

Successful	Sunday, August 25, 2002	Q313450: Security Update Read more...
Successful	Sunday, August 25, 2002	Q326830: Security Update (Windows XP) Read more...
Successful	Sunday, August 25, 2002	Q323759: August, 2002 Cumulative Patch for Internet Explorer 6 (Windows XP) Read more...
Successful	Saturday, July 27, 2002	Q320174: Recommended Update Read more...
Successful	Saturday, July 27, 2002	Q318138: Security Update (Windows XP) Read more...
Successful	Saturday, July 27, 2002	Q320920: Security Update (Windows Media Player for Windows XP) Read more...
Successful	Sunday, June 23, 2002	Ess Technology, Inc. Media Driver Version 5.12.01.1154; Released on September 28 2001.
Successful	Sunday, June 23, 2002	Ess Technology, Inc. Media Driver Version 5.12.01.1171
Successful	Sunday, June 23, 2002	Accton Net Driver Version 1.12.1016.2001
Successful	Sunday, June 23, 2002	Windows Movie Maker v1.2 Read more...
Successful	Sunday, June 23, 2002	Q321232: Security Update (Internet Explorer 6) Read more...
Successful	Sunday, April 21, 2002	Remote Assistance Connection
Successful	Sunday, April 21, 2002	Windows XP Application Compatibility Update, April 2002
Successful	Sunday, April 21, 2002	Q311967: Security Update
Successful	Sunday, March 31, 2002	Windows XP Update Package, October 25, 2001
Successful	Sunday, March 31, 2002	Critical Update, November 19, 2001
Successful	Sunday, March 31, 2002	Windows XP Application Compatibility Update
Successful	Sunday, March 31, 2002	Critical Update, February 9, 2002
Successful	Sunday, March 31, 2002	Critical Update, February 10, 2002
Successful	Sunday, March 31, 2002	Security Update, February 12, 2002

Successful	Sunday, March 31, 2002	Security Update, February 13, 2002 (MSXML 2.6 and 3.0)
Successful	Sunday, March 31, 2002	System Recovered Error Message Update
Successful	Sunday, March 31, 2002	Windows Messenger 4.6 Connectivity Update
Successful	Sunday, March 31, 2002	Security Update, February 14, 2002 (Internet Explorer 6)
Successful	Sunday, March 31, 2002	Security Update, March 4, 2002
Successful	Sunday, March 31, 2002	Security Update, March 28, 2002 (Internet Explorer 6)

© SANS Institute 2004, Author retains full rights.

Upcoming SANS Forensics Training

CLICK HERE TO
REGISTER NOW!

SANS New York City Winter 2018	New York, NY	Feb 26, 2018 - Mar 03, 2018	Live Event
SANS London March 2018	London, United Kingdom	Mar 05, 2018 - Mar 10, 2018	Live Event
SANS Secure Singapore 2018	Singapore, Singapore	Mar 12, 2018 - Mar 24, 2018	Live Event
SANS Paris March 2018	Paris, France	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS San Francisco Spring 2018	San Francisco, CA	Mar 12, 2018 - Mar 17, 2018	Live Event
Mentor Session - FOR500	Minneapolis, MN	Mar 13, 2018 - May 01, 2018	Mentor
SANS Northern VA Spring - Tysons 2018	McLean, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Pen Test Austin 2018	Austin, TX	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Secure Canberra 2018	Canberra, Australia	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Munich March 2018	Munich, Germany	Mar 19, 2018 - Mar 24, 2018	Live Event
Mentor Session - FOR610	Milwaukee, WI	Mar 21, 2018 - May 02, 2018	Mentor
SANS Boston Spring 2018	Boston, MA	Mar 25, 2018 - Mar 30, 2018	Live Event
Community SANS Columbia FOR610	Columbia, MD	Mar 26, 2018 - Mar 31, 2018	Community SANS
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Abu Dhabi 2018	Abu Dhabi, United Arab Emirates	Apr 07, 2018 - Apr 12, 2018	Live Event
Community SANS Virginia Beach FOR508 @ SLAIT	Virginia Beach, VA	Apr 09, 2018 - Apr 14, 2018	Community SANS
SANS vLive - FOR578: Cyber Threat Intelligence	FOR578 - 201804,	Apr 10, 2018 - May 17, 2018	vLive
SANS London April 2018	London, United Kingdom	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Zurich 2018	Zurich, Switzerland	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
SANS Seattle Spring 2018	Seattle, WA	Apr 23, 2018 - Apr 28, 2018	Live Event
SANS vLive - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	FOR508 - 201804,	Apr 23, 2018 - May 30, 2018	vLive
SANS Riyadh April 2018	Riyadh, Saudi Arabia	Apr 28, 2018 - May 03, 2018	Live Event
Automotive Cybersecurity Summit & Training 2018	Chicago, IL	May 01, 2018 - May 08, 2018	Live Event
SANS vLive - FOR500: Windows Forensic Analysis	FOR500 - 201805,	May 08, 2018 - Jun 14, 2018	vLive
Security West 2018 - FOR578: Cyber Threat Intelligence	San Diego, CA	May 11, 2018 - May 15, 2018	vLive
Security West 2018 - FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
Security West 2018 - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
Security West 2018 - FOR500: Windows Forensic Analysis	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
Security West 2018 - FOR572: Advanced Network Forensics and Analysis	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
Security West 2018 - FOR518: Mac Forensic Analysis	San Diego, CA	May 11, 2018 - May 16, 2018	vLive