

CYBER SECURITY LABORATORY

EXPLORING TOOLS IN KALI LINUX

Objective: In this assignment, you are required to explore 2-3 tools available in Kali Linux. For each tool, provide a brief explanation (2-3 lines) detailing what the tool is and its primary use case. Additionally, include a screenshot of the tool's home page or interface.

Information Gathering:

Nmap: (*Network mapper*)

- Nmap is a powerful open-source tool used for network discovery and security auditing.
- It can scan large networks quickly, identifying live hosts, open ports, services, and operating systems.
- Nmap is widely used by network administrators and security professionals to perform reconnaissance and vulnerability assessments.

Primary Use Cases:

- Scanning networks to discover hosts and services.
- Identifying open ports and running services on target machines.
- Detecting vulnerabilities and misconfigurations.

```
varshini@kali: ~
```

```
$ nmap
Nmap 7.94SVM ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ext: scannme, randomize, targetports, 192.168.0.1; 10.0.0-255.1-254
  -iL: List of files/hosts: Input from list of hosts/networks
  -iR: CNAME hosts?: Choose random targets
  --exclude host1[,host2][,host3]...: Exclude hosts/networks
  --excludefile <exclude file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan: simply list targets to scan
  -sn: Only scan port 0
  -pN: Treat all hosts as online -- skip host discovery
  -PS/PV/PU/PV[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-A: Never resolve hostnames/addresses resolve [default: sometimes]
  --nameservers <ns1[,ns2,...]>: Specify custom DNS servers
  --systems-dns: Use OS's DNS resolution
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/T/A/S/W/M: TCP SYN/Connect() /ACK/Window/Maimon scans
  -sU: UDP Scan
  -sF: FIN Scan
  -sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI: zombie host[:probeport]: Idle scan
  -sV/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b: Bounce scan
  -B: Bounce scan with FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p<port ranges>: Only scan specified ports
  Ex: -p22; -p1-65535; -p U:53,111,137,T:2-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports sequentially - don't randomize
  --top-ports <numports>: Scan <numports> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Try only the most likely service (Intensity 2)
  --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
```

Amass

- Amass is an open-source tool developed by the OWASP Foundation for performing advanced network mapping and asset discovery.
 - It is primarily used for subdomain enumeration, discovering external assets, and mapping attack surfaces by gathering

information from public sources and active reconnaissance techniques.

Primary Use Cases:

- Discovering subdomains associated with a target domain.
 - Mapping an organization's external infrastructure.
 - Performing reconnaissance to identify potential attack vectors.

Vulnerability Analysis:

unix-privesc-check

- unix-privesc-check is a command-line tool designed to identify potential privilege escalation paths on Unix-based systems.
 - It scans the system for misconfigurations, improper file permissions, and other vulnerabilities that can allow an attacker to elevate privileges from a standard user to root.
 - It is particularly useful for post-exploitation analysis.

Primary Use Cases:

- Identifying potential privilege escalation vulnerabilities.
 - Auditing system configurations for insecure settings.
 - Assisting penetration testers in finding paths to root access.

Nikto:

- Nikto is an open-source web server scanner that performs comprehensive tests to detect vulnerabilities, misconfigurations, and outdated software on web servers.
 - Developed in Perl, Nikto quickly scans for over 6,700 potentially dangerous files or programs, checks for outdated versions of server software, and identifies common security issues.

Primary Use Cases:

- Identifying vulnerabilities in web servers.

- Checking for misconfigurations and outdated software.
- Performing security assessments on web applications.

```
varshini@kali:~$ nikto -h
Option host requires an argument

Options:
  -ask+           Whether to ask about submitting updates
                  yes  Ask about each (default)
                  no   Don't ask, don't send
  -check#         Check for versioning (connects to ipv6.google.com or value set in nikto.conf)
  -cgidirs+       Scan these CGI dirs: "none", "all", or values like "/cgi/ /cgi-a/*"
  -config+        Use this config file
  -Display+       Turn on/off display outputs:
                  1 Show redirects
                  2 Show cookies received
                  3 Show all 200/OK responses
                  4 Show pages which require authentication
                  D Debug output
                  E Display all HTTP errors
                  P Print progress to STDOUT
                  S Scrub output of IPs and hostnames
                  V Verbose output
  -dbcheck        Check database and other key files for syntax errors
  -evasion+       Encoding technique:
                  1 Random URI encoding (non-UTF8)
                  2 Directory self-reference (./.)
                  3 Premature URL ending
                  4 Prefix long random string
                  5 Random spacer
                  6 TAB as request spacer
                  7 Change the case of the URL
                  8 Use Windows directory separator (\)
                  A Use a carriage return (0xd) as a request spacer
                  B Use a zero value 0xb0 as a request spacer
  -followredirects Follow 3xx redirects to new location
  -Format+        Save file (-o) format:
                  csv  Comma-separated-value
                  json JSON Format
                  htm  HTML Format
                  nbs  Nmap NSE 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)
  -Help            This help information
  -http+          HTTP Only
  -id+            Host authentication to use, format is id:pass or id:pass:realm
  -ipv4           IPv4 Only
  -ipv6           IPv6 Only

Dec 9 00:19
```

```
varshini@kali:~$ wpscan --help

Options:
  -nolookup    Disables DNS lookups
  -nossl       Disables the use of SSL
  -noslash     Strip trailing slash from URL (e.g., '/admin/' to '/admin')
  -no404       Disables nikto attempting to guess a 404 page
  -dot          Over-ride dotfiles in nikto.conf, can be issued multiple times
  -output+      Write output to this file ('-' for auto-name)
  -Pause+       Pause between tests (seconds)
  -plugins+    List of plugins to run (default: ALL)
  -port+        Port to use (default 80)
  -rScert+     Certificate to use
  -Save         Save positive responses to this directory ('.' for auto-name)
  -ssl          Force ssl mode on port
  -Tuning+     Scan tuning:
              1 Interesting File / Seen in logs
              2 Misconfiguration / Default File
              3 Interesting Directories
              4 Injection (XSS/Script/HTML)
              5 Remote File Retrieval - Inside Web Root
              6 Denial of Service
              7 Remote File Retrieval - Server Wide
              8 Command Execution / Remote Shell
              9 Software Detection
              0 File Upload
              a Authentication Bypass
              b Software Identification
              c Remote Source Inclusion
              d WebServices
              e Admin/Interactive Console
              x Reverse Tuning Options (i.e., include all except specified)
  -timeout+    Timeout for requests (default 10 seconds)
  -Userdbs     Load only user databases, not the standard databases
              all  Disable standard dbs and load only user dbs
              test Disable only db_tests and load udb_tests
  -useragent   Overrides the default user agent
  -until       Run until the specified time or duration
  -url+        Target host/URL (alias of -host)
  -usecookies  Use cookie from responses in future requests
  -useproxy+   Use the proxy defined in nikto.conf, or argument http://server:port
  -version+    Prints nikto database versions
  -what+       Virtual host (for fast testing)
  -404code     Ignore these HTTP codes as negative responses (always). Format is "302,301".
  -404string   Ignore this string in response body content as negative response (always). Can be a regular expression.
              + requires a value

zsh: corrupt history file /home/varshini/.zsh_history
[1]+ 14788 Stopped                 wpscan --help
```

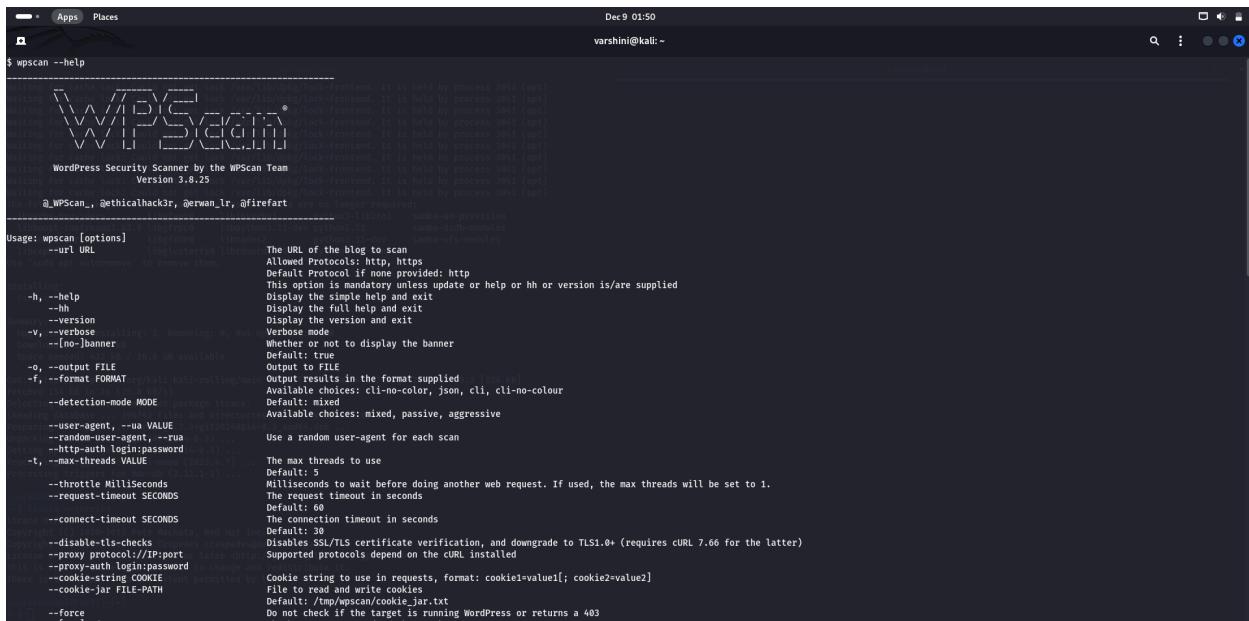
Web Application Analysis:

WPScan

- WPScan is a specialized WordPress vulnerability scanner designed to detect security issues within WordPress installations.
- It helps security professionals, penetration testers, and website administrators identify vulnerabilities in WordPress core files, plugins, themes, and configurations.
- WPScan is effective in detecting outdated versions, misconfigurations, and potential attack vectors on WordPress websites.

Primary Use Cases:

- Scanning WordPress websites for vulnerabilities.
- Identifying outdated plugins and themes with known security issues.
- Detecting weak user passwords and security misconfigurations.



```
varshini@kali:~$ wpScan --help
[...]
[WPScan] Version 3.8.25
[WPScan] Usage: wpScan [options] --url URL
[WPScan] Options:
[WPScan]   -h, --help           Help
[WPScan]   -hh, --hh            Help
[WPScan]   -v, --version        Version
[WPScan]   -vv, --verbose       Verbose mode
[WPScan]   -n, --no-banner      No banner
[WPScan]   -o, --output FILE    Output to FILE
[WPScan]   -f, --format FORMAT  Output results in the format supplied
[WPScan]   -detection-mode MODE Detection mode
[WPScan]   -user-agent, --ua VALUE User agent
[WPScan]   --random-user-agent, --rua Random user agent
[WPScan]   --http-auth login:password
[WPScan]   -t, --max-threads VALUE Maximum threads to use
[WPScan]   --throttle Milliseconds
[WPScan]   --request-timeout SECONDS Request timeout in seconds
[WPScan]   --connect-timeout SECONDS Connection timeout in seconds
[WPScan]   --disable-tls-checks
[WPScan]   --proxy protocol://IP:port
[WPScan]   --proxy-auth login:password
[WPScan]   --cookie-string COOKIE  Cookie string to use in requests, format: cookie1=value1; cookie2=value2
[WPScan]   --cookie-jar FILE PATH  File to read and write cookies
[WPScan]   --force
[WPScan]   [...]
```

Skipfish

- Skipfish is an open-source web application security scanner designed for fast and efficient vulnerability scanning.
- It is written in C and focuses on scanning websites for common web security flaws such as cross-site scripting (XSS), SQL injection, and other critical issues.
- Skipfish generates detailed reports and provides an interactive interface for analyzing the results.

Primary Use Cases:

- Scanning web applications for vulnerabilities.
- Identifying common web security flaws such as XSS, SQL injection, and information leakage.
- Quickly assessing the security of websites and web applications.



```

$ skipfish -h
skipfish web application scanner - version 2.10b
Usage: skipfish [ options ... ] -w wordlist -o output_dir start_url [ start_url2 ... ]

Authentication and access options:
-A user:pass      - use specified HTTP authentication credentials
-F host:IP        - pretend that 'host' resolves to 'IP'
-C name=val       - append a custom cookie to all requests
-H name=val       - append a custom HTTP header to all requests
-B [if|fp]         - use http/https consistent with MSIE / Firefox / iPhone
-N               - do not accept any new cookies
--auth-form url  - form authentication URL
--auth-user user  - form authentication user
--auth-pass pass  - form authentication password
--auth-verify-url - URL for in-session detection

Crawl scope options:
-d max_depth      - maximum crawl tree depth (10)
-c max_child       - maximum children to index per node (512)
-x max_desc        - maximum descendants to index per branch (8192)
-r r_limit         - maximum number of requests to send (100000000)
-p protocols       - protocols and link crawl probability (0.000000)
-q hex             - repeat probabilistic scan with given seed
-I string          - only follow URLs matching 'string'
-X string          - exclude URLs matching 'string'
-K string          - do not fuzz parameters named 'string'
-D domain          - crawl all subdomains under the specified domain
-B domain          - trust, but do not crawl, another domain
-Z                - do not descend into 5xx locations
-O                - do not submit any forms
-P                - do not parse HTML, etc, to find new links

Reporting options:
-o dir            - write output to specified directory (required)
-M               - log warnings about mixed content / non-SSL passwords
-E               - log all HTTP/1.0 / HTTP/1.1 caching intent mismatches
-U               - log all external URLs and e-mails seen
-Q               - completely suppress duplicate nodes in reports
-w               - be quiet, disable realtime progress stats
-v               - enable runtime logging (to stderr)

Dictionary management options:
-W wordlist        - use a specified read-write wordlist (required)
-S wordlist        - load a supplemental read-only wordlist

```

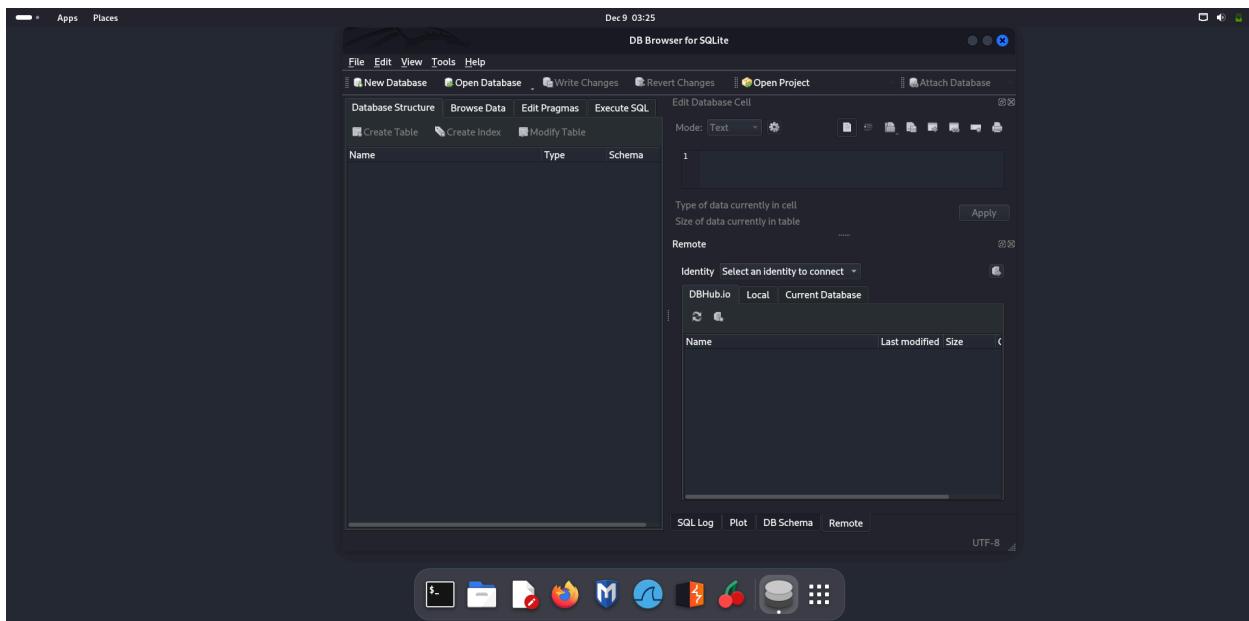
Database Assessment:

SQLite Database Browser

- SQLite Database Browser (also known as DB Browser for SQLite) is a free, open-source tool that provides an intuitive interface for managing SQLite databases.
- It allows users to create, design, and edit SQLite database files, perform queries, and browse through stored data.
- It is especially useful for developers, testers, and security professionals working with SQLite-based applications.

Primary Use Cases:

- Viewing, editing, and managing SQLite databases.
- Running SQL queries directly on the database.
- Analyzing and manipulating data within SQLite databases.
- Extracting information from SQLite databases during penetration testing and forensic investigations.



SQLmap

- SQLmap is an open-source penetration testing tool used for automating the process of detecting and exploiting SQL injection vulnerabilities in web applications.
- It is highly effective at identifying and exploiting SQL injection flaws and gaining access to underlying databases.
- SQLmap supports a wide range of database management systems (DBMS) such as MySQL, PostgreSQL, MSSQL, Oracle, and others.

Primary Use Cases:

- Detecting SQL injection vulnerabilities in web applications.
- Exploiting vulnerabilities to extract data from databases.
- Bypassing web application firewalls (WAFs) and other protective mechanisms.
- Performing database fingerprinting to identify the type and version of the underlying DBMS.
- Dumping data (such as user credentials and other sensitive information) from vulnerable databases.



```

$ Apps Places
Dec 9 03:39
varshini@kali: ~
$ sqlmap --wizard
{1.0.5#stable}
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 03:37:15 /2024-12-09/
[03:37:15] [INFO] starting wizard interface
Please enter full target URL (-u): http://google.com/page?id=1
POST or GET data? (Enter for None): none
[!] injection difficulty (--level/--risk). Please choose:
[1] Normal (default)
[2] Medium
[3] Hard
> 1
Enumeration (--banner/--current-user/etc). Please choose:
[1] Basic (default)
[2] Intermediate
[3] All
> 1

sqlmap is running, please wait..

[03:38:01] [CRITICAL] connection reset to the target URL. sqlmap is going to retry the request(s)
[03:38:01] [CRITICAL] connection reset to the target URL

[*] ending @ 03:38:01 /2024-12-09/
[~] varshini@kali: ~

```

Password Attacks:

John the Ripper

- John the Ripper is a fast, open-source password-cracking tool designed to detect weak passwords and recover lost passwords.
 - It supports various encryption algorithms and password hash types, making it a popular tool among penetration testers and security professionals.
 - John the Ripper can perform dictionary-based attacks, brute-force attacks, and rainbow table attacks.

Primary Use Cases:

- Cracking passwords to test the strength of user credentials.
 - Recovering lost passwords for system and file access.
 - Auditing password security in systems.
 - John the Ripper is highly effective for auditing password security and identifying weak passwords.

```
• Apps Places
Def 9 00:26
varshini@kali: ~
$ john
Created directory: /home/varshini/.john
Using John the Ripper 1.9-0-limbo-1bleeding-aec1328d6c 2021-11-02 10:45:52 +0100 OMP [linux-gnu 64-bit x86_64 SSE2 AC]
Copyright (c) 1996-2021 by Solar Designer and others
Homepage: https://www.openwall.com/john/
Usage: john [OPTIONS] [PASSWORD-FILES]

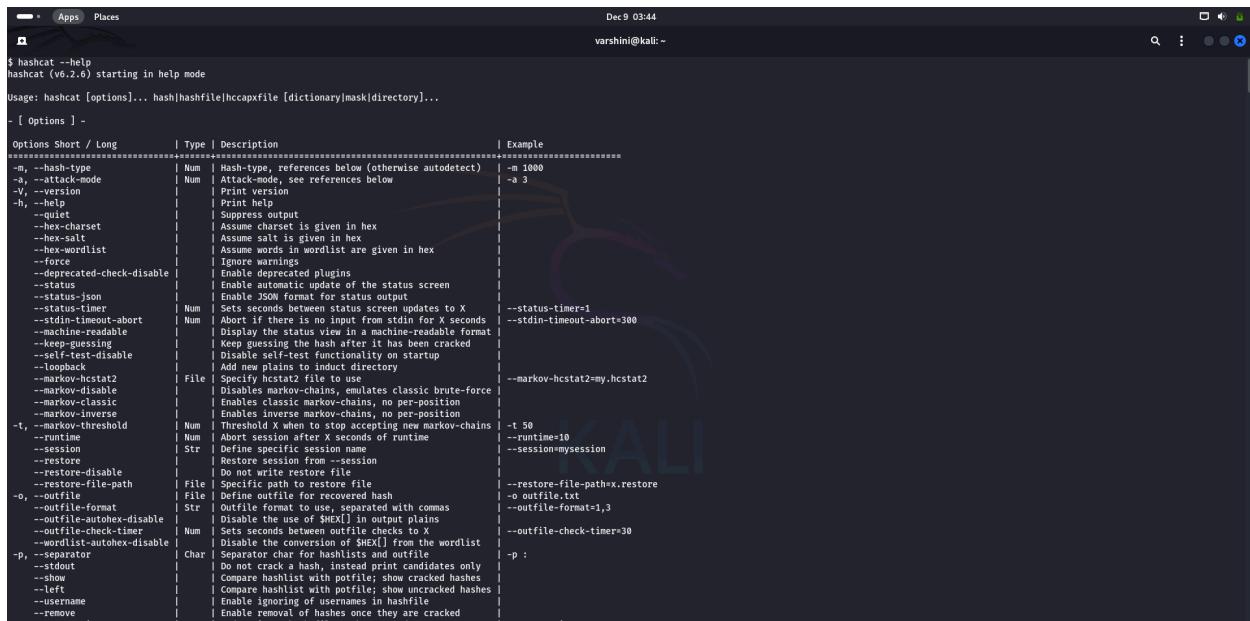
Use --help to list all available options.
[...]
[-] varshini@kali: ~]
[-]$ john -bsdi=fmts
descrypt, bsdcrypt, md5crypt, md5crypt-long, bcrypt, scrypt, LM, AFS,
tripcode, AndroidBackup, adcrypt, agilekeychain, aix-sha1, aix-ssha256,
aix-ssha512, andOTP, ansible,argon2, a5e00-des, a5e00-sha1, asa-md5,
asf-md5, asf-sha1, asf-sha256, asf-sha512, asuswrt, ats, ats-sha1,
bitshares, bitwarden, bks, Blackberry-ES1, wpaNSP, Blockchain, chap,
Clipperz, cloudkeychain, dynamic_n, cq, CRC32, cryptosafe, sha1crypt,
sha256crypt, sha512crypt, Citrix_N10, dahua, dashlane, diskryptor, Django,
django-scrypt, dmd5, dmg, dominoes, dominoes8c, DPAPImk, dragonfly32,
dragonfly364, dragonfly32, dragonfly464, Drupal7, cryptfs, eigrp,
F5, Fortinet, FortiAuthenticator, FortiAuthenticator2, FortiAuthenticator3,
Fortigate, FortiSSLP, FVDE, kali_gost, GPG, HAWAII-128+, HAWAII-256+, hdaa,
hMailServer, hsrp, IKE, iub2, itunes-backup, iwork, KeePass, keychain,
keyring, keystore, known_hosts, krb4, krb5, krb5aspass, krb5pa-sha1, krb5tg,
krb5t-17, krb5t-18, krb5t-3, kwllet, lp, lpc1l, leet, lotus, lotus85, LUKS,
MD2, md5c, Mediawiki, monero, money, MongoDB, scram, Mozilla, msasn1,
msas, msas-md5, msas-sha1, msas-sha256, msas-sha512, mssql12,
metasploit, myqlash, mysql, net-sha1, net-sha256, netlm, netlmv2,
net-md5, netntlm2, netntlm, netntlm-naive, net-sha1, nk, notes, netdns,
nsec3, NT, o10glogon, oologon, oslogon, ODF, Office, oldoffice,
OpenBSD-SofTRAI, openssl-enc, oracle, oracle11, Oracle12c, osc, ospf,
Padlock, Palshop, Panam, PBKDF2-HMAC-MD5, PBKDF2-HMAC-SHA1,
PBKDF2-HMAC-SHA1-160, PBKDF2-HMAC-SHA1-256, PGP, PEM, pfx, pgp, pgpda,
pgpkey, pkcs12, PMSH, pmsht, pkcs7, PKCS12, PST, PSTX, PSTX-V,
pwafse, qux, RACF, RACF-KOFAES, radius, RadMin, RAKP, rar, RAR5, Raw-SHA512,
Raw-BlaKE2, Raw-Kecccak-256, Raw-MD5, Raw-MD5s, Raw-SHA1,
Raw-SHA1-Acrypt, Raw-SHA1-linkedin, Raw-SHA224, Raw-SHA256,
Raw-SHA384, restic, ripemd-128, ripemd-160, rsvp, RVARY, Siemens-S7,
SipHash, SipHash2-32, SipHash3-64, SipHash3-96, SipHash3-128, SIP,
sk1eln-256, sk1eln-512, sky1, Slnfrit-128, Slnfrit-256, LaxPass, SNMP,
solardriven, SSH, ssp, STRIP, SunMD5, SybaseASE, Sybase-PROT, tacacs-plus,
tcp-md5, telegram, tezos, Tiger, tc_aes_xts, tc_ripemd160, tc_ripemd160boot,
tc_sha1, tc_whirlpool, vdi, OpenVM, vmx, VNC, vtp, wbds, whirlpool,
whirlpool, whirlpool, wpapsk, wpapsk-pmk, xmp-pscram, xsha, xsha512, zed,
ZIP, Zipmonster, plaintext, ripemd-160, HMAC-MD5, HMAC-SHA1, HMAC-SHA224,
HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, dummy, crypt
        formats (>9 dynamic formats shown as just 'dynamic_n' here)
```

Hashcat:

- Hashcat is a powerful, open-source password cracking tool designed to crack various hash types using a wide range of attack methods, such as brute-force, dictionary, and rule-based attacks.
- It is widely regarded as one of the fastest password recovery tools and supports GPU acceleration, making it significantly faster than CPU-based cracking tools.
- Hashcat is used by security professionals and penetration testers to audit password security.

Primary Use Cases:

- Cracking hashed passwords to recover lost passwords or test password strength.
- Auditing password databases to identify weak passwords.
- Performing dictionary, brute-force, and hybrid attacks on password hashes.
- Testing password security in various cryptographic hash algorithms.



The screenshot shows a terminal window on a Kali Linux desktop environment. The title bar reads "varshini@kali: ~". The terminal displays the Hashcat help menu, which includes usage instructions and a detailed table of command-line options with their descriptions and examples. The table has columns for Option, Short / Long, Type, Description, and Example.

Options Short / Long	Type	Description	Example
-h, --hash-type	Num	Hash-type, references below (otherwise autodetect)	-m 1000
-a, --attack-mode	Num	Attack-mode, see references below	-a 3
-V, --version		Print version	
-h, --help		Print help	
-q, --quiet		Suppress output	
-n, --hex-charset		Assume charset is given in hex	
-s, --salt		Assume salt is given in hex	
-w, --hex-wordlist		Assume words in wordlist are given in hex	
--force		Ignore warnings	
--deprecated-check-disable		Enable deprecated plugins	
--status		Enable automatic update of the status screen	
--status-json		Ends JSON format for status output	
--status-timer		Ends session status timer updates to X	--status-timer=1
--stdin-timeout-abort	Num	Abort if there is no input from stdin for X seconds	--stdin-timeout-abort=300
--machine-readable		Display the status view in a machine-readable format	
--keep-guessing		Keep guessing the hash after it has been cracked	
--self-test-disable		Disable self-test functionality on startup	
--loopback		Add new plain-text directory	
--markov-hcstat2	File	Specify a state file to	--markov-hcstat2=my.hcstat2
--markov-disable		Disables markov-chains, emulates classic brute-force	
--markov-classic		Enables classic markov-chains, no per-position	
--markov-inverse		Enables inverse markov-chains, no per-position	
-t, --markov-threshold	Num	Threshold X when to stop accepting new markov-chains	-t 50
--runtime	Num	Aborts session after X seconds of runtime	--runtime=10
--session	Str	Defines specific session name	--session=mysession
--restore		Restore session from --session	
--restore-disable		Do not write restore file	
--restore-file-path	File	Specific path to restore file	--restore-file-path=x.restore
-o, --outfile	File	Define outfile for recovered hash	-o outfile.txt
--outfile-format	Str	outfile format, use ; separated with commas	--outfile-format=1,3
--outfile-dashes-disable		Disable use of '-' as output plain	
--outfile-check-timer	Num	Sets seconds between outfile checks to X	--outfile-check-timer=30
--wordlist-autohex-disable		Disable the conversion of SHEX[] from the wordlist	
-P, --separator	Char	Separator char for hashlists and outfile	-p :
--stdout		Do not crack a hash, instead print candidates only	
--stats		Compare hashlist with potential; show cracked hashes	
--left		Compare hashlist with leftfile; show cracked hashes	
--username		Enable ignoring of usernames in hashfile	
--remove		Enable removal of hashes once they are cracked	

Wireless Attacks:

Wifite

- Wifite is an open-source. It is specifically designed for ease of use, automating the process of capturing handshakes and performing attacks on wireless networks.
- Wifite can conduct a variety of attacks, including dictionary-based and brute-force attacks on WPA/WPA2 networks, making it a valuable tool for penetration testers and security researchers.

Primary Use Cases:

- Cracking WEP and WPA/WPA2 encryption on wireless networks.
- Capturing handshakes to attempt offline password cracking.
- Automating attacks on weak or poorly configured wireless networks.
- Penetration testing and security assessments of wireless networks.



```

$ wifite --help
      . . . . . wifite2 2.7.0
      . . . . . a wireless auditor by devR82
      . . . . . maintained by kimocoder
      . . . . . https://github.com/kimocoder/wifite2

options:
-h, --help                      show this help message and exit

SETTINGS:
-v, --verbose                    Shows more options (-h -v). Prints commands and outputs. (default: quiet)
-i [interface]                  Wireless interface to use, e.g. wlanmon0 (default: ask)
-c [channel]                     Wireless channel to scan e.g. 1,3-6 (default: all 2GHz channels)
-inf, --infinite                 Enable infinite attack mode. Modify scanning time with -p (default: off)
-mac, --random-mac              Randomize wireless card MAC address (default: off)
-p [scan_time]                  Pillage: Attack all targets after scan time (seconds)
-kill                           Kill processes that conflict with Airmon/Airodump (default: off)
-pow [min_power], --power [min_power] Attacks any targets with at least min_power signal strength
--skip-crack                   Skip cracking captured handshakes/pmkid (default: off)
--first [attack_max], --first [attack_max] Attack first attack_max targets
--ignore-cracked               Ignores previously-cracked targets (default: off)
--clients-only                 Only show targets that have associated clients (default: off)
--nodeauths                     Passive mode: Never deauthenticated clients (default: deauth targets)
--daemon                        Puts device back in managed mode after quitting (default: off)

WEP:
--wep                          Show only WEP-encrypted networks
--require-fakeauth             Fails attacks if fake-auth fails (default: off)
--keep-ivs                      Retain .IVS files and reuse when cracking (default: off)

WPA:
--wpa                          Show only WPA-encrypted networks (includes WPS)
--new-hs                        Captures new handshakes, ignores existing handshakes in hs (default: off)
--dict [file]                    File containing passwords for cracking (default: /usr/share/dict/wordlist-probable.txt)

WPS:
--wps                          Show only WPS-enabled networks
--wps-only                      Only use WPS PIN & Pixie-Dust attacks (default: off)
--bully                         Use bully program for WPS PIN & Pixie-Dust attacks (default: reaver)
--reaver                        Use reaver program for WPS PIN & Pixie-Dust attacks (default: reaver)

```

Kismet:

- Kismet is an advanced, open-source wireless network detector, sniffer, and intrusion detection system (IDS).
- It is designed to work with a wide variety of wireless network interfaces and provides detailed information about Wi-Fi networks, including their signal strength, encryption type, and potential vulnerabilities.

Primary Use Cases:

- Detecting and monitoring Wi-Fi networks in the vicinity.
- Detecting hidden networks (i.e., networks that do not broadcast their SSID).
- Intrusion detection in wireless environments by identifying rogue access points and unauthorized clients.



```
$ 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-curses-wrapper  Disable server curses wrapper
--no-curses        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 line wrap 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-type <type>  Override activated log types
-L, --log-title <title>  Override default log title
-D, --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
(varshini@kali)-~]
```

Reverse Engineering:

Clang

- Clang is part of the LLVM (Low Level Virtual Machine) project and provides an alternative to the GCC (GNU Compiler Collection).
- Clang is designed to be highly efficient, fast, and feature-rich, providing better diagnostics (error and warning messages), and producing optimized, high-performance executables.

Primary Use Cases:

- Compiling C, C++, and Objective-C code for various platforms.
- Static analysis and code diagnostics to identify bugs and security vulnerabilities.
- Debugging and profiling using integration with tools like gdb or lldb.

```

$ clang --help
OVERVIEW: clang LLVM compiler
USAGE: clang [options] file...

OPTIONS:
-###[[          Print (but do not run) the commands to run for this compilation
--andgpu-arch-tool=value[[ Tool used for detecting AMD GPU arch in the system.
--analyzer-output <value>[[ Static analyzer report output format (html|plist|plist-multi-file|plist-html|sarif|sarif-html|text).
--analyze[[ Run the static analyzer
--arcmt-migrate-emit-errors[[ Emit ARC errors even if the migrator can fix them
--arcmt-migrate-report-output <value>[[ Output path for the plist report
-B $prefix[[ Search $prefix for symbols, libraries, and data files. If $prefix is a directory, search $prefix/$file
-b <arg>[[ Pass <arg> to the linker on AIX
-cc[[ Include comments from within macros in preprocessed output
-cl-denorms-are-zero[[ OpenCL only. Allow denormals to be flushed to zero.
-cl-ext-value[[ OpenCL only. Enable or disable OpenCL extensions/optional features. The argument is a comma-separated sequence of one or more extension names, each prefixed by '+' or '-'.
-cl-fast-relaxed-math[[ OpenCL only. Sets -cl-finite-math-only and -cl-unsafe-math-optimizations, and defines __FAST_RELAXED_MATH__.
-cl-finite-math-only[[ OpenCL only. Allow floating-point optimizations that assume arguments and results are not NaNs or +Inf.
-cl-ip02-correctly-rounded-divide-sqrt[[ OpenCL only. Specify that single precision floating-point divide and sqrt used in the program source are correctly rounded.
-cl-kernel-arg-info[[ OpenCL only. Generate kernel argument metadata.
-cl-mad-enable[[ OpenCL only. Allow use of less precise MAD computations in the generated binary.
-cl-no-signed-zeros[[ OpenCL only. Allow use of less precise no signed zeros computations in the generated binary.
-cl-no-hc[[ OpenCL only. Disables all standard includes containing non-native compiler types and functions.
-cl-opt-disable[[ OpenCL only. This option disables all optimizations. By default optimizations are enabled.
-cl-single-precision-constant[[ OpenCL only. Treat double precision floating-point constant as single precision constant.
-cl-std=value[[ OpenCL language standard to compile for.
-cl-strict-aliasing[[ OpenCL only. This option is added for compatibility with OpenCL 1.0.
-cl-uniform-work-group-size[[ OpenCL only. Defines that the global work-size be a multiple of the work-group size specified to clEnqueueNDRangeKernel
--config<file>[[ Specify configuration file
--cuda-compile-host-device[[ Compile CUDA code for both host and device (default). Has no effect on non-CUDA compilations.
--cuda-device-only[[ Compile CUDA code for device only
--cuda-feature=<value>[[ Manually specify the CUDA feature to use
--cuda-host-only[[ Compile CUDA code for host only. Has no effect on non-CUDA compilations.
--cuda-include-ptx=<value>[[ Include PTX for the following GPU architecture (e.g. sm_35) or 'all'. May be specified more than once.
--cuda-noopt-device-debug[[ Enable device-side debug info generation. Disables ptxas optimizations.

```

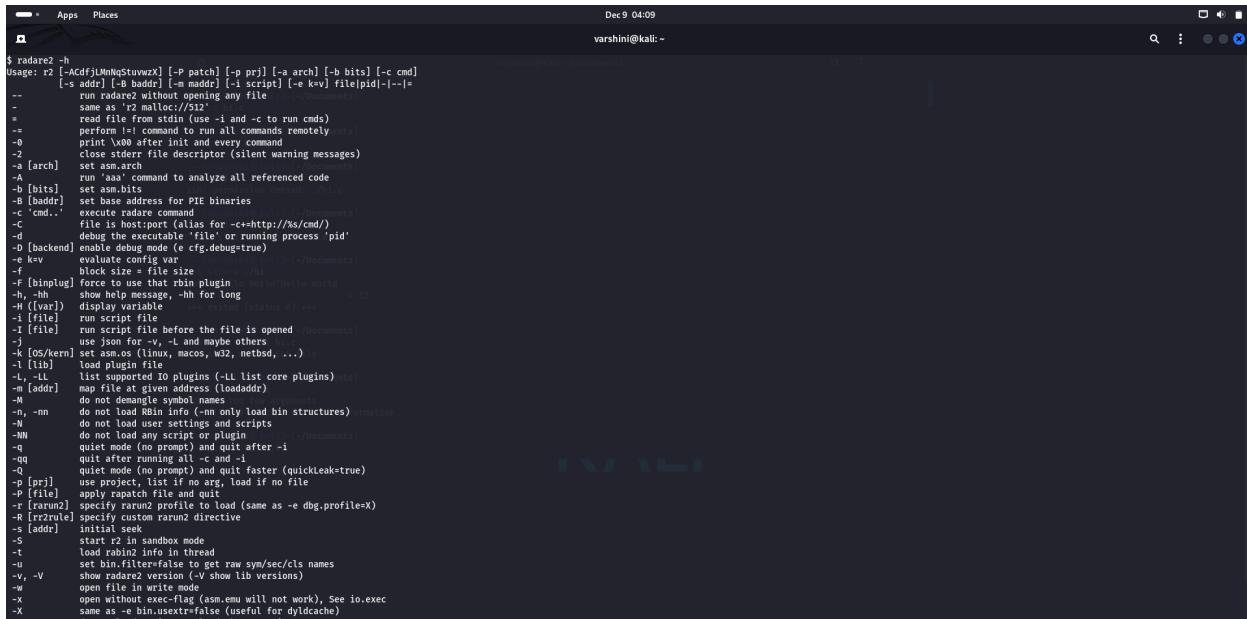
Radare2

- Radare2 is an open-source, reverse engineering framework that provides a comprehensive set of tools for analyzing and disassembling binaries, debugging, and performing low-level analysis on executable files.
- Radare2 is highly flexible and can be used for a wide range of tasks, from simple file inspection to complex vulnerability research.

Primary Use Cases:

- Disassembly and decompilation of machine code into human-readable formats.
- Malware analysis to understand the functionality and behavior of potentially malicious software.
- Binary patching to modify executable files for testing and debugging.

- Security auditing and vulnerability research to identify weaknesses in compiled code.



```

$ radare2 -h
Usage: r2 [-ACdfi] [file|stw|wrx] [-p patch] [-p prj] [-a arch] [-b bits] [-c cmd]
[-s size] [-S baddr] [-e enddr] [-i script] [-e kev] file|pid|--|=|
-- run radare2 without opening any file
- same as 'r2 malloc://$size'
= read file from stdin (use -i and -c to run cmds)
-= perform |=! command to run all commands remotely
-0 print \x00 after init and every command
-> close standard file descriptor (silent warning messages)
-a [arch] set asm.arch
-A run 'aaa' command to analyze all referenced code
-b [bits] set asm.bits
-B [baddr] set base address for PIE binaries
-c 'cmd...' execute radare command
-d file or host:alias for <-c>http://xs/cmd/
-e debug the executable 'file' or running process 'pid'
-o [backend] enable debug mode (e cfg.debug=true)
-e kev evaluate config var
-f block size = file size
-f [binplug] force to use that bin plugin
-h help message, -hn for long
-i [(var)] display variable
-i [file] run script file
-i [file] run script file before the file is opened
-j use json for -v, -l and maybe others
-k [OS kern] set asm.os (linux, macos, w32, netbsd, ...)
-l [lib] list plugin lib
-L list supported IO plugins (-L list core plugins)
-m [addr] map file at given address (loadaddr)
-M do not demangle symbol names
-n, -nn do not load RBin info (-nn only load bin structures)
-N do not load user settings and scripts
-NH do not load user script or plugin (-NH -N)
-q quiet mode (no progress bar) quit after -i
-qq quit after running all -c and -i
-Q quiet mode (no prompt) and quit faster (quickleak=true)
-p [prj] use project, list if no arg, load if no file
-p [file] apply patch file and quit
-r [cmd] specific radare2 profile to load (same as -e dbg.profile=X)
-s [r2rule] specify system r2rule directive
-s [addrs] initial seek
-S start r2 in sandbox mode
-t load rabin2 info in thread
-u set bin.filter=false to get raw sym/sec/cls names
-U show user settings (-U 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.usextr=false (useful for dyldcache)

```

Exploitation Tools:

Metasploit Framework

- Metasploit is an open-source penetration testing framework.
- It is widely used by ethical hackers and security professionals to identify, exploit, and validate vulnerabilities in networks, systems, and applications.

Primary Use Cases:

- Identifying vulnerabilities in systems and networks.
- Developing and testing exploits.
- Conducting penetration tests.
- Metasploit supports both command-line and GUI-based interfaces (such as Armitage).

```

$ sudo msfdb init & msfconsole
[sudo] password for varshini:
[*] Starting database
[*] Creating database user 'msf'
[*] Creating databases
[*] Creating database 'msf_test'
[*] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[*] Creating initial database schema
Metasploit tip: Metasploit can be configured at startup, see msfconsole
--help to Learn more

IIIIII dTb,dTb
II   4'  V  B
II   6.  P
II   "I. ;P'
II   "I; P'
II   YWP

I love shells --egypt

[*] msf6 > show exploits
      current Disclosure Date Rank Check Description
-----+-----+-----+-----+-----+
      1 ibstat 2013-09-24 excellent Yes ibstat SPATH Privilege Escalation
      2 insvcout 2023-04-24 excellent Yes insvcout RPM Privilege Escalation
      3 Xorg_X11_Socket 2018-05-10 great Yes Xorg X11 Server Remote Privilege Escalation
      4 AIX_rpsvc 2009-06-17 great No AIX Remote Procedure Call (RPC) Daemon (rpsvc) Opcode 21 Buffer Overflow
      5 ToolTalk_rpc_ttbservd_It_Internal_relpPath 2016-01-01 excellent Yes ToolTalk rpc_ttbservd_It_Internal_relpPath Buffer Overflow (AIX)
      6 Android_ADB_Server_Remote_Payload_Execution 2014-11-12 excellent Yes Android ADB Server Remote Payload Execution
      7 Samsung_Galaxy_KNOX_Android_Browser_RCE 2015-08-13 normal No Samsung Galaxy KNOX Android Browser RCE
      8 Android_Browser_and_WebView_addJavaScriptInterface 2012-12-21 excellent No Android Browser and WebView addJavaScriptInterface Code Execution
      9 Android_Binder_Use_After_Free_Exploit 2013-05-15 good No Android Binder Use After Free Exploit
      10 Android_Binder_Use_After_Free_Exploit 2019-09-26 excellent No Android Binder Use After Free Exploit
      11 Android_Janus_APK_Signature_Bypass 2014-05-03 excellent Yes Android 'Janus' APK Signature bypass
      12 Android_TowerRoot_Futex_Queue_Kernel_Exploit 2017-07-31 manual Yes Android TowerRoot Futex Queue Kernel Exploit
      13 Android_get_user_put_user_Exploit 2013-09-06 excellent No Android get_user/put_user Exploit
      14 Android_Selinux_Privilege_Escalation 2017-08-31 manual No Android SELinux Privilege Escalation
      15 Safari_WebKit_LibTiff 2006-08-25 good No Safari WebKit LibTIFF Exploit for iOS 7.1.2
      16 Apple_iOS_MobileSafari_LibTiff 2006-08-01 good No Apple iOS MobileSafari LibTIFF Buffer Overflow
      17 Safari_WebKit_Proxy_Object_Type_Confusion 2018-03-15 manual No Safari Webkit Proxy Object Type Confusion
      18 WebKit_no_number_defineProperties_UAF 2016-08-25 manual No WebKit no_number defineProperties UAF
      19 Apple_iOS_email_mobilemail_libtiff 2006-08-01 good No Apple iOS MobileMail LibTIFF Buffer Overflow
      20 Apple_iOS_ssh_cyberduck_ssh 2007-04-19 excellent No Apple OS DefenseSSH Password Vulnerability
      21 Mozilla_Netscape_Finished_BDF 2004-11-02 normal No Mozilla Netscape Finished BDF Buffer Overflow
      22 Mercante_SoftCart_CGI_Overflow 2004-08-19 great No Mercante SoftCart CGI Overflow
      23 System_V_Derived/bin/login_Extraneous_Arguments_Buffer_Overflow 2001-12-12 good No System V Derived /bin/login Extraneous Arguments Buffer Overflow
      24 Firefox_Exec_Shellcode_from_Privileged_Javascript_Shell 2014-03-10 excellent No Firefox Exec Shellcode from Privileged Javascript Shell
      25 ProFTPD_1.3.2rc1_IAC_Buffer_Overflow(FreeBSD) 2010-11-01 great Yes ProFTPD 1.3.2rc1 IAC Buffer Overflow (FreeBSD)
      26 Citrix_ADC_NetScaler_Forum_Logout_RCE 2010-07-17 excellent Yes Citrix ADC (NetScaler) Forum Logout RCE
      27 Junos_PMPRC_Environment_Variable_Manipulation_RCE 2023-07-16 normal No Junos OS PMPRC Environment Variable Manipulation RCE
      28 Watchguard_XCS_Remote_Command_Execution 2023-08-17 excellent Yes Watchguard XCS Remote Command Execution
      29 Watchguard_XCS_Remote_Command_Execution 2015-06-29 excellent Yes Watchguard XCS Remote Command Execution
      30 FreeBSD_Intel_SYSENTER_Privilege_Escalation 2012-06-12 great Yes FreeBSD Intel SYSENTER Privilege Escalation
      31 FreeBSD_ip6_Setsockopt_Use_After_Free_Privilege_Escalation 2020-07-07 great Yes FreeBSD ip6_setsockopt Use-After-Free Privilege Escalation
      32 FreeBSD_RIDL_Stack_Memory_Manipulation_Privilege_Escalation 2013-04-18 great Yes FreeBSD RIDL Stack Memory Manipulation Privilege Escalation
      33 FreeBSD_RIDL_Stack_Memory_Manipulation_Privilege_Escalation 2013-11-19 excellent Yes FreeBSD RIDL Stack Memory Manipulation Privilege Escalation
      34 Watchguard_XCS_FixCorruptMail_Local_Privilege_Escalation 2015-06-29 manual Yes Watchguard XCS FixCorruptMail Local Privilege Escalation
      35 Citrix_Netscaler_SOAP_Handler_Remote_Code_Execution 2014-09-22 normal Yes Citrix Netscaler SOAP Handler Remote Code Execution
      36 Samba_transOpen_Overflow_(#BSD_x86) 2003-04-07 great No Samba transOpen Overflow (#BSD x86)
      37 XTACACS_Report()_Buffer_Overflow 2008-01-08 average No XTACACS Report() Buffer Overflow

```

CrackMapExec (CME)

- CrackMapExec (CME) is an open-source, post-exploitation tool primarily used for network penetration testing and Windows environment assessments.
- It automates the assessment of large Active Directory (AD) networks by providing a simple and effective way to perform tasks like SMB enumeration, credential validation, command execution, and more.

Primary Use Cases:

- Network enumeration (SMB, RDP, and other services).
- Credential validation across multiple machines in a Windows domain.
- Dumping password hashes and performing Pass-the-Hash (PTH) attacks.
- Reconnaissance and exploitation in Active Directory (AD) environments.

```
$ crackmapexec -h
/usr/lib/python3/dist-packages/cme/cli.py:33: SyntaxWarning: invalid escape sequence '\'
...
/usr/lib/python3/dist-packages/cme/protocols/winnm.py:324: SyntaxWarning: invalid escape sequence '\S'
self._conn.execute_cmd('reg save HKLM\%SAM C:\windows\temp\%SAM 86 reg save HKLM\SYSTEM C:\windows\temp\%SYSTEM')
/usr/lib/python3/dist-packages/cme/protocols/winrm.py:104: SyntaxWarning: invalid escape sequence '\x'
self._conn.execute_cmd('reg save HKLM\SECURITY C:\windows\temp\%SECURITY 66 reg save HKLM\SYSTEM C:\windows\temp\%SYSTEM')
/usr/lib/python3/dist-packages/cme/protocols/smb/smbexec.py:67: SyntaxWarning: invalid escape sequence '\p'
stringbinding = 'ncacn_np:[\\pipe\\svcctl]' % self._host
/usr/lib/python3/dist-packages/cme/protocols/smb/smbexec.py:93: SyntaxWarning: invalid escape sequence '\'
    command = self._shell + 'echo * > %TEMP%\{} & %COMSPEC% /Q /C %TEMP%\{} & %COMSPEC% /Q /C del %TEMP%\{}'.format(self._share_name, self._output, self._batchFile, self._batchFile)
usage: crackmapexec [-h] [-t THREADS] [-timeout TIMEOUT] [--jitter INTERVAL]
                     [--darrell] [--verbose]
                     {ldap,ssh,winnm,mssql,ftp,rdp,smb} ...
A Swiss Army Knife for pentesting networks
Forged by Gh0st3d3r and 3m0rgn_X4 using the power of dank memes
Exclusive release for Porchetta Industries users
https://porchetta.industries/
Version : 5.4.0
Command: Indestructible GoThM0g

options:
-h, --help            show this help message and exit
-t THREADS           set how many concurrent threads to use (default: 100)
--timeout TIMEOUT   max time in seconds of each thread (default: None)
--jitter INTERVAL    sets a random jitter between each connection (default: None)
--darrell            give Darrell a hand
--verbose            enable verbose output

protocols:
available protocols
{ldap,ssh,winnm,mssql,ftp,rdp,smb}
ldap          own stuff using LDAP
ssh           own stuff using SSH
winnm         own stuff using WINRM
mssql         own stuff using MSSQL
```

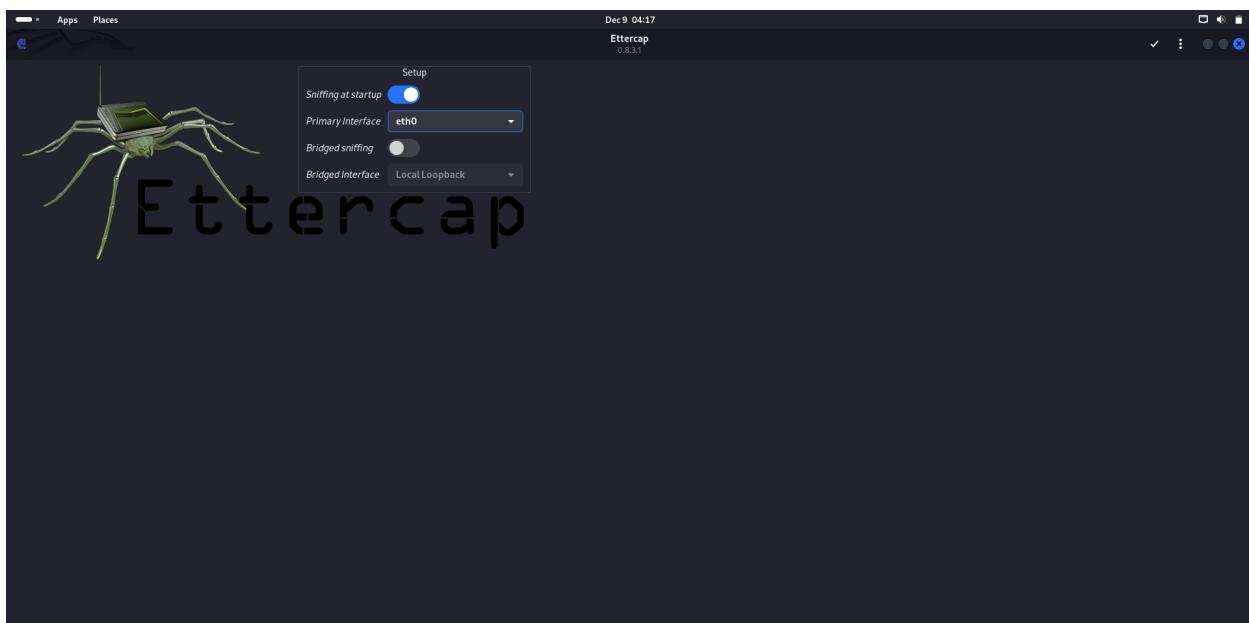
Sniffing & Spoofing:

Ethercap-Graphical

- Ethercap-Graphical is the graphical user interface (GUI) version of Ettercap, a well-known network sniffing and man-in-the-middle (MITM) attack tool.
- Ettercap is widely used for packet sniffing, network traffic analysis, and performing MITM attacks on local area networks (LANs).

Primary Use Cases:

- Man-in-the-Middle (MITM) attacks, including ARP poisoning and DNS spoofing.
- Packet sniffing and analysis on local networks to monitor traffic.
- Intercepting and modifying network traffic in real-time (e.g., injecting custom responses or altering data in transit).
- Session hijacking to take control of active user sessions.

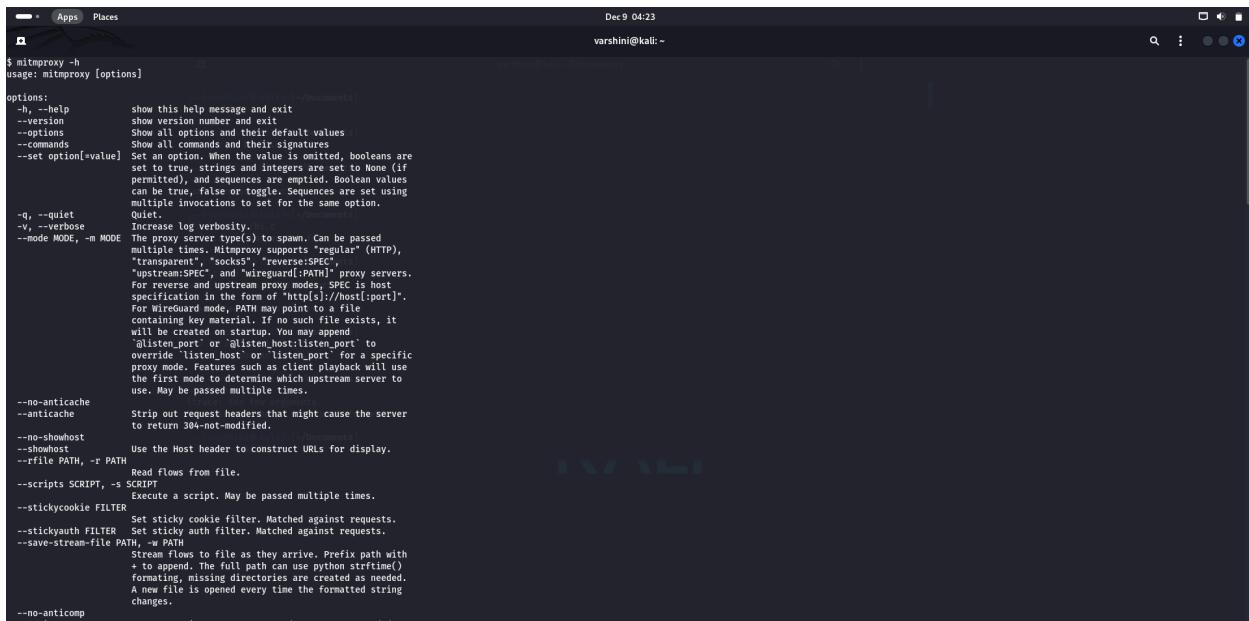


mitmproxy

- mitmproxy is an open-source, interactive proxy tool used for intercepting, inspecting, modifying, and debugging HTTP and HTTPS traffic in real-time.
- It supports both HTTP/1.x and HTTP/2 protocols, and provides a console-based interface (mitmproxy), a web-based interface (mitmweb), and an API for automated scripting.

Primary Use Cases:

- Intercepting and modifying HTTP/HTTPS traffic between clients and servers.
- Capturing and analyzing web traffic for sensitive data like login credentials, tokens, etc.
- Testing web applications for vulnerabilities such as cross-site scripting (XSS), SQL injection, and other web security issues.
- Debugging and analyzing web services by inspecting headers, cookies, and response data.



```
$ mitmproxy -h
usage: mitmproxy [options]

options:
-h, --help      show this help message and exit
--version       show version number and exit
--options       Show all options and their default values
--commands     Show all commands and their signatures
--set option[=value] Set an option. If no value is provided, booleans are
                  set to true, strings and integers are set to None (if
                  permitted), and sequences are emptied. Boolean values
                  can be true, false or toggle. Sequences are set using
                  multiple invocations to set for the same option.
-q, --quiet     Quiet.
-v, --verbose    Increase log verbosity.
--mode MODE, -m MODE  The proxy server type(s) to spawn. Can be passed
                  multiple times. Mitmproxy supports "regular" (HTTP,
                  "transparent", "socks5", "reverse:SPEC",
                  "upstream:SPEC", and "wireguard:[PATH]" proxy servers.
                  For reverse and upstream proxy modes, SPEC is host
                  specification in the form of "http://[host]:[port]".
                  For wireguard mode, PATH is point to a file
                  containing key material. If no such file exists, it
                  will be created at startup. You may append
                  '@listen_port' or '@listen_host:listen_port' to
                  override 'listen_host' and 'listen_port' for a specific
                  proxy mode. Feeding such an client playback will use
                  the first mode to determine which upstream server to
                  use. May be passed multiple times.
--no-anticache
--anticache      Strip out request headers that might cause the server
                  to return 304-not-modified.
--no-showhost
--showhost       Use the Host header to construct URLs for display.
--rfile PATH, -r PATH
                  Read flows from file.
--scripts SCRIPT, -s SCRIPT
                  Execute a script. May be passed multiple times.
--stickycookie FILTER
                  Set sticky cookie filter. Matched against requests.
--stickyauth FILTER
                  Set sticky auth filter. Matched against requests.
--save-stream-file PATH, -w PATH
                  Stream flows to file as they arrive. Prefix path with
                  '+' to append. The full path can use python strftime()
                  tokens. Missing directories are created as needed.
                  A new file is opened every time the formatted string
                  changes.
--no-anticomp
```

Post Exploitation:

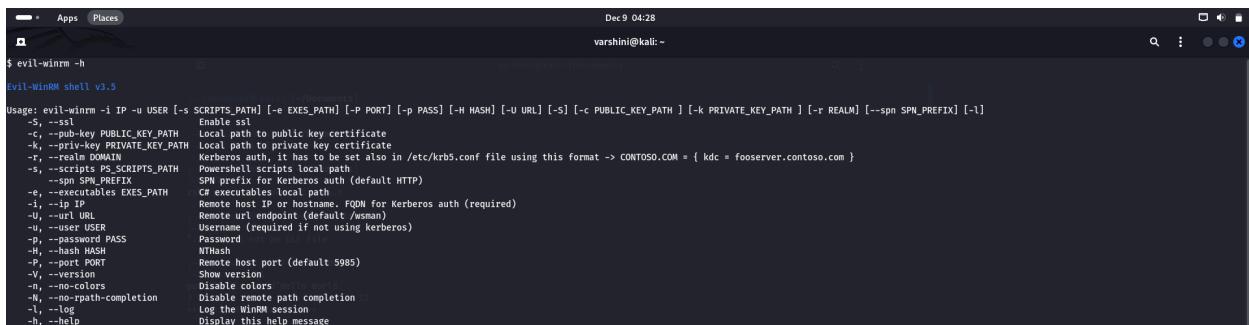
Evil-WinRM

- Evil-WinRM is a powerful open-source tool used for Windows Remote Management (WinRM) exploitation and post-exploitation in penetration testing and red team operations.

- Evil-WinRM is often employed in lateral movement across networks and post-exploitation activities, as it enables attackers to gain command-line access to Windows machines, bypassing traditional remote access tools like RDP.

Primary Use Cases:

- Remote command execution on Windows systems over WinRM, particularly in Active Directory (AD) environments.
- Remote administration and troubleshooting for legitimate penetration testing scenarios (with permission).
- Pivoting and lateral movement in a network to gain access to other machines and sensitive data.



```
$ evil-winrm -h
Evil-WinRM shell v3.5
Usage: evil-winrm [-i IP] [-u USER [-s SCRIPTS_PATH] [-e EXES_PATH] [-P PORT] [-p PASS] [-H HASH] [-U URL] [-S] [-c PUBLIC_KEY_PATH] [-k PRIVATE_KEY_PATH] [-r REALM] [-s SPN_SUFFIX] [-l]
      -S, --ssl           Enable ssl
      -c, --pub-key PUBLIC_KEY_PATH Local path to public key certificate
      -k, --priv-key PRIVATE_KEY_PATH Local path to private key certificate
      -r, --realm DOMAIN Kerberos auth, it has to be set also in /etc/krb5.conf file using this format -> CONTOSO.COM = { kdc = fooserver.contoso.com }
      -s, --scripts PS_SCRIPTS_PATH Powershell scripts local path
      -s, --spn SPN_PREFIX SPN prefix for Kerberos auth (default HTTP)
      -e, --executables EXES_PATH CF executables local path
      -i, --ip IP          Remote host IP or FQDN for Kerberos auth (required)
      -U, --url URL        Remote host endpoint (default /wsman)
      -u, --user USER      Username (required if not using kerberos)
      -p, --password PASS  Password
      -H, --hash HASH      NTHash
      -P, --port PORT     Remote host port (default 5985)
      -y, --yes            Show progress
      -n, --no-colors      Disable colors
      -w, --no-path-completion Disable remote path completion
      -l, --log             Log the WinRM session
      -h, --help            Display this help message
```




```
(varshini㉿kali)-~$ evil-winrm -i 192.168.1.10 -u username -p password
Evil-WinRM shell v3.5
Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine
Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion
Info: Establishing connection to remote endpoint
```

Netcat (nc)

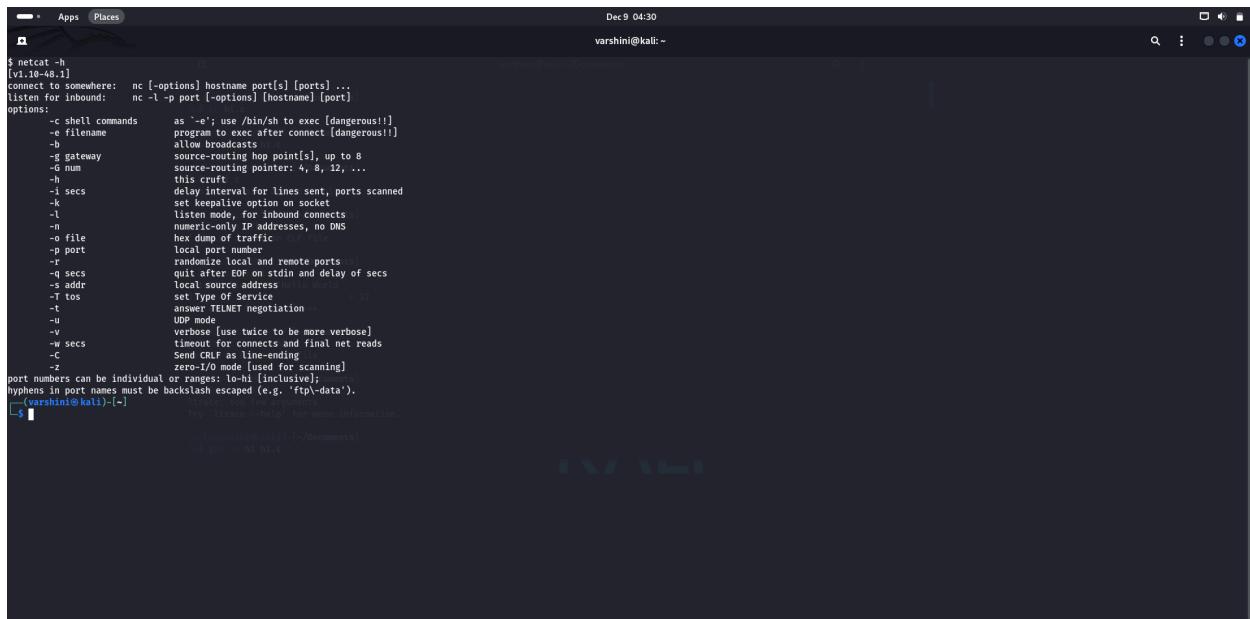
- Netcat (nc) is a versatile and widely used networking utility that can create both TCP and UDP connections between two machines.
- Often referred to as the "Swiss army knife" of networking tools, Netcat can be used for a wide range of tasks, such as

port scanning, banner grabbing, file transfers, remote access, and network debugging.

- Netcat is commonly used in penetration testing and red teaming activities due to its ability to create backdoors and facilitate data exfiltration.

Primary Use Cases:

- Banner grabbing to identify services and versions running on remote systems.
- Port scanning to identify open ports on target systems.
- Data exfiltration by sending files over the network.
- Simple file transfer between systems.



The screenshot shows a terminal window on a Kali Linux system. The command entered is \$ netcat -h. The output provides detailed documentation for the netcat command, listing various options and their descriptions. Key options include -e for executing a shell, -l for listening mode, -p for specifying a port, and -w for setting a timeout. The terminal also shows the user's path (varshini@kali: ~) and the current date and time (Dec 9 04:30).

```
$ netcat -h
[vi:10-48.1]
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound: nc -l -p port [-options] [hostname] [port]
options:
  -c shell commands      as "-e"; use /bin/sh to exec [dangerous!!]
  -e filename            program to exec after connect [dangerous!!]
  -b                   allow broadcasts
  -g gateway             maximum number of hop points], up to 8
  -o num                source-routing pointer: 4, 8, 32, ...
  -h                   this script
  -i secs               delay interval for lines sent, ports scanned
  -k                   set keepalive option on socket
  -l                   listen mode, for inbound connects
  -n                   numeric addresses, no DNS
  -o file               hex dump of traffic
  -p port               local port number
  -r                   randomize local and remote ports
  -q secs               quit after EOF on stdin and delay of secs
  -s addr              local source address
  -T tos               strict TOS
  -t                   answer TELNET negotiation
  -u                   UDP mode
  -v                   verbose [use twice to be more verbose]
  -w secs              timeout for connects and final net reads
  -C                  send CRLF as line-ending
  -z                  zero-byte detection [scanning]
port numbers can be individual or ranges, lo-hi [inclusive];
hyphens in port names must be backslash escaped (e.g. 'ftp\`data').
[~] try 'trace --help' for more information.
[~] varshini@kali: ~
```

Forensics:

Autopsy

- Autopsy is an open-source, digital forensics platform used for investigating and analyzing data from computers, mobile devices, and storage media.
- Autopsy is commonly used by forensic professionals, law enforcement, and cybersecurity experts to recover, analyze, and report on data from various devices.
- It supports file analysis, metadata extraction, timeline analysis, and much more.

Primary Use Cases:

- Digital forensics investigations, particularly in law enforcement and incident response scenarios.
- File recovery and examination, including deleted files and hidden data.
- Evidence tracking for ensuring that all digital evidence is preserved and properly documented.
- Malware analysis by extracting and analyzing data from infected systems or drives.

```

[sudo] password for varshini:
=====
Autopsy Forensic Browser
http://www.sleuthkit.org/Autopsy/
ver 2.24
=====
Evidence Locker: /var/lib/autopsy
Start Time: Mon Dec 9 04:33:05 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
=====
[1] 1186 pts/0 0:00 /bin/sh * 12
*** deleted (status 0) ***
=====
[1] 1186 pts/0 0:00 /bin/sh * 12
*** trace in blc ***
'thc.c' is not an ELF file
=====
[1] 1186 pts/0 0:00 /bin/sh * 12
Trace: too few arguments
try 'ltrace --help' for more information.
=====
[1] 1186 pts/0 0:00 /bin/sh * 12
getpid() H.H.C

```

Hashdeep

- Hashdeep is an open-source command-line tool for computing and verifying hashes of files, and for deep hashing across directories.
- It supports a variety of hash algorithms, including MD5, SHA-1, SHA-256, and more.
- Hashdeep is primarily used in file integrity checking, data verification, and forensic investigations to ensure the authenticity and integrity of files, especially when comparing hashes between different systems or performing audits on large collections of files.

Primary Use Cases:

- File integrity checking: Ensuring that files have not been altered, especially in digital forensics and cybersecurity investigations.
- Data verification: Comparing file hashes across different systems or environments to verify consistency.
- Forensic investigations: Checking the integrity of evidence by comparing hashes to known values or a hash database.
- Bulk hashing: Calculating hashes for many files in a directory or filesystem for auditing or forensic analysis.

```
$ hashdeep -h
hashdeep version 4.4 by Jesse Kornblum and Simson Garfinkel.
$ hashdeep [OPTION]... [FILE]...
-c [alg1,[alg2]] - Compute hashes only. Defaults are MD5 and SHA-256
-p <size> - piecwise mode. Files are broken into blocks for hashing
-r - recursive mode. All subdirectories are traversed
-d - output in DFXML (Digital Forensics XML)
-k [file] - add a file of known hashes
-a - audit mode. Validates FILES against known hashes. Requires -k
-m - matching mode. Requires -k
-x - recursive matching mode. Requires -k
-w - in -m mode, displays which known file was matched
-M and -N act like -m and -x, but display hashes of matching files
-e - compute estimated time remaining for each file
-s - silent mode. Suppress all error messages
-b - prints paths to the bare minimum files; all path information is omitted
-l - prints relative paths for filenames
-i -I - only process files smaller than the given threshold
-o - only process certain types of files. See README/manpage
-v - verbose mode. Use again to be more verbose
-d - output in DFXML; -W FILE - write to FILE
-j [num] - use num threads (default 3)
(Varshini@kali)-[~]
$
```

The terminal window shows the usage of the hashdeep command. It lists various options such as -h (help), -c (compute hashes), -p (piecwise mode), -r (recursive mode), -d (DFXML output), -k (add known hashes), -a (audit mode), -m (matching mode), -x (recursive matching mode), -w (display matched files), -M and -N (display matching files), -e (compute estimated time), -s (silent mode), -b (print bare minimum files), -l (print relative paths), -i (process files smaller than a threshold), -o (process certain types of files), -v (verbose mode), -d (DFXML output), and -j (use num threads). Examples of file paths are shown at the bottom.

Reporting Tools:

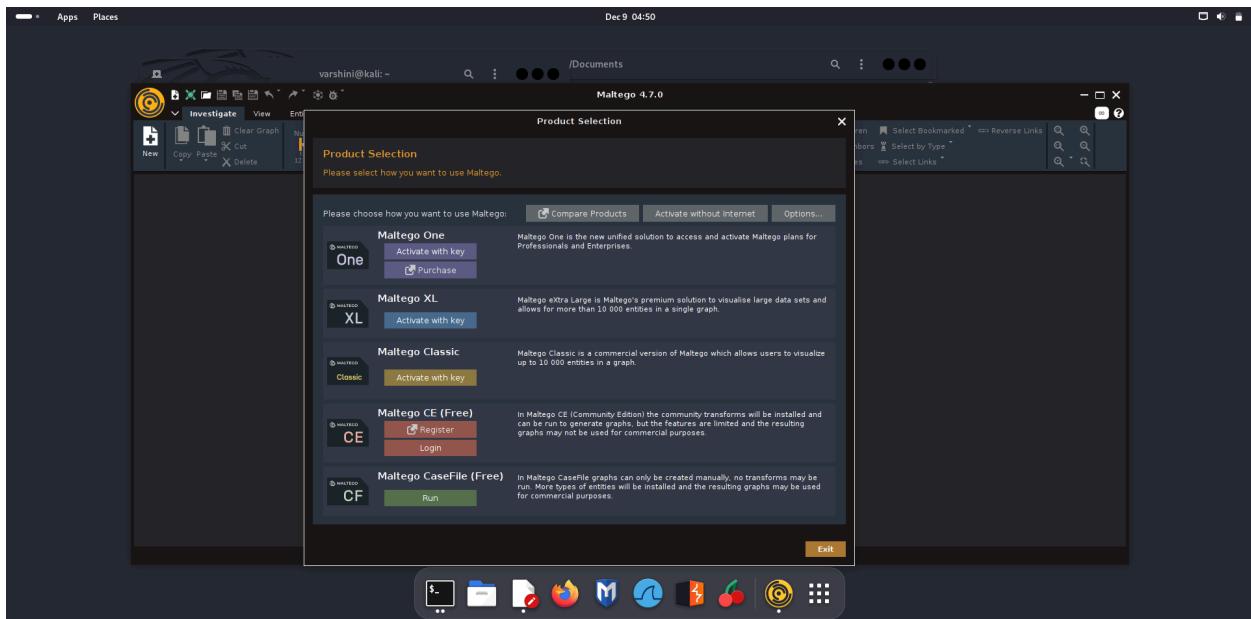
Maltego:

- Maltego is a powerful open-source intelligence (OSINT) and forensic investigation tool that focuses on mapping relationships between various entities such as people, organizations, domains, IP addresses, and more.
- It is used to conduct network analysis, data mining, and reconnaissance in cybersecurity and threat intelligence. Maltego excels at visualizing complex relationships and connections in a graph format, making it easier to identify patterns and understand the connections between different pieces of information.

Primary Use Cases:

- Network mapping to identify how domains, subdomains, IP addresses, and services are connected.

- Cyber threat intelligence to analyze the connections between different threats, such as attackers, malware, and victims.
- Investigative forensics for examining the relationships between people, organizations, and online activities in criminal investigations.
- Social engineering attacks, mapping out key individuals or targets and their networks.
- Phishing and fraud detection by mapping the links between email addresses, domains, and IP addresses.



Pipal

- Pipal is a password analysis tool used for statistical analysis of password datasets.
- It is primarily designed to evaluate and report on the strength and security of passwords within a given dataset.
- Pipal analyzes large collections of passwords (typically password dumps) and generates statistical reports,

identifying common password patterns, character distributions, and password length.

Primary Use Cases:

- Password analysis: Analyzing large datasets of passwords to identify common patterns and weak passwords.
 - Security audits: Evaluating the strength of passwords used within an organization by checking against common password lists and patterns.
 - Cracking password hashes: Assisting in cracking hashed passwords by identifying commonly used passwords and generating wordlists for further attacks.

```
$ pipal -h
pipal 3.4.0 Robin Wood (robin@digi.ninja) (http://digi.ninja)

Usage: pipal [OPTION] ... FILENAME
  --help, -h, -? show help
  --top, -t X: show the top X results (default 10)
  --output, -o <filename>: output to file
  --gkey <Google Maps API key>: to allow zip code lookups (optional)
  --list-checkers: Show the available checkers and which are enabled
  --verbose, -v: Verbose

  ...
  ... permission denied: ./h1.c
FILENAME: The file to count
(varshini㉿kali)-[~]
  ...
  ... ./h1.c is not an ELF file
  ...
  ... (varshini㉿kali)-[~]
  ... ./trace ./h1
  pipal "Hello world!Hello World"
  ...
  ... exited (status 0) ...
  ...
  ... (varshini㉿kali)-[~]
  ... ./trace -o h1.h1c
  'h1.c' is not an ELF file
  ...
  ... (varshini㉿kali)-[~]
  ... ./trace -o h1.h1c
  trace: too few arguments
  Try 'trace --help' for more information.
  ...
  ... (varshini㉿kali)-[~]
  ... ./trace -o h1.h1c
```

Social Engineering Tools

Social Engineering Toolkit:

- The Social Engineering Toolkit (SET) is an open-source penetration testing tool specifically designed for social engineering attacks.

- SET offers a wide range of attack vectors, including phishing, credential harvesting, and payload delivery mechanisms, making it a powerful tool for assessing and improving security awareness among employees.

Primary Use Cases:

- Phishing simulations: Crafting realistic phishing emails and fake websites to test if users will click on malicious links or provide credentials.
 - Credential harvesting: Capturing user credentials by creating fake login pages that resemble legitimate ones.
 - Penetration testing: Enhancing red team assessments by incorporating social engineering techniques.

```
  • Apps  Places   Dec 9 04:58
  Terminal

[...]
[...] The Social-Engineer Toolkit (SET) [--]
[...] Created by: David Kennedy (ReL1K) [--]
[...] Version: 0.8.3 [--]
[...] Codename: 'Maverick' [--]
[...] Follow us on Twitter: @TrustedSec [--]
[...] Follow me on Twitter: @HackingDave [--]
[...] homepage: https://www.trustedsec.com [--]
[...] Welcome to The Social-Engineer Toolkit (SET). [--]
[...] The one stop shop for all of your SE needs. [--]

The Social-Engineer Toolkit is a product of TrustedSec. [--]
Visit: https://www.trustedsec.com [--]

It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!
[...]
[...] Select from the menu: [--]
[...]      0) SET is not on SET file [--]
[...] 1) Social-Engineering Attacks [--]
[...] 2) Penetration Testing (Fast-Track) [--]
[...] 3) Third Party Modules [--]
[...] 4) Update the Social-Engineer Toolkit [--] usage --help for more information.
[...] 5) Update SET configuration [--]
[...] 6) Help, Credits, and About [--]
[...] 99) Exit the Social-Engineer Toolkit [--]

set> [--]
```

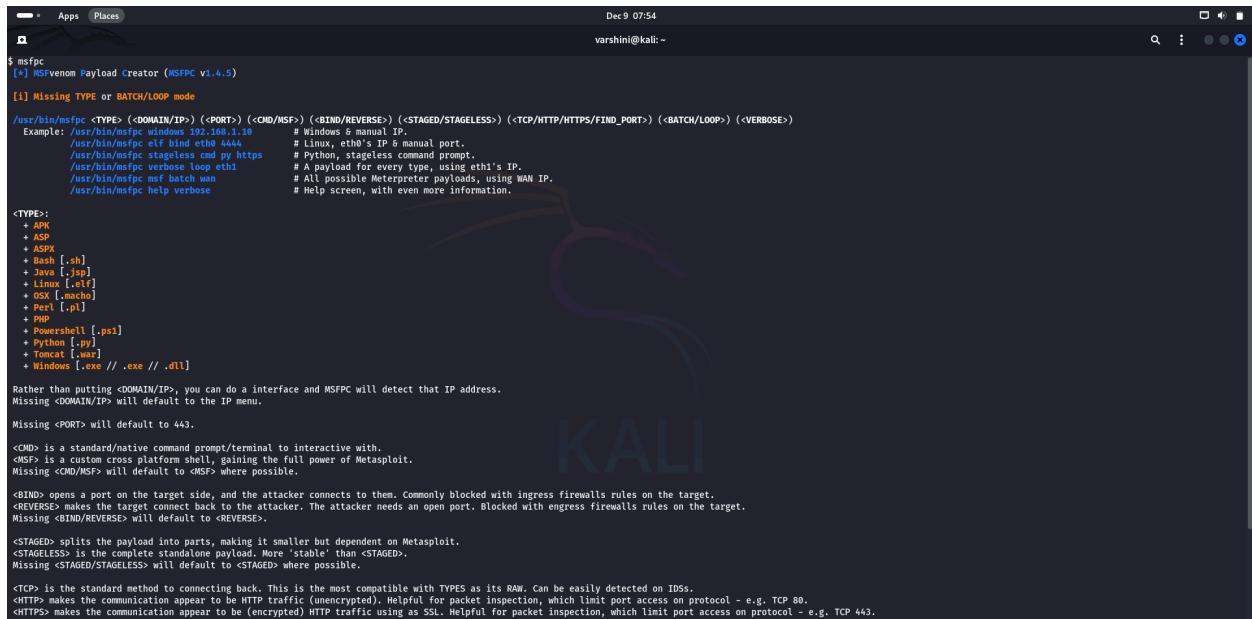
MSF Payload Creator (*MSFvenom Payload Creator - MSFPC*)

- MSF Payload Creator (MSFPC) is a script-based tool designed to simplify the creation of Metasploit payloads.

- It acts as a wrapper for MSFvenom, making it easier to generate payloads for different platforms and attack scenarios.

Primary Use Cases:

- Generating payloads for use with the Metasploit Framework.
- Simplifying payload creation by offering an intuitive, menu-driven interface.
- Automating reverse shell and bind shell payload generation for various operating systems (Windows, Linux, macOS, Android).



The screenshot shows a terminal window titled "MSFPC v1.4.5" running on Kali Linux. The user has typed "msfpc" and is presented with several options:

- [!] Missing TYPE or BATCH/LOOP mode**
- Example:** /usr/bin/msfpc windows 192.168.1.10 # Windows & manual IP.
- Available Payload Types:**
 - + APK
 - + ASP
 - + ASPX
 - + Bash [,.sh]
 - + Java [,.jsp]
 - + Linux [,.elf]
 - + OSX [,.macho]
 - + Perl [,.pl]
 - + Python [,.py]
 - + PowerShell [,.ps1]
 - + Tomcat [,.war]
 - + Windows [.exe // .dll]
- Notes:**
 - Rather than putting <DOMAIN/IP>, you can do a interface and MSFPC will detect that IP address.
 - Missing <DOMAIN/IP> will default to the IP menu.
 - Missing <PORT> will default to 443.
 - <CMD> is a standard/native command prompt/terminal to interactive with.
 - <MSF> is a custom cross platform shell, gaining the full power of Metasploit.
 - Missing <CMD/MSF> where possible.
 - <BIND> opens a port on the target side, and the attacker connects to them. Commonly blocked with ingress firewalls rules on the target.
 - <REVERSE> makes the target connect back to the attacker. The attacker needs an open port. Blocked with egress firewalls rules on the target.
 - Missing <BIND/REVERSE> will default to <REVERSE>.
 - <TCP> splits the payload into parts, making it smaller but dependent on Metasploit.
 - <STAGED> is the complete standalone payload. More 'stable' than <STAGED>.
 - Missing <STAGED/STAGELESS> will default to <STAGED> where possible.
 - <HTTP> is the standard method to connecting back. This is the most compatible with TYPES as it's RAW. Can be easily detected on IDS.
 - <HTTP> makes the communication appear to be HTTP traffic (unencrypted). Helpful for packet inspection, which limit port access on protocol - e.g. TCP 80.
 - <HTTPS> makes the communication appear to be (encrypted) HTTP traffic using as SSL. Helpful for packet inspection, which limit port access on protocol - e.g. TCP 443.

