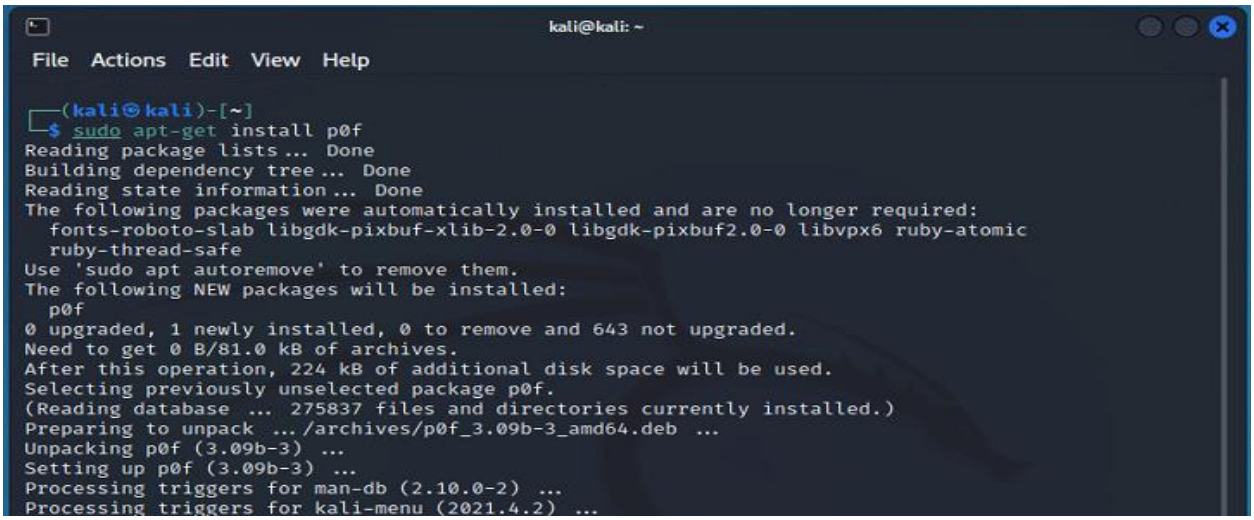
PART I IDENTIFYING DEVICES AND OSS WITH P0F

Step 1 & 2 & 3

In this step we are going to install p0f, which uses fingerprinting technique based on analyzing the structure of TCP/IP packet to determine the operating system.



P0f can use filters to include or exclude particular networks, hosts or packets.

Step 4

The main database of p0f fingerprinting is stored at the location of **/etc/p0f/p0f.fp.**

To read the stored file we are going to use "**cat**"

```
kali@kali: ~
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~]
└─$ cat /etc/p0f/p0f.fp
;
; p0f - fingerprint database
; ──────────────────────────
;
; See section 5 in the README for a detailed discussion of the format used here.
;
; Copyright (C) 2012 by Michal Zalewski <lcamtuf@coredump.cx>
;
; Distributed under the terms and conditions of GNU LGPL.
;

classes = win,unix,other

; ═══════════════
; MTU signatures
; ═══════════════

[mtu]

; The most common values, used by Ethernet-homed systems, PPP over POTS, PPPoA
; DSL, etc:

label = Ethernet or modem
sig   = 576
sig   = 1500

; Common DSL-specific values (1492 is canonical for PPPoE, but ISPs tend to
; horse around a bit):

label = DSL
sig   = 1452
```

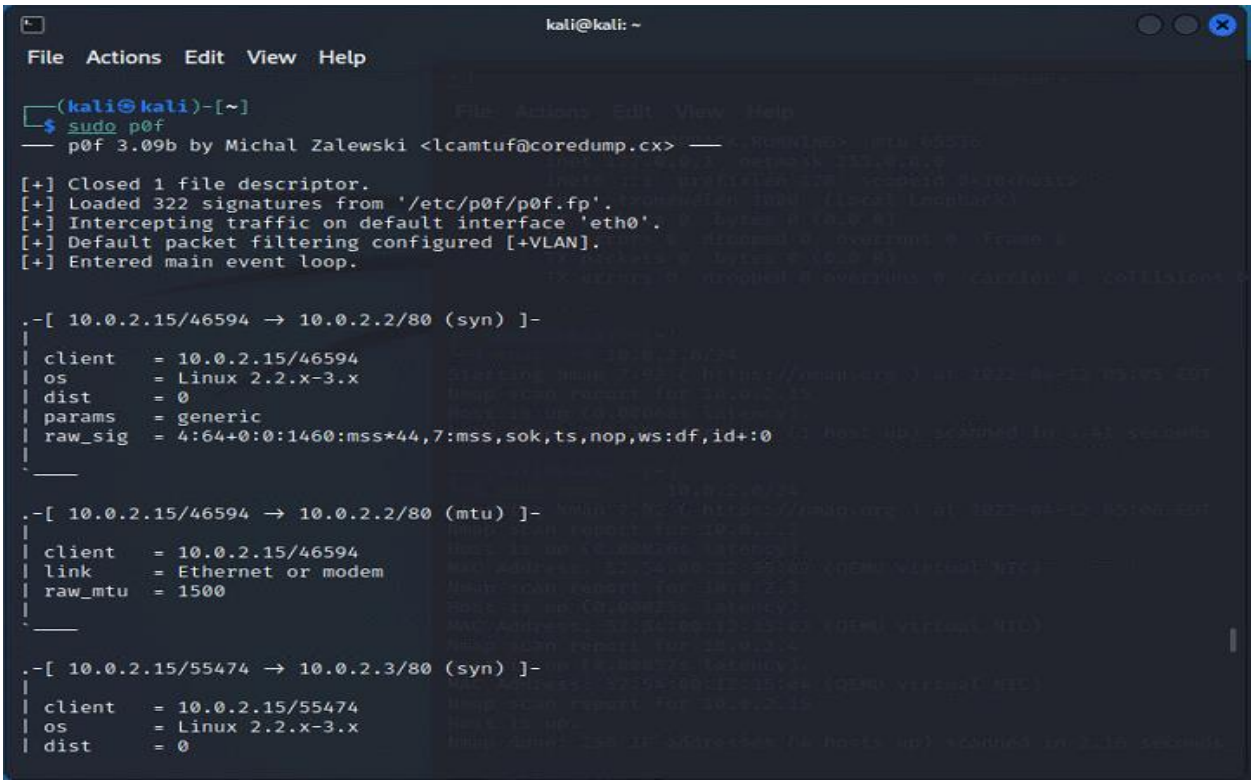
Step 5

P0f -L is going to display the following all of the interfaces of the device

```
kali@kali: ~
File  Actions  Edit  View  Help
All done. Processed 0 packets.

┌──(kali㉿kali)-[~]
└─$ p0f -L
─── p0f 3.09b by Michal Zalewski <lcamtuf@coredump.cx> ───


-- Available interfaces --

  0: Name        : eth0
     Description : -
     IP address  : 10.0.2.15

  1: Name        : any
     Description : Pseudo-device that captures on all interfaces
     IP address  : (none)

  2: Name        : lo
     Description : -
     IP address  : 127.0.0.1

  3: Name        : bluetooth-monitor
     Description : Bluetooth Linux Monitor
     IP address  : (none)

  4: Name        : nflog
     Description : Linux netfilter log (NFLOG) interface
     IP address  : (none)

  5: Name        : nfqueue
     Description : Linux netfilter queue (NFQUEUE) interface
     IP address  : (none)

  6: Name        : dbus-system
     Description : D-Bus system bus
```

Step 6

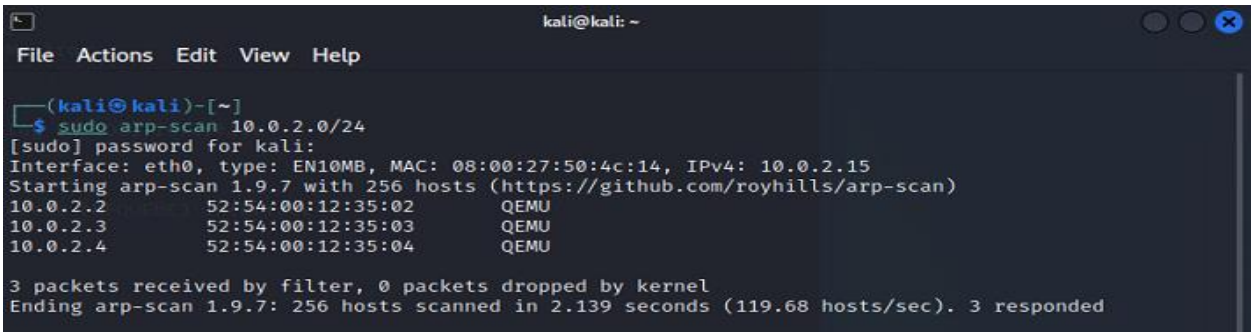
Using just **p0f** command without any flag is going to fingerprint the processes in the whole device.



PART II INFORMATION GATHERING AND FINGERPRINTING WITH ARP-SCAN & NMAP

Step 7

arp-scan is an amazing tool used to list down the contents of ARP table. It shows the number of devices on the network connected to the device.



Step 8

Another great and famous tool is nmap short for network mapping tool. Here we are using "**-sn**" flag which is going to ping all the devices on the network that are live.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sn 10.0.2.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-12 05:06 EDT
Nmap scan report for 10.0.2.2
Host is up (0.00026s latency).
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00025s latency).
MAC Address: 52:54:00:12:35:03 (QEMU virtual NIC)
Nmap scan report for 10.0.2.4
Host is up (0.00057s latency).
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)
Nmap scan report for 10.0.2.15
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.16 seconds
```

Step 9

Another flag of **nmap** is "**-sS**" that is used to perform SYN scan on the TCP ports and display the ports that are open

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-12 05:21 EDT
Nmap scan report for 10.0.2.4
Host is up (0.012s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
5357/tcp  open  wsdapi
6646/tcp  open  unknown
56738/tcp open  unknown
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 5.43 seconds
```

If the ports are been blocked by the firewall, just add rule to the firewall to allow the port's packet to be send and received.

This is done using the command "**ufw allow port/type**"

Using "**-sU**" flag in nmap scans all of the ports having udp transmission

```
kali@kali: ~
File  Actions  Edit  View  Help
└─$ sudo ufw allow 53/udp                                                    1 ×
Skipping adding existing rule
Skipping adding existing rule (v6)

┌──(kali㉿kali)-[~]
└─$ sudo nmap -sU 10.0.2.4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-12 05:23 EDT
Stats: 0:00:42 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 35.10% done; ETC: 05:25 (0:01:18 remaining)
Nmap scan report for 10.0.2.4
Host is up (0.00081s latency).
Not shown: 991 filtered udp ports (port-unreach)
PORT     STATE         SERVICE
67/udp   open|filtered dhcps
69/udp   open          tftp
137/udp  open|filtered netbios-ns
1900/udp open|filtered upnp
3702/udp open|filtered ws-discovery
4500/udp open|filtered nat-t-ike
5050/udp open|filtered mmcc
5353/udp open|filtered zeroconf
5355/udp open|filtered llmnr
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 128.86 seconds
```

Step 10

Nmap is used to perform a lot of tasks from scanning the network to finding vulnerabilities in it. To learn about nmap we need to use its manual and the command that shows it is **"man nmap"**

```
kali@kali: ~
File  Actions  Edit  View  Help
      --version-all: Try every single probe (intensity 9)
      --version-trace: Show detailed version scan activity (for debugging)
    SCRIPT SCAN:
      -sC: equivalent to --script=default
      --script=<Lua scripts>: <Lua scripts> is a comma separated list of
               directories, script-files or script-categories
      --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
      --script-args-file=filename: provide NSE script args in a file
      --script-trace: Show all data sent and received
      --script-updatedb: Update the script database.
      --script-help=<Lua scripts>: Show help about scripts.
               <Lua scripts> is a comma-separated list of script-files or
               script-categories.
    OS DETECTION:
      -O: Enable OS detection
      --osscan-limit: Limit OS detection to promising targets
      --osscan-guess: Guess OS more aggressively
    TIMING AND PERFORMANCE:
      Options which take <time> are in seconds, or append 'ms' (milliseconds),
      's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
      -T<0-5>: Set timing template (higher is faster)
      --min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
      --min-parallelism/max-parallelism <numprobes>: Probe parallelization
      --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies
          probe round trip time.
      --max-retries <tries>: Caps number of port scan probe retransmissions.
      --host-timeout <time>: Give up on target after this long
      --scan-delay/--max-scan-delay <time>: Adjust delay between probes
      --min-rate <number>: Send packets no slower than <number> per second
      --max-rate <number>: Send packets no faster than <number> per second
    FIREWALL/IDS EVASION AND SPOOFING:
      -f; --mtu <val>: fragment packets (optionally w/given MTU)
      -D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
      -S <IP_Address>: Spoof source address
 Manual page nmap(1) line 127 (press h for help or q to quit)
```

PART III: INFORMATION GATHERING WITH SWAP_DIGGER

Step 11

Swap_digger is an information gathering tool that is used to analyze Linux swap files to retrieve passwords, usernames, credentials and much more. It is not built-in tool in linux so we have to git clone its repository

```
┌──(kali㉿kali)-[~/Desktop/work]
└─$ git clone https://github.com/sevagas/swap_digger.git
Cloning into 'swap_digger' ...
remote: Enumerating objects: 147, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 147 (delta 15), reused 21 (delta 11), pack-reused 117
Receiving objects: 100% (147/147), 357.52 KiB | 1.20 MiB/s, done.
Resolving deltas: 100% (69/69), done.
```

-S flag in swap_digger looks for all available swap device files.

```
kali@kali: ~/Desktop/work/swap_digger
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/work]
└─$ cd swap_digger

┌──(kali㉿kali)-[~/Desktop/work/swap_digger]
└─$ ls
assets  LICENSE  README.md  swap_digger.sh

┌──(kali㉿kali)-[~/Desktop/work/swap_digger]
└─$ sudo chmod +x swap_digger.sh

┌──(kali㉿kali)-[~/Desktop/work/swap_digger]
└─$ sudo ./swap_digger.sh

 - SWAP Digger -

[!] Error: Specify one or more options such as:
    -S   search for swap devices!
    -a   mine for application data
    -p   mine for system passwds
    -h   view all options

┌──(kali㉿kali)-[~/Desktop/work/swap_digger]
└─$ sudo ./swap_digger.sh -S                                    1 ✗

 - 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

 SWAP Digger end, byebye!
```

-a flag in swap_digger is used to search all data related to applications that are there

```
kali@kali: ~/Desktop/work/swap_digger
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/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 ═══

[+] 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...

  ═══ XML data ═══

[+] Looking for xml passwords ...

  ═══ WiFi ═══

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

[+] Looking for potential Wifi passwords....
  [-] Potential wifi password list (use them to crack above networks)

[+] Looking for potential Wifi passwords method 2....
  [-] Potential wifi password list (use them to crack above networks)

  ═══ Mining most accessed resources ═══

[+] TOP 30 HTTP/HTTPS URLs (domains only)
  →      374 https://firefox-settings-attachments.cdn.mozilla.net
  →      163 http://www.w3.org
  →      110 https://www.digicert.com
  →       79 http://crl.usertrust.com
  →       79 http://ocsp.usertrust.com0
  →       79 http://www.freedesktop.org
  →       78 http://crt.usertrust.com
  →       73 http://crl3.digicert.com
```

```
kali@kali: ~/Desktop/work/swap_digger
File  Actions  Edit  View  Help

[+] TOP 30 FTP URLs

[+] TOP 30 .onion urls

[+] TOP 30 files
  →        4 file:///media/kali/bitcoin/101016.pdf
  →        4 file:///media/kali/bitcoin/101019.pdf
  →        2 file:///media/kali/bitcoin/101018.pdf

[+] TOP 30 smb shares

[+] TOP 30 IP addresses (lots of false positives, ex. file versions)
  →        3 0.99.7.1
  →        3 10.0.2.15
  →        3 255.255.255.0
  →        2 0.0.0.0
  →        2 10.1.0.0
  →        2 10.2.0.0
  →        2 1.2.5.1
  →        2 192.168.0.2
  →        2 4.2.9.1
  →        1 0.5.1.2
  →        1 1.1.2.1
  →        1 1.18.99.901
  →        1 1.2.189.0
  →        1 1.2.198.1
  →        1 1.2.2.3
  →        1 127.0.0.1
  →        1 127.0.1.1
  →        1 1.2.99.3
  →        1 1.4.99.1
  →        1 2.11.1.0
  →        1 3.10.0.2

  ═══ Mining hashes ═══

[-] No MD5-hashes found
[-] No SHA1-hashes found
[-] No SHA256-hashes found
[-] No SHA512-hashes found
[-] No Blowfish-hashes found

 SWAP Digger end, byebye!
```

Step 12

-p flag in swap_digger is used to dig passwords in the linux device

```
kali@kali: ~/Desktop/work/swap_digger
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/work/swap_digger]
└─$ 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 d
epending on swap usage) [y/n] y

 [+] Digging linux accounts credentials method 2 ... (dictionary attack)
   [-] Generating wordlist file ...
   [-] Digging passwords in wordlist ... (This may take 5min to few hours!)


  SWAP Digger end, byebye!

/home/kali/Desktop/work/swap_digger
```

PART IV: PASSWORD DUMPING WITH MIMIPENGUIN

Step 13

Mimipenguin is an opensource tool that is used to dump passwords

```
kali@kali: ~/Desktop/work/mimipenguin
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~]
└─$ cd Desktop/work

┌──(kali㉿kali)-[~/Desktop/work]
└─$ git clone https://github.com/huntergregal/mimipenguin.git
Cloning into 'mimipenguin'...
remote: Enumerating objects: 529, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 529 (delta 0), reused 1 (delta 0), pack-reused 525
Receiving objects: 100% (529/529), 182.38 KiB | 1.51 MiB/s, done.
Resolving deltas: 100% (241/241), done.

┌──(kali㉿kali)-[~/Desktop/work]
└─$ cd mimipenguin

┌──(kali㉿kali)-[~/Desktop/work/mimipenguin]
└─$ sudo ./mimipenguin.sh
[sudo] password for kali:
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 rkhunter --check --rwo                                                    100 ✗
Warning: Checking for prerequisites                [ Warning ]
         The file of stored file properties (rkhunter.dat) does not exist, and should be created.
To do this type in 'rkhunter --propupd'.
Warning: WARNING! It is the users responsibility to ensure that when the '--propupd' option
         is used, all the files on their system are known to be genuine, and installed from a
         reliable source. The rkhunter '--check' option will compare the current file properties
         against previously stored values, and report if any values differ. However, rkhunter
         cannot determine what has caused the change, that is for the user to do.
Warning: The command '/usr/bin/lwp-request' has been replaced by a script: /usr/bin/lwp-request: P
erl script text executable

Warning: The following suspicious (large) shared memory segments have been found:
         Process: /usr/bin/xfdesktop    PID: 945    Owner: kali    Size: 2.0MB (configured size al
lowed: 1.0MB)
Warning: Changes found in the group file for group 'stunnel4':
         User 'stunnel4' has been added to the group
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: ~
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~]
└─$ sudo chkrootkit
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
Checking `dirname' ...                                      not infected
Checking `echo' ...                                         not infected
Checking `egrep' ...                                        not infected
Checking `env' ...                                          not infected
Checking `find' ...                                         not infected
Checking `fingerd' ...                                      not found
Checking `gpm' ...                                          not found
Checking `grep' ...                                         not infected
Checking `hdparm' ...                                       not infected
Checking `su' ...                                           not infected
Checking `ifconfig' ...                                     not infected
Checking `inetd' ...                                        not infected
Checking `inetdconf' ...                                    not found
Checking `identd' ...                                       not found
Checking `init' ...                                         not infected
Checking `killall' ...                                      not infected
Checking `ldsopreload' ...                                  not infected
Checking `login' ...                                        not infected
Checking `ls' ...                                           not infected
Checking `lsof' ...                                         not infected
Checking `mail' ...                                         not infected
Checking `mingetty' ...                                     not found
Checking `netstat' ...                                      not infected
Checking `named' ...                                        not found
Checking `passwd' ...                                       not infected
Checking `pidof' ...                                        not infected
Checking `pop2' ...                                         not found
Checking `pop3' ...                                         not found
Checking `ps' ...                                           not infected
Checking `pstree' ...                                       not infected
Checking `rpcinfo' ...                                      not infected
Checking `rlogind' ...                                      not found
Checking `rshd' ...                                         not found
Checking `slogin' ...                                       not infected
Checking `sendmail' ...                                     not infected
Checking `sshd' ...                                         not infected
Checking `syslogd' ...                                      not tested
Checking `tar' ...                                          not infected
Checking `tcpd' ...                                         not found
```

```
kali@kali: ~
File  Actions  Edit  View  Help
Searching for Fu rootkit default files ...                  nothing found
Searching for ESRK rootkit default files ...                nothing found
Searching for rootedoor ...                                 nothing found
Searching for ENYELKM rootkit default files ...             nothing found
Searching for common ssh-scanners default files ...         nothing found
Searching for Linux/Ebury - Operation Windigo ssh ...       nothing found
Searching for 64-bit Linux Rootkit  ...                     nothing found
Searching for 64-bit Linux Rootkit modules ...              nothing found
Searching for Mumblehard Linux  ...                         nothing found
Searching for Backdoor.Linux.Mokes.a  ...                   nothing found
Searching for Malicious TinyDNS  ...                        nothing found
Searching for Linux.Xor.DDoS  ...                           nothing found
Searching for Linux.Proxy.1.0  ...                          nothing found
Searching for CrossRAT  ...                                 nothing found
Searching for Hidden Cobra  ...                             nothing found
Searching for Rocke Miner  ...                              nothing found
Searching for PWNLNX4 lkm ...                               nothing found
Searching for PWNLNX6 lkm ...                               nothing found
Searching for Umbreon lrk ...                               nothing found
Searching for Kinsing.a backdoor ...                        nothing found
Searching for RotaJakiro backdoor ...                       nothing found
Searching for suspect PHP files ...                         nothing found
Searching for anomalies in shell history files ...          nothing found
Checking `asp' ...                                          not infected
Checking `bindshell' ...                                    not infected
Checking `lkm' ...                                          chkproc: nothing detected
chkdirs: nothing detected
Checking `rexedcs' ...                                      not found
Checking `sniffer' ...                                      Output from ifpromisc:
lo: not promisc and no packet sniffer sockets
eth0: PACKET SNIFFER(/usr/sbin/NetworkManager[84406], /usr/sbin/NetworkManager[84406])
Checking `w55808' ...                                       not infected
Checking `wted' ...                                         1 deletion(s) between Sun Feb  6 00:48
:03 2022 and Sun Feb  6 01:38:32 2022
1 deletion(s) between Wed Feb  9 23:34:40 2022 and Tue Feb 15 01:25:27 2022
Checking `scalper' ...                                      not infected
Checking `slapper' ...                                      not infected
Checking `z2' ...                                           user kali deleted or never logged from
 lastlog!
Checking `chkutmp' ...                                      The tty of the following process(es) w
as not found in /var/run/utmp:
! RUID           PID TTY    CMD
! kali          1173 pts/0  /usr/bin/zsh
! kali        279877 pts/2  sudo chkrootkit
! kali          3869 pts/2  /usr/bin/zsh
chkutmp: nothing deleted
Checking `OSX_RSPLUG' ...                                   not tested
```

Step 16

Now we are going to display asci table using ascii tool, we can also asci values for “**hello**”

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install ascii
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
ascii is already the newest version (3.18-5).
The following packages were automatically installed and are no longer required:
  fonts-roboto-slab libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.0-0 libvpx6 python3-ipaddr
  python3-twisted-bin ruby-atomic ruby-thread-safe
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 244 not upgraded.

┌──(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
   04 EOT    14 DC4    24 $    34 4    44 D    54 T    64 d    74 t
   05 ENQ    15 NAK    25 %    35 5    45 E    55 U    65 e    75 u
   06 ACK    16 SYN    26 &    36 6    46 F    56 V    66 f    76 v
   07 BEL    17 ETB    27 '    37 7    47 G    57 W    67 g    77 w
   08 BS     18 CAN    28 (    38 8    48 H    58 X    68 h    78 x
   09 HT     19 EM     29 )    39 9    49 I    59 Y    69 i    79 y
   0A LF     1A SUB    2A *    3A :    4A J    5A Z    6A j    7A z
   0B VT     1B ESC    2B +    3B ;    4B K    5B [    6B k    7B {
   0C FF     1C FS     2C ,    3C <    4C L    5C \    6C l    7C |
   0D CR     1D GS     2D -    3D =    4D M    5D ]    6D m    7D }
   0E SO     1E RS     2E .    3E >    4E N    5E ^    6E n    7E ~
   0F SI     1F US     2F /    3F ?    4F O    5F _    6F o    7F DEL
```

Step 17

xxd is linux built-in command that is used to display hexadecimal values and **head** command is used to output only header

```
┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ xxd -g 1 work.tar| head
00000000: 77 6f 72 6b 2f 73 77 61 70 5f 64 69 67 67 65 72  work/swap_digger
00000010: 2f 2e 67 69 74 2f 6c 6f 67 73 2f 72 65 66 73 2f  /.git/logs/refs/
00000020: 72 65 6d 6f 74 65 73 2f 6f 72 69 67 69 6e 2f 48  remotes/origin/H
00000030: 45 41 44 00 00 00 00 00 00 00 00 00 00 00 00 00  EAD.............
00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060: 00 00 00 00 30 30 30 30 36 34 34 00 30 30 30 31  ....0000644.0001
00000070: 37 35 30 00 30 30 30 31 37 35 30 00 30 30 30 30  750.0001750.0000
00000080: 30 30 30 30 32 35 33 00 31 34 32 32 35 32 34 33  0000253.14225243
00000090: 30 36 37 00 30 32 31 31 31 37 00 20 30 00 00 00  067.021117. 0...
```

Step 18

String is simple command that grep and display strings in a file

```
┌──(kali㉿kali)-[~/Desktop]
└─$ strings -t x index.jpeg | grep "j"
    5ed ZCkj\
    aa7 Zl>Sj
   106b z@x.~<>?*kj
   1401 jF?h
   1651 0FjI
   173d Zpj%S
   1d56 w<jE
   1fc8 duj~
   24d4 n!@hj
```

Step 19 & 20

The **Sleuth Kit** is a collection digital forensic tools that can be used to analyze disk images and recover files from them:

- **fsstat:** Display general details
- **fls:** List files and directories
- **ils:** List inode information
- **img_cat:** Output contents of an image file
- **fiwalk:** Print the filesystem details

```
┌──(kali㉿kali)-[~/Desktop/work/images]
└─$ ls
nssal-thumb-fs.dd.bz2

┌──(kali㉿kali)-[~/Desktop/work/images]
└─$ bzip2 -dk nssal-thumb-fs.dd.bz2

┌──(kali㉿kali)-[~/Desktop/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)-[~/Desktop/work/images]
└─$ fsstat -i raw nssal-thumb-fs.dd                                  1 ×
FILE SYSTEM INFORMATION
--------------------------------------------

File System Type: FAT16

OEM Name: MSDOS5.0
Volume ID: 0x14d06139
Volume Label (Boot Sector): NO NAME
Volume Label (Root Directory):
File System Type Label: FAT16

Sectors before file system: 233

File System Layout (in sectors)
Total Range: 0 - 999702
* Reserved: 0 - 7
** Boot Sector: 0
* FAT 0: 8 - 251
* FAT 1: 252 - 495
* Data Area: 496 - 999702
** Root Directory: 496 - 527
** Cluster Area: 528 - 999695
** Non-clustered: 999696 - 999702

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)
--------------------------------------------
```

fsstat: Display general details

ils: List inode information

```
kali@kali: ~/Desktop/work/images
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~/Desktop/work/images]
└─$ fsstat -i raw nssal-thumb-fs.dd
FILE SYSTEM INFORMATION
--------------------------------------------

File System Type: FAT16

OEM Name: MSDOS5.0
Volume ID: 0x14d06139
Volume Label (Boot Sector): NO NAME
Volume Label (Root Directory):
File System Type Label: FAT16

Sectors before file system: 233

File System Layout (in sectors)
Total Range: 0 - 999702
* Reserved: 0 - 7
** Boot Sector: 0
* FAT 0: 8 - 251
* FAT 1: 252 - 495
* Data Area: 496 - 999702
** Root Directory: 496 - 527
** Cluster Area: 528 - 999695
** Non-clustered: 999696 - 999702

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)-[~/Desktop/work/images]
└─$ ils nssal-thumb-fs.dd | head
class|host|device|start_time
ils|kali||1649760284
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
```

fiwalk: Print the filesystem details

```
kali@kali: ~/Desktop/work/images
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/work/images]
└─$ img_stat nssal-thumb-fs.dd
IMAGE FILE INFORMATION
--------------------------------------------
Image Type: raw

Size in bytes: 511847936
Sector size:    512

┌──(kali㉿kali)-[~/Desktop/work/images]
└─$ fiwalk nssal-thumb-fs.dd
image_filename: nssal-thumb-fs.dd
fiwalk_version: 4.11.1
start_time: Tue Apr 12 06:45:16 2022
tsk_version: 4.11.1
# fs start: 0
partition_offset: 0
sector_size: 512
block_size: 8192
ftype: 4
ftype_str: fat16
block_count: 999703
first_block: 0
last_block: 999702

parent_inode: 2
filename: _hatever
partition: 1
id: 1
name_type: r
filesize: 511573308
unalloc: 1
used: 1
inode: 3
meta_type: 1
mode: 511
nlink: 0
uid: 0
gid: 0
mtime: 1236020810
mtime_txt: 2009-03-02T19:06:50
atime: 1235970000
atime_txt: 2009-03-02T05:00:00
crtime: 1236020595
crtime_txt: 2009-03-02T19:03:15
md5: cbbcbd0c7251db64038eb3e89079cdbd
sha1: 537bacc2aa7ae309819e10c11356afb602883400

parent_inode: 2
filename: 3323673964_94e64ebddd_b.jpg
partition: 1
```

```
kali@kali: ~/Desktop/work/images
File  Actions  Edit  View  Help
mode: 0
nlink: 1
uid: 0
gid: 0
md5: 796d55af78b2e3441ac5899274c662d5
sha1: 448d8643ffdabd48cbec8aac89901d4cdca6d06a

parent_inode: 2
filename: $FAT2
partition: 1
id: 37
name_type: v
filesize: 124928
alloc: 1
used: 1
inode: 15987317
meta_type: 10
mode: 0
nlink: 1
uid: 0
gid: 0
md5: 796d55af78b2e3441ac5899274c662d5
sha1: 448d8643ffdabd48cbec8aac89901d4cdca6d06a

parent_inode: 2
filename: $OrphanFiles
partition: 1
id: 38
name_type: V
filesize: 0
alloc: 1
used: 1
inode: 15987318
meta_type: 11
mode: 0
nlink: 1
uid: 0
gid: 0

# end of volume
# clock: 6.409157
utime: 5.673115
stime: 0.403981
maxrss: 15940
minflt: 1694
majflt: 5
nswap: 0
inblock: 17496
oublock: 0
clocktime: 6.409157
# stop_time: Tue Apr 12 06:45:22 2022
# =EOF=
```

SUMMARY

This lab was all about learning **Linux Forensics** from Identifying devices using different tool to using STK (**Sleuth Toolkit**). In the First part, we learned to do identification of devices using **p0f** tool. In the Second part, we learned about **arp-scan** and **nmap** scan, as they are used for network and device information gathering. In the Third part, we used **swap_digger**, which is another amazing tool for information gathering of a device, it extracts passwords and much more. In the Fourth part, we tried to do password dumping using **mimipenguin** tool that is available on github. In Fifth part, we did Digital Forensics using **rkhunter** tool, **chkrootkit**, **ascii**, **xxd** and **strings**. In the last part, we used **Sleuth Toolkit** to analyze disk images and recover files.