CYBER502x Computer Forensics

Unit 2: Linux/Unix Forensics Acquisition

Investigating Linux/Unix systems

- Four basic forensics steps
  - Collect
  - Preserve
  - Analyze
  - Present (report)

Preparation

- A tool box (CD or USB) containing trusted forensic tools
- A powerful machine with forensic tools installed and clean-wiped hard drive to store acquired evidence.

Remember!

- Always have your OWN tool sets ready !!
  - You are dealing with a compromised system
- Run tools from your own USB or device
- Save the output **outside** of the compromised system

Forensics tools in common

- Ensure forensically-sound operations
- Process data structure from the image bypassing kernel’s support
- Work on both images and live systems

Basic imaging steps

- Obtain volatile data (including RAM) –According to policy and the case nature
- Acquire non-volatile data (image drives and removable media)

Acquire volatile information

- System information
- Memory usage
- Running processes
- Logged in users
- Network connections
- Network interface configuration (promiscuous mode?)
- ….

Volatile Evidence

- Most volatile → Least volatile → Nonvolatile
  - Memory
  - Swap space or gagefile
  - Network status and connections
  - Processes running
  - File opening
  - Hard drive media
  - Removable media (CD, Zip, USB, etc.)

lsof

- Ls open files
  - a regular file, a directory, a block special file, a character special file, an executing text reference, a library, a stream or a network file
- Lsof [options] [filename/pid]

Command	Lists…
lsof	All open files belonging to all active processes
lsof –i [ipaddress]	Internet connections belonging to the given ipaddress
lsof -i 4 -a -p 1234	All open IPv4 network files in use by the process whose PID is 1234
lsof /dev/hda3	All open files on device /dev/hda3,
lsof /u/abe/foo	Finds the process that has /u/abe/foo open

lsof – find hidden disk spaces

- Create a process that opens a file; unlink the file; continue to write to the file
  - Disk resources remain in use
  - File is invisible to the ls command
- How to find them?
  - Use lsof +L1 #show you all open files that have a link count less than 1
- What is in this hidden file?
  - Find which process opens this file using lsof
  - cd to /proc/<PID>/cwd

Command examples to collect information from a live system

To display…	Command
Current system date and time	date
When was the system rebooted	uptime
System information	uname –a
Network interface running in promiscuous mode	ifconfig
Look for unusual processes and services	ps –eaf *or* top
Network connections	netstat –at *or* lsof –i 4
Logged in users	w *or* who *or* users
Find SUID programs	find / -uid 0 –perm -4000 2>/dev/null
Logs	more –f var/log/messages
Find executable files that were modified in one day	find /directory_path –type f –a=x –mtime -1
Display amount of free and used memory in system	free

Use netcat (or cryptcat)

- to transfer the retrieved data to a forensic workstation over the network
- Setting up the netcat listener on the forensic workstation (192.168.0.2)
  - nc –l 2222 > meaningful_Name
- Sending the info to the forensic workstation
  - who | nc 192.168.0.2 2222

Acquire RAM with physical access to the system

- Memdump for Linux, Unix, FreeBSD, Solaris
  - http://www.porcupine.org/forensics/tct.html
  - Action is not guaranteed due to a restricted range of addresses
- Linux Memory Extractor (LiME)
  - A Loadable Kernel Module for acquiring Linux/Android physical memory
  - https://github.com/504ensicslabs/lime
- Fmem
  - A kernel module fmem.ko that creates device /dev/fmem
  - sudo dd if=/dev/fmem of=mem.dd (no restriction)

F-Response for remote acquisition

- *F-Response* from https://www.f-response.com/
- Use dual-dongle to conduct remote forensic acquisition of memory and disks.
  - One dongle for subject system
  - One dongle for examiner system

Forensic imaging of hard drives

- Acquire non-volatile data
  - Bit-stream copy gets every single bit of every byte on hard drives
  - FTK Imager (covered before)
  - Many high-speed forensic Imagers exist
    - Tableau series from Guidance Software
  - Unix utility: *dd*
    - *man dd*

Tableau TD2u from Guidance Software

https://www.guidancesoftware.com/video/demo/TD2u-Informational-and-Setup

How does it work?

- Reads input blocks one @ time from block level device and puts them into a buffer (memory)
- outputs from buffer to desired location
  - Default block size = 512 bytes (4096 bits)
- Simply moving “chunks” of bits from a device to some other place
- dd will copy metadata and data blocks in their entirety (bit-by-bit) – regardless of whether they are allocated to an active file or not

Syntax

- A simple example:
  - dd if=<what-to-copy> of=<where-to-put>
- Default send to stdout
- Can redirect via PIPE to netcat or cryptcat
- dd if=/dev/fd0 | nc 192.168.1.2 2222

Options

- bs=n (bytes)
  - Input and output blocksize of n bytes
  - bs=nk (n kilobytes)
  - Larger block size (up to ~ 8k) can decrease imaging time
- ibs=q obs=r
  - Input block size q (bytes)
  - Output block size r (bytes)
- Count=s (blocks)
  - Stop after you have transferred *s* INPUT blocks of data

Options

- Carving data w/dd
  - skip=*n* (blocks)
    - skip n BLOCKs ibs-sized blocks at start of input file before copying
  - seek=*n* (blocks)
    - Skip n BLOCKs obs-sized blocks in the output file before copying

Separate each partitions

- 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

- # dd if=sda.dd skip=32 count=1884128 of=sda1.dd
- # dd if=sda.dd skip=1884160 count=212992 of=sda2.dd

Options

- ‘conv=conversion[,conversion]...'
  - Convert the file as specified by the conversion argument(s). (No spaces around any comma(s).)
- Conversions:

ascii	lcase	ucase
ebcdic	swab	noerror
ibm	notrunc	sync
block	unblock	

What about errors?

- If dd encounters an error while reading an input block, the copy process STOPS!
  - Can force it to continue (using the *conv=noerror* flag)
  - Include the *sync* along with the *noerror* flag to pad zeros in place of the errors encountered
    - *conv=noerror,sync*
- dd if=/dev/hdb1 of=/case1/hdb1.dd conv=noerror,sync

Other uses for dd

- Sterilize media
  - if=/dev/zero of=TARGET
    overwrites target with zeros
  - if=/dev/random of=TARGET
    overwrites target with random data

How do we access the source drive?

- Physically remove the drive from the suspect computer and connect it to a forensic machine (a write block should be used)
  - dd if=/dev/hda of=/dev/hdb OR
  - dd if=/dev/hda of=/case1/evidence.dd
- Imaging over a network/firewire connection
  - using dd and nc

Use nc to acquire image over the crossover cable

- Forensics machine listens on port 8888. Once data is received,
  - it is stored in a drive
    - nc –l 8888 > /dev/hdb
  - Or it is saved as an image file
    - nc –l 8888 > evidence.dd
- Suspect machine sends data
  - dd if=/dev/hda | nc ipaddr 8888 –w 3

sdd and dcfldd

- sdd
  - It is faster than dd in cases where input block size (ibs) is not equal to the output block size (obs).
  - Statistics are more easily understood than those from 'dd'.
  - It reports the number of bytes copied and how much of the last block was copied.
- dcfldd
  - by the U.S. Department of Defense computer forensics lab
  - It is an enhanced version of dd
  - It provides the option to generate hash of the transmitted data
  - It has a progress bar showing how much data has been sent

Essential Tools for Acquisition

- Advanced material if you are interested – not required
- http://malwarefieldguide.com/LinuxChapter1.html
  - Physical Memory Acquisition (locally and remotely)
  - System, user logon, network connections, process details collection
- *Malware Forensics Field Guide for Linux Systems: Digital Forensics Field Guides* by Cameron Malin Eoghan Casey, ad James Aquilina, Syngress, 2014