

**Amrita Vishwa Vidyapeetham**

Name: karthikeyan G

Roll No: cb.sc.p2cys24008

**24CYS682 – Cyber Security Lab**

**Assignment 1: Exploring Tools in Kali Linux**

# Contents

1 Information Gathering	3
2 Vulnerability Analysis	4
3 Web Application Analysis	6
4 Database Assessment	7
5 Password Attacks	8
6 Wireless Attacks	9
7 Reverse Engineering	10
8 Exploitation Tools	11
9 Sniffing & Spoofing	12
10 Post Exploitation	13
11 Forensics	14
12 Social Engineering Tools	14

# ☒ 1 Information Gathering

## 1. Nmap:

- Nmap is used to scan networks and find active hosts, open ports, and services running on those ports.
- It helps analyze system vulnerabilities by mapping the network's structure.
- Security analysts rely on it for reconnaissance before deeper penetration testing.

```
--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
-e <iface>; Use specified interface
-g/-source-port <portnum>; Use given port number
--proxies <url1,[url2],...>; Relay connections through HTTP/SOCKS4 proxies
--data <hex string>; Append a custom payload to sent packets
--data-string <string>; Append a custom ASCII string to sent packets
--data-length <num>; Append random data to sent packets
--ip-options <options>; Send packets with specified ip options
--ttl <val>; Set IP time-to-live field
--spoof-mac <mac address/prefix/vendor name>; Spoof your MAC address
--badsum: Send packets with a bogus TCP/UDP/SCTP checksum
OUTPUT:
-oN/-oX/-oS/-oG <file>; Output scan in normal, XML, s|<rIpt kIddi3,
    and Grepable format, respectively, to the given filename.
-oA <basename>; Output in the three major formats at once
-v: Increase verbosity level (use -vv or more for greater effect)
-d: Increase debugging level (use -dd or more for greater effect)
--reason: Display the reason a port is in a particular state
--open: Only show open (or possibly open) ports
--packet-trace: Show all packets sent and received
--iflist: Print host interfaces and routes (for debugging)
--append-output: Append to rather than clobber specified output files
--resume <filename>; Resume an aborted scan
--noninteractive: Disable runtime interactions via keyboard
--stylesheet <path/URL>; XSL stylesheet to transform XML output to HTML
--webxml: Reference stylesheet from Nmap.Org for more portable XML
--no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
-6: Enable IPv6 scanning
-A: Enable OS detection, version detection, script scanning, and traceroute
--datadir <dirname>; Specify custom Nmap data file location
--send-eth--send-ip: Send using raw ethernet frames or IP packets
--privileged: Assume that the user is fully privileged
--unprivileged: Assume the user lacks raw socket privileges
-V: Print version number
-h: Print this help summary page.
EXAMPLES:
nmap -v -A scanme.nmap.org
nmap -v -sn 192.168.0.0/16 10.0.0.0/8
nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
[karthikeyan@kali:~]
$
```

## 2. Recon-ng:

- Recon-ng automates data gathering about domains, IPs, or emails using open-source intelligence.
- It provides modules for seamless API integrations, like Shodan and VirusTotal.
- Its lightweight framework is perfect for organizing and reporting findings.

### 3. Maltego:

- Maltego is a visualization tool that links data points like domains, IPs, and email addresses.
  - It's great for mapping relationships during investigations, particularly in OSINT.
  - Investigators use it to uncover connections that aren't immediately visible.



## 2 Vulnerability Analysis

#### 4. Nikto:

- Nikto scans web servers to find outdated software, insecure configurations, or vulnerabilities.
  - It provides detailed information about headers, subdomains, and security missteps.
  - It's a fast and effective way to identify low-hanging security risks.

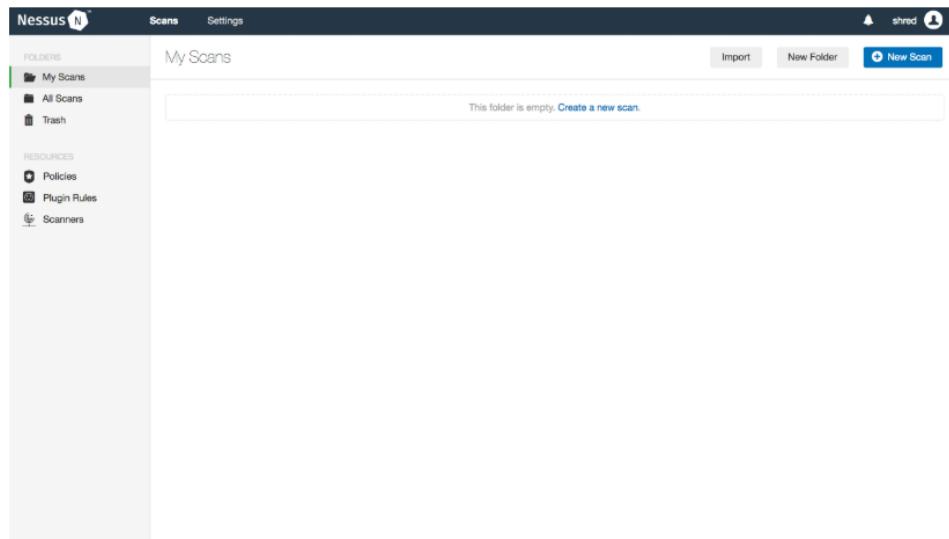
```

$ nikto --Help
Options:
  -ask*           Whether to ask about submitting updates
                  yes Ask about each (default)
                  no Don't ask, don't send
  -check*         Scan these CGI first: 'none', 'alt', or values like "/cgi-bin/test.cgi"
  -cgidirs*       Use this to scan for specific CGI Directories
  -config*        Test on/off display outputs:
                  1 Show redirects
                  2 Show all received
                  3 Show all 200/OK responses
                  4 Show URLs which require authentication
                  D Don't show errors
                  E Display all HTTP errors
                  P Print progress to STDOUT
                  S Scan for specific IPs and hostnames
                  V Verbose output
  -dbcheck*       Check database and other key files for syntax errors
  -encoding*      Encoding detection:
                  1 Random URI encoding (non-UTF8)
                  2 Directory self-reference ('./')
                  3 Postscript self-reference
                  4 Prepend long random string
                  5 Fake parameter
                  6 TAA (Temporary Acceptable Address)
                  7 Change the case of the URL
                  8 Use Windows directory separator (\)
                  A Use binary value 0x0B as a request spacer
                  B Use binary value 0x0B as a request spacer
  -followedirects Follow 3xx redirects to new location
  -formats*       Scan file formats:
                  csv Comma-separated-value
                  json JSON Format
                  hml HTML Format
                  nbs Nessus NBS format
                  sql Generic SQL (see docs for schema)
                  txt Plain Text
                  xml XML Format
                  (if not specified the format will be taken from the file extension passed to -output)

```

## 5. Nessus:

- Nessus is a powerful vulnerability scanning tool designed to identify weaknesses in systems, applications, and networks.
- It provides in-depth reports with recommendations for mitigating identified risks and prioritizes vulnerabilities based on criticality.
- Security professionals frequently use Nessus to perform comprehensive assessments and ensure compliance with industry standards.
- It supports scanning for misconfigurations, outdated software, and known vulnerabilities across various platforms.
- Nessus is widely regarded for its extensive plugin library, which allows it to detect over 50,000 vulnerabilities, including zero-day threats.



## 6. Lynis:

- Lynis performs security audits on Linux systems, scanning for misconfigurations and weak points.
- It generates suggestions for improving hardening and reducing risk.
- Its command-line interface makes it ideal for advanced system assessments.

```
(karthikayan㉿kali):[~]
$ lynis
[ lynis 3.1.2 ]
=====
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it under the terms of the GNU General Public License.
See the file "LICENSE" for details on how to use this software.

2022-2024, CISQIY - https://cisyfy.com/lynis
Enterprise support available (compliance, plugins, interface and tools)
=====

[*] Initializing program
-----
Usage: lynis command [options]

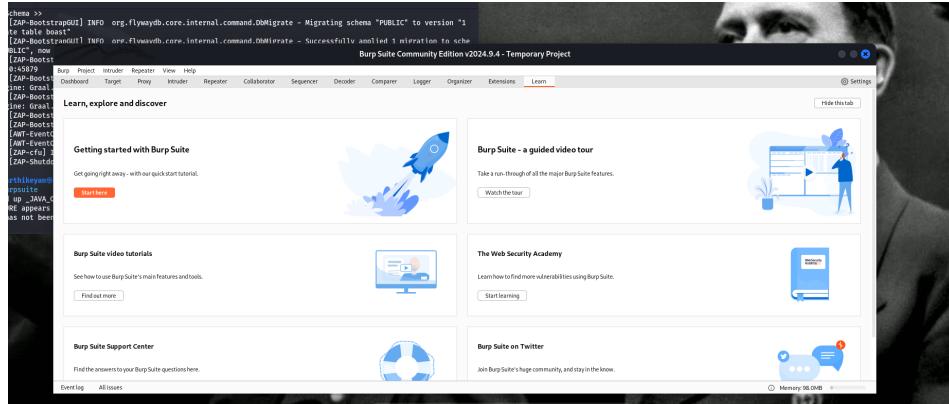
Command:
  audit      audit system           : Perform local security scan
  audit system remote ghost*       : Remote security scan
  audit dockerfile file*         : Analyze Dockerfile
  show       show                  : Show all commands
  show version        : Show lynis version
  show help          : Show help
  update      update info         : Show update details

Options:
  Alternative system audit modes
  --forensics      : Perform forensics on a running or mounted system
  --pentest        : Non-privileged, show points of interest for pentesting
```

## 3 Web Application Analysis

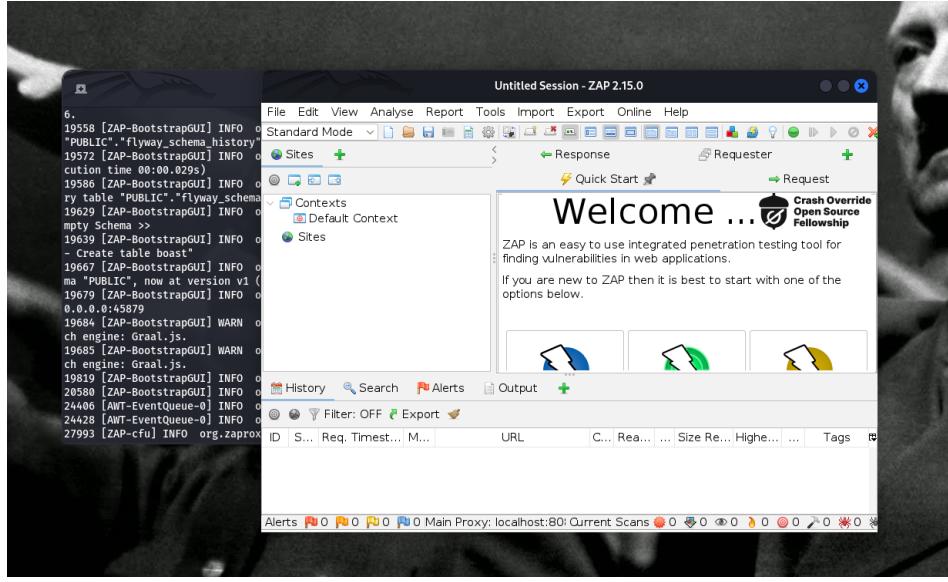
### 7. Burp Suite:

- Burp Suite intercepts and manipulates web traffic to identify vulnerabilities in web applications.
- Testers use it to simulate attacks like injection or authentication bypasses.
- It also allows for automated scans alongside manual testing.



### 8. OWASP ZAP:

- OWASP ZAP is an open-source tool for finding vulnerabilities in web applications.
- It simulates attacks to expose issues like XSS or SQL injection.
- Its automated spider feature helps map the entire application structure.



## 9. Wapiti:

- Wapiti scans web applications for vulnerabilities like file inclusion, XSS, and CRLF injections.
- It works by crawling the application and launching various payloads.
- It's lightweight but still provides a thorough vulnerability report.

```
$ wapiti -h
Usage: wapiti [OPTIONS] [-s scope] [-o folder, domain, url, punk]
  -s MODULES [--list-modules] [-u update] [-l LEVEL]
  [-p PROXY_URL] [-t tor] [-a CREDENTIALS]
  [-c auth-type [basic,digest,kerberos,ntlm,post]] [-c COOKIE_FILE]
  [-r proxy-crash] [-r rescan-crash] [-e exploit]
  [-f store-config-store-path] [-s store-config PATH]
  [-s URL] [-x URL] [-r PARAMETER] [-s skip PARAMETER] [-d DEPTH]
  [-max-links-per-page MAX] [-max-files-per-dir MAX]
  [-max-threads MAX] [-max-parallel MAX]
  [-max-parameters MAX] [-s force] [-t seconds] [-h header]
  [-a agent] [-v verify-ssl {0,1}] [-c color] [-v level] [-f format]
  [-o output_path] [-e external-endpoint EXTERNAL_ENDPOINT_URL]
  [-i internal-endpoint INTERNAL_ENDPOINT_URL]
  [-e endpoint ENDPOINT_URL] [-m merge-report] [-v version]

wapiti-3.0.4: Web application vulnerability scanner
```

## 4 Database Assessment

### 10. sqlmap:

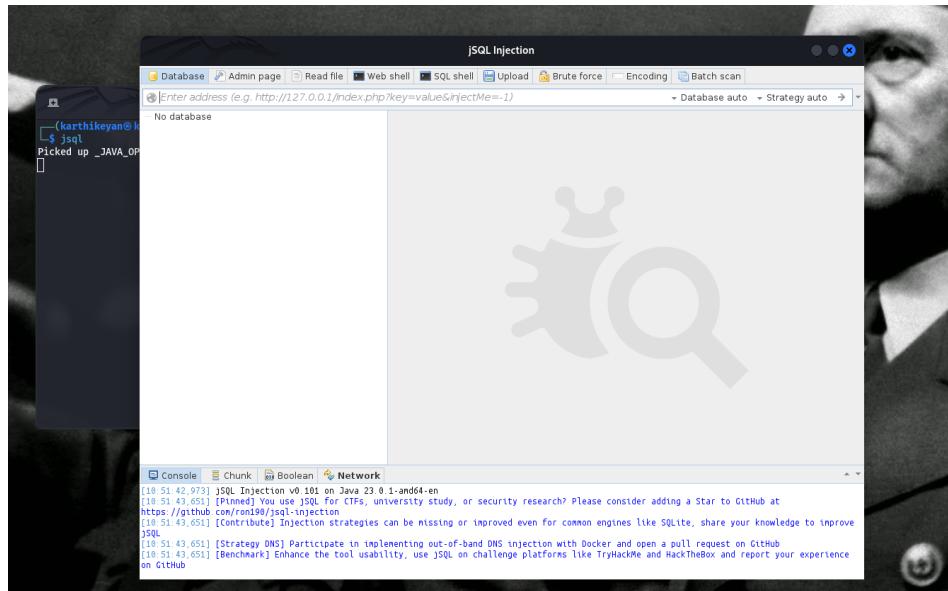
- sqlmap automates SQL injection attacks to exploit database vulnerabilities.
- It can dump databases, test blind SQL injection, and even crack hashes.
- It's perfect for both detection and exploitation of database flaws.

```
[karthikeyan@kali:~]
$ sqlmap
H
{1.8.11#stable}
https://sqlmap.org

Usage: python3 sqlmap [options]
sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, --wizard, --shell, --update, --purge, --list-tampers or --dependencies). Use -h for basic
and -hh for advanced help
```

## 11. jSQL Injection:

- jSQL is a lightweight tool for detecting and exploiting SQL injection vulnerabilities.
  - It supports different attack types, like blind, time-based, and error-based.
  - It's beginner-friendly while still being effective for testing.



## 5 Password Attacks

## 12. Hydra:

- Hydra performs brute-force attacks on network protocols like SSH, FTP, or Telnet.
  - It's highly configurable with options to test usernames, passwords, or both.
  - It's a go-to tool for testing weak or default credential setups.

### 13. John the Ripper:

- John the Ripper cracks hashed passwords using brute-force or dictionary attacks.

- It supports a variety of password formats, including MD5 and SHA.
- Security testers use it to ensure proper password policies are enforced.

```

$ /usr/share/kali-memu/helper-scripts/john
Created directory: /home/karthikyan/john
John the Ripper 1.9.0-jumbo-1+leeding-sec1328d6c 2021-11-02 30:45:52 +0100 OMP [linux-gnu 64-bit x86_64 SSE AC]
Copyright (c) 1996-2021 by Solar Designer and others
homepage: https://www.openwall.com/john/
Usage: john [OPTIONS] [PASSWORD-FILES]

--help          Print usage summary
--single[=SECTION...]  "Single crack" mode, using default or named rules
--single[=rule]  Same, using "immediate" file
--single[=wordlist,WORD] Same, using wordlist[FILE] for all salts in single mode
--single[=wordlist[FILE]] <Shorts wordlist with static seed words/morphemes>
--single[=user-seed[FILE]] Wordlist with seeds per username (user:password[s])
--force         Force crack
--single-pair-maxN  Override max. number of word pairs generated (6)
--no-single-pair Disable single word pair generation
--no-single-retest-dict  Override config for SingleRetestSection
--wordlist[=FILE]  Read wordlist from FILE or stdin
--pipe          Like --stdin, but bulk reads, and allows rules
--rules[=SECTION...]  Enable word mangling rules (for wordlist or PRINCE)
--rules[=rule[...]]   Same, using "immediate" rules
--rules[=rulef[...]]  Same, using "immediate" rules
--rules[=stack:SECTION...] Stacked rules, applied after regular rules or to
nodes that otherwise don't support rules
--rules[=stack:rule[...]] Same, using "immediate" rules
--rules[=skip-nop] Skip any NOP "-;" rules (you already ran w/o rules)
--loopback[=FILE]  Like --wordlist, but extract words from a .pot file
--size-threshold[SIZE]  Size threshold for wordlist (default: 2048 kB)
--dupe-suppression  Suppress all dupes in wordlist (and force prdict)
--incremental[=MODE]  "Incremental" mode (using section MODE)
--incremental-prdict[=WORDLIST]  Mask mode using WORDLIST (or default from john.conf)
--mask[=MASK]      "Markov" mode (see doc/MARKOV)
--markov[=OPTIONS] Mask mode options
--maskf[=FILE]     Mask mode file
--prince-loopback[=FILE] Fetch words from a .pot file
--prince[=WORDLIST] Mask mode, read words from WORDLIST
--prince-elem-cut-max[-l] Maximum number of elements per chain (-l is
relative to word length) (8)
--prince-skip-N  Initial skip
--prince-list=N  Limit N of candidates generated
--prince-wl-dist-len Calculate length distribution from wordlist
--prince-wl-max=N Load only N words from input wordlist
--prince-case-permute Permute case in output letters
--prince-map=mapfile  Memory map infile (not available with case permute)

```

#### 14. Hashcat:

- Hashcat is used to crack password hashes via brute-force, hybrid, or rule-based attacks.
- It supports GPU acceleration, making it extremely fast for cracking tasks.
- It's widely used in practical password recovery or policy testing.

```

$ hashcat -h
hashcat (v6.2.6) starting in help mode
Usage: hashcat [options]... hash[hashfile|hccapxfile [dictionary|mask|directory]...
- [ Options ] -
Options Short / Long           | Type | Description                                | Example
-----+-----+-----+-----+-----+-----+
--m, --hash-type    Num   | Hash-type, references below (otherwise autodetect) | -m 1000
--auto-mode        Num   | Auto mode, see references below                | -o 3
--v, --version      Num   | Print version
--h, --help          Print help
--quit             Num   | Stop cracking
--hex-salt          Str   | Assume salt is given in hex
--hex-charset       Str   | Assume charset is given in hex
--hex-salt          Str   | Assume salt is given in hex
--hex-wordlist      Str   | Assume words in wordlist are given in hex
--wordlist          Str   | Assume words in wordlist are given in hex
--depreciated       Str   | Ignored
--deprecated-check-disable Str   | Enable deprecated plugins
--status            Str   | Enable automatic update of the status screen
--status-json        Str   | Enable JSON output for the status screen
--status-timer       Num   | Sets seconds between status screen updates to X
--stdin-timeout-abort Num   | Abort if there is no input from stdin for X seconds
--machine-readable  Str   | Machine readable output format
--keep-crash        Str   | Keep crashing the hash after it has been cracked
--self-test-disable Str   | Disable self-test functionality on startup
--loopback          Str   | Add more plausible attack vectors
--mask-hoststat2    File  | Load hoststat2 file to use
--markov-disable    Str   | Disables markov-chains, emulates classic brute-force
--markov-classic   Str   | Enables classic markov-chains, no position
--markov-threshold Str   | Threshold X when to stop accepting new markov-chains
--t, --markov-threshold Num   | Threshold X when to stop accepting new markov-chains | -t 50
--runtime           Num   | Abort session after X seconds of runtime
--session           Num   | Abort session after X seconds of runtime
--restore           Str   | Restore session from session
--restore-from-session Str   | Restore session from session
--restore-disable  Str   | Do not write restore file
--restore-file-path File  | Specific path to restore file
--o, --outfile       File  | Output file to save cracked hash
--outfile-format    Str   | Outfile format to use, separated with commas
--outfile-autodec-h  Str   | Disable the use of SHEX() in output plains
--outfile-check-timer Num   | Set timer for output file
--wordlist-to-hex-disable Str   | Disable the conversion of SHEX() from the wordlist
-p, --separator      Char  | Separator char for wordlists and outfile
--stdout            Str   | Do not crack, just list cracked passwords only
--show              Str   | Compare wordlist with outfile, show cracked hashes
--left              Str   | Compare hashlist with outfile; show uncracked hashes
--username          Str   | Enable ignoring of usernames in hashfile
--source            Str   | Enable removal of hashes once they are cracked

```

## 6 Wireless Attacks

#### 15. Aircrack-ng:

- Aircrack-ng is used to capture and analyze Wi-Fi packets for cracking WEP/WPA keys.
- It relies on techniques like dictionary or statistical attacks for key recovery.
- It's a go-to tool for assessing Wi-Fi network security weaknesses.

```
[~] -> aircrack-ng [-]
[~] -> aircrack-ng

Aircrack-ng 1.7 - (C) 2006-2022 Thomas d'Otreppe
https://www.aircrack-ng.org

usage: aircrack-ng [options] <input file(s)>

Common options:

--<mode> : force attack mode (1/WEП, 2/WPA-PSK)
--essid   : target selection: network identifier
--b bssid   : target selection: wireless interface
--p corepu : # of CPU to use (default all CPUs)
--q        : enable quiet mode (no status output)
-C names  : merge the given APs to a virtual one
--l <file> : write key to file. Overwrites file.

Static WEP cracking options:

-c          : search alphanumeric characters only
-i          : search binary characters only
-h          : search the numeric key for FritzBOX
-d <mask>  : use masking of the key (A1:XX:Cf:YY)
--e <mac>   : MAC address to filter incoming packets
-n <chbits> : set key length to 40/64/128/152
-i <index>  : WEP key index (1 to 4), default: any
-f <fudge>  : bruteforce fudge factor, default: 2
-k <key>    : disable one attack mode (1 to 17)
-x <key>    : disable all attack modes for last selected
-x1         : last keybyte bruteforcing (default)
-x2         : enable last 2 keybytes bruteforcing
-X          : disable bruteforce multithreading
-y          : symmetric multithreaded attack mode
-K          : use only old Korek attacks (pre-PW)
-s          : show the key in ASCII while cracking
-M <num>   : specify maximum number of IVs to use
-J          : use Jumbo IVs attack, skip small IV streams
-P <num>   : PTW debug, disable Kleptobot PTW
-1          : run only 1 try to crack key with PTW
-V          : run in visual inspection mode

WEP and WPA-PSK cracking options:
```

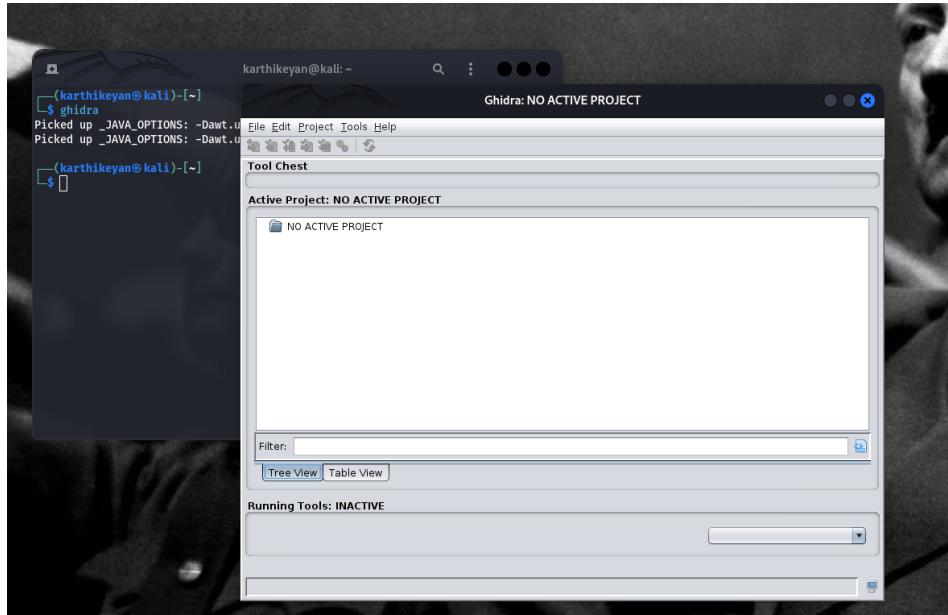
## 16. Wifite:

- Wifite automates Wi-Fi network attacks, focusing on WEP, WPA, and WPS.
  - It's a simple solution for launching deauthentication or handshake-capture attacks.
  - Security testers use it to quickly assess multiple networks.

## 7 Reverse Engineering

17. Ghidra:

- Ghidra is a reverse engineering tool that decompiles binaries into human-readable code.
  - Analysts use it to study malware behavior or analyze software.
  - It's highly modular, allowing for plugin development.



## 18. Radare2:

- Radare2 disassembles binary code for analysis and debugging.
- It supports various architectures and is highly customizable for advanced users.
- It's ideal for analyzing complex malware or unauthorized code.

```
$ radare2 -h
Usage: r2 [-Acpu] [-Wnogitversion] [-P patch] [-g prj] [-a arch] [-b bits] [-c cmd]
          [-s addr] [-B baddr] [-n mode] [-i script] [-e key] [file|pid|-|-|-]
--          run radare2 without opening any file
-s          same as `r2 malloc://$size` (useful for memory dump)
--          read file from standard input and -c to run cmds)
-p          port number to bind to run all commands remotely
-o          print \x00 after init and every command
-2          close stderr file descriptor (silent warning messages)
-a [arch]  set arch
-A          run 'asm' command to analyze all referenced code
-b [bits]  set asm.bits
-B [baddr] set base address for PIE binaries
-c [cmd..] run radare2 command
-C          file to start script (alias for -c<http://?s/cmd>
-d          debug the executable 'file' or running process 'pid'
-d [backend] enable debug mode (e.g., debug=true)
-e kw     enable keyword view
-f          block size = file size
-F [bingplug] force to use that rbin plugin
-h, -hh    show help message, -hh for long
-H [var]   display variable
-i [file]  run script file
-I [file]  run script file before the file is opened
-j [jsfunc] use json for -v, -I and maybe others
-k [os/kern] load kernel module (macos, w2k, netbsd, ...)
-l [lib]   load plugin file
-L, -Ll   list supported IO plugins (-L list core plugins)
-m [addr]  map file at given address (loadaddr)
-N        do not load user settings
-n, -m    do not load RBin Info (-mn only load bin structures)
-N        do not load user settings and scripts
-NN       do not load any script or plugin
-q        quiet mode (no output), -qf to defer -i
-qq      quit after running all -c and -i
-Q        quiet mode (no prompt) and quit faster (quickleak=true)
-p [prj]   use project, -p t for no arg, load if no file
-P [prj]   update patch file and -p
-r [ravunl] specify ravr2 profile to load (same as -e dbg.profile=x)
-R [rrrule] specify custom rr rule directive
-s [addr]  initial seek
-S        start in sandbox mode
-t        load rabin2 info in thread
-u        set bin.filter=false to get raw sym/sec/cls names
-v, -V    show version and versions (-V show lib versions)
-w        open file in write mode
-x        open without exec-flag (asm.emu will not work). See io.exec
-x        same as -e bin.usetrue=false (useful for dildcache)
```

## 8 Exploitation Tools

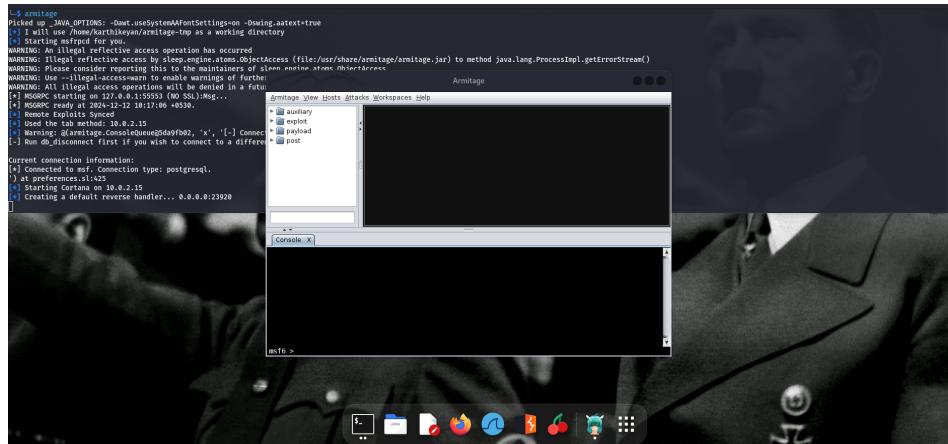
### 19. Metasploit Framework:

- Metasploit is a powerful tool for discovering, exploiting, and verifying vulnerabilities.

- It includes a massive library of pre-built exploits and payloads.
  - It's widely used for penetration testing and red team activities.

## 20. Armitage:

- Armitage provides a GUI interface to the Metasploit Framework for managing exploits.
  - It's beginner-friendly and makes collaboration on exploitation tasks easier.
  - Its interface visualizes attack paths and system weaknesses.

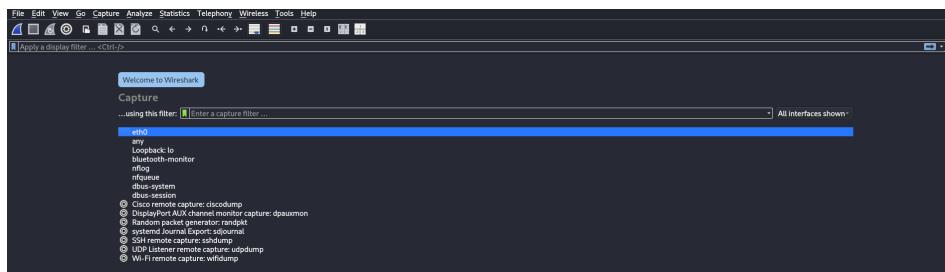


## 🛠️ 9 Sniffing & Spoofing

## 21. Wireshark:

- Wireshark captures and analyzes network traffic in real time.
  - It's perfect for identifying malicious packets or debugging network issues.

- Its filtering system makes pinpointing specific data straightforward.



## 22. Bettercap:

- Bettercap performs network attacks like ARP spoofing and DNS poisoning.
- It also monitors traffic for credentials or other sensitive data.
- It's a versatile tool for advanced red team operations.

```
$ bettercap -h
Usage of bettercap:
-autostart string
    Comma-separated list of modules to auto start. (default "events.stream")
-caplet string
    Read module code from this file and execute them in the interactive session.
-caplets-path string
    Specify an alternative base path for caplets.
-cpu-profile file
    The CPU profile file.
-debug
    Print debug messages.
-env-file string
    Load environment variables from this file if found, set to empty to disable environment persistence.
-eval string
    Run one or more commands separated by ; in the interactive session, used to set variables via command line.
-gateway-override string
    Set the provided IP address instead of the default gateway. If not specified or invalid, the default gateway will be used.
-iface string
    Network interface to bind to, if empty the default interface will be auto selected.
-no-profile file
    Write memory profile to file.
-no-colors
    Disable color output color effects.
-no-history
    Disable interactive session history file.
-pcap-buffer-size int
    PCAP buffer size, leave to 0 for the default value. (default -1)
-script string
    Load a session script.
-silent
    Suppress all logs which are not errors.
-version
    Print the version and exit.
```

## 10 Post Exploitation

### 23. Empire:

- Empire is a post-exploitation framework that helps maintain persistent access.
- It supports modules for keylogging, privilege escalation, and data exfiltration.
- Attackers use it for stealthy, long-term system compromise.

```
$ sudo powershell-empire -h
[sudo] password for karthikeyan:
Create mysql database empire
usage: empire.py [-h] {server,client} ...

positional arguments:
  {server,client}
    server      Launch Empire Server
    client      Launch Empire CLI

options:
  -h, --help      show this help message and exit
  [karthikeyan@kali)-[~]
  $
```

11 Forensics

#### **24. Autopsy:**

- Autopsy helps recover deleted files, analyze hard drives, and extract metadata.
  - It's commonly used in digital forensic investigations for evidence collection.
  - Its user-friendly interface simplifies the forensic process.

```
$ sudo autopsy
[sudo] password for karthikeyan:
=====
Autopsy Forensic Browser
http://www.sleuthkit.org/autopsy/
ver 2.24
=====
Evidence Locker: /var/lib/autopsy
Start Time: Thu Dec 12 10:06:48 2024
Remote Host: localhost
Local Port: 9999

Open an HTML browser on the remote host and paste this URL in it:
  http://localhost:9999/autopsy

Keep this process running and use <ctrl-c> to exit
*CEnd Time: Thu Dec 12 10:06:52 2024
```

## 12 Social Engineering Tools

## 25. Social Engineering Toolkit (SET):

- SET is a framework for creating phishing attacks, malicious payloads, and fake websites.
  - It's tailored for exploiting human vulnerabilities in security.
  - Organizations use it to test the awareness of their employees.