

# **AI FOR CYBERSECURITY WITH IBM QRADAR**

## **ASSIGNMENT – 2**

### **USAGE OF DIFFERENT CYBER SECURITY TOOLS**

**NAME: SHAIK MUHAMMED FAIZAAN ALI**

**BRANCH: ELECTRONICS AND  
COMMUNICATION ENGINEERING**

**CAMPUS: VIT-VELLORE**

**EMAIL ADDRESS:**

**shaikmuhammed.faizaan2021@vitstudent.acin**

## 13 SECTIONS WITH DIFFERENT TOOLS WITH RESPECTIVE USAGE:

	01 - Information Gathering
	02 - Vulnerability Analysis
	03 - Web Application Analysis
	04 - Database Assessment
	05 - Password Attacks
	06 - Wireless Attacks
	07 - Reverse Engineering
	08 - Exploitation Tools
	09 - Sniffing & Spoofing
	10 - Post Exploitation
	11 - Forensics
	12 - Reporting Tools
	13 - Social Engineering Tools

### 1. Information Gathering

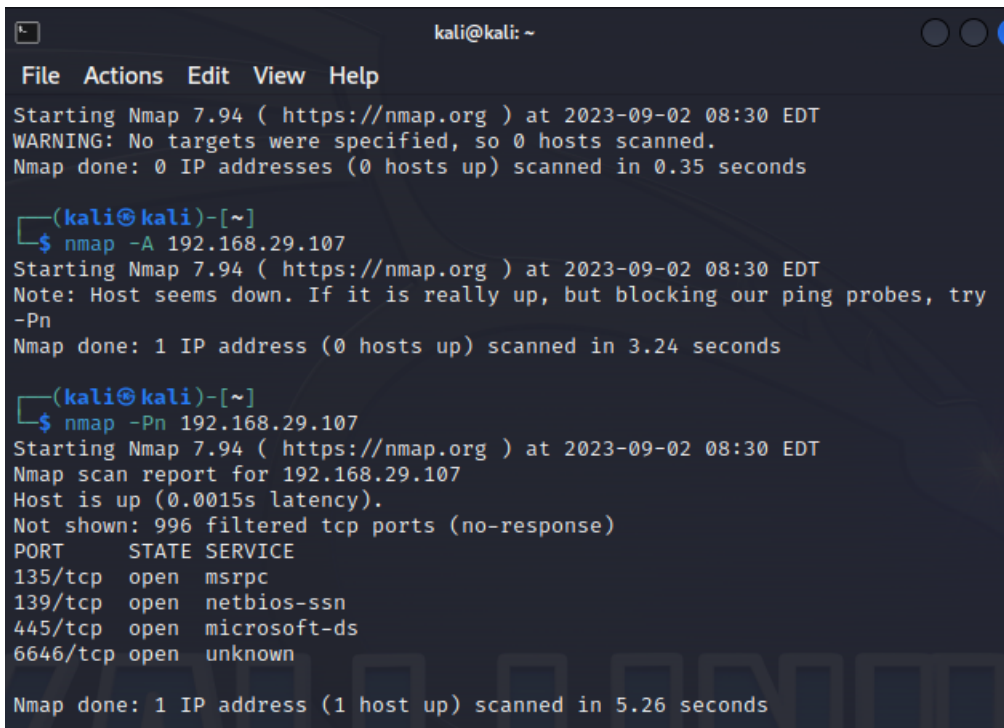
*These are all the tools available under this section.*

▶ 	• DNS Analysis
▶ 	• IDS/IPS Identification
▶ 	• Live Host Identification
▶ 	• Network & Port Scanners
▶ 	• OSINT Analysis
▶ 	• Route Analysis
▶ 	• SMB Analysis
▶ 	• SMTP Analysis
▶ 	• SNMP Analysis
▶ 	• SSL Analysis
	amass
	dmitry
	ike-scan
	legion (root)
	maltego (installer)
	netdiscover
	nmap
	recon-ng
	spiderfoot

### Nmap:

***Nmap ("Network Mapper") is an open-source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.***

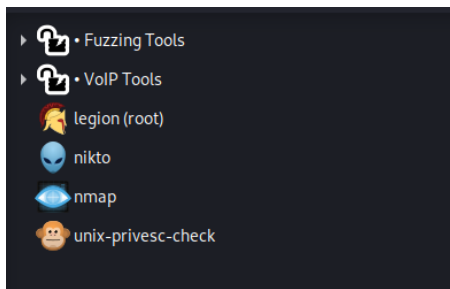
***The output from Nmap is a list of scanned targets, with supplemental information on each depending on the options used. Key among that information is the "interesting ports table". That table lists the port number and protocol, service name, and state. The state is either open, filtered, closed, or unfiltered. Open means that an application on the target machine is listening for connections/packets on that port.***



```
kali@kali: ~  
File Actions Edit View Help  
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-02 08:30 EDT  
WARNING: No targets were specified, so 0 hosts scanned.  
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.35 seconds  
  
(kali@kali)-[~]  
$ nmap -A 192.168.29.107  
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-02 08:30 EDT  
Note: Host seems down. If it is really up, but blocking our ping probes, try  
-Pn  
Nmap done: 1 IP address (0 hosts up) scanned in 3.24 seconds  
  
(kali@kali)-[~]  
$ nmap -Pn 192.168.29.107  
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-02 08:30 EDT  
Nmap scan report for 192.168.29.107  
Host is up (0.0015s latency).  
Not shown: 996 filtered tcp ports (no-response)  
PORT      STATE SERVICE  
135/tcp   open  msrpc  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
6646/tcp  open  unknown  
  
Nmap done: 1 IP address (1 host up) scanned in 5.26 seconds
```

## **2. Vulnerability analysis**

***These are all the tools available under this section.***



### **Unix-privesc-check:**

***It does a vulnerability analysis check on the system its running and gives a detailed information of the vulnerabilities that are found.***

```
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
script didn't find any problems.

(kali@kali)~$ unix-privesc-check standard
Assuming the OS is: linux
Starting unix-privesc-check v1.4 ( http://pentestmonkey.net/tools/unix-privesc-check )

This script checks file permissions and other settings that could allow
local users to escalate privileges.

Use of this script is only permitted on systems which you have been granted
legal permission to perform a security assessment of. Apart from this
condition the GPL v2 applies.

Search the output below for the word 'WARNING'. If you don't see it then
this script didn't find any problems.

#####
Recording hostname
#####
kali

#####
Recording uname
#####
Linux kali 6.3.0-kali1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.3.7-1kali1 (2023-06-29) x86_64 GNU/Linux

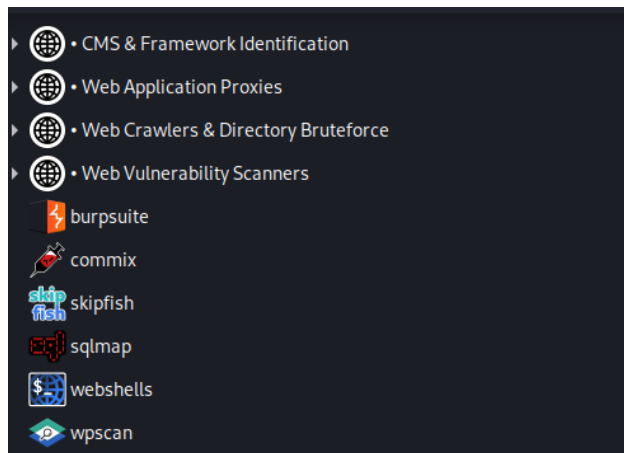
#####
Recording Interface IP addresses
#####
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.156.123 netmask 255.255.255.0 broadcast 192.168.156.255
    inet6 fe80::a00:27ff:fe13:1227 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:13:12:27 txqueuelen 1000 (Ethernet)
    RX packets 36 bytes 2737 (2.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 25 bytes 3220 (3.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

#####
```

### **3. Web Application analysis**

***These are all the tools available under this section.***



**Skip fish:**

*skip fish is an active web application security reconnaissance tool. It prepares an interactive sitemap for the targeted site by carrying out a recursive crawl and dictionary-based probes. The resulting map is then annotated with the output from a few active (but hopefully non-disruptive) security checks. The final report generated by the tool is meant to serve as a foundation for professional web application security assessments.*

```
File Actions Edit View Help
skipfish version 2.10b by lcantuf@google.com

- google.com -

Scan statistics:
  Scan time : 0:00:41.948
  HTTP requests : 1694 (40.5/s), 2301 kB in, 313 kB out (62.3 kB/s)
  Compression : 95 kB in, 115 kB out (9.4% gain)
  HTTP faults : 0 net errors, 0 proto errors, 0 retried, 0 drops
  TCP handshakes : 43 total (40.4 req/conn)
  TCP faults : 0 failures, 0 timeouts, 3 purged
  External links : 20 skipped
  Reqs pending : 42

Database statistics:
  Pivots : 11 total, 11 done (100.00%)
  In progress : 0 pending, 0 init, 0 attacks, 0 dict
  Missing nodes : 3 spotted
  Node types : 1 serv, 3 dir, 2 file, 2 pinfo, 3 unkn, 0 par, 0 val
  Issues found : 13 info, 1 warn, 2 low, 0 medium, 0 high impact
  Dict size : 8 words (8 new), 1 extensions, 117 candidates
  Signatures : 77 total

[!] Scan aborted by user, bailing out!
[+] Copying static resources ...
[+] Sorting and annotating crawl nodes: 11
[+] Looking for duplicate entries: 11
[+] Counting unique nodes: 9
[+] Saving pivot data for third-party tools ...
[+] Writing scan description ...
[+] Writing crawl tree: 11
[+] Generating summary views ...
[+] Report saved to 'google/index.html' [0xeeae0159].
[+] This was a great day for science!
```

A new directory named google has been created. And the report that has been generated is stored in index.html file.

```
(kali㉿kali)-[~]
$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos google
```

```
(kali㉿kali)-[~/google]
$ ls
_i0 _i3 _i6 _n1 i_high.png i_note.png issue_index.js n_clone.png n_failed.png n_unlinked.png p_param.png p_unknown.png samples.js
_i1 _i4 _i7 c0 i_low.png i_warn.png mime_entry.png n_collapsed.png n_maybe_missing.png p_dir.png p_pinfo.png p_value.png sf_name.png
_i2 _i5 _n0 child_index.js i_medium.png index.html n_children.png n_expanded.png n_missing.png p_file.png p_serv.png pivots.txt summary.js
```

**The index file is displayed in the browser:**

The screenshot shows a Kali Linux terminal on the left and a web browser window on the right. The terminal displays Skipfish scan statistics, including HTTP requests, TCP handshakes, and database statistics. The browser window shows the Skipfish interface with a list of crawl results and a document type overview. The document type overview lists various issues found during the scan, such as SSL certificate host name mismatch, self-signed SSL certificate, and response varies randomly.

```
Scan time : 0:00:41.948
HTTP requests : 1694 (40.5/s), 2301 kB in, 313 kB out
Compression : 95 kB in, 115 kB out (9.4% gain)
HTTP faults : 0 net errors, 0 proto errors, 0 retr
TCP handshakes : 43 total (40.4 req/conn)
TCP faults : 0 failures, 0 timeouts, 3 purged
External links : 20 skipped
Reqs pending : 42

Database statistics:
Pivots : 11 total, 11 done (100.00%)
In progress : 0 pending, 0 init, 0 attacks, 0 dict
Missing nodes : 3 spotted
Node types : 1 serv, 3 dir, 2 file, 2 pinfo, 3 unk
Issues found : 13 info, 1 warn, 2 low, 0 medium, 0 h
Dict size : 8 words (8 new), 1 extensions, 117 co
Signatures : 77 total

[!] Scan aborted by user, bailing out!
[*] Copying static resources...
[*] Sorting and annotating crawl nodes: 11
[*] Looking for duplicate entries: 11
[*] Counting unique nodes: 9
[*] Saving pivot data for third-party tools...
[*] Writing scan description...
[*] Writing crawl tree: 11
[*] Generating summary views...
[*] Report saved to 'google/index.html' [0xeeae0159].
[*] This was a great day for science!

(kali@kali)~$ man skipfish
(kali@kali)~$ ls
Desktop Documents Downloads Music Pictures Public
(kali@kali)~$ cd google
(kali@kali)~/google$ ls
_10 _13 _16 _m1 i_high.png i_note.png
_11 _14 _17 c0 i_low.png i_warn.png
_12 _15 _m0 child_index.js i_medium.png index.html
(kali@kali)~/google$ firefox index.html
```

**Crawl results - click to expand:**

- https://google.com/ (2) (1) (13) (7)

**Document type overview - click to expand:**

- application/xhtml+xml (1)
- text/html (3)

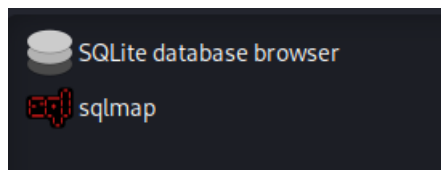
**Issue type overview - click to expand:**

- SSL certificate host name mismatch (1)
- Self-signed SSL certificate (1)
- Response varies randomly, skipping checks (1)
- Hidden files / directories (6)
- Resource not directly accessible (1)
- New 404 signature seen (2)
- New 'X-' header value seen (6)
- New 'Server' header value seen (4)

NOTE: 100 samples maximum per issue or document type.

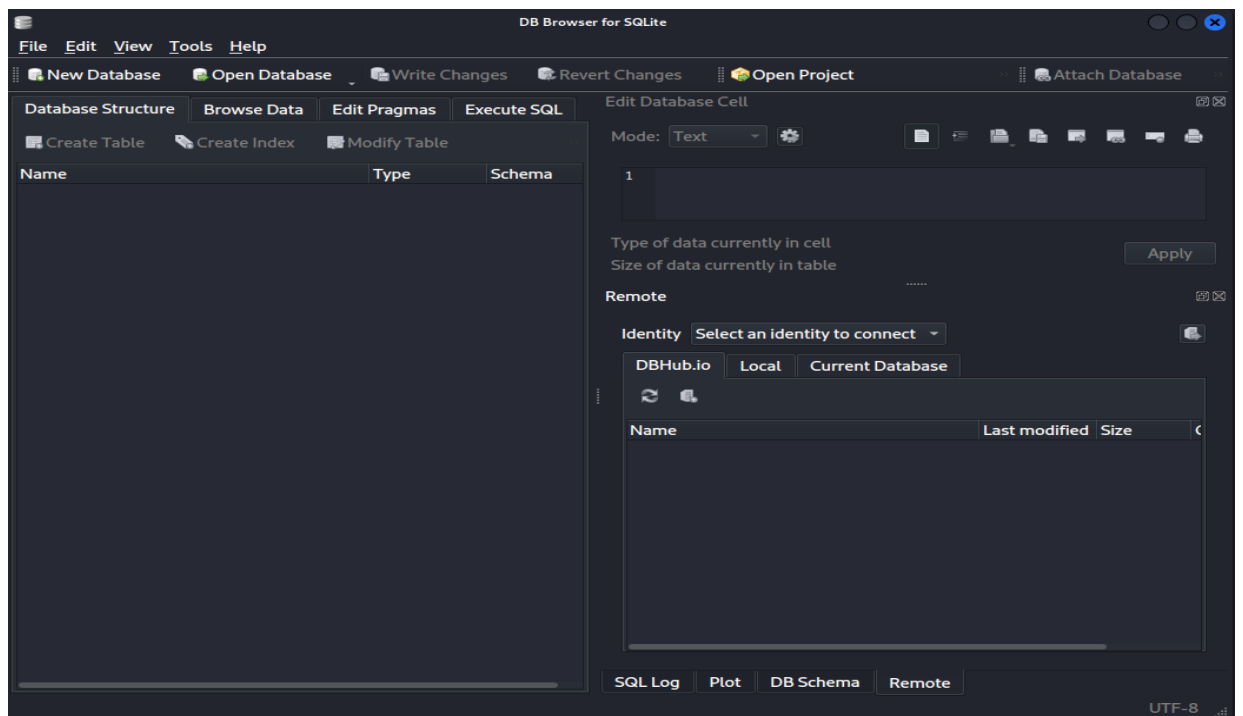
#### 4. Database assessment

**These are all the tools available under this section.**



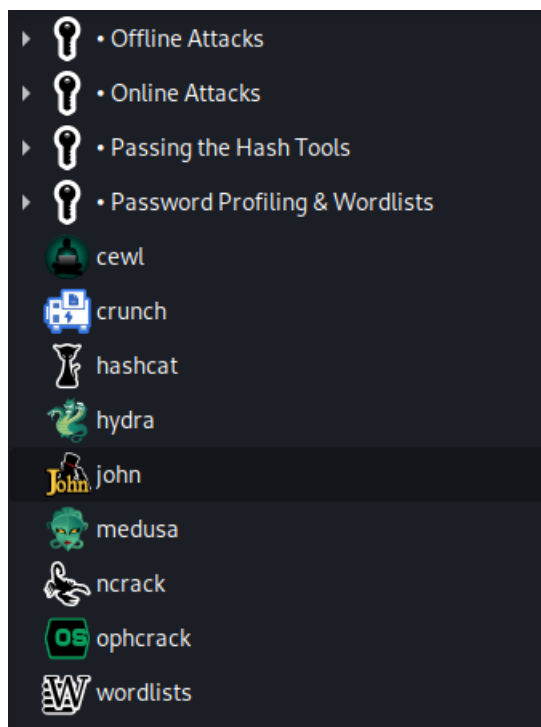
##### SQLite database browser:

**SQLite Database Browser, also known as DB Browser for SQLite, is an open-source graphical user interface (GUI) tool for working with SQLite databases on Linux and other platforms. SQLite is a popular and lightweight relational database management system often used in embedded systems, mobile applications, and desktop software.**



## 5. Password attacks

*These are all the tools available under this section.*



### John the ripper:

*This manual page documents briefly the john command. This manual page was written for the Debian GNU/Linux distribution because the original program does not have a manual page. john, better known as John the Ripper, is a tool to find weak passwords of users in a server. John can use a dictionary or some search pattern as well as a password file to check for passwords. John supports different cracking*



*modes and understands many ciphertext formats, like several DES variants, MD5 and blowfish. It can also be used to extract AFS and Windows NT passwords.*

*To use John, you just need to supply it a password file and the desired options. If no mode is specified, john will try "single" first, then "wordlist" and finally "incremental".*

*Once John finds a password, it will be printed to the terminal and saved into a file called ~/.john/john.pot. John will read this file when it restarts so it doesn't try to crack already done passwords.*

*First, create a text file with some md5 hash generated text in it.*

```
(kali㉿kali)-[~/Desktop]
$ ls
password.txt









(kali㉿kali)-[~/Desktop]
$ cat password.txt
5f4dcc3b5aa765d61d8327deb882cf99
```

*Now we have to use John the ripper to decrypt the text .*

```
(kali㉿kali)-[~/Desktop]
$ john password.txt --format=Raw-MD5
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password (?)
1g 0:00:00:00 DONE 2/3 (2023-09-05 05:51) 20.00g/s 7680p/s 7680c/s 7680C/s 123456..larry
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

## 6. Wireless attacks

*These are all the tools available under this section.*

- ▶  • 802.11 Wireless Tools
- ▶  • Bluetooth Tools
- ▶  aircrack-ng
- ▶  fern wifi cracker (root)
- ▶  kismet
- ▶  pixiewps
- ▶  reaver
- ▶  wifite

**Kismet:**

Nearly all these options are run-time overrides for values in the kismet's configuration file. Permanent changes should be made to the configuration file.

```
$ kismet -h
usage: kismet [OPTION]
Nearly all of these options are run-time overrides for values in the
kismet.conf configuration file.  Permanent changes should be made to
the configuration file.
*** Generic Options ***
-v, --version                Show version
-h, --help                  Display this help message
    --no-console-wrapper    Disable server console wrapper
    --no-ncurses-wrapper    Disable server console wrapper
    --no-ncurses            Disable server console wrapper
    --debug                 Disable the console wrapper and the crash
                             handling functions, for debugging
-c <datasource>             Use the specified datasource
-f, --config-file <file>    Use alternate configuration file
    --no-line-wrap          Turn off linewrapping of output
                             (for grep, speed, etc)
-s, --silent                Turn off stdout output after setup phase
    --daemonize             Spawn detached in the background
    --no-plugins            Do not load plugins
    --homedir <path>       Use an alternate path as the home
                             directory instead of the user entry
    --confdir <path>       Use an alternate path as the base
                             config directory instead of the default
                             set at compile time
    --datadir <path>       Use an alternate path as the data
                             directory instead of the default set at
                             compile time.
    --override <flavor>    Load an alternate configuration override
                             from {confdir}/kismet_{flavor}.conf
                             or as a specific override file.
*** Logging Options ***
-T, --log-types <types>    Override activated log types
-t, --log-title <title>    Override default log title
-p, --log-prefix <prefix>  Directory to store log files
-n, --no-logging           Disable logging entirely
*** Device Tracking Options ***
    --device-timeout=n     Expire devices after N seconds
(kali@kali)~$
```

It scans for the available wireless connection

```

(kali@kali)-[~]
└─$ sudo kismet -c wlan0mon
[sudo] password for kali:
INFO: Including sub-config file: /etc/kismet/kismet_httpd.conf
INFO: Including sub-config file: /etc/kismet/kismet_memory.conf
INFO: Including sub-config file: /etc/kismet/kismet_alerts.conf
INFO: Including sub-config file: /etc/kismet/kismet_80211.conf
INFO: Including sub-config file: /etc/kismet/kismet_logging.conf
INFO: Including sub-config file: /etc/kismet/kismet_filter.conf
INFO: Including sub-config file: /etc/kismet/kismet_uav.conf
INFO: Loading config override file '/etc/kismet/kismet_package.conf'
INFO: Optional sub-config file not present: /etc/kismet/kismet_package.conf
INFO: Loading config override file '/etc/kismet/kismet_site.conf'
INFO: Optional sub-config file not present: /etc/kismet/kismet_site.conf
INFO: Local config and cache directory '/root/.kismet/' does not exist;
      creating it.

INFO: Enabling channel hopping by default on sources which support channel
      control.
INFO: Setting default channel hop rate to 5/sec
INFO: Enabling channel list splitting on sources which share the same list
      of channels
INFO: Enabling channel list shuffling to optimize overlaps
INFO: Sources will be re-opened if they encounter an error
INFO: Saving datasources to the Kismet database log every 30 seconds
INFO: Launching remote capture server on 127.0.0.1 3501
INFO: Data sources passed on the command line (via -c source), ignoring
      source= definitions in the Kismet config file.
INFO: Probing interface 'wlan0mon' to find datasource type
INFO: Opened kismetdb log file './Kismet-20230905-10-11-20-1.kismet'
INFO: Saving packets to the Kismet database log.
INFO: GPS track will be logged to the Kismet logfile
ALERT: ROOTUSER Kismet is running as root; this is less secure. If you
       are running Kismet at boot via systemd, make sure to use `systemctl
       edit kismet.service` to change the user. For more information, see
       the Kismet README for setting up Kismet with minimal privileges.
INFO: Starting Kismet web server...
INFO: HTTP server listening on 0.0.0.0:2501
INFO: Could not open system plugin directory (/usr/lib/x86_64-linux-gnu/kis
      met/), skipping: No such file or directory
INFO: Did not find a user plugin directory (/root/.kismet//plugins/),
      skipping: No such file or directory
ERROR: Unable to find driver for 'wlan0mon'. Make sure that any required
       plugins are loaded, the interface is available, and any required
       Kismet helper packages are installed.
ERROR: Data source 'wlan0mon' failed to launch: Unable to find driver for
       'wlan0mon'. Make sure that any required plugins are loaded, the
       interface is available, and any required Kismet helper packages are
       installed.

```

## 7. Reverse engineering

These are all the available tools under this section.

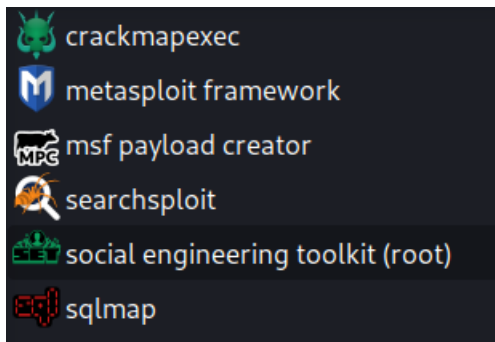
```

└─ clang
└─ clang++
└─ NASM shell
└─ radare2

```

## 8. Exploitation tools

*These are all the available tools under this section*



## 9. Sniffing and Spoofing

*These are all the available tools under this section*



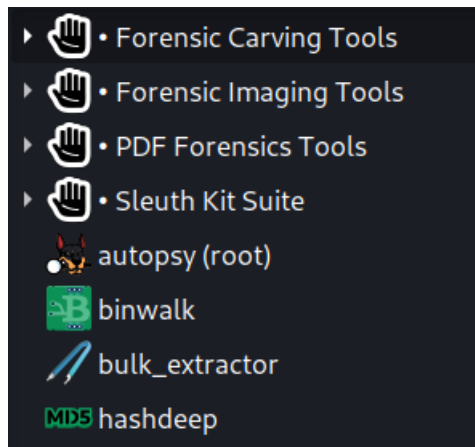
## 10. Post Exploitation:

*These are all the available tools under this section*



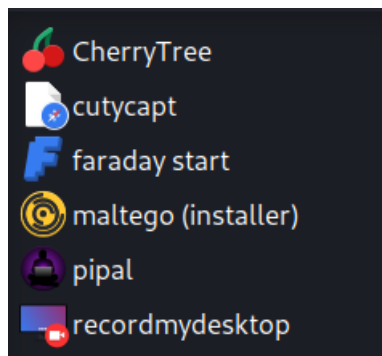
## 11. Forensics

*These are all the available tools under this section*



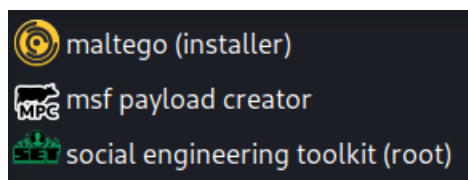
## 12. Reporting tools

*These are all the available tools under this section*



## 13. Social Engineering tools

*These are all the available tools under this section*



\*\*\*\*\* **THE END** \*\*\*\*\*