

# File System Forensics

[jims@bluenotch.com](mailto:jims@bluenotch.com)

# Today's Agenda

- File System Forensics for ext2/3, ufs1/2
  - Getting to the File System
    - Partitions and Volumes
    - Acquisition Tools
  - Media Analysis
    - Sleuth Kit tools
    - Methodology
  - Gotchas
    - Things that go boom and ruin your day

# Getting to the File System

- Live – Helix ISO
  - Sometimes needed, but you damage evidence and risk making/letting things get worse
  - Acquire with OS tools (netcat and dd)
  - Third Party tools (dcfldd, other static compiled)
- Dead – Helix ISO, too
  - Easier, the dead stop complaining
  - Repeatable
  - Can split up tasks

# Live Acquisitions

- Try memory first (best case)
  - memdump or dd the kernel memory
  - send to save place (`nc -l -p 1111 > sda.img`)
  - `dd if=/dev/kmem | nc 10.10.10.10 1111`
- Acquire by device (in order of most volatile)
  - `dd if=/dev/sdd` (assuming `sdd1` has `/var`)
  - Getting partitions instead of devices might miss something, plus we can break them out later

# Dead Acquisitions

- Sometimes you can't get there right away
  - Have the local pull the plug so the problem doesn't get worse—don't shutdown gracefully
  - Can duplicate with faster hardware
  - Sometimes you don't have the right tools live
    - Box is weird, Pwned, or just really hardened
    - “bits is bits” on a dead system
- `dd if=/dev/sda of=/mnt/sdb1/sda.img`

# Making the image

- Do more than dd “if=<source> of=<dest>”
  - add conv=noerror to pass bad sectors
  - add conv=sync to put zeroes in place of bad
  - pipe through gzip or bzip2 for compressed image  
dd if=/dev/sda | gzip | nc 10.10.10.10 1111
  - collect 1 MB at a time
    - dd if=/dev/sda bs=1024 count=1024
    - dd if=/dev/sda bs=1024 count=1024 skip=1024
- Use a better dd (dfcldd or dc3dd)
  - hashwindow=0 (calculate md5 sum of all bits)

# Exceptions to think about

- Software RAID / LVM / md is a huge pain
  - Good luck reconstructing if you missed a portion of the image, better off acquiring logical devices
  - `dd if=/dev/sda1`
- Hardware RAID? Keep the controller with the drives
- Did you manufacturer steal some storage?
  - HBA, DCO
  - can unlock with proper tools

# Making sense of the partitions

- mmls from The Sleuth Kit is your friend
  - tells you start, end, and type of each part.
- Or just break out the specifications and start counting ones and zeroes
- Use dd again to carve out the partition from the image
- `dd bs=512 skip=63 count=15952482 if=/mnt/sdb1/sda.img`

# Using the logical images

- mount's loop option lets you mount a file
- be sure to do everything possible so you don't accidentally change the evidence
- now if you had a clue to investigate you can

```
#mount -t ext2 /mnt/sdb1/sda1.img /mnt/a/1 -o  
loop,ro,noatime,noexec,nosuid,noatime,nodev
```

# How to look at unallocated space

- Using the raw image we can look for ASCII strings  
`-radix=d /mnt/sdb1/sda1.img /`  
`> /mnt/sdb1/sda1.str`
- Now build a “dirty-word list”
  - any keywords associated with the investigation
  - grep is also your friend, but you already knew  
`grep -f words.txt -i /mnt/sdb1/sda1.str`
  - Now you have the offset (in bytes) where the match is in the raw image

# Finding where the match was

- The Sleuthkit tools help us work with all layers of media analysis:
- First letter is the “layer” rest is function
  - output commands: dcat, icat, fcat, jcat
  - list commands: mmls, dls, ils, fls, jls
  - stat commands: fsstat, dstat, istat
  - find commands: ifind, ffind
  - a few others: hfind, mactime, sorter

# Using The Sleuth Kit (TSK) on Data

- List unallocated data blocks
  - `dls /mnt/sdb1/sda1.img > /mnt/sdb1/sda1.dls`
- Calculate where in unallocated space our hit from our wordlist was
  - if sector size is 512 (normal), divide the byte offset by 512 to get block offset (drop remain.)
  - `dcalc -u <blockoffset> /mnt/sdb1/sda1.img`
    - Now we know where in the image the deleted match was, investigate the area

# Crossing Layers with TSK

- Now we know a block offset we can operate on the datablock layer for a while, guessing this file is 1K in size we might try:
  - `dcat /mnt/sdb1/sda1.img 135 2 > somefile.dat`
- We can find the inode that this block belongs
  - `ifind -o 135 /mnt/sdb1/sda1.img`
  - inodes are metadata (ie:change,access,modify)
- We can find the filename if it exists:
  - `ffind -i 3254 /mnt/sdb1/sda1.img`

# When flying blind . . . .

- If we knew what to look for, that's not too bad
- To get a good idea of what happened, let's make a timeline of changes to the filesystem
- list out dates and times of inodes and files, merged into one big document
- like tracks in the snow, if somebody else stomped over them you only see the last time/date

# Creating a “body” for a timeline

- List out all inodes for each filesystem image
  - use a bunch of ils, fls commands with -m to get mactimes
  - `ils -m /mnt/sdb1/sda1.img > /mnt/sdb1/sda1.ils`
  - `ils -m /mnt/sdb1/sda2.img > /mnt/sdb1/sda2.ils`
  - `fls -m /mnt/sdb1/sda1.img > /mnt/sdb1/sda1.fl`
  - `fls -m /mnt/sdb1/sda2.img > /mnt/sdb1/sda2.fl`
  - `cat /mnt/sdb1/sda?.?ls > /mnt/sdb1/body.dat`

# Creating the actual timeline

- `mactime -b /mnt/sdb1/body.dat > timeline.txt`
- could have extracted the password and group files and mactime can match uid, guid to username, groupnames (sometimes misleading)
- Now we have the closest thing to a timeline (will have gaps if a file was touched twice)

# Getting Forensics Done

- A lot of data to go through
- Finding a new clue sometimes means you reprocess from an earlier image
- Autopsy is a web GUI to TSK
- Many commercial options (FTK, Encase, etc.)
- Practice, take notes, think about what you want to look for and what each command you type actually does (assumptions kill)

# Overall Gotchas

- Lenovo likes to write stuff to the last 512 bytes of a drive
- Drive encryption complicates things, but an Enterprise can escrow keys for emergencies
- Knowing how an OS allocates data to a filesystem can be useful
- EXT/UFS larger files use indirect blocks
  - Is that a datablock or addresses for more data?

# Summary

- You must get your hands dirty if you want to learn - TSK is free and powerful
- Think about what you are doing, a simple mistake can drive you crazy
- Practice on random systems on random gear
- Virtualization can help you get started
- Read!

# References / For More Information

- SANS Security 508, written by Rob Lee and Brian Carrier (legal by Richard Salgado)
- “File System Forensics” by Brian Carrier
- <http://www.sleuthkit.org>
- <http://www.cooldrives.com>
- <http://www.geeks.com>
- <http://www.e-fense.com/helix/>