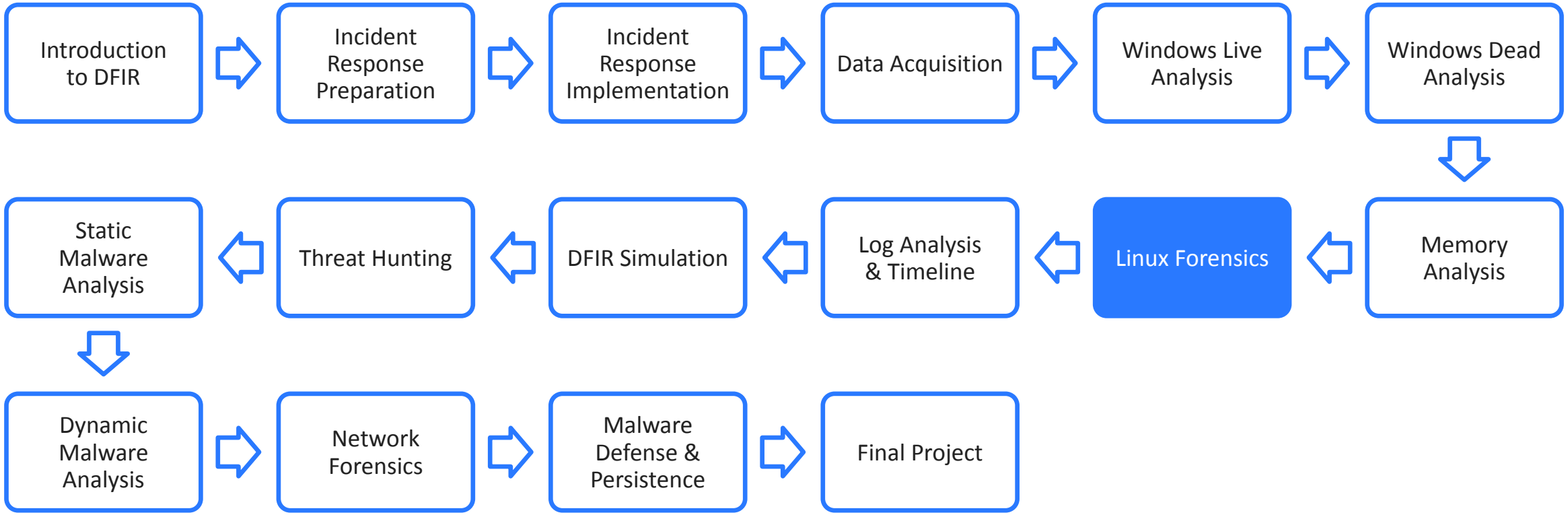
Cybersecurity Professional Program

Linux Forensics

Digital Forensics & Incident Response

Digital Forensics & Incident Response

Course Path



Objectives

The object of this lesson is to learn about the forensic methods applied during Linux live and dead analysis.

- Linux Live Forensics
- Linux Live Acquisition
- Linux File Systems
- File System Analysis
- Linux Memory Forensics
- Process Investigation

Linux Forensics

Linux Live Forensics

Linux Forensics Methods

- In Linux, everything has file representation: memory, running processes, etc.
- Dead and live analysis are similar in many aspects.
- Most Linux-based data is not binary.

Linux Forensics Acquisition

- Static binaries are used for a minimal footprint on the system.
- Binaries should be loaded from a live CD.
- In most cases, such Rescue CDs are custom made.

Various *nix tools built as statically-linked binaries

75 commits | 1 branch | 0 packages | 0 releases | 5 contributors

Branch: master | New pull request | Find file | Clone or download

andrew-d Merge pull request #15 from FireFart/socat ... Latest commit 530df97 on Jul 10, 2017

binaries	Add binary for nano	3 years ago
binutils	Switch to using separate musl-cross images	5 years ago
file/arm	Add "file" command for Linux/ARM (and magic number file)	5 years ago
ht	Switch to using separate musl-cross images	5 years ago
make	WIP on Makefile build system	5 years ago
nano	Updates nano SBUILD	3 years ago
nmap	Switch to using separate musl-cross images	5 years ago
p0f	Switch to using separate musl-cross images	5 years ago
pv/arm	Switch to using separate musl-cross images	5 years ago
python	Switch to using separate musl-cross images	5 years ago
sbuild	Add sbuild work	5 years ago
socat	switch to alpine and use current versions	3 years ago
strace/arm	Add strace for arm	5 years ago
tcpdump/arm	Add tcpdump for Linux/ARM	5 years ago
the_silver_searcher	Version bump on the src packages for the build	3 years ago

Static Binaries

- Live CDs should be mounted as RO.
- Although the binaries are static, some may rely on other binaries to work.
- Busybox is also commonly used in such cases.

```
johnd@ubuntu:~$ sudo mount -o loop,ro /dev/sr0 /mnt/
johnd@ubuntu:~$ ls /mnt/
acl-2.2.52
acpid-2.0.22
aespipe-2.4c
aircrack-ng-1.2-rc1
alsa-lib-1.0.29
alsa-utils-1.0.29
argp-standalone-1.3
arptables-0.0.4
at-3.1.13
atk-2.16.0
attr-2.4.47
audiofile-0.3.6
aumix-2.8
axel-2.4
bash-4.3.30
bc-1.06.95
```

Extraction Over NC

```
johnd@ubuntu:~$ cat /etc/os-release | /mnt/netcat-0.7.1/netcat -c
172.16.0.11 1337
johnd@ubuntu:~$
```

- To prevent writing to disk, data acquisition should be done over the network.
- Netcat is typically used to send and receive the data.
- On the compromised host, a static binary of nc (Netcat) is used.

```
sansforensics@siftworkstation -> ~
$ nc -lp 1337 > os-release.capture
sansforensics@siftworkstation -> ~
$ cat os-release.capture
NAME="Ubuntu“
VERSION="18.04.3 LTS (Bionic Beaver)“
ID=ubuntu
ID_LIKE=Debian
PRETTY_NAME="Ubuntu 18.04.3 LTS“
VERSION_ID="18.04“
HOME_URL="https://www.ubuntu.com/"
```

Linux Forensics

Linux Live Acquisition

Network Data Extraction

```
johnd@ubuntu:~$ /mnt/busybox-1.23.2/bin/netstat -ant |
/mnt/netcat-0.7.1/netcat -c 172.16.0.11 1337
johnd@ubuntu:~$
```

- Netstat can be used to obtain network information.
- Netstat can show open and established sockets.
- This is useful when attempting to identify backdoors.

```
sansforensics@siftworkstation -> ~
$ nc -lp 1337 > netstat.capture
sansforensics@siftworkstation -> ~
$ cat netstat.capture
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 ::::80                  :::*                    LISTEN
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN
tcp        0      0 ::::443                 :::*                    LISTEN
```

Process Acquisition

```
johnd@ubuntu:~$ /mnt/lsof-4.88/lsof -n -P -l | /mnt/netcat-0.7.1/netcat
-c 172.16.0.11 1337
johnd@ubuntu:~$
```

ps is used in Linux to acquire process information.

However, from a forensics perspective, ***lsof*** is better.

Lsof lists are based on the files used by processes.

```
sansforensics@siftworkstation -> ~
$ nc -lp 1337 > lsof.capture
sansforensics@siftworkstation -> ~
$ cat lsof.capture | sort -u -k1,1
accounts-  614 0 cwd unknown /proc/614/cwd (readlink: Permission denied)
acpid      608 0 cwd unknown /proc/608/cwd (readlink: Permission denied)
acpi_ther   86 0 cwd unknown /proc/86/cwd (readlink: Permission denied)
alsa-sink 1424 0 cwd unknown /proc/1424/cwd (readlink: Permission denied)
alsa-sour 1425 0 cwd unknown /proc/1425/cwd (readlink: Permission denied)
apache2   1016 0 cwd unknown /proc/1016/cwd (readlink: Permission denied)
vahi-dae  598  0 cwd unknown /proc/598/cwd (readlink: Permission denied)
bash      2425          1000  cwd       DIR                8,1     4096
3145730 /home/johnd
```

Kernel Modules

```
johnd@ubuntu:~$ cat /proc/modules | /mnt/netcat-0.7.1/netcat -c
172.16.0.11 1337
johnd@ubuntu:~$
```

Inspecting the kernel modules may reveal malicious activity.

Kernel modules can be hidden and require a more thorough investigation.

```
sansforensics@siftworkstation -> ~
$ nc -lp 1337 > kernel_modules.capture
sansforensics@siftworkstation -> ~
$ cat kernel_modules.capture
isofs 45056 2 - Live 0x0000000000000000
rfcomm 77824 4 - Live 0x0000000000000000
bnep 20480 2 - Live 0x0000000000000000
crct10dif_pclmul 16384 0 - Live 0x0000000000000000
crc32_pclmul 16384 0 - Live 0x0000000000000000
ghash_clmulni_intel 16384 0 - Live 0x0000000000000000
pcbc 16384 0 - Live 0x0000000000000000
```

File Acquisition

File extraction is possible over the network using ***dd***.

dd can copy files and partitions byte by byte.

Using piping and input redirection, the data can be sent over the network.

```
johnd@ubuntu:~$ dd < /etc/passwd | /mnt/netcat-0.7.1/netcat -c
172.16.0.11 1337
4+1 records in
4+1 records out
2511 bytes (2.5 kB, 2.5 KiB) copied, 0.000146646 s, 17.1 MB/s
```

```
sansforensics@siftworkstation -> ~
$ nc -lp 1337 > passwd.capture
sansforensics@siftworkstation -> ~
$ cat passwd.capture
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

DFIR-08-L1

Forensic Acquisition

15–25 Min.

Mission

Create a forensic acquisition CD with static binaries and use it to extract information from a Linux OS.

Steps

- Download the static binaries.
- Create an ISO image.
- Load the image in RO mode.
- Extract forensic information.

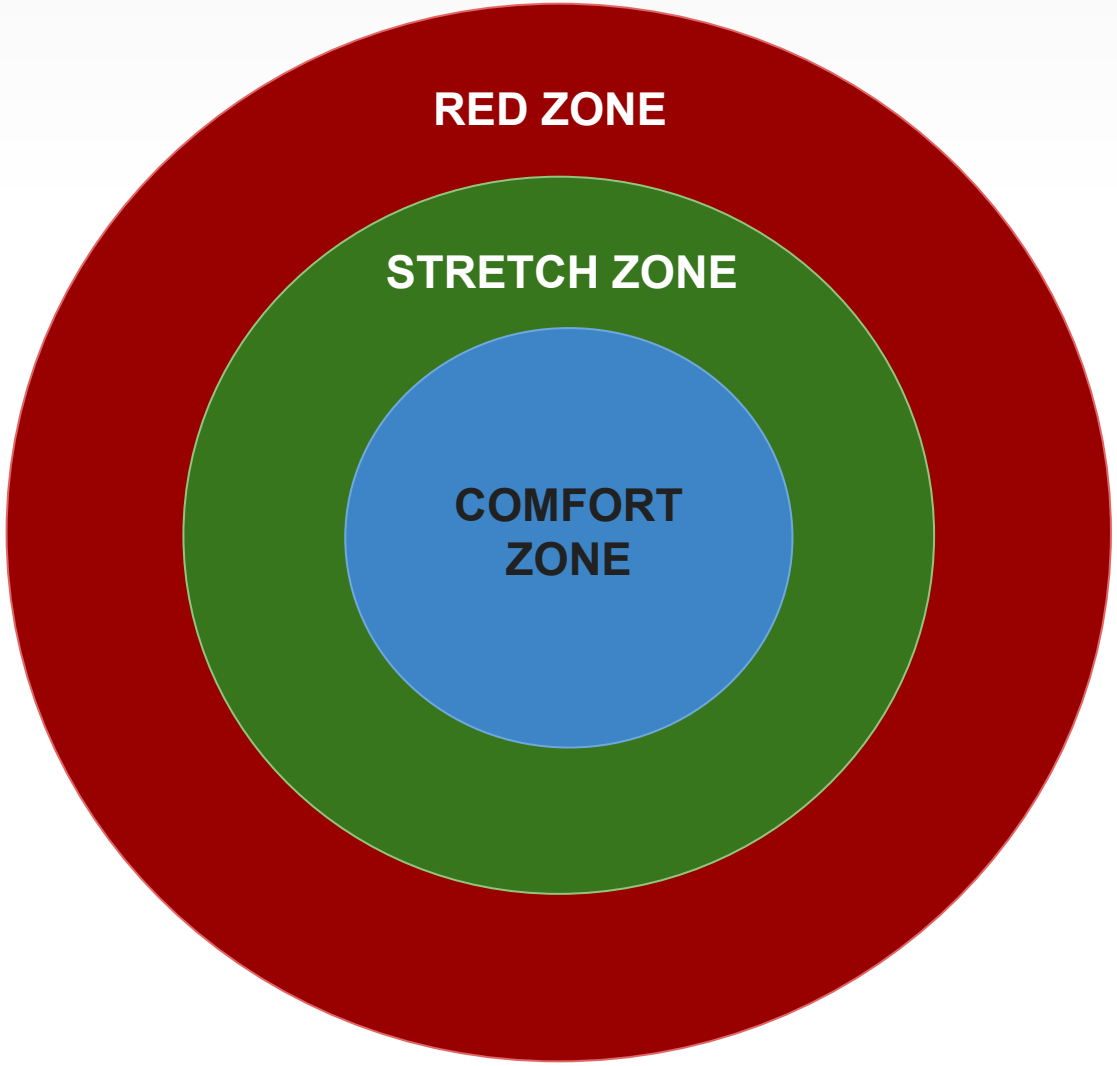
Env. & Tools

- SIFT Workstation
- Ubuntu 16.04+
- ***LinuxRescueCD.iso***

Related Files

- Lab document

Pulse Check



Linux Forensics

Linux File Systems

Different Distribution File Systems

Ubuntu

Debian

Linux Mint

Red Hat

Extended File System

- A family of file systems that includes Ext2, Ext3, and Ext4
- Ext4 is the most common file system in Linux distributions.

> Ext4 includes many features, such as journaling, space allocation, and others.

Other File Systems

XFS

- Designed to span multiple storage devices
- Divides the file system into mapped blocks of data

BTRFS

- Space-efficient file system
- Supports compression and snapshots

File System Comparison

XFS		EXT4		BTRFS
Architecture: B+ Tree		Architecture: Hashed B Tree		Architecture: Extent Based
Introduced: 1994		Introduced: 2006		Introduced: 2009
Max volume size: 8 Ebytes	VS	Max volume size: 1 Ebytes	VS	Max volume size: 16 Ebytes
Max file size: 8 Ebytes		Max file size: 16 Tbytes		Max file size: 16 Ebytes
Snapshots: Planned		Snapshots: No		Snapshots: Yes

Multiple File Systems

- At any given time, Linux hosts multiple file systems.
- Among them are tmpfs, squashfs, and others.
- The systems can be viewed using ***df –T***.

```
johnd@ubuntu:~$ df -T
Filesystem     Type      1K-blocks    Used Available Use% Mounted on
udev           devtmpfs    1985544       0   1985544   0% /dev
tmpfs          tmpfs        401592    2100    399492   1% /run
/dev/sda1      ext4       61663020 9624136  48876876  17% /
tmpfs          tmpfs       2007940       0   2007940   0% /dev/shm
tmpfs          tmpfs          5120       4      5116   1% /run/lock
tmpfs          tmpfs       2007940       0   2007940   0% /sys/fs/cgroup
/dev/loop0     squashfs      91264   91264         0 100% /snap/core/7917
/dev/loop1     squashfs     144128  144128         0 100% /snap/gnome-3
...
tmpfs          tmpfs        401588      16    401572   1% /run/user/120
tmpfs          tmpfs        401588      32    401556   1% /run/user/1000
/dev/sr0       iso9660     1214060 1214060         0 100% /media/CDROM
/dev/loop19    iso9660     1214060 1214060         0 100% /mnt
johnd@ubuntu:~$
```

Linux Forensics

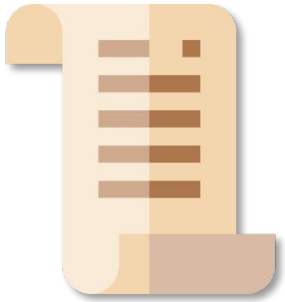
File System Analysis

Image Mounting

- Captured images can be mounted directly in Linux.
- The ***losetup*** command is used to create a loop device.
- Loop devices can be mounted the same way as other devices.

```
sansforensics@siftworkstation -> ~
$ sudo losetup -f -P Documents/Samples/dd/Web_Server.dd
sansforensics@siftworkstation -> ~
$ sudo mount -o loop,ro -t ext4 /dev/loop0p1 /mnt/dd/
sansforensics@siftworkstation -> ~
$ ls -la /mnt/dd/
total 88
drwxr-xr-x  19 root root  4096 Jan 16 09:49 .
drwxr-xr-x  20 root root  4096 Jan 16 20:02 ..
lrwxrwxrwx   1 root root     7 Jan 16 09:43 bin -> usr/bin
drwxr-xr-x   3 root root  4096 Jan 16 09:50 boot
drwx------   2 root root  4096 Jan 16 09:49 .cache
drwxr-xr-x   4 root root  4096 Jan 16 09:43 dev
drwxr-xr-x 119 root root  4096 Jan 16 10:02 etc
drwxr-xr-x   3 root root  4096 Jan 16 09:54 home
lrwxrwxrwx   1 root root     7 Jan 16 09:43 lib -> usr/lib
lrwxrwxrwx   1 root root     9 Jan 16 09:43 lib32 -> usr/lib32
lrwxrwxrwx   1 root root     9 Jan 16 09:43 lib64 -> usr/lib64
lrwxrwxrwx   1 root root    10 Jan 16 09:43 libx32 -> usr/libx32
drwx------   2 root root 16384 Jan 16 09:43 lost+found
...
```

Journaling



- Keeps track of changes not yet committed to the file system
- Journaling can be done on an entire file or just its metadata.
- Was introduced in Ext3 and improved in Ext4

JLS and JCAT

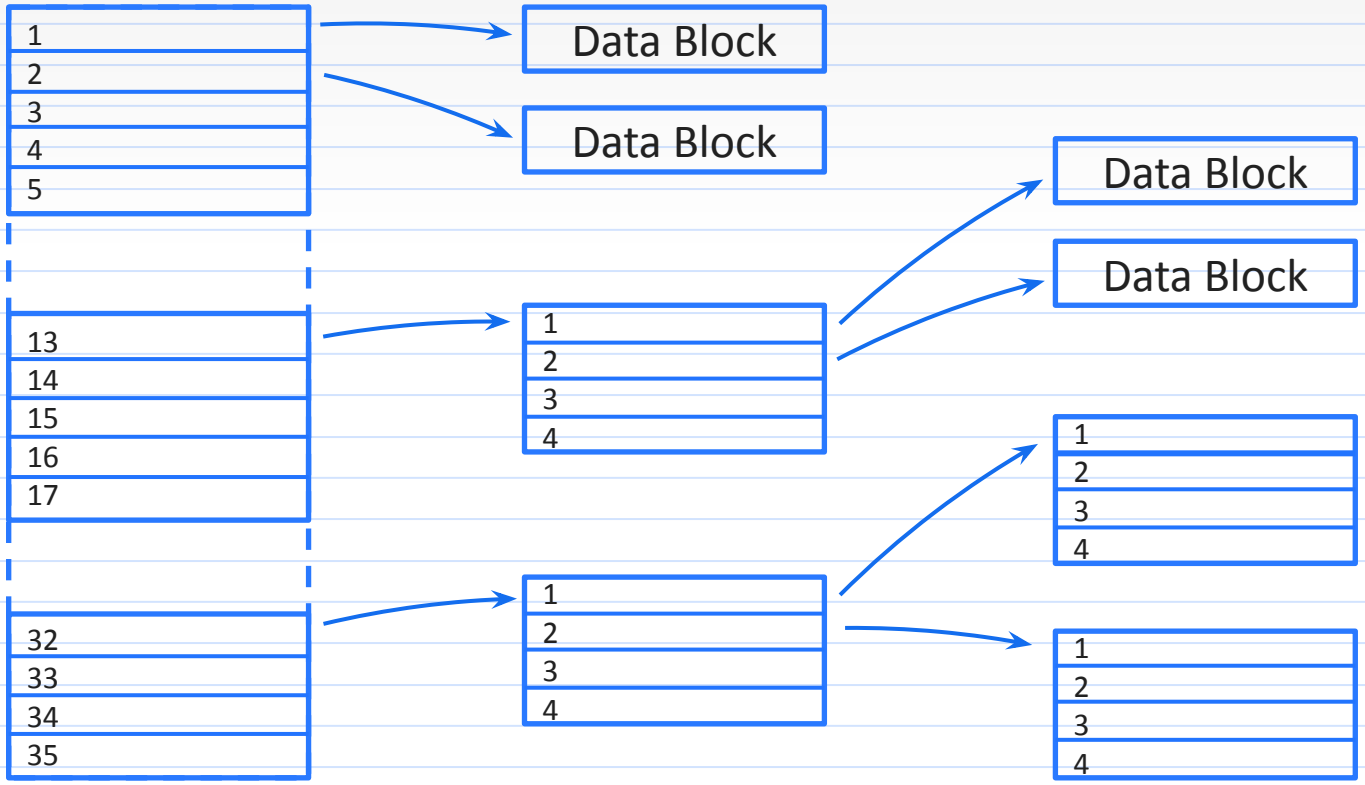
- The Sleuth kit provides tools to inspect the journal.
- JLS lists all the blocks, while JCAT prints information of a given block.
- The block data is usually unreadable but may include file names.

```
sansforensics@siftworkstation -> ~
$ jcat -f ext4 -o 2048 Web_Server.dd 4009
⍰s
   .⍰2
         ..⍰s
xmlrpc.php⍰swp-blog-header.php⍰s
wp-signup.php⍰s      index.php⍰s readme.html⍰s,
                                    wp-cron.php7r .htaccessg.php.swphp⍰s

wp-login.php⍰swp-settings.php⍰s
                     license.txt⍰s
wp-contentLt
               wp-mail.phpMtwp-links-opml.phpNt
                                                    wp-load.phpOt

wp-includes⍰vwp-activatwp-config.phpminxwp-trackback.phxwp-comments-post.php⍰s4
              ⍰⍰%:
```

Inode Structure

Inodes are the Linux equivalent of MFT.

They map files to the system without file names and include time stamps.



Inodes

Inodes can be viewed using the ***ils*** and ***ffstat*** commands.

By default, ***ils*** only displays deleted nodes.
Inodes can be viewed more elaborately on live systems.

```
sansforensics@siftworkstation -> ~
$ ils -f ext4 -o 2048 Documents/Samples/dd/Hacked.dd
class|host|device|start_time
ils|siftworkstation||1579367260
st_ino|st_alloc|st_uid|st_gid|st_mtime|st_atime|st_ctime|st_crtime|st_mode|st_nlin
|st_size
22344|f|0|0|1579364887|1579364887|1579365328|1579364887|644|0|0
22345|f|0|0|1579169605|1579169605|1579169605|1579169605|755|0|0
22353|f|0|0|1579168209|1579168205|1579168209|1579168203|644|0|0
22354|f|0|0|1579168209|1579168203|1579168209|1579168203|755|0|0
22355|f|0|0|1579168209|1579168204|1579168209|1579168203|644|0|0
22356|f|0|0|1579168209|1579168205|1579168209|1579168203|644|0|0
22357|f|0|0|1579168209|1579168205|1579168209|1579168203|644|0|0
22358|f|0|0|1579168209|1579168204|1579168209|1579168203|644|0|0
22359|f|0|0|1579168209|1579168205|1579168209|1579168203|644|0|0
22360|f|0|0|1579168209|1579168203|1579168209|1579168203|755|0|0
22361|f|0|0|1579168209|1579168204|1579168209|1579168203|644|0|0
22362|f|0|0|1579168209|1579168205|1579168209|1579168203|644|0|0
22363|f|0|0|1579168209|1579168205|1579168209|1579168203|644|0|0
22364|f|0|0|1579168209|1579168205|1579168209|1579168203|644|0|0
22365|f|0|0|1579168209|1579168203|1579168209|1579168203|755|0|0
```

File System Debugging

Linux has a special utility to debug file systems called **debugfs**.

debugfs can also be used to recover files.

```
johnd@ubuntu:~$ echo "data" > file
johnd@ubuntu:~$ ls -li file
3147090 -rw-r--r-- 1 johnd johnd 5 Jan 15 10:19 file
johnd@ubuntu:~$ sudo debugfs /dev/sda1
debugfs 1.44.1 (24-Mar-2018)
debugfs: logdump -i <3147009>
Inode 3147009 is at group 384, block 12582984, offset 0
Journal starts at block 1, transaction 1137048
  FS block 12582984 logged at sequence 1137091, journal block 3298 (flags 0x2)
    (inode block for inode 3147009):
    Inode: 3147009   Type: regular        Mode:  0600   Flags: 0x80000
    Generation: 2008725915    Version: 0x00000000:00000001
    User:     0   Group:     0   Project:     0   Size: 1612
    File ACL: 0
    Links: 1   Blockcount: 8
    Fragment:  Address: 0    Number: 0    Size: 0
     ctime: 0x5c2dd781:ebf6c1e0 -- Thu Jan  3 01:36:01 2019
     atime: 0x5c2dd781:eb02a058 -- Thu Jan  3 01:36:01 2019
     mtime: 0x5c2dd781:ebf6c1e0 -- Thu Jan  3 01:36:01 2019
    crtime: 0x5c2dd781:eb02a058 -- Thu Jan  3 01:36:01 2019
...
```

DFIR-08-L2

Server Investigation

15–20 Min.

Mission

Investigate a server image and identify the deleted or modified files.

Steps

- Mount the hacked image.
- Identify the deleted files.
- Identify the modified files.

Env. & Tools

- SIFT Workstation

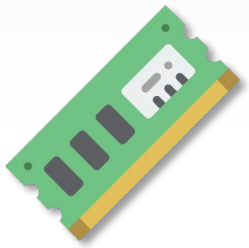
Related Files

- Lab document
- ***DFIR-08-L2 Hacked.rar***

Linux Forensics

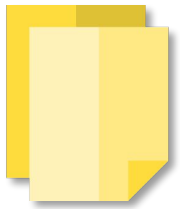
Linux Memory Forensics

Linux Memory



RAM

- Linux uses RAM in a similar way to Windows.
- RAM can be investigated using Volatility.



SWAP

- The Linux equivalent to page file
- Can be a file or an entire partition

> The ***swapon -s*** command can be used to check the location of the swap.

fmem Kernel Module

One way to create a memory dump is to use the ***fmem*** kernel module.

The kernel module creates ***/dev/fmem***, which can be captured.

Because the memory is dynamic, issues may arise when using ***dd***.

```
johnd@ubuntu:/opt/fmem$ sudo ./run.sh
Module: insmod fmem.ko a1=0xffffffffbd098030 : OK
Device: /dev/fmem
----Memory areas: -----
reg00: base=0x000000000 (    0MB), size= 2048MB, count=1: write-back
reg01: base=0x080000000 ( 2048MB), size= 1024MB, count=1: write-back
reg02: base=0x100000000 ( 4096MB), size= 4096MB, count=1: write-back
reg03: base=0x200000000 ( 8192MB), size= 8192MB, count=1: write-back
reg04: base=0x400000000 (16384MB), size=16384MB, count=1: write-back
reg05: base=0x800000000 (32768MB), size=32768MB, count=1: write-back
reg06: base=0x1000000000 (65536MB), size=65536MB, count=1: write-back
-----------------------
!!! Don't forget add "count=" to dd !!!
johnd@ubuntu:/opt/fmem$ sudo dd if=/dev/fmem of=/tmp/memdump.raw bs=1024
count=1024000
1024000+0 records in
1024000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 2.28005 s, 460 MB/s
johnd@ubuntu:/opt/fmem$
```

Linux Memory Forensics

LiME

A more stable tool for memory dumping is **LiME**.

LiME also supports mobile devices.

A Python utility called **LiMEaide** enables remote memory dumping.

```
johnd@ubuntu:/opt/LiME/src$ sudo make
make -C /lib/modules/4.15.0-74-generic/build M="/opt/LiME/src" modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-74-generic'
  CC [M]  /opt/LiME/src/tcp.o
  CC [M]  /opt/LiME/src/disk.o
  CC [M]  /opt/LiME/src/main.o
  CC [M]  /opt/LiME/src/hash.o
  CC [M]  /opt/LiME/src/deflate.o
  LD [M]  /opt/LiME/src/lime.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /opt/LiME/src/lime.mod.o
  LD [M]  /opt/LiME/src/lime.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-74-generic'
strip --strip-unneeded lime.ko
mv lime.ko lime-4.15.0-74-generic.ko
johnd@ubuntu:/opt/LiME/src$ insmod lime-4.15.0-74-generic.ko "path=/tmp/memdump.raw
format=raw"
@ubuntu:/opt/LiME/src$
```

Swap Digger

Swap Digger is a Bash script that automates swap analysis.

The script looks up passwords and URLs.

Swap Digger can operate on live systems and mounted captures.

```
sansforensics@siftworkstation -> ~
$ sudo ./swap_digger.sh -vx -s /home/swap.capture

  - SWAP Digger -

 [+] Using /home/swap.capture as swap partition
 [+] Dumping swap strings in /tmp/swap_dump.txt ... (this may take some time)
    [-] Swap dump size: 3.9M

   ==== Web entered passwords and emails ===

 [+] Looking for web passwords method 1 (password in GET/POST)...

 [+] Looking for web passwords method 2 (JSON) ...

 [+] Looking for web passwords method 3 (HTTP Basic Authentication) ...

 [+] Looking for web entered emails...
```

Short Practice

Swap Inspection

10–15 Min.

Mission

Investigate a swap and extract useful data.

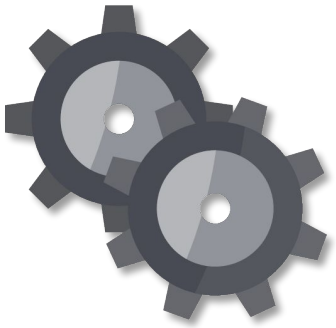
Steps

- Inspect the contents of the swap using Swap Digger.
- Identify the location of the swap on the system.
- Load the ***swap/drive*** file.
- Investigate the capture using Swap Digger and compare the output.

Linux Forensics

Process Investigation

Process Investigation



- Processes in Linux have file representations.
- Process files include metadata associated with the process.
- Processes are mapped in the ***/proc/*** directory.

> The ***/proc/*** directory uses tmpfs, meaning the files are saved in volatile memory.

Proc Directory

Each process listed by ***ps*** or ***lsof*** is mapped in ***/proc/***.

Process directories are based on their PIDs.

Each folder contains additional files required for the processes to run.

```
johnd@ubuntu:~$ ps -aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.4 167852  9860 ?        Ss   06:47   0:02 /sbin/init
root         2  0.0  0.0      0     0 ?        S    06:47   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   06:47   0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   06:47   0:00 [rcu_par_gp]
...
johnd@ubuntu:~$ cd /proc
johnd@ubuntu:/proc$ ls -la
total 40
dr-xr-xr-x 232 root       root                    0 Dec 31 06:47 .
drwxr-xr-x  19 root       root                36864 Aug  8 02:10 ..
dr-xr-xr-x   9 root       root                    0 Dec 31 06:47 1
dr-xr-xr-x   9 root       root                    0 Dec 31 06:47 10
dr-xr-xr-x   9 root       root                    0 Dec 31 06:47 100
dr-xr-xr-x   9 root       root                    0 Dec 31 06:48 1004
dr-xr-xr-x   9 root       root                    0 Dec 31 06:47 101
dr-xr-xr-x   9 root       root                    0 Dec 31 06:48 1010
dr-xr-xr-x   9 root       root                    0 Dec 31 06:48 1014
dr-xr-xr-x   9 root       root                    0 Dec 31 06:48 1017
dr-xr-xr-x   9 root       root                    0 Dec 31 06:48 1023
```

/proc/[PID] Content

The folder structure in ***/proc/*** includes useful information about processes.

Directory	Description
/proc/PID/cmdline	Command-line arguments
/proc/PID/cpu	Current and last CPU in which it was executed
/proc/PID/cwd	Link to the current working directory
/proc/PID/environ	Environment variable values
/proc/PID/exe	Link to the process executable
/proc/PID/fd	Directory that contains all file descriptors

Process Investigation

The first step in process investigation is to understand how the process is executed.

The required information can be found in ***comm*** and ***cmdline***.

```
johnd@ubuntu:~$ ps -aux
johnd@ubuntu:~$ netstat -naltp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address       Foreign Address     State       PID/Program
tcp        0      0 127.0.0.1:3306      0.0.0.0:*           LISTEN      -
tcp        0      0 127.0.0.53:53       0.0.0.0:*           LISTEN      -
tcp        0      0 0.0.0.0:22          0.0.0.0:*           LISTEN      -
tcp        0      0 127.0.0.1:631       0.0.0.0:*           LISTEN      -
tcp        0      0 0.0.0.0:1337        0.0.0.0:*           LISTEN      3930/./x99
tcp6       0      0 :::80               :::*                LISTEN      -
tcp6       0      0 :::22               :::*                LISTEN      -
tcp6       0      0 ::1:631             :::*                LISTEN      -
johnd@ubuntu:~$ cat /proc/3930/comm
x99
johnd@ubuntu:~$ cat /proc/3930/cmdline
./x99-l1337
johnd@ubuntu:~$
```

Maps and Descriptors

Two other useful commands are ***maps*** and ***fd***.

The ***maps*** command lists all loaded libraries.

The ***fd*** command lists all file descriptors.

```
johnd@ubuntu:~$ ls -la /proc/3930/fd
total 0
dr-x------ 2 lionk lionk  0 Jan 18 10:04 .
dr-xr-xr-x 9 lionk lionk  0 Jan 18 10:04 ..
lrwx------ 1 lionk lionk 64 Jan 18 10:04 0 -> /dev/pts/0
l-wx------ 1 lionk lionk 64 Jan 18 10:04 1 -> /dev/null
lrwx------ 1 lionk lionk 64 Jan 18 10:04 2 -> /dev/pts/0
lrwx------ 1 lionk lionk 64 Jan 18 10:04 3 -> 'socket:[61151]
lionk@ubuntu:/tmp$ cat /proc/3930/maps
55ad3744d000-55ad37455000 r-xp 00000000 08:01 1311077          /tmp/x99 (deleted)
55ad37654000-55ad37655000 r--p 00007000 08:01 1311077          /tmp/x99 (deleted)
55ad37655000-55ad37656000 rw-p 00008000 08:01 1311077          /tmp/x99 (deleted)
55ad37656000-55ad376d6000 rw-p 00000000 00:00 0
55ad377fc000-55ad3781d000 rw-p 00000000 00:00 0                [heap]
7fd841d8c000-7fd841da6000 r-xp 00000000 08:01 398746
/lib/x86_64-linux-gnu/libpthread-2.27.so
7fd841da6000-7fd841fa5000 ---p 0001a000 08:01 398746
/lib/x86_64-linux-gnu/libpthread-2.27.so
7fd841fa5000-7fd841fa6000 r--p 00019000 08:01 398746
/lib/x86_64-linux-gnu/libpthread-2.27.so
7fd841fa6000-7fd841fa7000 rw-p 0001a000 08:01 398746           /lib/x86_64-linux-
```

Extracting the Executable

The executable for each process can be extracted using a ***cp***.

Extraction will work even if the executable was deleted.

```
johnd@ubuntu:~$ ls -la /proc/3930/
total 0
dr-xr-xr-x   9 lionk lionk 0 Jan 18 10:04 .
dr-xr-xr-x 302 root  root  0 Jan 18 09:22 ..
dr-xr-xr-x   2 lionk lionk 0 Jan 18 10:04 attr
-rw-r--r--   1 lionk lionk 0 Jan 18 10:05 autogroup
-r--------   1 lionk lionk 0 Jan 18 10:05 auxv
-r--r--r--   1 lionk lionk 0 Jan 18 10:05 cgroup
--w-------   1 lionk lionk 0 Jan 18 10:05 clear_refs
-r--r--r--   1 lionk lionk 0 Jan 18 10:04 cmdline
-rw-r--r--   1 lionk lionk 0 Jan 18 10:05 comm
-rw-r--r--   1 lionk lionk 0 Jan 18 10:05 coredump_filter
-r--r--r--   1 lionk lionk 0 Jan 18 10:05 cpuset
lrwxrwxrwx   1 lionk lionk 0 Jan 18 10:05 cwd -> /tmp
-r--------   1 lionk lionk 0 Jan 18 10:05 environ
lrwxrwxrwx   1 lionk lionk 0 Jan 18 10:05 exe -> '/tmp/x99 (deleted)'
dr-x------   2 lionk lionk 0 Jan 18 10:04 fd
johnd@ubuntu:~$ cp /proc/3930/exe x99
johnd@ubuntu:~$ md5sum x99
3dd534fc7f982d3d79391e8c26bcf023  x99
johnd@ubuntu:~$
```

DFIR-08-L3

Process Investigation

20–25 Min.

Mission

Mimic bind shell behavior and investigate it via live analysis of the ***/proc/*** directory content.

Steps

- Simulate bind shell behavior.
- Identify the process in ***/proc/***.
- Investigate the process.
- Recover the executable's binary.

Env. & Tools

- Ubuntu 16.04+

Related Files

- Lab document

Thank You

Questions?