Digital Forensics For Unix

The SANS Institute

John Green – john@cybersecuritysciences.com

Hal Pomeranz – hal@deer-run.com

Forensics in a Nutshell

- Evidence seizure
- Investigation and analysis
- Reporting results

"Gathering and analyzing data in a manner as free from distortion or bias as possible to reconstruct data or what has happened in the past on a system."

Farmer and Venema, 1999 *www.fish.com/security/forensics.html*

Major Challenges

- Rapid action.
- Recording the scene without disturbing it.
- Maintaining the integrity of the evidence.
- Investigating a potential compromise when the software you're using can't be trusted.

Guiding Principles

- Work to minimize evidence loss.
- Take great notes in excruciating detail.
- Collect all the evidence that you can.
- Analyze everything you collect.
- Be ready to prove that evidence and process integrity has been maintained.
- Learn lessons from each incident.

Drive Imaging

- Need to capture the *entire* drive, including the free disk blocks (use [**`dcfl`**]**`dd`**, not **`tar`**)
- Need an uncorrupted image, so drive must not be active (do forensics off-line)
- *Never* write data to disks of system being investigated (don't destroy evidence)
- Only analyze copies, never the original media

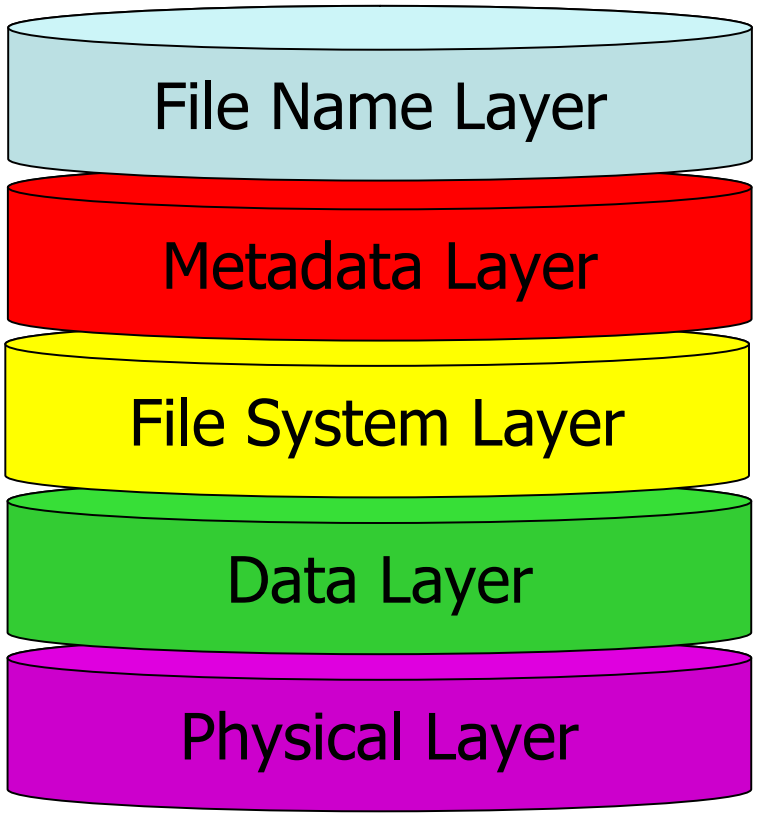
dcfldd Screenshot

```
# ./dcfldd if=/dev/sda hashwindow=0 |\
      ./nc 192.168.1.1 31337
2097152 blocks (1024Mb) written.
Total: 1e504a2e1202e3d01a92a32eb8978afa
2097152+0 records in
2097152+0 records out
```

Collected the entire disk!

A Conceptual Model for File System Organization

File systems can be organized around a model based on 5 layers



- File Names, Directories
- Inodes
- Superblock
- Sectors, Fragments, Blocks
- VToC, Partitions

Physical Layer and Disk Partitions

- A disk can be segmented into partitions.
  - 2 types: logical and extended
- Each partition is treated as an independent device by the OS.
- A partition table (aka disk label or VToC) at the beginning of the disk provides a map.
- A partition usually contains a **file system** or swap.

Data Layer

- The **Data Layer** is where the zeros and ones are actually written on disk.
- The basic storage unit is a [disk] block.
  - Blocks are composed of fragments (usually 4)
  - Fragments are composed of sectors (usually 2)
  - Sectors are the smallest unit and are 512 bytes
- For efficiency, consecutive sectors are organized and allocated together when possible.

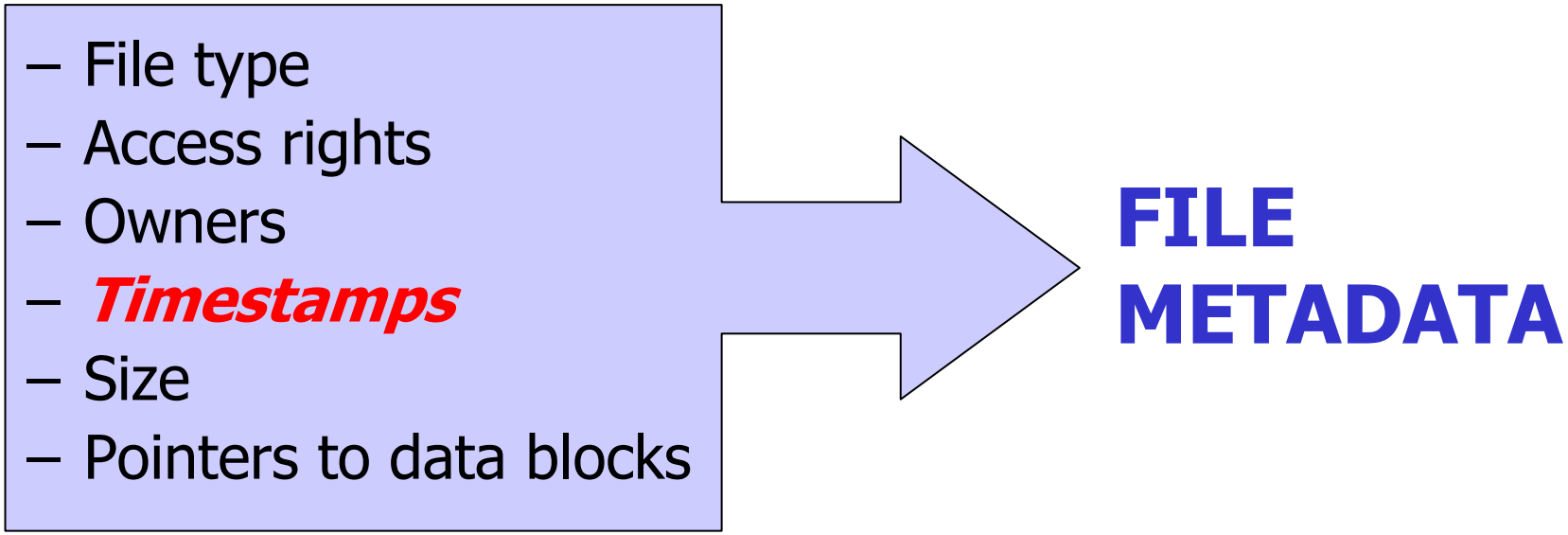
File System Layer

- The **File System Layer** contains the data that describes the file system within a partition.

- It refers to a data structure known as the **superblock** which contains the following data:
  - FS type, status (clean or dirty), and size
  - Pointer to the inode corresponding to the root of the FS
  - Modification time
  - Disk block size
  - List of free disk blocks and total number
  - List of free **inodes** and total number

Metadata Layer

- The **Metadata Layer** refers to the data structures that describe files.
- These structures are called **inodes** and each files has one.



Human Interface Layer

- The **Human Interface (File Name) Layer** associates a file name with each allocated inode.
- Although the metadata layer completely defines a file, humans do not like referring to files by their inode number.
- This structure is typically known as a directory.

The Sleuth Kit (TSK)

- The Sleuth Kit is a collection of 16 highly specialized file system analysis tools.
  - Combines and enhances collection and analysis tools from earlier packages.
  - Tools are organized by file system layers and follow a mnemonic naming convention.

TSK Programs

- File System Layer Tools
  - fsstat Displays details about the file system
- Data Layer Tools
  - dcat Displays the contents of a disk block
  - dls Lists contents of deleted disk blocks
  - dcalc Maps between dd images and dls results
  - dstat Lists statistics associated with specific disk blocks
- Metadata Layer Tools
  - ils Displays inode details
  - istat Displays information about a specific inode
  - icat Displays contents of disk blocks allocated to an inode
  - ifind Determine which inode has allocated a block in an image

More TSK Programs

- Human Interface (File Name) Layer
  - fls Displays file and directory entries in a directory inode
  - ffind Determine which file has allocated an inode in an image
- Media Management (partitions)
  - mmls Displays list of partitions in a DISK image
- The remaining 6 tools do not directly process a file system image:
  - Hash database tools, the 'file' tool, 'sorter' tool, and timeline tools

Partition Extraction Screenshot

```
# mmls -t dos dev_sda.dd
DOS Partition Table
Units are in 512-byte sectors

     Slot    Start        End          Length       Description
00:  -----   0000000000   0000000000   0000000001   Primary Table (#0)
01:  -----   0000000001   0000000031   0000000031   Unallocated
02:  00:00   0000000032   0001884159   0001884128   Linux (0x83)
03:  00:01   0001884160   0002097151   0000212992   Linux Swap (0x82)

# dd if=dev_sda.dd of=unallocated.dd bs=512 skip=1 count=31
31+0 records in
31+0 records out
# dd if=dev_sda.dd of=dev_sda1.dd bs=512 skip=32 count=1884128
1884128+0 records in
1884128+0 records out
# dd if=dev_sda.dd of=dev_sda2.dd bs=512 skip=1884160 count=212992
212992+0 records in
212992+0 records out
```

What Can You Do With These Images?

- Use **`fls`**/**`ils`**/**`mactime`** from TSK for *timeline analysis*
- Mount file systems (via loopback mounts) and use standard Unix tools like **`find`**
- Use **`grep`** to search for "dirty words" in raw disk blocks and then use TSK tools to find associated files
- Leverage specialized tools like Foremost, Lazarus, and TSK to recover deleted data from "free" disk blocks

Timeline Forensics – It Was FTP!

```
Oct 03 100 16:01:30      484 .a. -rw------- root     root     /etc/ftpaccess
                      153488 .a. -rwxr-xr-x root     root     /usr/sbin/in.ftpd
Oct 03 100 16:01:33      456 .a. -rw------- root     root     /etc/ftpconversions
Oct 03 100 16:01:34      104 .a. -rw------- root     root     /etc/ftphosts
                          79 .a. -rw------- root     root     /etc/ftpusers
                        4096 mac -rw-r--r-- root     root     /var/run/ftp.pids-all
Oct 03 100 16:01:54    42736 .a. -rwxr-xr-x root     root     /sbin/ifconfig
                       11868 .a. -rwxr-xr-x root     root     /usr/bin/cut
Oct 03 100 16:01:55     3070 m.c -rw-r--r-- root     root     /etc/inetd.conf
                       10160 .a. -rwxr-xr-x root     root     /usr/bin/killall
                        8860 .a. -r-xr-xr-x root     root     /usr/bin/w
Oct 03 100 16:20:37    20452 m.c -rwxr-xr-x root     root     /bin/systat
```

`mount` Usage for Images

- **`mount -t fstype [options]` *`device directory`***

  ***device* can be a disk partition or image file**

- [Useful Options]

-t	file system (ext2, ntfs, msdos, etx)
-ro	mount as read only
-loop	mount on a loop device (used for image files)
-noexec	do not execute files from mounted partitions
-noatime	do not modify access times on mounted partitions

- Example: Mount an image file at /mnt/hacked and protect it

```
# mount -o ro,loop /casedata29/dev_sda1.dd /mnt/hacked/
```

Searching the File System

- Mount the image and search the compromised file system for:
  - Extra or incorrect /etc/passwd entries
  - Log files and history files
  - Any directory beginning with "."
  - Regular files in /dev
  - SUID/SGID files
  - Recently modified binaries
  - Recently created files (derived from timeline)

Hidden Files or Directories

```
# find /mnt/hacked -name ".*" -type d -print
/mnt/hacked/lib/.x
/mnt/hacked/root/.ssh
/mnt/hacked/root/.links
# ls /mnt/hacked/lib/.x
cl          hide.log      install.log       log    sk
hide        inst          ip                s
#
# find /mnt/hacked/dev -type f -print
/mnt/hacked/dev/MAKEDEV
/mnt/hacked/dev/shm/k
/mnt/hacked/dev/ttyop
/mnt/hacked/dev/ttyoa
/mnt/hacked/dev/ttyof
/mnt/hacked/dev/hdx1
/mnt/hacked/dev/hdx2
```

Modified Binaries

```
# ls -lai | sort -n
  :     :     :    :
45669 -rwxr-xr-x   1 root root    9468 Jul 24  2001 true
45670 -rwxr-xr-x   1 root root   10844 Jul 24  2001 uname
45755 -rwxr-xr-x   1 rpm  rpm  1580104 Sep  7  2001 rpm
45758 -rwxr-xr-x   1 root root    2872 Aug 27  2001 arch
45759 -rwxr-xr-x   1 root root    4252 Aug 27  2001 dmesg
45760 -rwxr-xr-x   1 root root    7964 Aug 27  2001 kill
45761 -rwxr-xr-x   1 root root   17740 Aug 27  2001 login
45762 -rwxr-xr-x   1 root root   23372 Aug 27  2001 more
45817 -rwxr-xr-x   1 root root    2708 Sep  9  2001 doexec
45818 -rwxr-xr-x   1 root root   23744 Sep  9  2001 ipcalc
45819 -rwxr-xr-x   1 root root   19748 Sep  9  2001 usleep
92011 -rwxr-xr-x   1 root root   32756 Dec 14  2001 ps
92013 -rwxr-xr-x   1 root root   30640 Dec 14  2001 netstat
92022 -rwxr-xr-x   1 root root   36692 Dec 14  2001 ls
92032 -rwxr-xr-x   1  506  506  165136 Jan 19  2002 pico
total 5924
```

Search for Strings

- Much like antivirus signatures, rootkits and other malware often contain signature "strings" that we can search for.
- Searches can be performed at the filesystem layer or at the disk block layer.
- Examples might include words like:
  - r00tk1t, gr33tz, password, login, profanity, Romanian words
- **The goal of string searching is to find disk "hotspots" to zero-in on for further investigation and analysis.**

What File System and Block Size?

```
# fsstat dev_sda1.dd
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: EXT2FS
Volume Name: /
Last Mount: Sat Aug  6 21:01:32 2004
Last Write: Sat Aug  6 21:01:32 2004
Last Check: Mon Jun 14 09:07:26 2004
Unmounted properly
Last mounted on:
Operating System: Linux
:        :              :
CONTENT-DATA INFORMATION
--------------------------------------------
Fragment Range: 0 - 4727125
Block Size: 4096
Fragment Size: 4096
```

Now, Where's Waldo? Block 170388!

```
# grep -abi waldo dev_sda1.dd > /tmp/found_waldo.txt
# less /tmp/found_waldo.txt
   :   :   :
689702819:^@^@<8D^@submit@bugs.kde.org^@(c) 2003
Waldo Bastian^@Author^@bastian@kde.org^@No
696076095:Do you want to save the changes or
discard them?^@editor^@0.5^@submit@bugs.kde.org^@KDE
Menu Editor^@kmenuedit^@bastian@kde.org^@Waldo
Bastian^@sandrini@kde.org(C) 2000-2003, Waldo
697911472:Waldo
697911478:Waldorf
   :   :   :
```

697911472(byte offset) / 4096(blocksize) = 170388(block)

I'd Like to See that in Context, Please!

```
# dcat -h -f linux-ext2 dev_sda1.dd 170388
:    :    :
2144    0a77616b 656e0a77 616b656e 65640a77     .wak en.w aken ed.w
2160    616b656e 696e670a 77616b65 730a7761     aken ing. wake s.wa
2176    6b657570 0a77616b 696e670a 57616c62     keup .wak ing. Walb
2192    72696467 650a5761 6c636f74 740a5761     ridg e.Wa lcot t.Wa
2208    6c64656e 0a57616c 64656e73 69616e0a     lden .Wal dens ian.
2224    57616c64 6f0a5761 6c646f72 660a5761     Wald o.Wa ldor f.Wa
2240    6c64726f 6e0a7761 6c65730a 57616c66     ldro n.wa les. Walf
2256    6f72640a 57616c67 7265656e 0a77616c     ord. Walg reen .wal
2272    6b0a7761 6c6b6564 0a77616c 6b65720a     k.wa lked .wal ker.
2288    77616c6b 6572730a 77616c6b 696e670a     walk ers. walk ing.
2304    77616c6b 730a7761 6c6c0a57 616c6c61     walk s.wa ll.W alla
:    :    :
```

Cool, is it Allocated to a File?

```
# dstat -f linux-ext2 dev_sda1.dd 170388
Fragment: 170388
Allocated
Group: 5
```

OK, So Which Inode Does It Belong To?

```
# ifind -f linux-ext2 -d 170388 dev_sda1.dd
69739
```

What do the File's Attributes Look Like?

```
# istat -f linux-ext2 dev_sda1.dd 69739
inode: 69739
Allocated
Group: 4
uid / gid: 0 / 0
mode: -rw-r--r--
size: 409305
num of links: 1

Inode Times:
Accessed:        Tue Feb 17 19:47:53 2004
File Modified:   Tue Feb 17 19:47:53 2004
Inode Modified: Sun Jun 13 23:13:18 2004

Direct Blocks:
170290 170291 170292 170293 170294 170295 170296 170297
170298 170299 170300 170301 170303 170304 170305 170306
   :      :      :
```

What File Name Maps to that Inode?

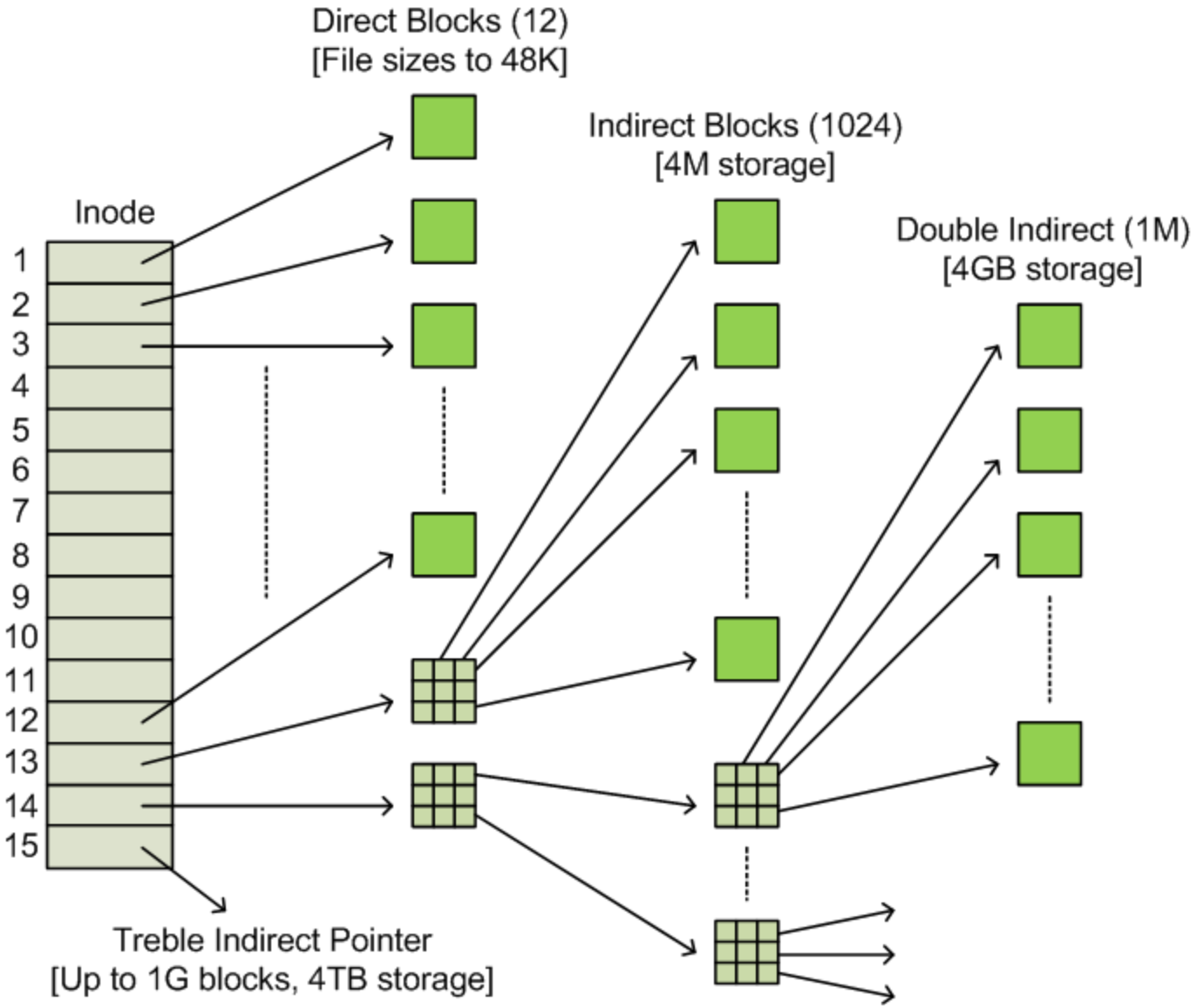
```
# ffind -a -f linux-ext2 dev_sda1.dd 69739
/usr/share/dict/linux.words
```

Linux Deleted Files

- A combination of the operating system and the file system determine the mechanics of file deletion.
  - EXT2 simply returns the inode to the free inode list in the superblock WITHOUT clearing its contents.
    - This makes complete file recovery easy! (use **`icat`**)
  - EXT3 clears the inodes contents (pointers to disk blocks) before returning to the free inode list.
    - This makes complete file recovery very difficult (though not always impossible)

`foremost`

- Excellent tool for finding popular (binary) file types (documents, images, ZIP, etc)
- Extensible and relatively fast
- Be sure to use **`-d`** option when searching Unix file systems!

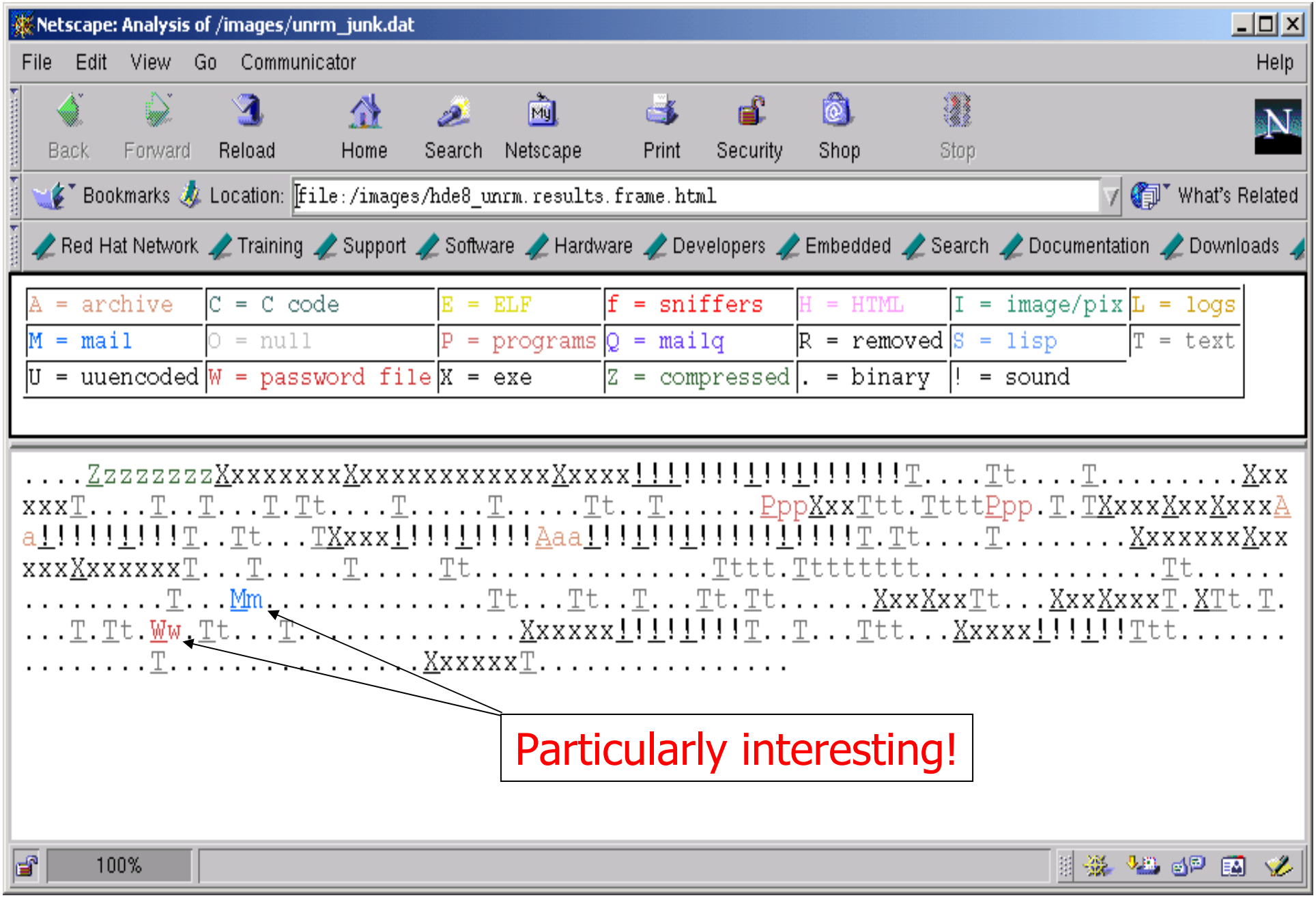

Direct Blocks (12)
[File sizes to 48K]
Indirect Blocks (1024)
[4M storage]
Double Indirect (1M)
[4GB storage]
Inode
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Treble Indirect Pointer
[Up to 1G blocks, 4TB storage]

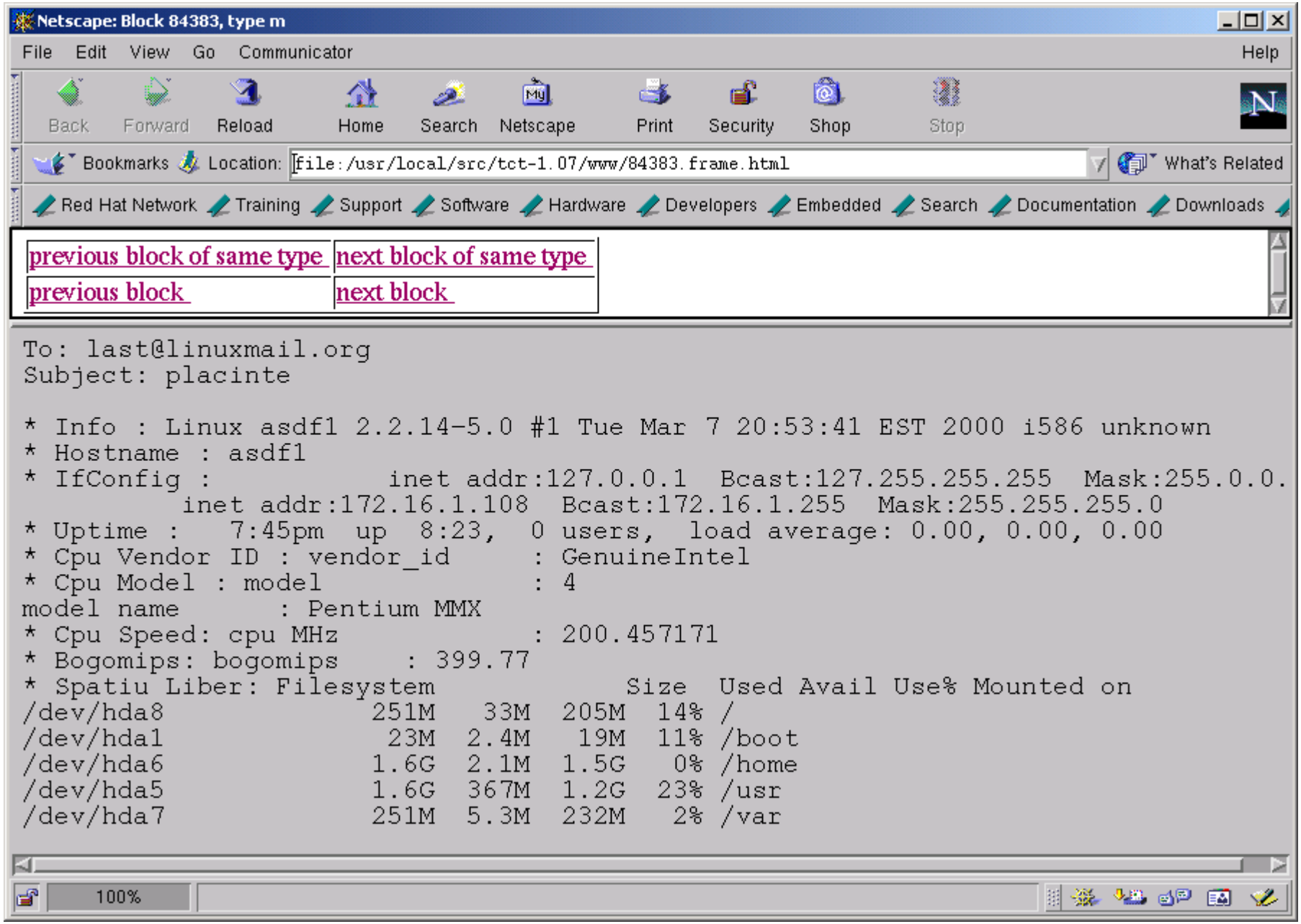
lazarus

- Analyzes raw data from dls, swap space, anything else…
- Attempts to identify disk blocks
  - Like "file" for disk blocks! ☺
- Provides a browser based display
- Think slow, slow, slow
- Part of TCT

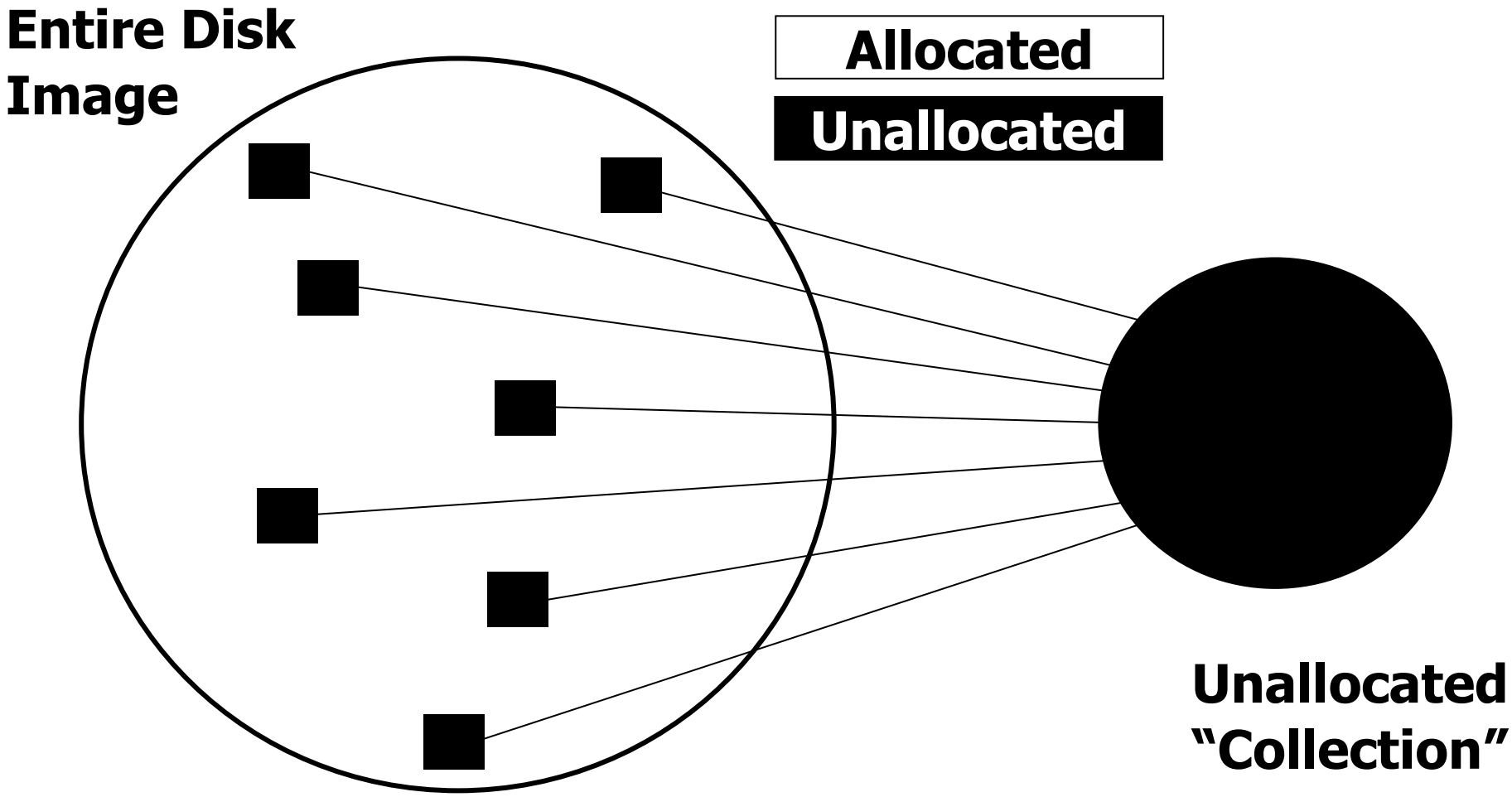
Using `lazarus`

- Use some method to collect raw disk data
  - Swap space, dd, dls
  - dls is preferable to dd because it reduces the amount of data to analyze
- Run lazarus with the –h flag to produce html
- Wait, wait, wait, then wait some more
- Lazarus creates two directories
  - www
  - blocks
- Use browser to view (*inputfilename*.frame.html)


Netscape: Analysis of /images/unrm_junk.dat
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: file:/images/hde8_unrm.results.frame.html What's Related
Red Hat Network Training Support Software Hardware Developers Embedded Search Documentation Downloads
A = archive C = C code E = ELF f = sniffers H = HTML I = image/pix L = logs
M = mail O = null P = programs Q = mailq R = removed S = lisp T = text
U = uuencoded W = password file X = exe Z = compressed . = binary ! = sound
Particularly interesting!
100%


Netscape: Block 84383, type m
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: file:/usr/local/src/tct-1.07/www/84383.frame.html What's Related
Red Hat Network Training Support Software Hardware Developers Embedded Search Documentation Downloads
previous block of same type
next block of same type
previous block
next block
To: last@linuxmail.org
Subject: placinte
* Info : Linux asdf1 2.2.14-5.0 #1 Tue Mar 7 20:53:41 EST 2000 i586 unknown
* Hostname : asdf1
* IfConfig : inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.
inet addr:172.16.1.108 Bcast:172.16.1.255 Mask:255.255.255.0
* Uptime : 7:45pm up 8:23, 0 users, load average: 0.00, 0.00, 0.00
* Cpu Vendor ID : vendor_id : GenuineIntel
* Cpu Model : model : 4
model name : Pentium MMX
* Cpu Speed: cpu MHz : 200.457171
* Bogomips: bogomips : 399.77
* Spatiu Liber: Filesystem Size Used Avail Use% Mounted on
/dev/hda8 251M 33M 205M 14% /
/dev/hda1 23M 2.4M 19M 11% /boot
/dev/hda6 1.6G 2.1M 1.5G 0% /home
/dev/hda5 1.6G 367M 1.2G 23% /usr
/dev/hda7 251M 5.3M 232M 2% /var
100%

Allocated VS Unallocated



Unallocated "Collections" with `dls`

- The **`dls`** tool *lists* content from data units and is most useful for the extraction of unallocated data.
  - Copies the block content to STDOUT
  - By default, only copies unallocated data.

- To create an unallocated subset image:

```
# dls -f linux-ext3 dev_sda1.dd >dev_sda1.dls
```

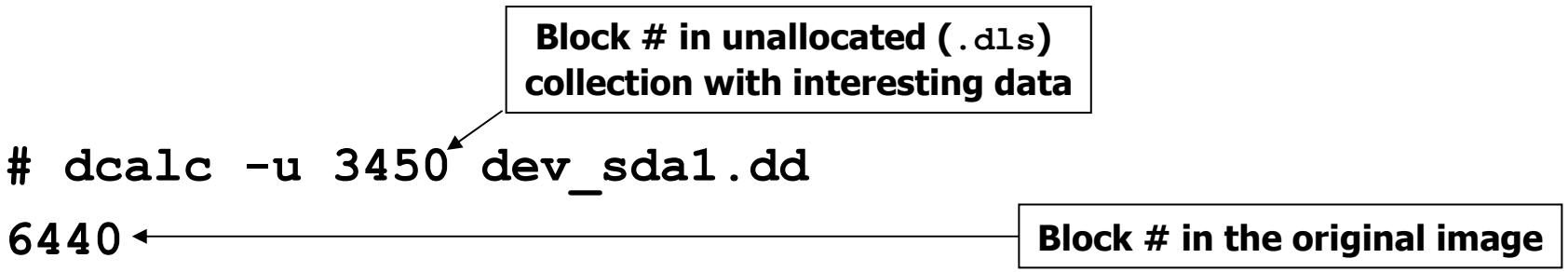
Note: This was called 'unrm' in TCT

Kicking It Old School
`dls` and `dcalc` Examples

- To extract all unallocated data:

```
# dls -f linux-ext2 dev_sda1.dd > dev_sda1.dls
```

- Use **`grep`** to find "interesting" blocks
- Once we find data in the **`.dls`** image, we need to figure out where it was in the original
- The **`dcalc`** tool maps between the two:



Conclusions

- This is just the tip of the iceberg
- Seek professional help!
- 30% preparation, 70% perspiration
- Requires diligence and persistence

http://sansforensics.wordpress.com/author/halpomeranz/

http://www.deer-run.com/~hal/

Thanks for listening!