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

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.

Part 1: Analyze an Unknown Binary

SANS Case Detail	
Tag	fl-160703-jp1
Media	3.5 inch TDK floppy disk
MD5 Hash	4b680767a2aed974cec5fbcbf84cc97a
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
4b680767a2aed974cec5fbcbf84cc97a  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 *strings* against *prog* 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 Google[9] 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 pages[10]:

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

The source code[11] 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 utility[13]. 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 *prog* 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
/dev/console
/dev/log
/etc/suid-debug
/proc/sys/kernel/osrelease
/usr/lib/gconv
/usr/lib/gconv/gconv-modules.cache
/usr/share/locale
/usr/share/locale
/locale.alias
/etc/localtime
/usr/share/zoneinfo
/proc/self/cwd
/proc
/etc/mtab
/etc/fstab
/cpuinfo
/meminfo
/lib/
/usr/lib/
/etc/ld.so.cache
/proc/self/exe
/usr/lib/locale
/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           ioctl
 21.54    0.005392        2696         2           open
 19.84    0.004965        1655         3           fcntl64
 14.83    0.003711        3711         1           read
  9.17    0.002294         459         5           write
  2.63    0.000659         330         2           close
  1.43    0.000358         119         3           lstat64
  1.06    0.000265          53         5           brk
  0.27    0.000068          68         1           _llseek
  0.24    0.000059          59         1           uname
  0.16    0.000040          40         1           geteuid32
  0.14    0.000036          36         1           getuid32
  0.14    0.000036          36         1           getegid32
  0.14    0.000035          35         1           getgid32
------ ----------- ----------- --------- --------- ----------------
100.00    0.025027                    37           total
```

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

[17] http://lcamtuf.coredump.cx/fenris

```
[root@localhost tmp]# strace -ff ./prog -s dd.tmp
execve("./prog", ["./prog", "-s", "dd.tmp"], [/* 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("dd.tmp", {st_mode=S_IFREG|0755, st_size=30772, ...}) = 0
open("dd.tmp", O_RDONLY|O_LARGEFILE)    = 3
ioctl(3, FIGETBSZ, 0xbffff4c4)          = 0
lstat64("dd.tmp", {st_mode=S_IFREG|0755, st_size=30772, ...}) = 0
lstat64("/dev/sda2", {st_mode=S_IFBLK|0660, st_rdev=makedev(8, 2), ...}) = 0
open("/dev/sda2", O_RDONLY|O_LARGEFILE) = 4
ioctl(3, FIGETBSZ, 0xbffff434)          = 0
brk(0x80c2000)                          = 0x80c2000
ioctl(3, FIBMAP, 0xbffff4c4)            = 0
ioctl(3, FIBMAP, 0xbffff4c4)            = 0
ioctl(3, FIBMAP, 0xbffff4c4)            = 0
ioctl(3, FIBMAP, 0xbffff4c4)            = 0
ioctl(3, FIBMAP, 0xbffff4c4)            = 0
ioctl(3, FIBMAP, 0xbffff4c4)            = 0
ioctl(3, FIBMAP, 0xbffff4c4)            = 0
ioctl(3, FIBMAP, 0xbffff4c4)            = 0
write(2, "getting from block 437672\n", 26) = 26
write(2, "file size was: 30772\n", 21)  = 21
write(2, "slack size: 1996\n", 17)      = 17
write(2, "block size: 4096\n", 17)      = 17
_llseek(4, 1792706612, [1792706612], SEEK_SET) = 0
read(4, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 1996) = 1996
write(1, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 1996) = 1996
close(3)                                = 0
close(4)                                = 0
_exit(0)                                = ?
```

To be sure, a third instance of *prog* was executed, this time with the command switch set to place data in slack space.

```
[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 *textfile*

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.

As previously discussed, the binary from the Price forensic image is undoubtedly a modification of the *bmap* utility. To compare binary function, *strace* was run one more time, this time against the *bmap* binary compiled from its source. According to the help output for each binary, the command option used in each case should result in the same operation (get data placed in slack space of target):

```
[root@localhost tmp]# strace -ff ./bmap -slack dd.tmp
execve("./bmap", ["./bmap", "-slack", "dd.tmp"], [/* 33 vars */]) = 0
uname({sys="Linux", node="localhost.localdomain", ...}) = 0
brk(0)                                  = 0x806df60
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x40016000
open("/etc/ld.so.preload", O_RDONLY)    = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=39295, ...}) = 0
old_mmap(NULL, 39295, PROT_READ, MAP_PRIVATE, 3, 0) = 0x40017000
close(3)                                = 0
open("/lib/tls/libc.so.6", O_RDONLY)    = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0360U\1"..., 1024) = 1024
fstat64(3, {st_mode=S_IFREG|0755, st_size=1547996, ...}) = 0
old_mmap(0x42000000, 1257224, PROT_READ|PROT_EXEC, MAP_PRIVATE, 3, 0) = 0x42000000
mprotect(0x4212e000, 20232, PROT_NONE)  = 0
old_mmap(0x4212e000, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED, 3, 0x12e000) =
0x4212e000
old_mmap(0x42131000, 7944, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x42131000
close(3)                                = 0
set_thread_area({entry_number:-1 -> 6, base_addr:0x40016a00, limit:1048575, seg_32bit:1,
contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0, useable:1}) = 0
munmap(0x40017000, 39295)               = 0
lstat64("dd.tmp", {st_mode=S_IFREG|0755, st_size=30772, ...}) = 0
open("dd.tmp", O_RDONLY|O_LARGEFILE)    = 3
ioctl(3, FIGETBSZ, 0xbfffdd54)          = 0
lstat64("dd.tmp", {st_mode=S_IFREG|0755, st_size=30772, ...}) = 0
lstat64("/dev/sda2", {st_mode=S_IFBLK|0660, st_rdev=makedev(8, 2), ...}) = 0
open("/dev/sda2", O_RDONLY|O_LARGEFILE) = 4
ioctl(3, FIGETBSZ, 0xbfffdcc4)          = 0
brk(0)                                  = 0x806df60
brk(0x806ef60)                          = 0x806ef60
brk(0)                                  = 0x806ef60
brk(0x806f000)                          = 0x806f000
ioctl(3, FIBMAP, 0xbfffdd54)            = 0
ioctl(3, FIBMAP, 0xbfffdd54)            = 0
ioctl(3, FIBMAP, 0xbfffdd54)            = 0
ioctl(3, FIBMAP, 0xbfffdd54)            = 0
ioctl(3, FIBMAP, 0xbfffdd54)            = 0
ioctl(3, FIBMAP, 0xbfffdd54)            = 0
ioctl(3, FIBMAP, 0xbfffdd54)            = 0
ioctl(3, FIBMAP, 0xbfffdd54)            = 0
write(2, "getting from block 437672\n", 26getting from block 437672
) = 26
write(2, "file size was: 30772\n", 21file size was: 30772
)  = 21
write(2, "slack size: 1996\n", 17slack size: 1996
)      = 17
write(2, "block size: 4096\n", 17block size: 4096
)      = 17
_llseek(4, 1792706612, [1792706612], SEEK_SET) = 0
read(4, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 1996) = 1996
write(1, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 1996) = 1996
close(3)                                = 0
close(4)                                = 0
exit_group(0)                           = ?
```

Some noteworthy differences in operation, evidenced by the *strace* comparison, are:

1. *strace* shows fewer functional steps for the *prog* binary

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:smtp *:*                     LISTEN
udp        0      0 *:1024                  *:*
udp        0      0 *:sunrpc                *:*
udp        0      0 *:631                   *:*
udp        0      0 *:892                   *:*
```

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 *lsof*:

```
[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]# lsof -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 /pathto/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.rpmfind.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 evidence[18]), 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 run[20], 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/mtab
/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 *Mikemsg.doc*, 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 36203230 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 site[23], 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:

Sect-num.gif from Google cache md5sum: `fd8cb5272589dc32a4d88a22fe60d0b0`
Sect-num.gif 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·:·\·D·o·c·u·m·e·n·t·s·
·a·n·d··S·e·t·t·i·n·g·s·\·A·d·m·i·n·i·s·t·r·a·t·o·r·\·D·e·s·k·t·o·p·\·M·i·k·e·m·s·g·.·d·o·c

J·o·h·n· ·P·r·i·c·e·:·C·:·\·D·o·c·u·m·e·n·t·s· ·a·n·d·
·S·e·t·t·i·n·g·s·\·A·d·m·i·n·i·s·t·r·a·t·o·r·\·D·e·s·k·t·o·p·\·L·e·t·t·e·r·.·d·o·c
```

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 conducted[24] 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:

[25]
```
alert 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 data[26]:
```
\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 *bmap*
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 utility[29]. 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.shtm

[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.shtm

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

[30] http://www.mountimage.com/

[31] http://www.guidancesoftware.com/products/EnCaseForensic/purchase.shtm

[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 Challenge[35] 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 *dd* generation and hashing
dd on Linux	4.5.3	Image file generation
md5sum on Linux	4.5.3	Hash generation
dd for Windows (cygwin)	3,16,2,1029	Image file generation for mounted image files
Md5sum 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 *isosize*.[37]

```
[root@localhost tmp]# isosize -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 *isosize* 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.

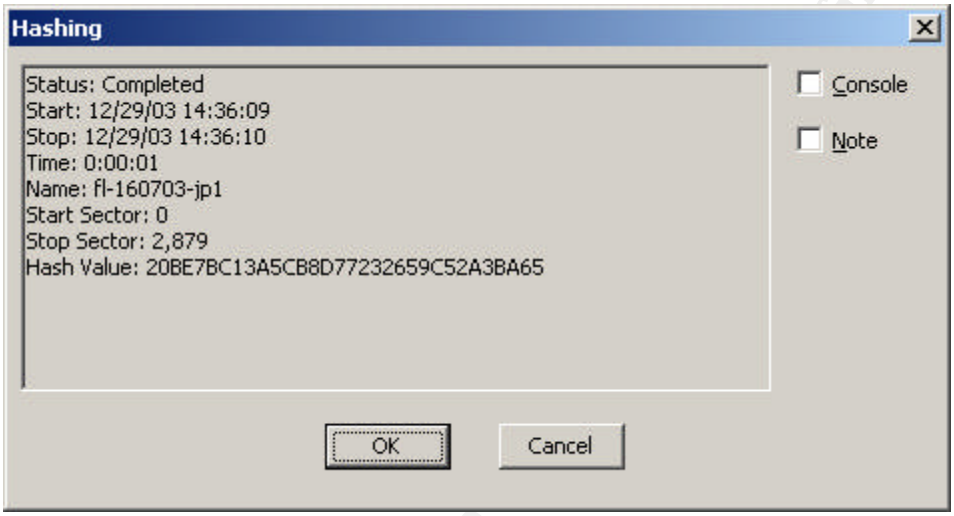
The command sequence to restore the floppy image follows:

[36] http://www.mountimage.com/downloads/mountimagepro-manual.pdf
[37] http://www.ussg.iu.edu/hypermail/linux/kernel/0112.3/0602.html

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

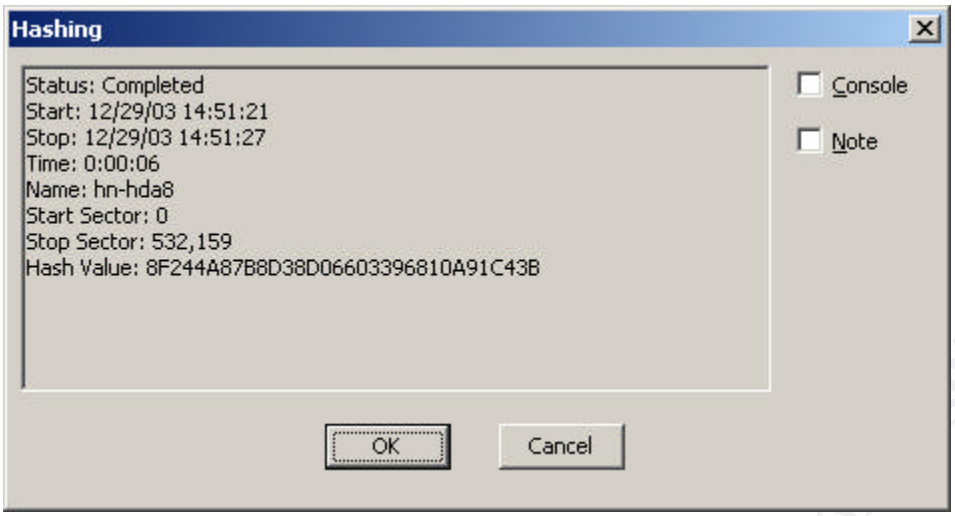
The matching *md5sum* hashes indicates a successful restore of the *dd* image to the floppy disk. The floppy diskette write-protect window was opened to enable write-protection, as soon as the restore was completed (before the hash was done).

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:

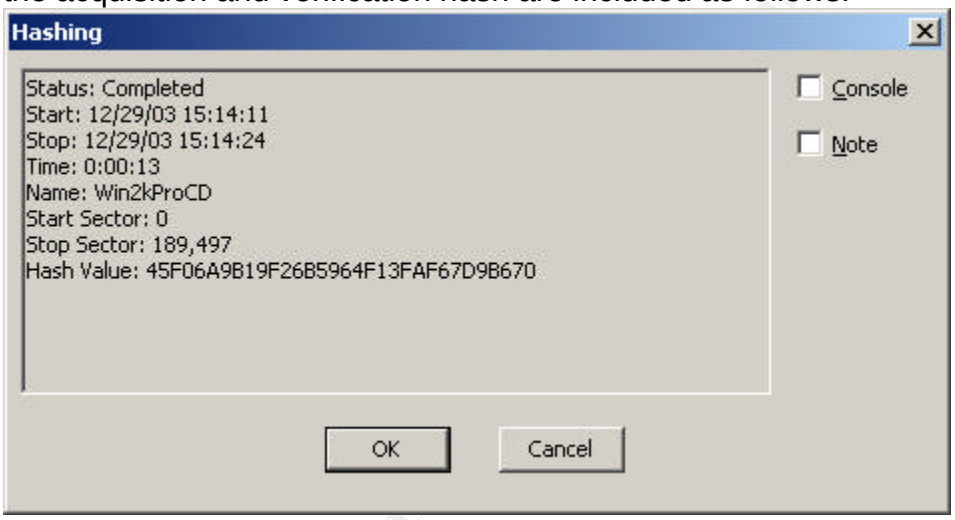


Image Preparation Summary		
Data Source	**Image Type**	**MD5 Hash**
Win2kProCD	dd	45f06a9b19f26b5964f13faf67d9b670
Win2kProCD	EnCase	45f06a9b19f26b5964f13faf67d9b670
fl-160703-jp1	dd	20be7bc13a5cb8d77232659c52a3ba65
fl-160703-jp1	EnCase	20be7bc13a5cb8d77232659c52a3ba65
hn-hda8	dd	8f244a87b8d38d06603396810a91c43b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

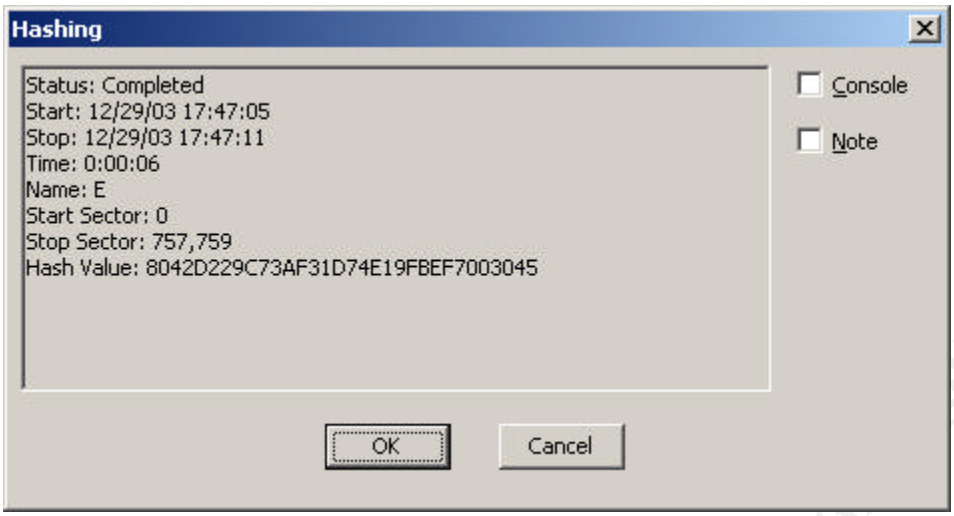
All mounted drives were unmounted, via Mount Image Pro, before the next image file was mounted.

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=Win2
kProCD-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.d
d
Based on original version developed by Paul Rubin, David MacKenzie, and Stuart K
emp
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\Win2kPr
oCD-EnC.dd...

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

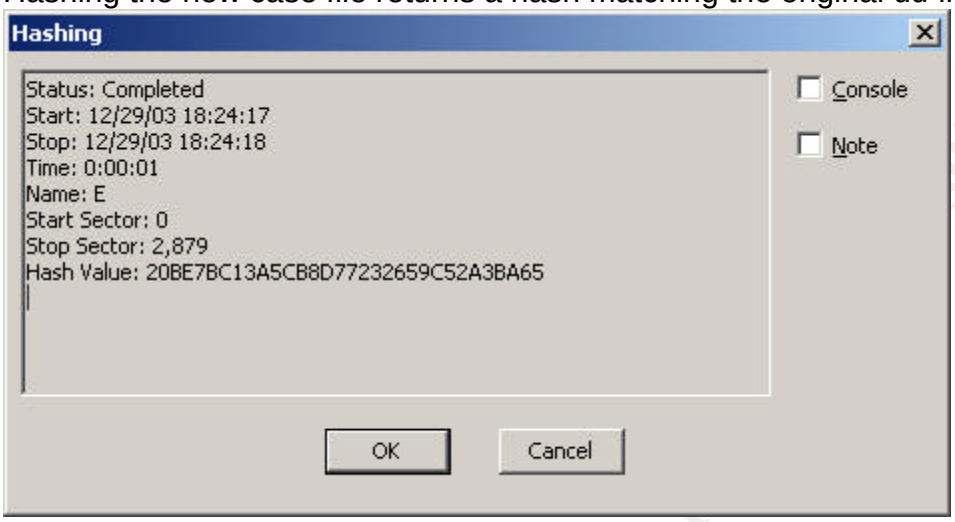
The new *dd* image file, created from the mounting of the EnCase image, results in a mismatched hash.

```
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=fl-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=fl-160703-jp1-EnC.dd
Based on original version developed by Paul Rubin, David MacKenzie, and Stuart K
emp
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\fl-160703-jp1-EnC.dd...Failed!
Incorrect function.
unable to display device infoCopying \\.\e: to C:\VirtualMachines\Shared\fl-1607
03-jp1-EnC.dd...
c:\dos\tools\dd.exe:
        \\.\e:: Invalid argument

Output C:\VirtualMachines\Shared\fl-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.

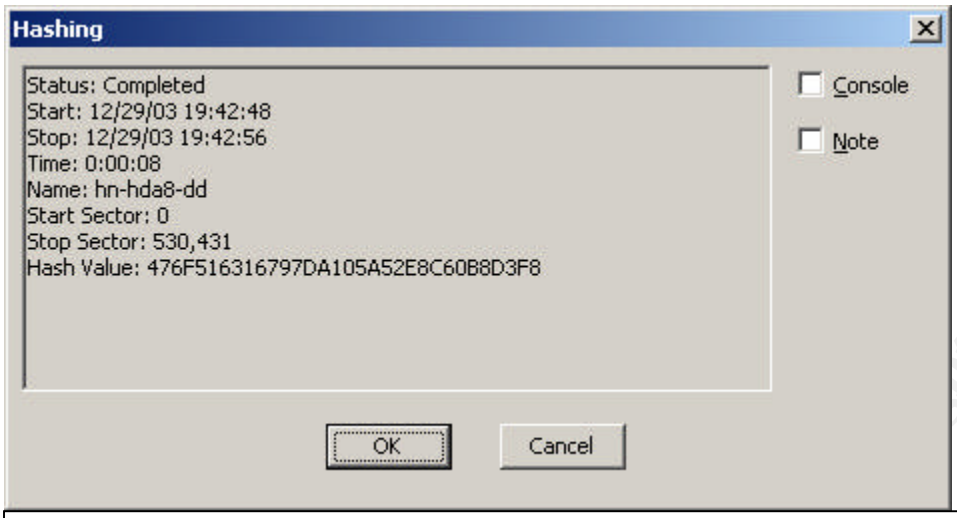
```
C:\VirtualMachines\Shared>c:\dos\tools\md5sum fl-160703-jp1-EnC.dd
\20be7bc13a5cb8d77232659c52a3ba65 *C:\\VirtualMachines\\Shared\\fl-160703-jp1-EnC.dd
```

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.r
aw
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 K
emp
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 (compr
essed/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

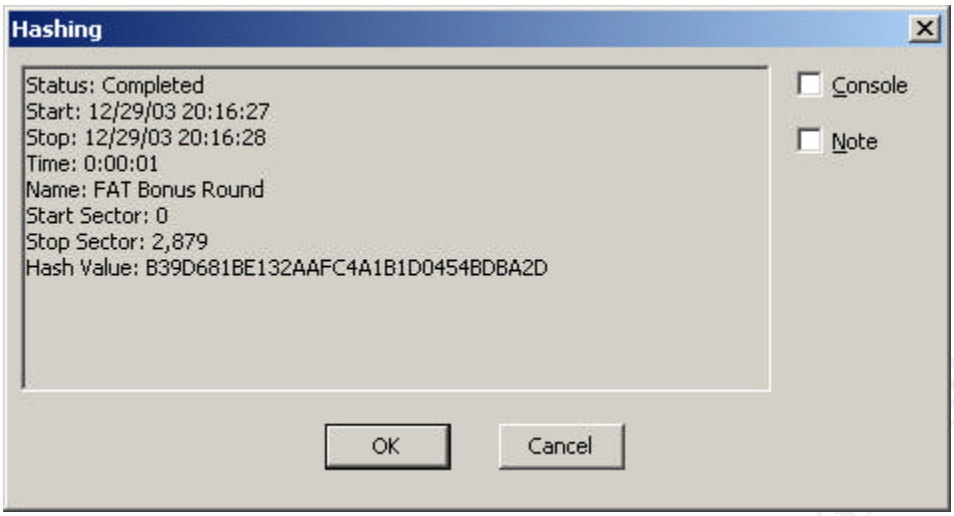
A floppy disk was wiped with EnCase (using hex 00 for the overwriting character) and then quick-formatted with the FAT file system using Windows XP Professional. The binary *Fport*, version 1.33 from Foundstone, was copied to the newly formatted diskette, along with the EICAR antivirus test file.[38] 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 K
emp
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
```

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 K
emp
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?**

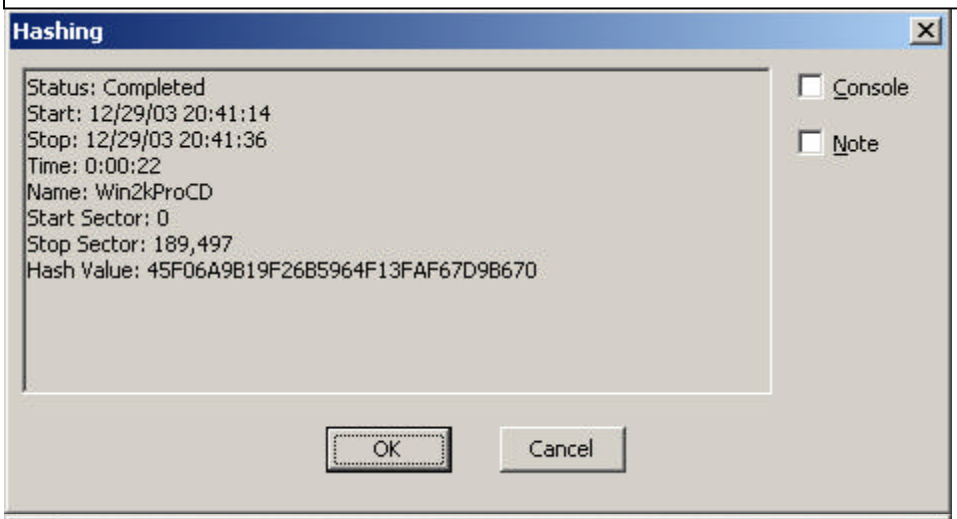
ISO9660	dd to EnCase	No
ISO9660	EnCase to dd	No
Ext2	dd to EnCase	Yes
Ext2	EnCase to dd	Yes
Ext2	dd to EnCase	No
Ext2	EnCase to dd	Yes
FAT	EnCase to dd	Yes

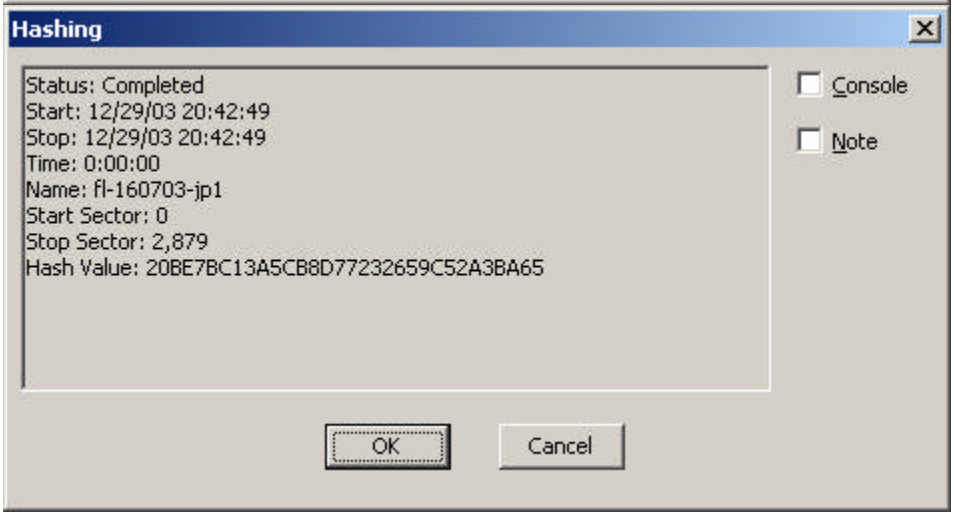
The following is screen output from the image file re-hashing:

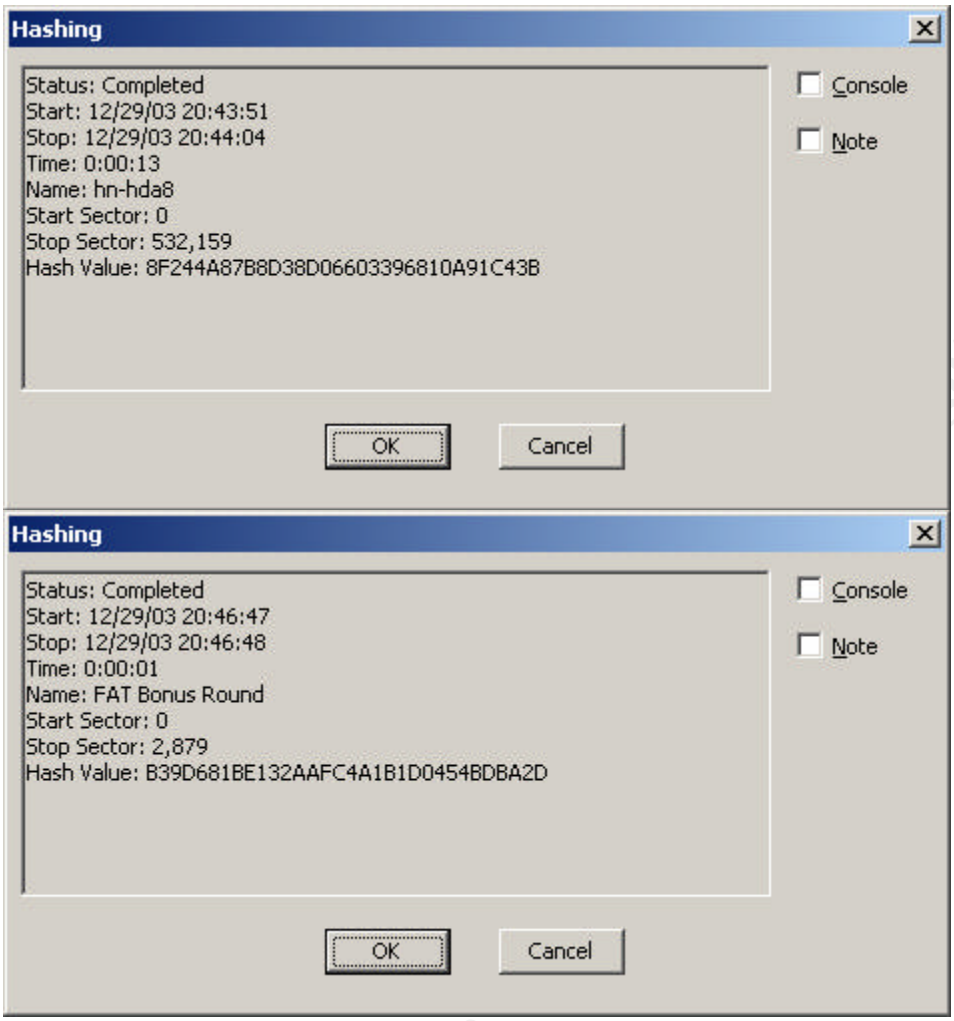
```
C:\VirtualMachines\Shared\original>c:\dos\tools\md5sum fl-160703-jp1.dd.raw
\20be7bc13a5cb8d77232659c52a3ba65 *C:\\VirtualMachines\\Shared\\original\\fl-160
703-jp1.dd.raw

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 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 site 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<      v
h@=                     RPSQ                    F\xX                    Z[^_]
h@=                     RPSQ                    [^_]                    C<      w+
h!>                     [^_]                    P0+H                    0<      v
h=>                     h@N                     [^_]                    [^_]
hP>                     I,RPSQ                  L       dQ              Rj@WS
0hm>                    h_N                     C +C                    <       w1
h@=                     p8hxN                   RPWV                    0<      v
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?                     <       v61             w&;=                    [^_]
hn?                     ug;]                    [^_]                    [^_]
hs?                     s=;]                    [^_]                    Qh`b
hy?                     PSh|                    [^_]                    Rh`b
Ph @                    [^_]                    PWVS                    [^_]
hg@                     XZSV                    [^_]                    [^_]
0h A                    VPVS                    tB;u                    [^_]
0h@A                    [^_]                    $v?V                    t       B<:u
0h B                    [^_]                    [^_]                    AF<:te1
0h B                    [^_]                    [^_]                    t       B<:u
0h&C                    <bt!<b                  [^_]                    VQRP
0h5C                    RPSW                    [^_]                    Vh`b
0hDC                    /FBH~                   [^_]                    SQRP
0hXC                    [^_]                     @t-                    [^_]
0h_C                    ;C tU                   [^_]                    [^_]
0hbC                    t$QVPS                  [^_]                    [^_]
hhC                     PVRS                    AELD                    [^_]
0hnC                    ;S t<                   ;AELD                   SRVW
hzC                     [^_]                    [^_]                    [^_]
h,D                     G +G                    [^_]                    [^_]
0h7D                    G +G                    @t5=                    [^_]
0h`D                    t';]                    [^_]                    @F;E
0h_C                    V +V                    [^_]                    [^_]
0hbC                    FDHP                    [^_]                    @F;E
hzC                     [^_]                    [^_]                    @G;E
h,D                     [^_]                    [^_]                    [^_]
0h-E                    t PS                    [^_]                    @G;E
h@I                     VPRQ                    RSVP                    [^_]
hdI                     [^_]                    [^_]                    FB;u
hxI                     VPRQ                    [^_]                    FB;u
h)J                     ~"PSV                   SPWQ                    BG;U
hCJ                     CDHP                    [^_]                    tS;}
h`J                     [^_]                    [^_]                    AC;M
h K                     F0+V                    [^_]                    [^_]
PhAK                    [^_]                    Gu~1                    AC;M
PhHK                    V$PQ)                   GuL1                    @F;E
hLK                     ~VQSV                   [^_]                    [^_]
hbK                     [^_]                    [^_]                    @F;E
huK                     FAJy                    \[^_]                   AC;M
hbK                     ~>PS                    CX9C                    [^_]
huK                     [^_]                    \[^_]                   AC;M
```

```
@F;E                    [^_]                    [^_]                    [^_]
[^_]                    t SVj                   Gu[1                    tLPj
@F;E                    QVj                     Gu#1                    [^_]
[^_]                    t*QVj                   t0@Nt                   tJPVj
[^_]                    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       9P              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                    [^_]                    [^_]
```

```
P ;U
[^_]
[^_]
Ph4n
[^_]
Wh4n
Sj:P
[^_]
tQ9u
=S@P
=S@P
[^_]
SQRP
/GBH~
/GBH~
[^_]
[^_]
/SYS
[^_]
|M9u
<[^_]
[^_]
[^_]
w_;M
[^_]
wA;U
Jt.P
X[^_
Bt(P
X[^_
Jt}1
[^_]
WVUS
[]^_
JtTG
[^_]
JtPG
[^_]
SWVV
RSWV
RSWV
Jte1
[^_]
Jtl1
Jtd1
[^_]
Jtd1
[^_]
R Iu
WVUS
[]^_
L[^_]
\[^_]
[^_]
[^_]
tC;E
[^_]
[^_]
[^_]
[^_]
[^_]
t&PS
RSVP
[^_]
[^_]
[^_]
H%T=
<\t~<\tn
[^_]
[^_]
< tZ<   tB<\t2
```

```
[^_]
/j      hs
[^_]
@t0R
XZh@
[^_]
[^_]
[^_]
PSRW
QPRW
[^_]
R Iu
WVUS
[]^_
[^_]
<*tm<'ti<Ite
<*t><*
*<'t
PVh|
[^_]
QQQQQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQj
t4Qj
t>Qj
x Rj
[^_]
RPh`
VQSW
RPh`
QSVW
@bQs
[^_]
[^_]
[^_]
[^_]
[^_]
tmPh
tiPh
[^_]
[^_]
}+;M
[^_]
t(;U
<       v;1
uc;u
s=;u
}+;M
[^_]
t(;U
<       v;1
uc;u
s=;u
[^_]
t%Pj
0<      v
0<      w
<       v$
VUUU
~3SVRR
 t0Wj
0<      v
t       N;
[^_]
t%Pj
0<      v
0<      w
<       v$
VUUU
 t0Wj
0<      v
```

```
0<      v
t       N;
[^_]
t%Pj
0<      v
0<      w
<       v$
VUUU
 t0Wj
0<      v
t       N;
[^_]
(^_]
u^9u
8^_]
mft_getopt
no index
invalid index %d
argv[%d] is NULL
argv[%d] (%s) is not an
option
examining a filename or
url!
%s is a well-formed
argument
checking against %s
flag-
flagized option
invokation
examining an enum!
matched against an enum
val
examining a venum!
matched against an
venum val
arg matches against %s
process_match
true
matches against %s
invalid value for enum
mft_log_init
nbd-server
MFT_LOG_THRESH
none
fatal
error
info
branch
progress
entryexit
mft_log_shutdown
unspecified
enter
exit
%s: %s
violet
blue
green
yellow
orange
white
%s: %s
<table
bgcolor=%s><tr><td>%s:
%s</td></tr></table><br
>
<table
bgcolor=%s><tr><td>%s</
td></tr></table><br>
```

```
<table
bgcolor=%s><tr><td></td
></tr></table><br>
Brazil
.TH %s "%d" "%s" "%s"
"%s"
.SH NAME
%s \- %s
.SH SYNOPSIS
.B %s
[\fIOPTION\fR]...
.SH DESCRIPTION
\fB\-\-%s\fR %s
\fB\-\-%s\fR \fIARG\fR
%s
\fB\-\-%s\fR \fIINT\fR
%s
\fB\-\-%s\fR
\fIFILENAME\fR %s
\fB\-\-%s\fR
\fIVALUE\fR %s
\fIVALUE\fR can be one
of:
       \fB%s\fR
 | \fB%s\fR
       \fBSHORTHAND
INVOKATION:\fR
Any of the valid values
for \fB--%s\fR can be
supplied directly as
options. For instance,
\fB--%s\fR can be used
in place of \fB--
%s=%s\fR.
       \fB%s\fR %s
--%s %s
.SH REPORTING BUGS
Report bugs to %s.
Usage: %s [OPTION]...
 [<%s-filename>]
--%s    %s
--%s <arg> %s
--%s <int> %s
--%s <filename> %s
--%s <
 | %s
> %s
--%s VALUE
  where VALUE is one
of:
  %s  %s
<tt>%s</tt> invocation
<tt>%s
[&lt;OPTIONS&gt;]
 [&lt;%s-filename&gt;]
</tt>
Where <bf>OPTIONS</bf>
may include any of:
<descrip>
<tag>--%s</tag>
       %s
<tag>--%s
&lt;arg&gt;</tag> %s
<tag>--%s
&lt;int&gt;</tag> %s
<tag>--%s
&lt;filename&gt;</tag>
%s
<tag>--%s &lt;
&gt;</tag> %s
<tag>--%s VALUE</tag>
```

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

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

```
/dev/sdi2
/dev/sdi15
/dev/sdi14
/dev/sdi13
/dev/sdi12
/dev/sdi11
/dev/sdi10
/dev/sdi1
/dev/sdi
/dev/sdh9
/dev/sdh8
/dev/sdh7
/dev/sdh6
/dev/sdh5
/dev/sdh4
/dev/sdh3
/dev/sdh2
/dev/sdh15
/dev/sdh14
/dev/sdh13
/dev/sdh12
/dev/sdh11
/dev/sdh10
/dev/sdh1
/dev/sdh
/dev/sdg9
/dev/sdg8
/dev/sdg7
/dev/sdg6
/dev/sdg5
/dev/sdg4
/dev/sdg3
/dev/sdg2
/dev/sdg15
/dev/sdg14
/dev/sdg13
/dev/sdg12
/dev/sdg11
/dev/sdg10
/dev/sdg1
/dev/sdg
/dev/sdf9
/dev/sdf8
/dev/sdf7
/dev/sdf6
/dev/sdf5
/dev/sdf4
/dev/sdf3
/dev/sdf2
/dev/sdf15
/dev/sdf14
/dev/sdf13
/dev/sdf12
/dev/sdf11
/dev/sdf10
/dev/sdf1
/dev/sdf
/dev/sde9
/dev/sde8
/dev/sde7
/dev/sde6
/dev/sde5
/dev/sde4
/dev/sde3
/dev/sde2
/dev/sde15
/dev/sde14
/dev/sde13
/dev/sde12
/dev/sde11
/dev/sde10
/dev/sde1
/dev/sde
/dev/sddx9
/dev/sddx8
/dev/sddx7
/dev/sddx6
/dev/sddx5
/dev/sddx4
/dev/sddx3
/dev/sddx2
/dev/sddx15
/dev/sddx14
/dev/sddx13
/dev/sddx12
/dev/sddx11
/dev/sddx10
/dev/sddx1
/dev/sddx
/dev/sddw9
/dev/sddw8
/dev/sddw7
/dev/sddw6
/dev/sddw5
/dev/sddw4
/dev/sddw3
/dev/sddw2
/dev/sddw15
/dev/sddw14
/dev/sddw13
/dev/sddw12
/dev/sddw11
/dev/sddw10
/dev/sddw1
/dev/sddw
/dev/sddv9
/dev/sddv8
/dev/sddv7
/dev/sddv6
/dev/sddv5
/dev/sddv4
/dev/sddv3
/dev/sddv2
/dev/sddv15
/dev/sddv14
/dev/sddv13
/dev/sddv12
/dev/sddv11
/dev/sddv10
/dev/sddv1
/dev/sddv
/dev/sddu9
/dev/sddu8
/dev/sddu7
/dev/sddu6
/dev/sddu5
/dev/sddu4
/dev/sddu3
/dev/sddu2
/dev/sddu15
/dev/sddu14
/dev/sddu13
/dev/sddu12
/dev/sddu11
/dev/sddu10
/dev/sddu1
/dev/sddu
/dev/sddt9
/dev/sddt8
/dev/sddt7
/dev/sddt6
/dev/sddt5
/dev/sddt4
/dev/sddt3
/dev/sddt2
/dev/sddt15
/dev/sddt14
/dev/sddt13
/dev/sddt12
/dev/sddt11
/dev/sddt10
/dev/sddt1
/dev/sddt
/dev/sdds9
/dev/sdds8
/dev/sdds7
/dev/sdds6
/dev/sdds5
/dev/sdds4
/dev/sdds3
/dev/sdds2
/dev/sdds15
/dev/sdds14
/dev/sdds13
/dev/sdds12
/dev/sdds11
/dev/sdds10
/dev/sdds1
/dev/sdds
/dev/sddr9
/dev/sddr8
/dev/sddr7
/dev/sddr6
/dev/sddr5
/dev/sddr4
/dev/sddr3
/dev/sddr2
/dev/sddr15
/dev/sddr14
/dev/sddr13
/dev/sddr12
/dev/sddr11
/dev/sddr10
/dev/sddr1
/dev/sddr
/dev/sddq9
/dev/sddq8
/dev/sddq7
/dev/sddq6
/dev/sddq5
/dev/sddq4
/dev/sddq3
/dev/sddq2
/dev/sddq15
/dev/sddq14
/dev/sddq13
/dev/sddq12
/dev/sddq11
/dev/sddq10
/dev/sddq1
/dev/sddq
/dev/sddp9
/dev/sddp8
/dev/sddp7
/dev/sddp6
/dev/sddp5
/dev/sddp4
/dev/sddp3
/dev/sddp2
/dev/sddp15
/dev/sddp14
/dev/sddp13
/dev/sddp12
/dev/sddp11
/dev/sddp10
/dev/sddp1
/dev/sddp
/dev/sddo9
/dev/sddo8
/dev/sddo7
/dev/sddo6
/dev/sddo5
/dev/sddo4
/dev/sddo3
/dev/sddo2
/dev/sddo15
/dev/sddo14
/dev/sddo13
/dev/sddo12
/dev/sddo11
/dev/sddo10
/dev/sddo1
/dev/sddo
/dev/sddn9
/dev/sddn8
/dev/sddn7
/dev/sddn6
/dev/sddn5
/dev/sddn4
/dev/sddn3
/dev/sddn2
/dev/sddn15
/dev/sddn14
/dev/sddn13
/dev/sddn12
/dev/sddn11
/dev/sddn10
/dev/sddn1
/dev/sddn
/dev/sddm9
/dev/sddm8
/dev/sddm7
/dev/sddm6
/dev/sddm5
/dev/sddm4
/dev/sddm3
/dev/sddm2
/dev/sddm15
/dev/sddm14
/dev/sddm13
/dev/sddm12
/dev/sddm11
/dev/sddm10
/dev/sddm1
/dev/sddm
/dev/sddl9
/dev/sddl8
/dev/sddl7
/dev/sddl6
/dev/sddl5
/dev/sddl4
/dev/sddl3
/dev/sddl2
/dev/sddl15
/dev/sddl14
/dev/sddl13
/dev/sddl12
/dev/sddl11
/dev/sddl10
/dev/sddl1
/dev/sddl
/dev/sddk9
/dev/sddk8
/dev/sddk7
```

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

```
/dev/sdcu
/dev/sdct9
/dev/sdct8
/dev/sdct7
/dev/sdct6
/dev/sdct5
/dev/sdct4
/dev/sdct3
/dev/sdct2
/dev/sdct15
/dev/sdct14
/dev/sdct13
/dev/sdct12
/dev/sdct11
/dev/sdct10
/dev/sdct1
/dev/sdct
/dev/sdcs9
/dev/sdcs8
/dev/sdcs7
/dev/sdcs6
/dev/sdcs5
/dev/sdcs4
/dev/sdcs3
/dev/sdcs2
/dev/sdcs15
/dev/sdcs14
/dev/sdcs13
/dev/sdcs12
/dev/sdcs11
/dev/sdcs10
/dev/sdcs1
/dev/sdcs
/dev/sdcr9
/dev/sdcr8
/dev/sdcr7
/dev/sdcr6
/dev/sdcr5
/dev/sdcr4
/dev/sdcr3
/dev/sdcr2
/dev/sdcr15
/dev/sdcr14
/dev/sdcr13
/dev/sdcr12
/dev/sdcr11
/dev/sdcr10
/dev/sdcr1
/dev/sdcr
/dev/sdcq9
/dev/sdcq8
/dev/sdcq7
/dev/sdcq6
/dev/sdcq5
/dev/sdcq4
/dev/sdcq3
/dev/sdcq2
/dev/sdcq15
/dev/sdcq14
/dev/sdcq13
/dev/sdcq12
/dev/sdcq11
/dev/sdcq10
/dev/sdcq1
/dev/sdcq
/dev/sdcp9
/dev/sdcp8
/dev/sdcp7
/dev/sdcp6
/dev/sdcp5
/dev/sdcp4
/dev/sdcp3
/dev/sdcp2
/dev/sdcp15
/dev/sdcp14
/dev/sdcp13
/dev/sdcp12
/dev/sdcp11
/dev/sdcp10
/dev/sdcp1
/dev/sdcp
/dev/sdco9
/dev/sdco8
/dev/sdco7
/dev/sdco6
/dev/sdco5
/dev/sdco4
/dev/sdco3
/dev/sdco2
/dev/sdco15
/dev/sdco14
/dev/sdco13
/dev/sdco12
/dev/sdco11
/dev/sdco10
/dev/sdco1
/dev/sdco
/dev/sdcn9
/dev/sdcn8
/dev/sdcn7
/dev/sdcn6
/dev/sdcn5
/dev/sdcn4
/dev/sdcn3
/dev/sdcn2
/dev/sdcn15
/dev/sdcn14
/dev/sdcn13
/dev/sdcn12
/dev/sdcn11
/dev/sdcn10
/dev/sdcn1
/dev/sdcn
/dev/sdcm9
/dev/sdcm8
/dev/sdcm7
/dev/sdcm6
/dev/sdcm5
/dev/sdcm4
/dev/sdcm3
/dev/sdcm2
/dev/sdcm15
/dev/sdcm14
/dev/sdcm13
/dev/sdcm12
/dev/sdcm11
/dev/sdcm10
/dev/sdcm1
/dev/sdcm
/dev/sdcl9
/dev/sdcl8
/dev/sdcl7
/dev/sdcl6
/dev/sdcl5
/dev/sdcl4
/dev/sdcl3
/dev/sdcl2
/dev/sdcl15
/dev/sdcl14
/dev/sdcl13
/dev/sdcl12
/dev/sdcl11
/dev/sdcl10
/dev/sdcl1
/dev/sdcl
/dev/sdck9
/dev/sdck8
/dev/sdck7
/dev/sdck6
/dev/sdck5
/dev/sdck4
/dev/sdck3
/dev/sdck2
/dev/sdck15
/dev/sdck14
/dev/sdck13
/dev/sdck12
/dev/sdck11
/dev/sdck10
/dev/sdck1
/dev/sdck
/dev/sdcj9
/dev/sdcj8
/dev/sdcj7
/dev/sdcj6
/dev/sdcj5
/dev/sdcj4
/dev/sdcj3
/dev/sdcj2
/dev/sdcj15
/dev/sdcj14
/dev/sdcj13
/dev/sdcj12
/dev/sdcj11
/dev/sdcj10
/dev/sdcj1
/dev/sdcj
/dev/sdci9
/dev/sdci8
/dev/sdci7
/dev/sdci6
/dev/sdci5
/dev/sdci4
/dev/sdci3
/dev/sdci2
/dev/sdci15
/dev/sdci14
/dev/sdci13
/dev/sdci12
/dev/sdci11
/dev/sdci10
/dev/sdci1
/dev/sdci
/dev/sdch9
/dev/sdch8
/dev/sdch7
/dev/sdch6
/dev/sdch5
/dev/sdch4
/dev/sdch3
/dev/sdch2
/dev/sdch15
/dev/sdch14
/dev/sdch13
/dev/sdch12
/dev/sdch11
/dev/sdch10
/dev/sdch1
/dev/sdch
/dev/sdcg9
/dev/sdcg8
/dev/sdcg7
/dev/sdcg6
/dev/sdcg5
/dev/sdcg4
/dev/sdcg3
/dev/sdcg2
/dev/sdcg15
/dev/sdcg14
/dev/sdcg13
/dev/sdcg12
/dev/sdcg11
/dev/sdcg10
/dev/sdcg1
/dev/sdcg
/dev/sdcf9
/dev/sdcf8
/dev/sdcf7
/dev/sdcf6
/dev/sdcf5
/dev/sdcf4
/dev/sdcf3
/dev/sdcf2
/dev/sdcf15
/dev/sdcf14
/dev/sdcf13
/dev/sdcf12
/dev/sdcf11
/dev/sdcf10
/dev/sdcf1
/dev/sdcf
/dev/sdce9
/dev/sdce8
/dev/sdce7
/dev/sdce6
/dev/sdce5
/dev/sdce4
/dev/sdce3
/dev/sdce2
/dev/sdce15
/dev/sdce14
/dev/sdce13
/dev/sdce12
/dev/sdce11
/dev/sdce10
/dev/sdce1
/dev/sdce
/dev/sdcd9
/dev/sdcd8
/dev/sdcd7
/dev/sdcd6
/dev/sdcd5
/dev/sdcd4
/dev/sdcd3
/dev/sdcd2
/dev/sdcd15
/dev/sdcd14
/dev/sdcd13
/dev/sdcd12
/dev/sdcd11
/dev/sdcd10
/dev/sdcd1
/dev/sdcd
/dev/sdcc9
/dev/sdcc8
/dev/sdcc7
/dev/sdcc6
/dev/sdcc5
/dev/sdcc4
/dev/sdcc3
/dev/sdcc2
/dev/sdcc15
/dev/sdcc14
/dev/sdcc13
```

```
/dev/sdcc12
/dev/sdcc11
/dev/sdcc10
/dev/sdcc1
/dev/sdcc
/dev/sdcb9
/dev/sdcb8
/dev/sdcb7
/dev/sdcb6
/dev/sdcb5
/dev/sdcb4
/dev/sdcb3
/dev/sdcb2
/dev/sdcb15
/dev/sdcb14
/dev/sdcb13
/dev/sdcb12
/dev/sdcb11
/dev/sdcb10
/dev/sdcb1
/dev/sdcb
/dev/sdca9
/dev/sdca8
/dev/sdca7
/dev/sdca6
/dev/sdca5
/dev/sdca4
/dev/sdca3
/dev/sdca2
/dev/sdca15
/dev/sdca14
/dev/sdca13
/dev/sdca12
/dev/sdca11
/dev/sdca10
/dev/sdca1
/dev/sdca
/dev/sdc9
/dev/sdc8
/dev/sdc7
/dev/sdc6
/dev/sdc5
/dev/sdc4
/dev/sdc3
/dev/sdc2
/dev/sdc15
/dev/sdc14
/dev/sdc13
/dev/sdc12
/dev/sdc11
/dev/sdc10
/dev/sdc1
/dev/sdc
/dev/sdbz9
/dev/sdbz8
/dev/sdbz7
/dev/sdbz6
/dev/sdbz5
/dev/sdbz4
/dev/sdbz3
/dev/sdbz2
/dev/sdbz15
/dev/sdbz14
/dev/sdbz13
/dev/sdbz12
/dev/sdbz11
/dev/sdbz10
/dev/sdbz1
/dev/sdbz
/dev/sdby9
/dev/sdby8
/dev/sdby7
/dev/sdby6
/dev/sdby5
/dev/sdby4
/dev/sdby3
/dev/sdby2
/dev/sdby15
/dev/sdby14
/dev/sdby13
/dev/sdby12
/dev/sdby11
/dev/sdby10
/dev/sdby1
/dev/sdby
/dev/sdbx9
/dev/sdbx8
/dev/sdbx7
/dev/sdbx6
/dev/sdbx5
/dev/sdbx4
/dev/sdbx3
/dev/sdbx2
/dev/sdbx15
/dev/sdbx14
/dev/sdbx13
/dev/sdbx12
/dev/sdbx11
/dev/sdbx10
/dev/sdbx1
/dev/sdbx
/dev/sdbw9
/dev/sdbw8
/dev/sdbw7
/dev/sdbw6
/dev/sdbw5
/dev/sdbw4
/dev/sdbw3
/dev/sdbw2
/dev/sdbw15
/dev/sdbw14
/dev/sdbw13
/dev/sdbw12
/dev/sdbw11
/dev/sdbw10
/dev/sdbw1
/dev/sdbw
/dev/sdbv9
/dev/sdbv8
/dev/sdbv7
/dev/sdbv6
/dev/sdbv5
/dev/sdbv4
/dev/sdbv3
/dev/sdbv2
/dev/sdbv15
/dev/sdbv14
/dev/sdbv13
/dev/sdbv12
/dev/sdbv11
/dev/sdbv10
/dev/sdbv1
/dev/sdbv
/dev/sdbu9
/dev/sdbu8
/dev/sdbu7
/dev/sdbu6
/dev/sdbu5
/dev/sdbu4
/dev/sdbu3
/dev/sdbu2
/dev/sdbu15
/dev/sdbu14
/dev/sdbu13
/dev/sdbu12
/dev/sdbu11
/dev/sdbu10
/dev/sdbu1
/dev/sdbu
/dev/sdbt9
/dev/sdbt8
/dev/sdbt7
/dev/sdbt6
/dev/sdbt5
/dev/sdbt4
/dev/sdbt3
/dev/sdbt2
/dev/sdbt15
/dev/sdbt14
/dev/sdbt13
/dev/sdbt12
/dev/sdbt11
/dev/sdbt10
/dev/sdbt1
/dev/sdbt
/dev/sdbs9
/dev/sdbs8
/dev/sdbs7
/dev/sdbs6
/dev/sdbs5
/dev/sdbs4
/dev/sdbs3
/dev/sdbs2
/dev/sdbs15
/dev/sdbs14
/dev/sdbs13
/dev/sdbs12
/dev/sdbs11
/dev/sdbs10
/dev/sdbs1
/dev/sdbs
/dev/sdbr9
/dev/sdbr8
/dev/sdbr7
/dev/sdbr6
/dev/sdbr5
/dev/sdbr4
/dev/sdbr3
/dev/sdbr2
/dev/sdbr15
/dev/sdbr14
/dev/sdbr13
/dev/sdbr12
/dev/sdbr11
/dev/sdbr10
/dev/sdbr1
/dev/sdbr
/dev/sdbq9
/dev/sdbq8
/dev/sdbq7
/dev/sdbq6
/dev/sdbq5
/dev/sdbq4
/dev/sdbq3
/dev/sdbq2
/dev/sdbq15
/dev/sdbq14
/dev/sdbq13
/dev/sdbq12
/dev/sdbq11
/dev/sdbq10
/dev/sdbq1
/dev/sdbq
/dev/sdbp9
/dev/sdbp8
/dev/sdbp7
/dev/sdbp6
/dev/sdbp5
/dev/sdbp4
/dev/sdbp3
/dev/sdbp2
/dev/sdbp15
/dev/sdbp14
/dev/sdbp13
/dev/sdbp12
/dev/sdbp11
/dev/sdbp10
/dev/sdbp1
/dev/sdbp
/dev/sdbo9
/dev/sdbo8
/dev/sdbo7
/dev/sdbo6
/dev/sdbo5
/dev/sdbo4
/dev/sdbo3
/dev/sdbo2
/dev/sdbo15
/dev/sdbo14
/dev/sdbo13
/dev/sdbo12
/dev/sdbo11
/dev/sdbo10
/dev/sdbo1
/dev/sdbo
/dev/sdbn9
/dev/sdbn8
/dev/sdbn7
/dev/sdbn6
/dev/sdbn5
/dev/sdbn4
/dev/sdbn3
/dev/sdbn2
/dev/sdbn15
/dev/sdbn14
/dev/sdbn13
/dev/sdbn12
/dev/sdbn11
/dev/sdbn10
/dev/sdbn1
/dev/sdbn
/dev/sdbm9
/dev/sdbm8
/dev/sdbm7
/dev/sdbm6
/dev/sdbm5
/dev/sdbm4
/dev/sdbm3
/dev/sdbm2
/dev/sdbm15
/dev/sdbm14
/dev/sdbm13
/dev/sdbm12
/dev/sdbm11
/dev/sdbm10
/dev/sdbm1
/dev/sdbm
/dev/sdbl9
/dev/sdbl8
/dev/sdbl7
/dev/sdbl6
/dev/sdbl5
/dev/sdbl4
/dev/sdbl3
```

```
/dev/sdbl2
/dev/sdbl15
/dev/sdbl14
/dev/sdbl13
/dev/sdbl12
/dev/sdbl11
/dev/sdbl10
/dev/sdbl1
/dev/sdbl
/dev/sdbk9
/dev/sdbk8
/dev/sdbk7
/dev/sdbk6
/dev/sdbk5
/dev/sdbk4
/dev/sdbk3
/dev/sdbk2
/dev/sdbk15
/dev/sdbk14
/dev/sdbk13
/dev/sdbk12
/dev/sdbk11
/dev/sdbk10
/dev/sdbk1
/dev/sdbk
/dev/sdbj9
/dev/sdbj8
/dev/sdbj7
/dev/sdbj6
/dev/sdbj5
/dev/sdbj4
/dev/sdbj3
/dev/sdbj2
/dev/sdbj15
/dev/sdbj14
/dev/sdbj13
/dev/sdbj12
/dev/sdbj11
/dev/sdbj10
/dev/sdbj1
/dev/sdbj
/dev/sdbi9
/dev/sdbi8
/dev/sdbi7
/dev/sdbi6
/dev/sdbi5
/dev/sdbi4
/dev/sdbi3
/dev/sdbi2
/dev/sdbi15
/dev/sdbi14
/dev/sdbi13
/dev/sdbi12
/dev/sdbi11
/dev/sdbi10
/dev/sdbi1
/dev/sdbi
/dev/sdbh9
/dev/sdbh8
/dev/sdbh7
/dev/sdbh6
/dev/sdbh5
/dev/sdbh4
/dev/sdbh3
/dev/sdbh2
/dev/sdbh15
/dev/sdbh14
/dev/sdbh13
/dev/sdbh12
/dev/sdbh11
/dev/sdbh10
/dev/sdbh1
/dev/sdbh
/dev/sdbg9
/dev/sdbg8
/dev/sdbg7
/dev/sdbg6
/dev/sdbg5
/dev/sdbg4
/dev/sdbg3
/dev/sdbg2
/dev/sdbg15
/dev/sdbg14
/dev/sdbg13
/dev/sdbg12
/dev/sdbg11
/dev/sdbg10
/dev/sdbg1
/dev/sdbg
/dev/sdbf9
/dev/sdbf8
/dev/sdbf7
/dev/sdbf6
/dev/sdbf5
/dev/sdbf4
/dev/sdbf3
/dev/sdbf2
/dev/sdbf15
/dev/sdbf14
/dev/sdbf13
/dev/sdbf12
/dev/sdbf11
/dev/sdbf10
/dev/sdbf1
/dev/sdbf
/dev/sdbe9
/dev/sdbe8
/dev/sdbe7
/dev/sdbe6
/dev/sdbe5
/dev/sdbe4
/dev/sdbe3
/dev/sdbe2
/dev/sdbe15
/dev/sdbe14
/dev/sdbe13
/dev/sdbe12
/dev/sdbe11
/dev/sdbe10
/dev/sdbe1
/dev/sdbe
/dev/sdbd9
/dev/sdbd8
/dev/sdbd7
/dev/sdbd6
/dev/sdbd5
/dev/sdbd4
/dev/sdbd3
/dev/sdbd2
/dev/sdbd15
/dev/sdbd14
/dev/sdbd13
/dev/sdbd12
/dev/sdbd11
/dev/sdbd10
/dev/sdbd1
/dev/sdbd
/dev/sdbc9
/dev/sdbc8
/dev/sdbc7
/dev/sdbc6
/dev/sdbc5
/dev/sdbc4
/dev/sdbc3
/dev/sdbc2
/dev/sdbc15
/dev/sdbc14
/dev/sdbc13
/dev/sdbc12
/dev/sdbc11
/dev/sdbc10
/dev/sdbc1
/dev/sdbc
/dev/sdbb9
/dev/sdbb8
/dev/sdbb7
/dev/sdbb6
/dev/sdbb5
/dev/sdbb4
/dev/sdbb3
/dev/sdbb2
/dev/sdbb15
/dev/sdbb14
/dev/sdbb13
/dev/sdbb12
/dev/sdbb11
/dev/sdbb10
/dev/sdbb1
/dev/sdbb
/dev/sdba9
/dev/sdba8
/dev/sdba7
/dev/sdba6
/dev/sdba5
/dev/sdba4
/dev/sdba3
/dev/sdba2
/dev/sdba15
/dev/sdba14
/dev/sdba13
/dev/sdba12
/dev/sdba11
/dev/sdba10
/dev/sdba1
/dev/sdba
/dev/sdb9
/dev/sdb8
/dev/sdb7
/dev/sdb6
/dev/sdb5
/dev/sdb4
/dev/sdb3
/dev/sdb2
/dev/sdb15
/dev/sdb14
/dev/sdb13
/dev/sdb12
/dev/sdb11
/dev/sdb10
/dev/sdb1
/dev/sdb
/dev/sdaz9
/dev/sdaz8
/dev/sdaz7
/dev/sdaz6
/dev/sdaz5
/dev/sdaz4
/dev/sdaz3
/dev/sdaz2
/dev/sdaz15
/dev/sdaz14
/dev/sdaz13
/dev/sdaz12
/dev/sdaz11
/dev/sdaz10
/dev/sdaz1
/dev/sdaz
/dev/sday9
/dev/sday8
/dev/sday7
/dev/sday6
/dev/sday5
/dev/sday4
/dev/sday3
/dev/sday2
/dev/sday15
/dev/sday14
/dev/sday13
/dev/sday12
/dev/sday11
/dev/sday10
/dev/sday1
/dev/sday
/dev/sdax9
/dev/sdax8
/dev/sdax7
/dev/sdax6
/dev/sdax5
/dev/sdax4
/dev/sdax3
/dev/sdax2
/dev/sdax15
/dev/sdax14
/dev/sdax13
/dev/sdax12
/dev/sdax11
/dev/sdax10
/dev/sdax1
/dev/sdax
/dev/sdaw9
/dev/sdaw8
/dev/sdaw7
/dev/sdaw6
/dev/sdaw5
/dev/sdaw4
/dev/sdaw3
/dev/sdaw2
/dev/sdaw15
/dev/sdaw14
/dev/sdaw13
/dev/sdaw12
/dev/sdaw11
/dev/sdaw10
/dev/sdaw1
/dev/sdaw
/dev/sdav9
/dev/sdav8
/dev/sdav7
/dev/sdav6
/dev/sdav5
/dev/sdav4
/dev/sdav3
/dev/sdav2
/dev/sdav15
/dev/sdav14
/dev/sdav13
/dev/sdav12
/dev/sdav11
/dev/sdav10
/dev/sdav1
/dev/sdav
/dev/sdau9
/dev/sdau8
/dev/sdau7
```

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

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

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

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

```
/dev/hdb31
/dev/hdb30
/dev/hdb3
/dev/hdb29
/dev/hdb28
/dev/hdb27
/dev/hdb26
/dev/hdb25
/dev/hdb24
/dev/hdb23
/dev/hdb22
/dev/hdb21
/dev/hdb20
/dev/hdb2
/dev/hdb19
/dev/hdb18
/dev/hdb17
/dev/hdb16
/dev/hdb15
/dev/hdb14
/dev/hdb13
/dev/hdb12
/dev/hdb11
/dev/hdb10
/dev/hdb1
/dev/hdb
/dev/hda9
/dev/hda8
/dev/hda7
/dev/hda6
/dev/hda5
/dev/hda4
/dev/hda32
/dev/hda31
/dev/hda30
/dev/hda3
/dev/hda29
/dev/hda28
/dev/hda27
/dev/hda26
/dev/hda25
/dev/hda24
/dev/hda23
/dev/hda22
/dev/hda21
/dev/hda20
/dev/hda2
/dev/hda19
/dev/hda18
/dev/hda17
/dev/hda16
/dev/hda15
/dev/hda14
/dev/hda13
/dev/hda12
/dev/hda11
/dev/hda10
/dev/hda1
/dev/hda
/dev/gscd
/dev/fd7u830
/dev/fd7u820
/dev/fd7u800
/dev/fd7u720
/dev/fd7u3840
/dev/fd7u360
/dev/fd7u3520
/dev/fd7u3200
/dev/fd7u2880
/dev/fd7u1920
/dev/fd7u1840
/dev/fd7u1760
/dev/fd7u1743
/dev/fd7u1722
/dev/fd7u1680
/dev/fd7u1660
/dev/fd7u1440
/dev/fd7u1120
/dev/fd7u1040
/dev/fd7h880
/dev/fd7h720
/dev/fd7h420
/dev/fd7h410
/dev/fd7h360
/dev/fd7h1660
/dev/fd7h1494
/dev/fd7h1476
/dev/fd7h1440
/dev/fd7h1200
/dev/fd7d360
/dev/fd7CompaQ
/dev/fd7
/dev/fd6u830
/dev/fd6u820
/dev/fd6u800
/dev/fd6u720
/dev/fd6u3840
/dev/fd6u360
/dev/fd6u3520
/dev/fd6u3200
/dev/fd6u2880
/dev/fd6u1920
/dev/fd6u1840
/dev/fd6u1760
/dev/fd6u1743
/dev/fd6u1722
/dev/fd6u1680
/dev/fd6u1660
/dev/fd6u1440
/dev/fd6u1120
/dev/fd6u1040
/dev/fd6h880
/dev/fd6h720
/dev/fd6h420
/dev/fd6h410
/dev/fd6h360
/dev/fd6h1660
/dev/fd6h1494
/dev/fd6h1476
/dev/fd6h1440
/dev/fd6h1200
/dev/fd6d360
/dev/fd6CompaQ
/dev/fd6
/dev/fd5u830
/dev/fd5u820
/dev/fd5u800
/dev/fd5u720
/dev/fd5u3840
/dev/fd5u360
/dev/fd5u3520
/dev/fd5u3200
/dev/fd5u2880
/dev/fd5u1920
/dev/fd5u1840
/dev/fd5u1760
/dev/fd5u1743
/dev/fd5u1722
/dev/fd5u1680
/dev/fd5u1660
/dev/fd5u1440
/dev/fd5u1120
/dev/fd5u1040
/dev/fd5h880
/dev/fd5h720
/dev/fd5h420
/dev/fd5h410
/dev/fd5h360
/dev/fd5h1660
/dev/fd5h1494
/dev/fd5h1476
/dev/fd5h1440
/dev/fd5h1200
/dev/fd5d360
/dev/fd5CompaQ
/dev/fd5
/dev/fd4u830
/dev/fd4u820
/dev/fd4u800
/dev/fd4u720
/dev/fd4u3840
/dev/fd4u360
/dev/fd4u3520
/dev/fd4u3200
/dev/fd4u2880
/dev/fd4u1920
/dev/fd4u1840
/dev/fd4u1760
/dev/fd4u1743
/dev/fd4u1722
/dev/fd4u1680
/dev/fd4u1660
/dev/fd4u1440
/dev/fd4u1120
/dev/fd4u1040
/dev/fd4h880
/dev/fd4h720
/dev/fd4h420
/dev/fd4h410
/dev/fd4h360
/dev/fd4h1660
/dev/fd4h1494
/dev/fd4h1476
/dev/fd4h1440
/dev/fd4h1200
/dev/fd4d360
/dev/fd4CompaQ
/dev/fd4
/dev/fd3u830
/dev/fd3u820
/dev/fd3u800
/dev/fd3u720
/dev/fd3u3840
/dev/fd3u360
/dev/fd3u3520
/dev/fd3u3200
/dev/fd3u2880
/dev/fd3u1920
/dev/fd3u1840
/dev/fd3u1760
/dev/fd3u1743
/dev/fd3u1722
/dev/fd3u1680
/dev/fd3u1660
/dev/fd3u1440
/dev/fd3u1120
/dev/fd3u1040
/dev/fd3h880
/dev/fd3h720
/dev/fd3h420
/dev/fd3h410
/dev/fd3h360
/dev/fd3h1660
/dev/fd3h1494
/dev/fd3h1476
/dev/fd3h1440
/dev/fd3h1200
/dev/fd3d360
/dev/fd3H720
/dev/fd3H360
/dev/fd3H1440
/dev/fd3D720
/dev/fd3D360
/dev/fd3CompaQ
/dev/fd3
/dev/fd2u830
/dev/fd2u820
/dev/fd2u800
/dev/fd2u720
/dev/fd2u3840
/dev/fd2u360
/dev/fd2u3520
/dev/fd2u3200
/dev/fd2u2880
/dev/fd2u1920
/dev/fd2u1840
/dev/fd2u1760
/dev/fd2u1743
/dev/fd2u1722
/dev/fd2u1680
/dev/fd2u1660
/dev/fd2u1440
/dev/fd2u1120
/dev/fd2u1040
/dev/fd2h880
/dev/fd2h720
/dev/fd2h420
/dev/fd2h410
/dev/fd2h360
/dev/fd2h1660
/dev/fd2h1494
/dev/fd2h1476
/dev/fd2h1440
/dev/fd2h1200
/dev/fd2d360
/dev/fd2H720
/dev/fd2H360
/dev/fd2H1440
/dev/fd2D720
/dev/fd2D360
/dev/fd2CompaQ
/dev/fd2
/dev/fd1u830
/dev/fd1u820
/dev/fd1u800
/dev/fd1u720
/dev/fd1u3840
/dev/fd1u360
/dev/fd1u3520
/dev/fd1u3200
/dev/fd1u2880
/dev/fd1u1920
/dev/fd1u1840
/dev/fd1u1760
/dev/fd1u1743
/dev/fd1u1722
/dev/fd1u1680
/dev/fd1u1660
/dev/fd1u1440
/dev/fd1u1120
/dev/fd1u1040
/dev/fd1h880
/dev/fd1h720
/dev/fd1h420
```

```
/dev/fd1h410
/dev/fd1h360
/dev/fd1h1660
/dev/fd1h1494
/dev/fd1h1476
/dev/fd1h1440
/dev/fd1h1200
/dev/fd1d360
/dev/fd1H720
/dev/fd1H360
/dev/fd1H1440
/dev/fd1D720
/dev/fd1D360
/dev/fd1CompaQ
/dev/fd1
/dev/fd0u830
/dev/fd0u820
/dev/fd0u800
/dev/fd0u720
/dev/fd0u3840
/dev/fd0u360
/dev/fd0u3520
/dev/fd0u3200
/dev/fd0u2880
/dev/fd0u1920
/dev/fd0u1840
/dev/fd0u1760
/dev/fd0u1743
/dev/fd0u1722
/dev/fd0u1680
/dev/fd0u1660
/dev/fd0u1440
/dev/fd0u1120
/dev/fd0u1040
/dev/fd0h880
/dev/fd0h720
/dev/fd0h420
/dev/fd0h410
/dev/fd0h360
/dev/fd0h1660
/dev/fd0h1494
/dev/fd0h1476
/dev/fd0h1440
/dev/fd0d360
/dev/fd0H720
/dev/fd0H360
/dev/fd0h1200
/dev/fd0D720
/dev/fd0D360
/dev/fd0H1440
/dev/fd0
/dev/fd0CompaQ
/dev/cm206cd
/dev/cm205cd
/dev/cdu535
/dev/cdu31a
/dev/bpcd
/dev/aztcd
/dev/md15
/dev/md14
/dev/md13
/dev/md12
/dev/md11
/dev/md9
/dev/md8
/dev/md7
/dev/md6
/dev/md5
/dev/md4
/dev/md3
/dev/md2
/dev/md1
/dev/md0
/dev/md10
/dev/null
Wrong medium type
No medium found
Disk quota exceeded
Remote I/O error
Is a named type file
No XENIX semaphores
available
Not a XENIX named type
file
Structure needs
cleaning
Stale NFS file handle
Operation now in
progress
Operation already in
progress
No route to host
Host is down
Connection refused
Connection timed out
No buffer space
available
Connection reset by
peer
Network is unreachable
Network is down
Address already in use
Protocol family not
supported
Operation not supported
Socket type not
supported
Protocol not supported
Protocol not available
Message too long
Destination address
required
Too many users
Streams pipe error
Remote address changed
File descriptor in bad
state
Name not unique on
network
Bad message
RFS specific error
Multihop attempted
Protocol error
Communication error on
send
Srmount error
Advertise error
Link has been severed
Object is remote
Package not installed
Machine is not on the
network
Out of streams
resources
Timer expired
No data available
Device not a stream
Bad font file format
Invalid slot
Invalid request code
No anode
Exchange full
Invalid request
descriptor
Invalid exchange
Level 2 halted
No CSI structure
available
Protocol driver not
attached
Link number out of
range
Level 3 reset
Level 3 halted
Level 2 not
synchronized
Channel number out of
range
Identifier removed
No message of desired
type
Directory not empty
Function not
implemented
No locks available
File name too long
Resource deadlock
avoided
Numerical result out of
range
Broken pipe
Too many links
Read-only file system
Illegal seek
No space left on device
File too large
Text file busy
Too many open files
Too many open files in
system
Invalid argument
Is a directory
Not a directory
No such device
Invalid cross-device
link
File exists
Device or resource busy
Block device required
Bad address
Permission denied
Cannot allocate memory
No child processes
Bad file descriptor
Exec format error
Argument list too long
No such device or
address
Input/output error
Interrupted system call
No such process
No such file or
directory
Operation not permitted
Success
Too many references:
cannot splice
Cannot send after
transport endpoint
shutdown
Transport endpoint is
not connected
Transport endpoint is
already connected
Software caused
connection abort
Network dropped
connection on reset
Cannot assign requested
address
Address family not
supported by protocol
Protocol wrong type for
socket
Socket operation on
non-socket
Interrupted system call
should be restarted
Invalid or incomplete
multibyte or wide
character
Cannot exec a shared
library directly
Attempting to link in
too many shared
libraries
.lib section in a.out
corrupted
Accessing a corrupted
shared library
Can not access a needed
shared library
Value too large for
defined data type
Too many levels of
symbolic links
Numerical argument out
of domain
Inappropriate ioctl for
device
Resource temporarily
unavailable
,ccs=
TOP_PAD_
MMAP_MAX_
TRIM_THRESHOLD_
MMAP_THRESHOLD_
Arena %d:
system bytes     = %10u
in use bytes     = %10u
Total (incl. mmap):
max mmap regions = %10u
max mmap bytes   =
%10lu
malloc: top chunk is
corrupt
free(): invalid pointer
%p!
malloc: using debugging
hooks
realloc(): invalid
pointer %p!
Unknown error
ANSI_X3.4-
1968//TRANSLIT
syslog: unknown
facility/priority: %x
out of memory [
<%d>
%h %e %T
[%d]
/dev/console
/dev/log
```

```
apic
mtrr
cmov
pse36
clflush
acpi
fxsr
sse2
ia64
amd3d
i386
i486
i586
i686
LD_AOUT_LIBRARY_PATH
LD_AOUT_PRELOAD
LD_PRELOAD
LD_LIBRARY_PATH
LD_ORIGIN_PATH
LD_DEBUG_OUTPUT
LD_PROFILE
GCONV_PATH
HOSTALIASES
LOCALDOMAIN
LOCPATH
MALLOC_TRACE
NLSPATH
RESOLV_HOST_CONF
RES_OPTIONS
TMPDIR
TZDIR
LD_WARN
LD_LIBRARY_PATH
LD_BIND_NOW
LD_BIND_NOT
LD_DYNAMIC_WEAK
/etc/suid-debug
MALLOC_CHECK_
/proc/sys/kernel/osrele
ase
FATAL: kernel too old
FATAL: cannot determine
library version
/usr/lib/gconv
gconv-modules
=INTERNAL->ucs2reverse
=ucs2reverse->INTERNAL
=INTERNAL->ascii
=ascii->INTERNAL
=INTERNAL->ucs2
=ucs2->INTERNAL
=utf8->INTERNAL
=INTERNAL->utf8
=ucs4le->INTERNAL
=INTERNAL->ucs4le
UCS-4LE//
=ucs4->INTERNAL
=INTERNAL->ucs4
UCS-2BE// UNICODEBIG//
UCS-2LE// ISO-
10646/UCS2/
CSASCII// ANSI_X3.4-
1968//
CP367// ANSI_X3.4-
1968//
IBM367// ANSI_X3.4-
1968//
US-ASCII// ANSI_X3.4-
1968//
ISO646-US// ANSI_X3.4-
1968//
ISO-IR-6// ANSI_X3.4-
1968//
ANSI_X3.4// ANSI_X3.4-
1968//
OSF00010102// ISO-
10646/UCS2/
OSF00010101// ISO-
10646/UCS2/
OSF00010100// ISO-
10646/UCS2/
UCS-2// ISO-10646/UCS2/
UCS2// ISO-10646/UCS2/
OSF05010001// ISO-
10646/UTF8/
ISO-IR-193// ISO-
10646/UTF8/
UTF-8// ISO-10646/UTF8/
UTF8// ISO-10646/UTF8/
WCHAR_T// INTERNAL
OSF00010106// ISO-
10646/UCS4/
OSF00010105// ISO-
10646/UCS4/
OSF00010104// ISO-
10646/UCS4/
ISO-10646// ISO-
10646/UCS4/
CSUCS4// ISO-
10646/UCS4/
UCS-4BE// ISO-
10646/UCS4/
UCS-4// ISO-10646/UCS4/
alias
module
UNICODELITTLE// ISO-
10646/UCS2/
OSF00010020//
ANSI_X3.4-1968//
ISO_646.IRV:1991//
ANSI_X3.4-1968//
ANSI_X3.4-1986//
ANSI_X3.4-1968//
ISO-10646/UTF-8/ ISO-
10646/UTF8/
10646-1:1993/UCS4/ ISO-
10646/UCS4/
10646-1:1993// ISO-
10646/UCS4/
GCONV_PATH
/usr/lib/gconv/gconv-
modules.cache
gconv
gconv_init
gconv_end
toupper
tolower
upper
lower
alpha
digit
xdigit
space
print
graph
blank
cntrl
punct
alnum
libc
POSIX
ANSI_X3.4-1968
messages
/usr/share/locale
POSIX
LC_COLLATE
LC_CTYPE
LC_MONETARY
LC_NUMERIC
LC_TIME
LC_MESSAGES
LC_ALL
LC_XXX
LANGUAGE
charset=
OUTPUT_CHARSET
/usr/share/locale
/locale.alias
parse error
parser stack overflow
plural=
nplurals=
0123456789abcdefghijklm
nopqrstuvwxyz
(null)
(nil)

0000000000000000
%m/%d/%y
%Y-%m-%d
%H:%M
%I:%M:%S %p
%H:%M:%S
/etc/localtime
Universal
%[^0-9,+-]
%hu:%hu:%hu
M%hu.%hu.%hu%n
/usr/share/zoneinfo
TZDIR
posixrules
/proc/self/cwd
/proc
/etc/mtab
/etc/fstab
proc
/cpuinfo
processor
/meminfo
MemTotal: %ld kB
MemFree: %ld kB
/lib/
/usr/lib/
ORIGIN
PLATFORM
cannot allocate name
record
system search path
cannot stat shared
object
cannot read file data
cannot map zero-fill
pages
cannot create
searchlist
 search path=
                (%s from
file %s)
                (%s)
file too short
invalid ELF header
ELF file OS ABI invalid
ELF file ABI version
invalid
internal error
  trying file=%s
file=%s;  needed by %s
find library=%s;
searching
RPATH
RUNPATH
cannot create cache for
search path
cannot create
RUNPATH/RPATH copy
cannot create search
path array
file=%s;  generating
link map
cannot create shared
object descriptor
ELF load command
alignment not page-
aligned
ELF load command
address/offset not
properly aligned
failed to map segment
from shared object
cannot dynamically load
executable
cannot change memory
protections
cannot allocate memory
for program header
object file has no
dynamic section
  dynamic: 0x%0*lx
base: 0x%0*lx   size:
0x%0*Zx
    entry: 0x%0*lx
phdr: 0x%0*lx  phnum:
%*u
shared object cannot be
dlopen()ed
ELF file data encoding
not big-endian
ELF file data encoding
not little-endian
ELF file version ident
does not match current
one
ELF file version does
not match current one
ELF file's phentsize
not the expected size
only ET_DYN and ET_EXEC
can be loaded
cannot open shared
object file
AT_HWCAP:
/etc/ld.so.cache
 search cache=%s
ld.so-1.7.0
glibc-ld.so.cache1.1
undefined symbol:
symbol=%s;  lookup in
file=%s
file=%s;  needed by %s
(relocation dependency)
binding file %s to %s:
%s symbol `%s'
relocation error
```

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

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)**

Last Updated: February 25th, 2020

Upcoming SANS Forensics Training

SANS Northern VA - Reston Spring 2020	Reston, VA	Mar 02, 2020 - Mar 07, 2020	Live Event
SANS Secure Japan 2020	Tokyo, Japan	Mar 02, 2020 - Mar 14, 2020	Live Event
SANS Munich March 2020	Munich, Germany	Mar 02, 2020 - Mar 07, 2020	Live Event
SANS St. Louis 2020	St. Louis, MO	Mar 08, 2020 - Mar 13, 2020	Live Event
Dallas 2020 - FOR500: Windows Forensic Analysis	Dallas, TX	Mar 09, 2020 - Mar 14, 2020	vLive
SANS vLive - FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques	FOR610 - 202003,	Mar 09, 2020 - Apr 22, 2020	vLive
SANS Paris March 2020	Paris, France	Mar 09, 2020 - Mar 14, 2020	Live Event
SANS Dallas 2020	Dallas, TX	Mar 09, 2020 - Mar 14, 2020	Live Event
SANS London March 2020	London, United Kingdom	Mar 16, 2020 - Mar 21, 2020	Live Event
SANS Norfolk 2020	Norfolk, VA	Mar 16, 2020 - Mar 21, 2020	Live Event
SANS San Francisco Spring 2020	San Francisco, CA	Mar 16, 2020 - Mar 27, 2020	Live Event
SANS Secure Singapore 2020	Singapore, Singapore	Mar 16, 2020 - Mar 28, 2020	Live Event
SANS Oslo March 2020	Oslo, Norway	Mar 23, 2020 - Mar 28, 2020	Live Event
SANS Seattle Spring 2020	Seattle, WA	Mar 23, 2020 - Mar 28, 2020	Live Event
SANS Madrid March 2020	Madrid, Spain	Mar 23, 2020 - Mar 28, 2020	Live Event
SANS Secure Canberra 2020	Canberra, Australia	Mar 23, 2020 - Mar 28, 2020	Live Event
Mentor Session - FOR508	Sao Paulo, Brazil	Mar 25, 2020 - Mar 28, 2020	Mentor
SANS Abu Dhabi March 2020	Abu Dhabi, United Arab Emirates	Mar 28, 2020 - Apr 02, 2020	Live Event
SANS FOR585 Rome March 2020 (In Italian)	Rome, Italy	Mar 30, 2020 - Apr 04, 2020	Live Event
SANS Frankfurt March 2020	Frankfurt, Germany	Mar 30, 2020 - Apr 04, 2020	Live Event
SANS vLive - FOR508: Advanced Incident Response, Threat Hunting, and Digital Forensics	FOR508 - 202003,	Mar 31, 2020 - May 07, 2020	vLive
SANS 2020	Orlando, FL	Apr 03, 2020 - Apr 10, 2020	Live Event
SANS Riyadh April 2020	Riyadh, Kingdom Of Saudi Arabia	Apr 04, 2020 - Apr 16, 2020	Live Event
SANS Bethesda 2020	Bethesda, MD	Apr 14, 2020 - Apr 19, 2020	Live Event
SANS Minneapolis 2020	Minneapolis, MN	Apr 14, 2020 - Apr 19, 2020	Live Event
SANS London April 2020	London, United Kingdom	Apr 20, 2020 - Apr 25, 2020	Live Event
SANS Brussels April 2020	Brussels, Belgium	Apr 20, 2020 - Apr 25, 2020	Live Event
SANS Baltimore Spring 2020	Baltimore, MD	Apr 27, 2020 - May 02, 2020	Live Event
SANS Bucharest May 2020	Bucharest, Romania	May 04, 2020 - May 09, 2020	Live Event
SANS Security West 2020	San Diego, CA	May 06, 2020 - May 13, 2020	Live Event
Security West 2020 - FOR498: Battlefield Forensics & Data Acquisition	San Diego, CA	May 08, 2020 - May 13, 2020	vLive