LAB 12: LINUX FORENSICS (ARTIFACTS ANALYSIS)

Lab Requirements

1. Linux OS
2. Internet connection

Content

Part I: Identifying Devices and OSs with p0f

STEP 1: Install p0f as follows, if it does not come preinstalled on the Linux distribution.

```
# Install p0f
kali@kali [~] sudo apt-get install p0f
```

STEP 2: “p0f uses a fingerprinting technique based on [passively] analyzing the structure of a TCP/IP packet to determine the operating system and other configuration properties of a remote host.” [man p0f] Partial list of available **options**:

- **`-i device:`** Listen to a specific device (interface).
- **`-r:`** Read packets from `tcpdump` snapshot. This is an interesting option for forensics, where the output of `tcpdump` is parsed using `p0f`. ○ **`-o:`** Write results to a log file.
- **`-p:`** Switch card to promiscuous mode. ○ **`-L: List all available interfaces.`**

STEP 3: `p0f` can use filters to include or exclude particular networks, hosts or packets. Examples

- **`dst port 80`** ○ **`src host 172.6.16.101 or 172.6.16.102`**

STEP 4: The default fingerprint database of `p0f` is stored in `/etc/p0f/p0f.fp.` Use the command `cat` to display the signatures of various packets/protocols.

STEP 5: Display the available interfaces using the following command:

```
kali@kali [~] p0f -L
     --- p0f 3.09b by Michal Zalewski <lcamtuf@coredump.cx> ---
     -- Available interfaces --
       0: Name        : eth0
          Description : -
          IP address  : (none)

       1: Name        : eth1
          Description : -
          IP address  : 172.16.200.135
      ...
```

STEP 6: Using the `p0f` command without any options starts the fingerprint process on the local machine. It takes a few minutes to start displaying results.

```

kali@kali [~] sudo p0f
     --- p0f 3.09b by Michal Zalewski <lcamtuf@coredump.cx> ---
     [+] Closed 1 file descriptor.
     [+] Loaded 322 signatures from '/etc/p0f/p0f.fp'.
     [+] Intercepting traffic on default interface 'eth0'.
     [+] Default packet filtering configured [+VLAN].
     [+] Entered main event loop.

     .-[ 172.16.200.136/34884 -> 172.217.4.35/443 (syn) ]-
     |
     | client   = 172.16.200.136/34884
     | os     = Linux 2.2.x-3.x
     | dist   = 0
     | params   = generic
     | raw_sig  = 4:64+0:0:1460:mss*44,10:mss,sok,ts,nop,ws:df,id+:0
     |
     `----

     .-[ 172.16.200.136/34884 -> 172.217.4.35/443 (mtu) ]-
     |
     | client   = 172.16.200.136/34884
     | link  = Ethernet or modem
     | raw_mtu  = 1500
     |
     `----
     .-[ 172.16.200.136/34884 -> 172.217.4.35/443 (mtu) ]-
     |
     | server   = 172.217.4.35/443
     | link  = Ethernet or modem
     | raw_mtu  = 1500
     |
     `----
      ...
```

Part II: Information gathering and Fingerprinting with arp-scan & nmap

Step 7: `arp-scan` is used to list the ARP table content (ARP: Address resolution protocol) in the local network.

```

kali@kali [~] sudo arp-scan 172.16.145.1/24 ...
     Interface: eth0, type: EN10MB, MAC: 00:50:56:20:0d:60, IPv4: (none)
     WARNING: host part of 172.16.145.1/24 is non-zero
     Starting arp-scan 1.9.7 with 256 hosts
     (https://github.com/royhills/arpscan)
     172.16.145.1    a6:83:e7:d9:44:66       (Unknown: locally administered)
     172.16.145.254  00:50:56:eb:3a:18       VMware, Inc.

     2 packets received by filter, 0 packets dropped by kernel
     Ending arp-scan 1.9.7: 256 hosts scanned in 1.977 seconds (129.49
     hosts/sec). 2 responded
```

Step 8: `arp-scan` is used to list the ARP table content (ARP: Address resolution protocol) in the local network.

```
user@parrot [~] nmap -sn 172.16.145.1/24 nmap
     -sn 172.16.145.1/24
     Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-02 21:39 BST Nmap
     scan report for 172.16.145.1
     Host is up (0.0051s latency).
     Nmap scan report for 172.16.145.2
     Host is up (0.0067s latency).
     Nmap scan report for 172.16.145.137
     Host is up (0.0024s latency).
     Nmap scan report for 172.16.145.140
     Host is up (0.0017s latency).
     Nmap done: 256 IP addresses (4 hosts up) scanned in 2.54 seconds
```

STEP 9: To perform TCP port SYN scan, use the following command. You can open more ports using the `sudo utf allow p/tcp` (p is the port you wish to open).

```
user@parrot [~] sudo nmap -sS 172.16.145.137
     [sudo] password for user:
     Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-02 21:48 BST Nmap
     scan report for 172.16.145.137
     Host is up (0.023s latency).
     Not shown: 999 closed tcp ports (reset)
     PORT    STATE SERVICE
     902/tcp open  iss-realsecure
     MAC Address: 00:50:56:2A:DE:62 (VMware)

     Nmap done: 1 IP address (1 host up) scanned in 1.16 seconds
```

STEP 9: To perform UDP port scan, use the following command:

```
# On VM1 with IP address 172.16.145.140 (it could be different on your VM)
# Start the ufw firewall and allow the port 53/udp
user@parrot [~] sudo systemctl start ufw
user@parrot [~] sudo ufw allow 53/udp user@parrot
[~] sudo ufw status verbose

    Status: active
    Logging: on (low)
    Default: deny (incoming), allow (outgoing), disabled (routed)
    New profiles: skip

    To                          Action      From
    --                          ------      ----
    53/udp                      ALLOW IN    Anywhere
    53/udp (v6)                 ALLOW IN    Anywhere (v6)

# On VM2
user@parrot [~] sudo nmap -sU 172.16.145.137
    Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-02 22:04 BST Nmap
    scan report for 172.16.145.137
    Host is up (0.0019s latency).
    Not shown: 999 open|filtered udp ports (no-response)
    PORT   STATE  SERVICE
    53/udp closed domain
    MAC Address: 00:50:56:2A:DE:62 (VMware)

    Nmap done: 1 IP address (1 host up) scanned in 11.34 seconds
```

STEP 10: Discover more functionalities of `nmap` tool by display the man page of the command (`man nmap`).

Part III: Information Gathering with swap_digger

STEP 11: swap_digger perform analysis of the Linux swap file to retrieve system passwords, usernames, credentials, among others. Let us install swap_digger.

```
kali@kali [~] cd work kali@kali [~/work] git clone
https://github.com/sevagas/swap_ digger.git
     Cloning into 'swap_digger'... ...

kali@kali [~/work] cd swap_digger kali@kali
[~/work/swap_digger] sudo chmod +x swap_digger.sh

kali@kali [~/work/swap_digger] sudo ./swap_digger.sh -S
  - SWAP Digger -
 [+] Current swap file:
   -> /dev/sda5
 [+] /etc/fstab swap files:
   -> /dev/sda5
 [+] Looking for all available swap device files (will take some time):
   -> /dev/sda5

# dump application data
kali@kali [~/work/swap_digger] sudo ./swap_digger.sh -a
     - SWAP Digger -

      [+] Looking for swap partition
          -> Found swap at /dev/sda5
      [+] Dumping swap strings in /tmp/swap_dig/swap_dump.txt ... (this may take
     some time)

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

     ==== XML data ===

      [+] Looking for xml passwords ...
        -> n failea
     ials><username>admin</username><password>kali</password></credentials></aut
     henticate>
     293.236 0.0918274 293.228 0.
     ...

     ==== WiFi ===

      [+] Looking for wifi access points...
        [-] Potential wifi network list this computer accessed to:
     ...

     ==== Mining most accessed resources ===

      [+] TOP 30 HTTP/HTTPS URLs (domains only)
        ->    4213 https://lists.fedoraproject.org
        ->    2650 https://developer.huaweicloud.com
        ->    1771 https://bugs.mageia.org
        ->    1621 https://advisories.mageia.org    ->
     1372 https://www.suse.com ...

     [+] TOP 30 FTP URLs
        ->       3
     ftp://ftp.software.ibm.com/ps/products/db2/fixes/englishus/aparlist/db2_v82/APAR
     LIST.TXT ...

     [+] TOP 30 files                                                                 ->
     89 file:///usr/lib/firefox-esr/omni.ja
        ->      63 file:///usr/lib/firefox-esr/browser/omni.ja ...

     [+] TOP 30 IP addresses (lots of false positives, ex. file versions)
        ->   65999 1.3.6.1
        ->     496 3.6.1.4
        ->     430 2.6.8.1
        ->     406 1.4.1.2
        ->     406 6.1.4.1
        ->     384 09.09.09.09 ...

     ==== Mining hashes ===

      [-] No MD5-hashes found
      [-] No SHA1-hashes found
      [-] No SHA256-hashes found

```

```

      [-] No SHA512-hashes found
      [-] No Blowfish-hashes found

```

STEP 12: Other `swap_digger` options include `-p` (passwords: Linux system credentials).

```

kali@kali [~/work/swap_digger] sudo ./swap_digger.sh -S sudo
     ./swap_digger.sh -p
       - SWAP Digger -
      [+] Swap dump already available at /tmp/swap_dig/swap_dump.txt
        ==== Linux system accounts ===

      [+] Digging linux accounts credentials... (pattern attack)
      Passwords not found. Attempt dictionary based attack? (Can last from 5
     minutes to several hours depending on swap usage) [y/n]  ...

```

Part IV: Password Dumping with mimipenguin

STEP 13: Install and use `mimipenguin` as follows.

```
kali@kali [~/work] git clone https://github.com/huntergregal/mimipenguin.git
kali@kali [~/work] cd mimipenguin
kali@kali [~/work/mimipenguin] sudo ./mimipenguin.sh
     MimiPenguin Results:
```

Part V: Further Linux Digital Forensic Tools

STEP 14: Check for the presence of rootkits, suspicious files, or hidden directories using `rkhunter`.

```
kali@kali [~] sudo apt-get install rkhunter kali@kali
[~] sudo rkhunter –check -rwo
    Warning: The file '/usr/bin/mail' exists on the system, but it is not
    present in the 'rkhunter.dat' file.
    Warning: The command '/usr/bin/lwp-request' has been replaced by a script:
    /usr/bin/lwp-request: Perl script text executable
    Warning: The file '/usr/bin/bsd-mailx' exists on the system, but it is not
    present in the 'rkhunter.dat' file.
    Warning: The following suspicious (large) shared memory segments have been
    found:
             Process: /usr/bin/xfdesktop    PID: 1096    Owner: kali    Size:
    64MB (configured size allowed: 1.0MB)


             Process: /usr/bin/xfdesktop    PID: 1096    Owner: kali    Size:
    2.0MB (configured size allowed: 1.0MB)
    Warning: The SSH configuration option 'PermitRootLogin' has not been set.
             The default value may be 'yes', to allow root access. Warning:
    Hidden directory found: /etc/.java
```

STEP 15: Check for the presence of rootkits using `chkrootkit`.

```
kali@kali [~] sudo apt-get install chkrootkit kali@kali
[~] sudo rkhunter –check -rwo
    ROOTDIR is `/'
    Checking `amd'...                                          not found
    Checking `basename'...                                     not infected
    Checking `biff'...                                         not found
    Checking `chfn'...                                         not infected
    Checking `chsh'...                                         not infected
    Checking `cron'...                                         not infected
    Checking `crontab'...                                      not infected
    Checking `date'...                                         not infected
    Checking `du'...                                           not infected
    ...
```

STEP 16: Display ascii table using `ascii`.

```
kali@kali [~] sudo apt-get install ascii kali@kali
[~] ascii -s hello
     6/8   104   0x68   0o150   01101000
     6/5   101   0x65   0o145   01100101
     6/12   108   0x6C   0o154   01101100
     6/12   108   0x6C   0o154   01101100
     6/15   111   0x6F   0o157   01101111

kali@kali [~] ascii -x
00     NUL    10 DLE    20       30 0    40 @    50 P    60 `    70 p
01     SOH    11 DC1    21 !     31 1    41 A    51 Q    61 a    71 q
02     STX    12 DC2    22 "     32 2    42 B    52 R    62 b    72 r     03 ETX
13 DC3    23 #    33 3    43 C    53 S    63 c    73 s
   ...
```

Step 17: Display file signature using (and content) using `xxd` command. The following command displays the signature and the first 10 lines of the a .rar file. The signature displayed below is for Roshal ARchive compressed archive v1.50 onwards. (For v5.00 onwards, the signature is `52 61 72 21 1a 07 01 00`)

```
kali@kali [~] xxd -g 1 0zapftis.rar | head
     00000000: 52 61 72 21 1a 07 00 ce 99 73 80 00 0d 00 00 00  Rar!.....s......
     00000010: 00 00 00 00 a8 dc 2f ea 1b 70 d3 d0 02 45 55 1e  ....../..p...EU.
     00000020: c5 ac cb 85 9e f3 47 f3 69 c2 34 ec e6 ad 34 f1  ......G.i.4...4.
   00000030: 32 c5 8e b8 44 31 3f 92 14 17 a1 e3 19 96 ec 54  2...D1?........T
   00000040: e9 d5 e1 a0 36 da cd 8f c7 5e c6 84 b1 fc f2 19  ....6....^......
     00000050: d8 81 b6 99 ea 65 eb 71 b7 b3 4e 18 02 68 0f 7b  .....e.q..N..h.{
     00000060: bf da a4 14 fa 1f aa 83 66 ef 9a b6 6b b5 a0 69  ........f...k..i
   00000070: f2 06 35 53 01 5e a9 1d ab cc a8 77 2e 9c 50 6a  ..5S.^.....w..Pj
   00000080: 17 65 04 2a bc 2f d5 ea 9b ed fe 43 48 4b 0f cf  .e.*./.....CHK..
     00000090: ed 64 a8 5c 32 cc c2 6d 73 54 9e bb b7 c7 90 c5  .d.\2..msT......
```

Step 18: Use the command `strings` to look for a specific pattern within a non-text file. The image file was used in the previous lab. The `-t` options display the offset of the matched string with the file, with `d`, `o`, and `x` values refer to decimal, octal, and hexadecimal number of bits from the beginning of the file.

```
kali@kali [~] strings -t x terry-work-usb-2009-12-11.E01 | grep -i "jpg"
     12f44f jPg?O
     1217a1c jpG"
     13d81b9 LJpG
```

Part VI: Sleuth Toolkit (STK)

STEP 19: `The Sleuth Kit` is a collection digital forensic tools that can be used to analyze disk images and recover files from them [sleuthkit.org]. The kit includes several commands [http://wiki.sleuthkit.org/index.php?title=The_Sleuth_Kit_commands] including

○ **`fsstat:`** Display general details of the file system. ○ **`fls:`** List files and directories in the disk image.

○ **`ils:`** List inode information. ○ **`img_stat:`** Display details of an image file. ○ **`img_cat:`** Output contents of an image file.

○ **`fiwalk:`** Print the filesystem details.

STEP 20: To perform the following tasks, I use the image available at https://cfreds.nist.gov/all/DFRWS2009Challenge/DFRWS2009USBFlashDriveImages. You should extract the .dd file before performing any of the commands.

```
# Download the compressed image file
kali@kali [~/work/images] wget
http://old.dfrws.org/2009/challenge/imgs/nssalthumb-fs.dd.bz2

# Uncompress the image file
```

```
kali@kali [~/work/images] bzip2 -dk nssal-thumb-fs.dd.bz2
# List the content of the current directory kali@kali
[~/work/images] ls
0zapftis.vmem  Cfreds001A001.dd  nssal-thumb-fs.dd  nssal-thumb-fs.dd.bz2

kali@kali [~/work/images] fls nssal-thumb-fs.dd | head -n 3
     r/r * 3:      _hatever r/r * 7:
     3323673964_94e64ebddd_b.jpg r/r * 11:
     3323673964_94e64ebddd_b.jpg

kali@kali [~/work/images] fsstat -I raw nssal-thumb-fs.dd

     FILE SYSTEM INFORMATION
     --------------------------------------------
     File System Type: FAT16

     OEM Name: MSDOS5.0 Volume
     ID: 0x14d06139 ...
     METADATA INFORMATION
     --------------------------------------------
     Range: 2 - 15987318
     Root Directory: 2

     CONTENT INFORMATION
     --------------------------------------------
     Sector Size: 512
     Cluster Size: 8192
     Total Cluster Range: 2 - 62449

     FAT CONTENTS (in sectors)
     --------------------------------------------

kali@kali [~/work/images] ils nssal-thumb-fs.dd | head
     class|host|device|start_time ils|kali||1649008520
     st_ino|st_alloc|st_uid|st_gid|st_mtime|st_atime|st_ctime|st_crtime|st_mode|
     st_nlink|st_size
     3|f|0|0|1236020810|1235970000|0|1236020595|777|0|511573308
     7|f|0|0|1236021330|1235970000|0|1236021328|777|0|0
     11|f|0|0|1236021330|1235970000|0|1236021328|777|0|248179
     15|f|0|0|1236021364|1235970000|0|1236021363|777|0|0
     19|f|0|0|1236021366|1235970000|0|1236021363|777|0|743412
     23|f|0|0|1236021412|1235970000|0|1236021411|777|0|0
     27|f|0|0|1236021414|1235970000|0|1236021411|777|0|468985

kali@kali [~/work/images] img_stat nssal-thumb-fs.dd
```

```
     IMAGE FILE INFORMATION
     --------------------------------------------
     Image Type: raw

     Size in bytes: 511847936
     Sector size:  512

kali@kali [~/work/images] fiwalk nssal-thumb-fs.dd
     ... parent_inode: 2 filename:
     3316820191_4737c3edf4.jpg partition: 1 id: 34
     name_type: r filesize: 139264 unalloc: 1 used:
     1 inode: 132 meta_type: 1 mode: 511 nlink: 0
     uid: 0 gid: 0 mtime: 1236393458 mtime_txt:
     2009-03-07T02:37:38 atime: 1236315600
     atime_txt: 2009-03-06T05:00:00 crtime:
     1236393458 crtime_txt: 2009-03-07T02:37:38
     md5: ade94ab75ccf766087659e0287fabba2 sha1:
     edae17220e659c0d61a9c908303e3b5114535b16 ...
```