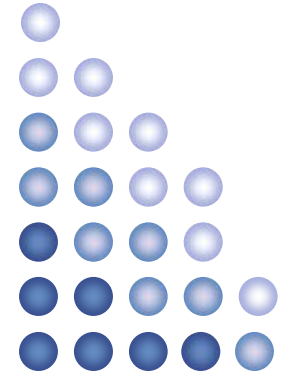
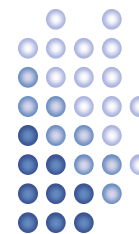


Computer Forensics: Basics



Lecture 6b
Evidence Acquisition

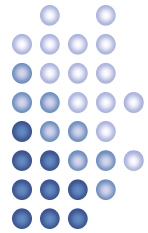
Agenda



- Objectives
- Why use images?
- Bitstream vs. backups
- Forensic imaging tools
- Forensic imaging methods (disk to disk, network)
- Preserving volatile data
- Lab: Evidence Acquisition

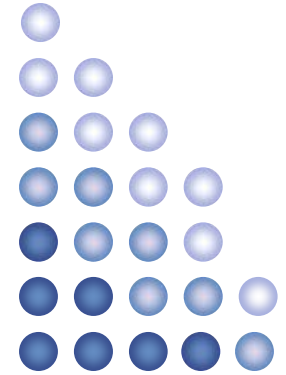


Learning Objectives



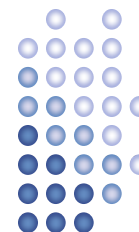
- At the end of this module, you will be able to:
 - Describe the difference between a forensic copy and a backup;
 - Explain the importance of capturing the “truest” state of the media as possible with today’s technology;
 - Describe the accepted procedure to ensure integrity of the images;
 - Discuss the issues surrounding data acquisition;
 - Demonstrate mastery of the topic by actually acquiring a forensic image.

The imaging process



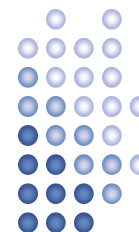
“Look Ma no DNA”

Rules of Thumb



- Make 2 copies of the original media
 - 1 copy becomes the working copy
 - 1 copy is a library/control copy
 - Verify the integrity of the copies to the original
- The working copy is used for the analysis
- The library copy is stored for disclosure purposes or in the event that the working copy becomes corrupted
- If performing a drive to drive imaging (not an image file) use clean media to copy to!
 - Shrink wrapped new drives
 - Next best, zero another drive
- Verify the integrity of all images!

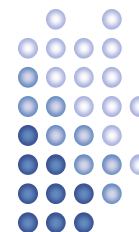
Statistics



- 69% of users use disk images rather than disk copies and 20% use partition images.
- 48% of copies and images are made in the field and 36% are made in laboratories.
- 57% of the drives imaged are larger than 8.4GB and 35% are less than that size.
- 50% of the drives imaged require IDE BIOS/Extended BIOS access and 63% require direct (ASPI) SCSI access.
- 25 to 33% of users sometimes mix IDE and SCSI drives in making images or copies, 25% often do so, and 13% always do.

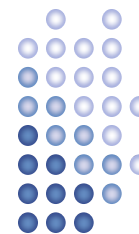
**Source NIST CFTT Project

Forensic Boot Disk



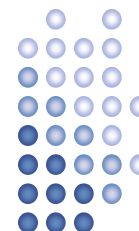
- General principles:
 - Used to boot suspect systems **safely**
 - Contains a filesystem and statically linked utilities (e.g., ls, fdisk, ps, nc, dd, ifconfig, etc.)
 - Recognizes large partitions (+2 or + 8 Gb)
 - Places the suspect media in a locked or read-only state
 - Does not swap any data to the suspect media

Forensic Boot Disk



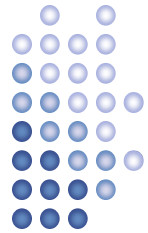
- Open source bootable images:
 - FIRE
(<http://biatchux.dmzs.com/?section=main>)
 - Linuxcare Bootable Business Cards
(<http://www.lnx-bbc.org/>)
 - Helix -SANS/NW3C using this
(<http://www.e-fense.com/helix/index2.html>)
 - Penguin STD

Drive Imaging



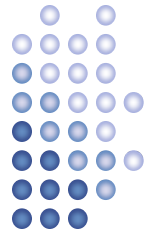
- We will consider the following 2 scenarios
 - System is off
 - System is live
- Examples will use open source tools (dd & netcat)
- What circumstances may require the system to remain live?
 - Software RAID
 - Manually mounted volumes/filesystems
 - ??

Tools - Disk Dump (dd)



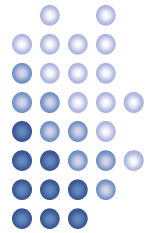
- dd is a tool for making disk images in the UNIX world
- dd has many flags suited to copying block-oriented devices such as tapes etc.
- Basic syntax:
 - dd if=(source) of=destination
 - if = input file, or evidence you want to copy (hard drive, diskette, tape etc.)
 - of = output file, where you want the evidence to be stored
 - example:
 - **dd if=/dev/hda of=/forensics/images/case1.dd**
 - This creates an image named case1.dd located in the forensics/images directory

Tools - Netcat



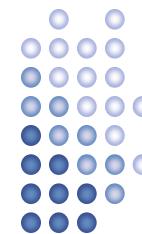
- Netcat
- Designed in 1995 as a network debugging tool
 - Some of the features of netcat are:
 - Outbound or inbound connections, TCP or UDP, to or from any ports
 - Full DNS forward/reverse checking, with appropriate warnings
 - Ability to use any local source port
 - Ability to use any locally-configured network source address
 - Built-in port-scanning capabilities, with randomizer
 - Built-in loose source-routing capability
 - Can read command line arguments from standard input
 - Slow-send mode, one line every N seconds
 - Optional ability to let another program service inbound connections

Drive Imaging



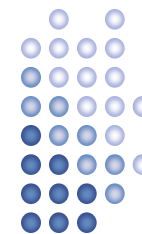
- System is off
 - Disk to Disk imaging
 - Imaging over the NIC

Imaging: Disk to Disk



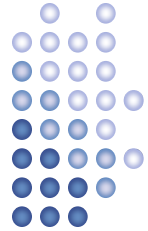
- Step 1:
 - Assumes the scene and system have been properly secured.
 - Remove suspect hard drive from suspect system
 - Place in mobile forensic system (MFS) running Linux or UNIX
- or**
- Connect power cable and ribbon from (MFS) to suspect drive
- Step 2:
 - Boot the MFS
 - Ensure that the suspect drive is recognized and is either not mounted or mounted READONLY.

Imaging: Disk to Disk



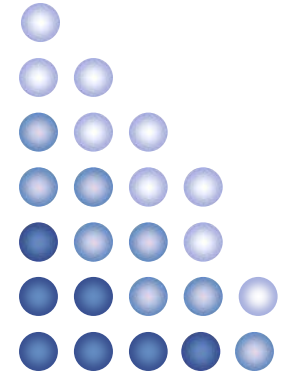
- Step 3
 - Create bitstream image of the suspect drive
 - `dd if=/dev/hdb of=/forensics/images/case1.dd`
 - Assumes hdb corresponds to the dev name of the suspect drive.
- Step 4
 - Ensure integrity of source and image
 - md5sum provides a 128 bit signature that is sensitive to bit changes.
 - **md5sum /dev/hdb**
a2c9e26fb92276cf57a59293401514b9 /dev/hdb
 - **md5sum /forensics/images/case1.dd**
a2c9e26fb92276cf57a59293401514b9 /forensics/images/case1.dd
 - The reported hashes should match

Imaging: Disk to Disk



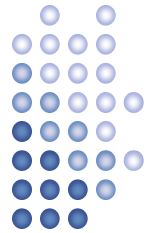
- Step 5
 - Disconnect suspect drive
 - Shut MFS down
 - Make detailed notes

Network based



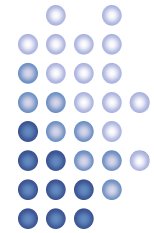
"Warp 9 Scotty"

Imaging – Network based



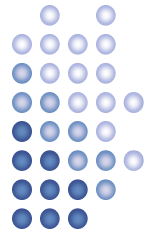
- Step 1
 - Connect cross over cable or hub to suspect & MFS
 - Boot suspect system from forensic boot disk
 - Start up the MFS (running Linux/UNIX)
 - Set IP addresses for both systems
 - `ifconfig eth0 10.1.1.2 netmask 255.255.255.0`
 - `ifconfig eth0 10.1.1.3 netmask 255.255.255.0`
 - Ping the one system to ensure connectivity
 - Verify the date & time reported on the suspect & MFS systems (Why is this NB)

Imaging – Network based



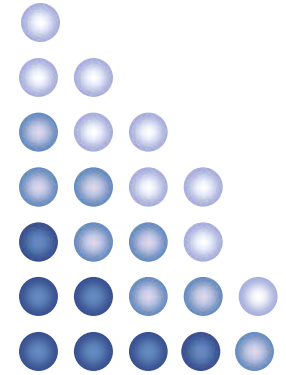
- Step 2
 - Listening host (MFS) run netcat in listening mode
 - `nc -l -p 10000 > /forensics/images/case1.dd`
 - -l = listening mode
 - -p = port address
 - > pipes the input to the specified file
 - Suspect Host
 - **`dd bs=1024 < /dev/hda1 | nc 10.1.1.3 10000 -w 3`**
 - Run dd set block size to 1024
 - Pipe the dd input (/dev/hda1) through netcat to the ip address 192.168.1.2 on port 10000

Imaging – Network based



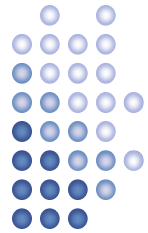
- Step 3
 - Ensure integrity of source and image (md5sum)
 - Hash totals should match
- Step 4
 - Shut down the MFS and the suspect system
 - Remove forensic boot disk
 - Disconnect cables etc.
 - Make detailed notes.

Live file system



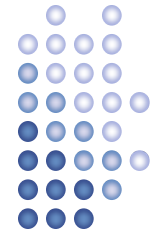
“There be dragons!”

Drive Imaging – Live System



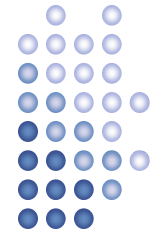
- Assumptions for our example
 - Suspect system is a UNIX filesystem
- Document everything!
- Use statically linked binaries
 - Diskette, CD-ROM
 - <http://www.incident-response.org/irtoolkits.htm>
- Tools = dd & Netcat

Drive Imaging – Live System



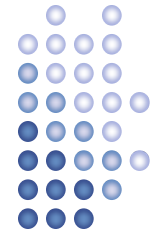
- Step 1
 - Connect cross over cable or hub to suspect & MFS
 - Start up the MFS
 - Set IP addresses for both systems
 - `ifconfig eth0 10.1.1.2 netmask 255.255.255.0`
 - `ifconfig eth0 10.1.1.3 netmask 255.255.255.0`
 - Ping the one system to ensure connectivity
 - Verify the date & time reported on the suspect & MFS systems (Why is this NB)
 - Mount CD with statically linked binaries
 - `#!/mount /dev/hdc /mnt`

Drive Imaging – Live System



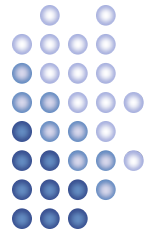
- Step 2
 - Use netcat & dd to image systems
 - Netcat syntax:
 - Listening host (system we are going to store the image on)
 - `nc -l -p 10000 > /forensics/images/case1.dd`
 - -l = listening mode
 - -p = port address
 - > pipes the input to the specified file
 - Suspect host (system we want to image)
 - From the CDROM!
 - `nc <ip address of listening host> <port number> -w 3`
 - `nc 10.1.1.3 10000 -w 3`
 - -w is timeout value (in our example 3 seconds)

Drive Imaging – Live System



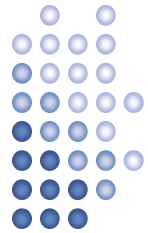
- Combining Netcat & dd
 - Listening host
 - `nc -l -p 10000 > /forensics/images/case1.dd`
 - Suspect host
 - **`dd bs=1024 < /dev/hda1 | nc 10.1.1.3 10000 -w 3`**
 - Run dd set block size to 1024
 - Pipe the dd input (/dev/hda1) through netcat to the ip address 10.1.1.3 on port 10000
 - If no data transmitted for 3 seconds then end the process
 - Our suspect image is now safely on our system and is called case1.dd

Drive Imaging – Live System



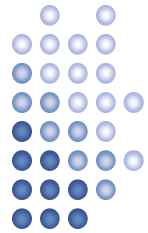
- Step 3
 - Ensure integrity of source and image (md5sum)
 - Hash totals should match
- Step 4
 - Shut down the MFS
 - Disconnect cables etc.
 - Make detailed notes.

Acquiring Volatile Data



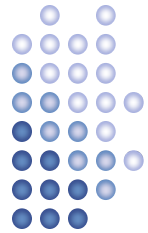
- The data that is held in temporary storage in the system's memory is called volatile data.
- The memory is dependant upon electrical power. When the power is shut off the memory is disrupted.
- Order of volatility:
 - Registers and Cache
 - Routing tables, ARP cache, process tables, kernel statistics
 - Contents of system memory
 - Temporary file systems
 - Data on disk
- Commands
 - Nstat -an (-rn)
 - Isof
 - Ifconfig
 - Ipconfig
 - pslist
 - Nbtstat
 - Top
 - Prstat
 - Arp -a

Summary



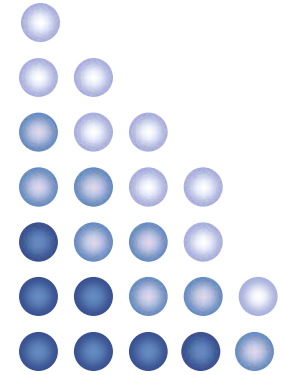
- Acquiring an exact copy of the suspect media is difficult. Bitstream copies are acceptable as long as you can demonstrate the integrity of the images.
- Acquiring digital evidence should adhere to the G8s second principle of digital forensics (actions should be taken not to change the evidence)
- There are several open source tools that can be used
- If possible avoid obtaining a image from a live system.
- Be conscious of volatile data on live systems
- Use a forensic boot disk
- Make detailed notes as the procedures you follow need to be both auditable and replicable.

Summary



- Test and validate all tools on a known system
- Re-test after upgrading to newer version
- Conduct "Before and After" comparisons
- Be prepared to testify to your methodology

Lab



“Cloning for fun and profit”