# Generelle Form eines aufbereiteten Commands

* Tool name / Bezeichnung des Commands (z.B. nmap)
* kurze Beschreibung des Commands
* General Syntax
* Syntax Options: hier nur in CTF relevante Options aufführen
* 1-2 Default Commands / Beispielcommands, mit Beschreibung (diese aus den CTF ziehen)

Anmerkung:

* Falls ein Command sehr komplex ist (sehr lange allgemeine Syntax, sehr viele Optionen hat) kann die General Syntax und Syntax Options weggelassen und auf einen Datenbankeintrag verwiesen werden
* bei häufig vorkommenden, gut dokumentierbaren Commands gerne die gesamte Form aufbereiten. Dies dann in der schriftlichen Arbeit methodisch begründen warum manche so ausführlich und andere knapp dargestellt wurden

Fragen und Anmerkungen:

* Quelle für Commands? 🡪 gute Datenbank?(Internet), Shell für Abfrage?
* Genereller Umfang der Commands?
  + Voller Umfang wie in Datenbanken
  + Nur Options, welche in den CTFs behandelt wurden
* Syntax: Welche Bedeutung haben Groß- und Kleinschreibung, kursiv, fett gedruckte Elemente etc.?
  + kann/darf man die verallgemeinern
* es wurden so viele Challenges genannt, um genannten Default Commands zu hinterlegen (1-3)
  + teilweise wurden commands auch nur in wenigen / vereinzelten Challenges verwendet

Für Limitations:

* 2 Commands in einer Zeile wie z.B.: find ./ -type f | grep -v .html

🡪 hier wird nur der erste in den Statistiken geführt

* Abgleich mit einer Datenbank
* Nicht alle tag-Kombinationen in der Commands Extraction berücksichtig
* Erkenntnis: viele Challenges können mit Basic Commands gelöst werden 🡪 Know how mit und im System sehr hoher Stellenwert (wenn man weiß wie man etwas umgeht ist die Umsetzung nicht mehr so komplex)
* Verschachtelte Commands werden nur als erster Command geführt, z.B. sudo perl… 🡪 hier wird nur sudo erkannt

To-Do’s

General Description muss teilweise noch ergänzt werden:

* auf CTF anwenden / in ctf Kontext beschreiben/erweitert
  + welchen Nutzen hat der Command in CTFs
* ggf. konkreter Beispiele und Erklärungen unter den bisherigen ergänzen
  + Wann und wieso wird der Command in ctf eingesetzt 🡪 in Description
* Commands-Zuteilung am Ende noch einmal abchecken

Reminder

To maintain conciseness within the scope of this context, a comprehensive exposition for all extracted commands is not provided. In certain instances, the omission of specific options in conjunction with the general syntax is due to either the considerable volume of options available, surpassing the intended focus, or the typical usage of the command without the need for further option specification in the considered use case. For a more exhaustive understanding, readers are advised to consult the corresponding contents accessible within the database, which offer more intricate details regarding the commands and their associated options.

ggf. ergänzen, dass wenn Options zwar in general Syntax gefhrt werden aber dann nicht genauer erklärt bzw. weggelassen werden 🡪 diese sind zu umfangreich oder im Kontext von CTF nicht relevant

# Basic Linux Commands

### **cat**

General Description:

Prints the contents of a file to the standard output

General Syntax:

### cat [*OPTION*]… [*FILE*]…

Syntax Options:

* -n, --number 🡪 number of all output lines
* -A, --show all 🡪 shows non-printing, ends and tabs

Default Commands:

* Print the contents of a file “example.txt” to the standard output and show the number of output files

### $ cat -n [example.txt]

* Concatenate several files (“root.txt” and “flag.txt”) into an output file (“collection.txt”)

### $ cat [root.txt] [flag.txt] > [collection.txt]

### **cp**

General Description:

Copies Source file to chosen destination

General Syntax:

### cp [OPTION]... [SOURCE]... [DESTINATION]…

Syntax Options:

* -f, --force 🡪 if an existing destination file cannot be opened, remove it and try again (this option is ignored when the -n option is also used)
* -l, --link 🡪 hard link files instead of copying

Default Commands:

* Copy a file to another location

### $ cp [path/to/source\_file.ext] [path/to/target\_file.ext]

* Copy path to the Bash Shell to the current working directory

### $ cp /bin/bash

### **wget**

General Description:

Download the contents of a URL to a file

General Syntax:

### wget [*option*]… [*URL*]…

Syntax Options:

* -o logfile, --Output-file = logfile 🡪 Log all messages to the logfile
* -i file, --Input-file = file 🡪 Reads URLs from a file (local or external) for further processing

Default Commands:

* Download the contents of a URL to a file (named "storage" in this case)

### $ wget [https://example.com/storage]

* Download a file from an HTTP server using Basic Authentication

### $ wget --user=[username] --password=[password] [https://example.com]

* Download files specified in the input-file (f.e. an URL-List)

### $ wget -i [/path/to/input-file.txt]

* Download a file from the specified URL and save it as “shell.jsp” in the given directory

### $ wget [http://example.com] -O [/path/to/directory/shell.jsp]

### **python**

General Description:

Executes Python scripts or launches the Python interactive interpreter.

Default Commands:

* Execute a specific python file

### $ python [path/to/file.py]

* Execute a pyhton expression

### $ python -c "[expression]"

* Run the script of the specified library module

### $ python -m [module] [arguments]

* Import external modules or libraries

### $ python import os

### **sudo**

General Description:

The sudo command allows a permitted user to execute a command as the superuser or an‐ other user, as specified by the security policy.

Default Commands:

* Listing of all permissions or privileges assigned to your user account

### $ sudo -l

* Temporarily switching to the root user or another user with administrative privileges

### $ sudo su

* Running a command as another userand/or group

### $ sudo --user=[user] --group=[group] [id -a]

* Executing a program or script with elevated privileges

### $ sudo [path/to/script]

### **type**

General Description:

Used to determine the type or location of a command or executable. It provides information about how a command will be interpreted or executed by the shell.

General Syntax:

### type [Option] [*Command*]

Syntax Options:

* -t 🡪 prints a single word describing the type of the command
* -a 🡪 displays all locations that contain the command
* -p 🡪 returns the path to the command only if the command is an executable file on the disk

Default Commands:

* Display the type of a command

### $ type [command]

* Display all locations containing the specified executable

### $ type -a [command]

* Display the name of the disk file that would be executed

### $ type -p [command]

### **whoami**

General Description:

Prints the users name associated with the current effective user ID

General Syntax:

### whoami

Deafult Commands:

* Display the currently logged username

### $ whoami

* Display the username after a change in the user ID

### $ sudo whoami

### **ls**

General Description:

Lists files and directories in a directory

General Syntax:

### ls [*OPTION*]… [*FILE*]

Syntax Options:

* -a, --all 🡪 do not ignore entries starting with .
* -l 🡪 use a long listing format
* -r, --reverse 🡪 reverse order while sorting
* -t, --time 🡪 sort by time (newest first)
* -1 🡪 list one file per line

Default Commands:

* Long format list (permissions, ownership, size, and modification date) of all files

### $ ls -la

### **id**

General Description:

Returns user identity

General Syntax:

### id [*OPTION*]... [*USER*]...

Syntax Options:

* -u, --user 🡪 print only the effective user ID
* -g, --group 🡪 print only the effective group ID

Default Commands:

* Display current user's ID (UID), group ID (GID) and groups to which they belong

### $ id

* Display the current user identity as a number

### $ id -u

* Display the current group identity as a number

### $ id -g

### **find**

General Description:

Used to search for files and directories in a directory hierarchy based on various criteria. It recursively traverses directories, starting from a specified path, and matches files or directories that meet specific conditions.

Default Commands:

* Find files by extension:

### $ find [root\_path] -name '[\*.ext]'

* Find files matching multiple path/name patterns

### $ find [root\_path] -path '[\*\*/path/\*\*/\*.ext]' -or -name '[\*pattern\*]'

### **locate**

General Description:

Find files by name

General Syntax:

### locate [OPTION]... PATTERN...

Default Commands:

* Find a specified file

### $ locate [filename]

### **cd**

General Description:

Change the current working directory

General Syntax:

### cd [-L|-P] [directory]

Default Commands:

* Go to the home directory of the current user

### $ cd

* Go to the specified directory

### $ cd [path/to/directory

* Go to the home directory of the specified user

### $ cd ~[username]

* Go to the root directory

### $ cd /

### **unzip**

General Description:

Lists, tests, or extracts files from a ZIP archive

Default Commands:

* Extract all files/directories from specific archives into the current directory

### $ unzip [path/to/archive1.zip path/to/archive2.zip ...]

* Extract files/directories from archives to a specific path

### $ unzip [path/to/archive1.zip path/to/archive2.zip ...] -d [path/to/output]

### **pwd**

General Description:

Returns working directory name

General Syntax:

### pwd [OPTION]...

Default Commands:

* Print the current directory

### $ pwd

### **less**

General Description:

Used to view the contents of a file. It breaks down the content into manageable sections, making it easier to read and navigate through large files

Default Commands:

* View the contents of a given file

### $ less [filename]

### **grep**

General Description:

Used for searching and filtering text. It allows you to search for specific patterns or regular expressions within files or command outputs

General Syntax:

### grep [OPTION. . .] PATTERNS [FILE. . .]

Syntax Options:

* -i, --ignore-case 🡪 Ignore case distinctions in patterns and input data, so that characters that differ only in case match each other
* -v, --invert-match 🡪 Invert the sense of matching, to select non-matching lines.
* -r, --recursive 🡪 Read all files under each directory, recursively, following symbolic links only if they are on the command line

Default Commands:

* Search for a pattern within a file

### $ grep "[search\_pattern]" [path/to/file]

### **exit**

General Description:

Cause the shell to exit from its current execution environment

Default Commands:

* Exit the shell with the exit code of the last command executed

### $ exit

### **powershell**

General Description:

Provides an interactive shell session which presents a flexible and efficient way to interact with the Windows operating system, perform administrative tasks, and write scripts for various purposes

Default Commands:

* Start an interactive shell session

### $ powershell

* Download content from a URL using an HTTP GET request and save it locally

### $ powershell -c iwr [URL] -outf [path/to/output/file]

Challenge Names:

* Anubis Hackthebox Walkthrough

### **mkdir**

General Description:

Create Directory(ies), if they do not already exist

General Syntax:

### mkdir [OPTION]... DIRECTORY...

Default Commands:

* Create specific directories

### $ mkdir [path/to/directory1 path/to/directory2 ...]

Challenge Names:

* Kenobi Tryhackme Walkthrough
* Hack The Box Challenge Joker Walkthrough

### **php**

General Description:

Parse and execute PHP filea

General Syntax:

### php [options] [ -f ] file [[--] args. . .]

Default Commands:

* Parse and execute a PHP script

### $ php [path/to/file]

* Start a PHP built-in web server in the current directory

### $ php -S [host:port]

Challenge Names:

* proteus-1-ctf-walkthrough
* Enubox Mattermost Vulnhub Walkthrough

### **g++**

General Description:

Takes C++ source code files as input and generates executable files that can be run on a compatible system

Default Commands:

* Compile a source code file into an executable binary

### $ g++ [path/to/source.cpp] -o [path/to/output\_executable]

Challenge Names:

* Hack The Lampiao 1 Ctf Challenge

### **if**

General Description:

Used to conditionally load another module

General Syntax:

### use if CONDITION, "MODULE", ARGUMENTS; no if CONDITION, "MODULE", ARGUMENTS;

Default Commands:

* Execute the specified commands if the condition command's exit status is zero

### $ if [condition\_command]; then [echo "Condition is true"]; fi

Challenge Names:

* Hack The Gemini Inc2 Ctf Challenge

# Network Scanning

### **nmap**

General Description:

The nmap command comes with many options and use cases depending on the situation at hand. In the considered CTF challenges the command is mainly used to:

* scan for open ports
* detect service and version information from the open ports
* perform an aggressive scan
* don’t ping
* UDP scan
* TCP Connect Scan
* default script scanning

General Syntax:

### nmap [*Scan Type…*] [*Options*] {*target* *specification*}

Syntax Options:

* Scan for open ports: nmap -p- <target>
* Scan for Service Version Detection: nmap -sV <target>
* Perform an aggressive scan: nmap -A <target>
* Don’t Ping: nmap -pN <target>
* UDP Scan: nmap -sU <target>
* Default Script scanning: nmap -sC <target>
* TCP Connect Scan: nmap -sT <target>
* <target> is the IP address of looked at system

Default Commands:

* Comprehensive port scanning an version detection. Writing results (‘-oN’) to output file

### $ nmap -p- -sV [address\_or\_addresses] -oN [ports.txt]

* Aggressive Scan that also enable scripts, service detection, OS fingerprinting and traceroute

### $ nmap -A [address\_or\_addresses]

### **ssh**

General description:

Establishes a secure shell (SSH) connections to remote machines. It provides a secure encrypted channel for communication between the local and remote systems.

Default Commands:

* Connect to a remote server

### $ ssh [username]@[remote\_host]

* Connect to a remote server using a specific port

### $ ssh [username]@[remote\_host] -p [2222]

### **sftp**

General Description:

Used for secure file transfer between computers over an SSH (Secure Shell) connection. It provides a way to transfer files securely and remotely, similar to FTP (File Transfer Protocol), but with encryption to protect the data during transmission

Default Commands:

* Connect to the remote server with given IP address, FTP username, custom port

### $ sftp -p [port] [ftp\_username]@[IP\_address]

Challenge Names:

* Hack The Box Nightmare Walkthrough

### **netdiscover**

General Description:

Used to discover and identify active hosts (IP-Address) on a local network

General Syntax:

### netdiscover [-i device] [-r range | -l file | -p] [-m file] [-F filter] [-s time] [-c count] [-n node] [-dfPLNS]

Default Commands:

* Discover and identify active hosts on the local network (Output: IP-Address)

### $ netdiscover

* Run netdiscover and scan a given range

### $ netdiscover -r [Starting\_Point/Range]

* Run netdiscover on a specific network interface

### $ netdiscover -i [Interface]

### **objdump**

General Description:

Used to display information about one or more object files

Default Commands:

* Display the file header information

### $ objdump -f [binary]

Challenge Names:

* proteus-1-ctf-walkthrough

### **arp-scan**

General Description:

Used for scanning and discovering devices within a local network. It allows you to retrieve information about the MAC (Media Access Control) addresses and IP addresses of devices connected to the same network

General Syntax:

### arp-scan [options] [hosts...]

Syntax Options:

* -l, --localnet 🡪 Use the network interface IP address and network mask to generate the list of target host addresses
* -i, <x>, --interval=<x> 🡪 Set minimum packet interval to <x>
* -g, --ignoredups 🡪 Don't display duplicate packets

Default Commands:

* Scan the current local network

### $ arp-scan -l

* Scan a specific network interface

### $ arp-scan -i [network\_interface\_name]

### **file**

General Description:

Determine file type

General Syntax:

file [option ] file ...

Default Commands:

* Give a description of the type of the specified file. Works fine for files with no file extension

### $ file [path/to/file]

### **base64**

General Description:

Used to encode or decode data in base64 format. This is commonly used to discover hidden information

General Syntax:

### base64 [OPTION]... [FILE]

Syntax Options:

* -d, --decode 🡪 decode data

Default Commands:

* Decode the base64 contents of a file and write the result to standard output

### $ base64 -d [path/to/file]

* Decode the base64 contents of a file and write the result to a specified file

### $ base64 -d [path/to/file] > [path/to/safe\_file]

### **base32**

General Description:

Used to encode or decode data in base32 format. This is commonly used to discover hidden informatio

General Syntax:

### base32 [OPTION]... [FILE]

Syntax Options:

* -d, --decode 🡪 decode data

Default Commands:

* Decode a file

### $ base32 -d [path/to/file]

* Decode the base64 contents of a file and write the result to a specified file

### $ base32 -d [path/to/file] > [path/to/safe\_file]

Challenge Names:

* Geisha1 Vulnhub Walkthrough
* hogwarts-dobby-vulnhub-ctf-walkthrough

### **ifconfig**

General Description:

Used to interact with the configuration settings of various services, applications, or systems involved to uncover potential hidden configuration details or secret settings

Default Commands:

* View network settings of an Ethernet adapter

### $ ifconfig eth0

### **tcpdump**

General Description:

Used to capture and analyze network traffic. It allows you to inspect the packets flowing through a network interface and provides valuable information about the communication happening between devices on a network

Default Commands:

* List available network interfaces

### $ tcpdump -D

* Capture network traffic on the loopback interface (lo) and writes it to a specified file

### $ tcpdump -w [filename] -i lo

* Captures network traffic, filtering the captured packets to only show those with a source or destination IP address

### $ tcpdump -A -n host [IP\_address] and arp

Challenge Names:

* Symfonos3 Vulnhub Walkthrough
* Jigsaw1 Vulnhub Walkthrough

### **telnet**

General Description:

Login to remote system host

General Syntax:

### telnet [OPTION...] [HOST [PORT]]

Default Commands:

* Telnet to the default port of a host

### $ telnet [host]

* Telnet to a specific port of a host

### $ telnet [ip\_address] [port]

Challenge Names:

* Hack Billy Madison Vm Ctf Challenge
* Casino Royale 1

### **sh**

General Description:

Command-line interpreter used to execute commands read from a command line string, the standard input, or a specified file

General Syntax:

### sh -c “[command]” [option]...

Default Commands:

* Execute a command and then exit

### $ sh -c [command]

* Execute a command in a modified environment, setting options for the passed path variable

### $ sh -c [command] -i [path\_variable]=[variable\_value\_1]:[variable\_value\_2]

Challenge Names:

* depth-1-ctf-walkthrough
* Hack The Box Writeup Walkthrough

### **smbmap**

General Description:

Used to enumerate samba share drives across an entire domain, simplifying searching for potentially sensitive data across large networks

General Syntax:

### smbmap [options]

Syntax Options:

* -H HOST 🡪 IP of host
* -u USERNAME 🡪 Username, if omitted null session assumed
* -p PASSWORD 🡪 Password or NTLM hash

Default Commands:

* Enumerate a specified host with NULL sessions enabled and open shares

### $ smbmap -H [host\_IP\_address]

* Enumerate a specified host and check SMB file permissions

### $ smbmap -H [host\_IP\_address] -u [username] -p [password]

Challenge Names:

* My File Server 1 Vulnhub Walkthrough
* Anubis Hack the Box

# Enumeration

### **echo**

General Description:

The echo command is commonly used to display text strings or command results as the standard output (messages)

General Syntax:

### echo [*OPTION*] [*STRING*]

Syntax Options:

* -n 🡪 Displays the output while omitting the newline after it / do not output the trailing newline
* -e 🡪 Enables the interpretation of (backlash) escape characters

Escape Characters:

* \\ backslash
* \n new line
* \t horizontal tab
* \v vertical tab

Default Commands:

* Print a chosen text message (f.e. an IP address)

### $ echo [192.168.1.10]

* Print a message without the trailing newline

### $ echo -n [php-reverse-shell.php]

* Append a message to the file

### $ echo [192.168.1.10] >> [ip\_addresses.txt]

### **rm**

General Description:

Removes directory entries

General syntax:

### rm [*OPTION*]… [*FILE*]…

Syntax Options:

* -f, --force 🡪 ignore nonexistent files and arguments, never prompt
* -i 🡪 prompt before every removal
* -r, -R, --recursive 🡪 remove directories and their contents recursively

Default Commands:

* Remove specific files

### $ rm [path/to/file1 path/to/file2 ...]

* Remove specific files ignoring nonexistent ones

### $ rm -f [path/to/file1 path/to/file2 ...]

* Remove specific files [i]nteractively prompting before each removal

### $ rm -i [path/to/file1 path/to/file2 ...]

* Remove specific files and directories [r]ecursively

### $ rm -r [path/to/file\_or\_directory1 path/to/file\_or\_directory2 ...]

### **mv**

General Description:

Rename a file or directory or move a file(s) to a directory.

General Syntax:

### mv [OPTION]... SOURCE... DIRECTORY

Default Commands:

* Rename a file or directory (when the target is not an existing directory)

### $ mv [original\_name] [target\_name]

* Move a file or directory into an existing directory

### $ mv [file.py] [name\_of\_existing\_directory]

### **dirb**

General Description:

The dirb command is a web application scanner used for directory brute-forcing. It helps to discover hidden directories or files on a web server by systematically testing different combinations.

General Syntax:

### **dirb** **<url base>** <*url base*> [*<wordlist\_file(s)>*] [*options*]

Syntax Options:

* -o <output file> 🡪 Save output to disk
* -X <extensions> 🡪 Amplify search with this extensions
* -u <username:password> 🡪 username and password to use
* -w 🡪 don’t stop on warning messages
* -p <proxy[:port]> 🡪 Use this proxy (Default port is 1080)
* -r 🡪 don’t search recursively
* -a <agent string> 🡪 Specify your custom USER\_AGENT (Default is: “Mozilla/4.0 (compatible MSIE 6.0; Windows NT 5.1)”)

Wordlist\_files(s):

Used to scan a webserver using a custom wordlist

Default Commands:

* Scan a webserver using the default wordlist

### $ dirb [https://example.org]

* Scan a webserver using a custom wordlist

### $ dirb [https://example.org] [path/to/wordlist.txt]

* Scan a webserver using a specified user-agent and cookie for HTTP-requests

### $ dirb [https://example.org] -a [user\_agent\_string] -c [cookie\_string]

### **sed**

General Description:

Sed is a stream editor used to perform basic text transformation on an input stream (a file or input from a pipeline)

General Syntax:

### sed [*OPTION*]… {*script-only-if-no-other-script*} [*input-file*]…

Syntax Options:

* -i[SUFFIX], --in-place[=SUFFIX] 🡪 edit files in place (Backup if Suffix is supplied)

Default Commands:

* In-place text substitution: Replaces the end of each line with “-upload.pdf” in the related file

### $ [command] | sed -i ['s/$/-upload.pdf/'] [file.txt]

* Execute a specific script [f]ile and print the result to stdout

### $ [command] | sed -f [path/to/script.sed]

### **exiftool**

General Description:

Reads and writes meta information in given files

Default Commands:

* Print the EXIF metadata for a given file

### $ exiftool [path/to/file]

* Modifying the comment metadata of given image file

### $ exiftool -[MetadataTag]='[NewValue]' [ImageFile]

### **gobuster**

General Description:

Brute-forces hidden paths on web servers and more to discover hidden or unprotected content

General Syntax:

### gobuster [command] [options]

Syntax Options:

Command:

* dir 🡪 Used for directory brute-forcing to discover hidden directories or files on a web server
* dns 🡪 Used for DNS subdomain brute-forcing to find subdomains associated with a given domain.
* vhost 🡪 Used for virtual host brute-forcing to identify virtual hosts configured on a web server.

Options:

* -u string 🡪 the target URL or domain
* -w string 🡪 path to the wordlist
* -t int 🡪 number of concurrent threads
* -x string 🡪 file extensions(s) to search for (dir mode only)
* -e 🡪 Expanded mode, print full URLs
* -s string 🡪 Proxy to use for requests [http(s)://host:port] (dir mode only)

Default Commands:

* Discover directories and files that match in the wordlist

### $ gobuster dir -u [https://example.com/] -w [path/to/wordlist]

* Discover subdomains that match in the wordlist

### $ gobuster dns --domain [https://example.com] -w [path/to/wordlist]

* Fuzz the value of a parameter that match in the wordlist

### $ gobuster fuzz -u [https://example.com/?parameter=FUZZ] -w [path/to/wordlist]

### **nano**

General Description:

Nano is a small and friendly editor that copies the look and feel of Pico and implements several features such as: opening multiple files, scrolling per line, undo/redo, syntax coloring, line numbering, and soft-wrapping overlong lines.

General Syntax:

### nano [*options*] [[+*line*[,*column*]] *file*]...

Syntax Options:

* -P, --Positionlog 🡪 For the 200 most recent files, log the last position of the cursor, and place it at that position again upon reopening such a file.

Default Commands:

* Start the editor

### $ nano

* Opena file for editing using the nano text editor

### $ nano [filename]

* Open a file and position the cursor at a specific line and column using the nano text editor

### $ nano +[line],[column] [path/to/file]

### **fping**

General Description:

Used to determine if a target host is responding to identify live hosts, network topology, and assessing the availability of services.

Default Commands:

* List alive hosts within a subnet generated from a netmask

### $ fping -a -q -g [starting\_IP\_address]/[IP\_address\_range\_in\_CIDR\_notation]

### **wpscan**

General Description:

Used to scan WordPress websites for security vulnerabilities, enumerate WordPress installations, plugins, and themes, and perform password brute-forcing and username enumeration

General Syntax:

### wpscan [*options*]

Syntax Options:

* -url URL 🡪 the url of the blog to scan
* -api-token TOKEN 🡪 displays vulnerability data, available at https://wpscan.com/profile
* -plugins-detection MODE 🡪 Use the supplied mode (mixed, passive, aggressive) to enumerate Plugins
* -e, --enumerate [OPTS] 🡪 Choosing Enumeration Process
* -U, --username LIST 🡪 List of usernames to use during the password attack.
* -P, --passwords FILE-PATH 🡪 List of passwords to use during the password attack

Default Commands:

* Scan a WordPress website

### $ wpscan --url [https://example.com/]

* Scan a WordPress website with enabled enumeration phase to perform various enumerations.

### $ wpscan --url [https://example.com/] -e

* Scan a WordPress website using a custom password list and a list of usernames

### $ wpscan --url [https://example.com/] -P [path/to/passwordlist.txt] -U [path/to/userlist.txt]

### **nikto**

General Description:

Examine a web server to find potential problems and security vulnerabilities, including: Server and software misconfigurations, Default files and programs, Insecure files and programs; Outdated servers and programs

General Syntax:

### nikto -h [Option] [Target\_URL]

Syntax Options:

* -userproxy 🡪 set specified settings for the proxy server

Default Commands:

* Perform a basic Nikto scan against a target host

### $ nikto -h [https://Target\_IP\_Address]

* Perform a basic Nikto scan with default settings to indicate the specified proxy server for the scan

### $ nikto -h [http://Target\_IP\_Address] -userproxy [https://Target\_IP\_Address[:Port]]

### **searchsploit**

General Description:

Exploit Database Archive Search: Allow you to search through exploits and shellcodes using one or more terms from Exploit-DB

General Syntax:

### searchsploit [options] term1 [term2] ... [termN]

Syntax Options:

* -m, --mirror 🡪 Mirror (aka copies) an exploit to the current working directory

Default Commands:

* Search for an exploit, shellcode, or paper

### $ searchsploit [search\_terms]

* Make a copy of the resource to the current directory (requires the number of the exploit)

### $ searchsploit -m [exploit\_number]

### **enum4linux**

General Description:

Performs various enumeration techniques to extract valuable information, such as user and group details, share information, password policies, domain information, and more. It utilizes different techniques like null sessions, LDAP queries, and SMB queries to gather as much information as possible from the target system

General Syntax:

### enum4linux [Option] [Target]

Target = IP-Address

Default Commands:

* Try to enumerate using all methods

### $ enum4linux -a [remote\_host]

* Enumerate using given login credentials

### $ enum4linux -u [user\_name] -p [password] [remote\_host]

### **uname**

General Description:

Used to obtain basic information about the underlying operating system and system architecture of a target system

General Syntax:

### uname [OPTION]...

Default Commands:

* Print all available system information

### $ uname -a

### **sqlmap**

General Description:

Automatic SQL Injection Tool used to exploit SQL injection vulnerabilities in web applications to gain unauthorized access or extract sensitive information

General Syntax:

### sqlmap [options]

Syntax Options:

* -u URL, --url=URL 🡪 specifies target URL
* -D Db\_Name 🡪 specifies database name to enumerate
* -T Table\_Name 🡪 specifies the table within the selected database
* -- dbs 🡪 enumerate databases available on the target
* -- dump 🡪 Dump the contents of the specified table
* -- batch 🡪 enables batch mode, which automatically selects default options and avoids user interaction
* -- forms 🡪 identify and analyze HTML forms on the target page.

Default Commands:

* Run sqlmap against a single target URL

### $ sqlmap -u [http://example.com]

* Run sqlmap from a request file containing the HTTP request

### $ sqlmap -r [request\_file.txt]

* Perform a batched SQL injection attack on a target URL, targeting a specific database and table

### $ sqlmap -u [http://example.com] -D [database\_name] -T [table\_name] --dump --batch

### **dir**

General Description:

List directory contents

General Syntax:

### dir [OPTION]... [FILE]...

Default Commands:

* Display a list of files and directories in the current working directory

### $ dir

* List subdirectories recursively

### $ dir --recursive

### **ffuf**

General Description:

Helps to find hidden directories and files on a web server by brute-forcing URLs using a specified wordlist. It works by replacing the FUZZ keyword in the given URL with words from the wordlist and sends HTTP requests to each generated URL

General Syntax:

### ffuf [options]

Syntax Options:

* -u 🡪 target URL
* -e 🡪 Comma separated list of extensions. Extends FUZZ keyword
* -w 🡪 Wordlist file path

Default Commands:

* Discover directories using a [w]ordlist on a target [u]rl with specified file [e]xtensions

### $ ffuf -u [http://example.com] -w [path/to/wordlist] -e [specified\_file\_extension]

### **ps**

General Description:

Displays information about a selection of the active processes which helps in identifying running services, applications, and potentially suspicious processes or/and in finding a specific process based on certain criteria.

General Syntax:

### ps [options]

Syntax Options:

* -a 🡪 Select all processes except both session leaders
* -u 🡪 Current user
* -x 🡪 Includes processes that do not have a controlling terminal (e.g. background processes)
* -e 🡪 Select all processes
* -f 🡪 full-format listing

Default Commands:

* List all running processes

### $ ps aux

* Display a detailed process listing in a hierarchical format, including all processes and their relationships

### $ ps eaf

### **dig**

General Description:

Performs DNS (Domain Name Systems) lookups and displays the answers that are returned from the name server(s) that were queried. Used for DNS enumeration, which involves systematically querying DNS servers to gather information about a target domain or network. It can be used to discover subdomains, identify DNS zone transfers, find DNS misconfigurations, or locate potential entry points for further exploration

Default Commands:

* Retrieve domain information of a known Domain with given IP-address

### $ dig @[IP\_address] [domain\_name]

* Perform a zone transfer providing all the DNS records associated with the specified domain (subdomains, IP addresses, mail server configurations, and other DNS resource record)

### $ dig axfr @[IP\_address] [domain\_name]

### **vim**

General Description:

Open a file in the vim text editor for quickly navigating and editing files that involve analyzing or modifying code, configurations, or other text-based data.

General Syntax:

### vim [options] [file]

Syntax Options:

* -r 🡪 Lists swap files and indicates that the file was previously edited and offers the option to recover it

Default Commands:

* Open a file

### $ vim [path/to/file]

* Recover unsaved changes made to a specified file in Vim

### $ vim -r [path/to/file]

### **gunzip**

General Description:

Decompresses a compressed file and restores it to its original form. This allows to access and work with the uncompressed data

General Syntax:

### gunzip [options] [filename]

Default Commands:

* Extract a file from an archive, replacing the original file if it exists

### $ gunzip [archive.tar.gz]

### **gzip**

General Description:

Compress or expand files

Default Commands:

* Decompress a file, replacing it with the original uncompressed version

### $ gzip -d [file.ext].gz

Challenge Names:

* Hack The Box Curling Walkthrough

### **zip**

General Description:

Creates Zip archives containing the compressed versions of the specified files or directories

General Syntax:

### zip [OPTIONS] [ARCHIVE\_NAME] [FILES]

Syntax Options:

* -r, --recurse-paths 🡪 Travel the directory structure recursively
* -# 🡪 Regulate the speed of compression using the specified digit #

Default Commands:

* Create a recursive ZIP archive in the current directory

### $ zip -r [Archive\_name]\*

* Store all files (no compression)

### $ zip -0 [Archive\_name] [file\_name]

### **bzip2**

General Description:

Used to compress and decompress files in a simple way. Each file is replaced by a compressed version of itself. The compressed files typically have the extension “.bz2”

Default Commands:

* Decompress a file

### $ bzip2 -d [path/to/compressed\_file.bz2]

Challenge Names:

* Hack The Box Curling Walkthrough

### **net**

General Description:

Uses to discover open ports, services, and their associated configurations on target systems

Default Commands:

* Display information about all user accounts on a system

### $ net user

* Display information about one specified user on a system

### $ net user [username]

Challenge Name:

* Driver Hack the Box
* Hack the Box Access

### **netstat**

General Description:

Print network connections, routing tables, interface statis‐ tics, masquerade connections, and multicast memberships

General Syntax:

netstat [options]

Syntax Options:

* -a, -all 🡪 show listening and non-listening sockets
* -n, --numeric 🡪 Show numerical addresses instead of trying to determine symbolic host, port or user names
* -t, --tcp 🡪 show only TCP connections
* -p, --program 🡪 Show the PID and name of the program to which each socket belongs
* -l, --listening 🡪 show only listening sockets
* -e, --extend 🡪 Display additional information. Use this option twice for maximum detail

Default Commands:

* Display all active TCP connections, along with their associated addresses, ports, and process information

### $ netstat -antp

* Display only listening sockets, along with their associated addresses, ports, and process information

### $ netstat -ntlp

* Display all active TCP connections, along with extended information

### $ netstat -aepn

Challenge Names:

* The Box Challenge Inception Walkthrough
* Tempus Fugit 1 Vulnhub Walkthrough

### **rpcclient**

General Description:

Used to interact with Remote Procedure Call (RPC) services on a network. Commonly used when its needed to explore, enumerate, or exploit RPC services for gaining unauthorized access, privilege escalation, or extracting information

Default Commands:

* Connect to a remote host

### rpcclient -U [username]%[password] [target\_ip]

Challenge Names:

* Blackfield Hackthebox Walkthrough
* Fuse Hackthebox Walkthrough

### **apk**

General Description:

Used to install new packages (f.e. on virtual machines)

Default Commands:

* Install a new package

### $ apk add [package]

Challenge Names:

* Tempus Fugit 1 Vulnhub Walkthrough

### **lynx**

General Description:

used to launch the Lynx web browser and access a web page located at the specified URL, allowing to browse and interact with the web content hosted at that address. Commonly used to browse text-based web interfaces, analyze source code and clues, inspect HTTP headers and responses, manipulate HTML elements, and interact with web-based challenges in a command-line environment

General Syntax:

### lynx [options] [URL]

Default Commands:

* Open the Lynx web browser and direct it to the specified URL

### $ lynx http://192.168.1.104:8080

Challenge Names:

* Tr0Ll 3 Vulnhub Walkthrough

### **ip**

General Description:

Show / manipulate routing, network devices, interfaces and tunnels

Default Commands:

* List interfaces with detailed info

### $ ip address

Challenge Names:

* Hack the Boc Carrier
* 2Much 1 Vulnhub Walkthrough

### **lftp**

General Description:

Used to establish an FTP connection to achieve information hidden in FTP services / target machines

Default Commands:

* Connects to the FTP server located at a specified IP address with a given username

### $ lftp [username]@[IP\_address]

Challenge Names:

* Tempus Fugit 1 Vulnhub Walkthrough

### **tftp**

General Description:

Used for transferring files to and from remote servers using the Trivial File Transfer Protocol (TFTP). Can be used to achieve information hidden in TFTP target machines

Default Commands:

* Connect to a TFTP server specifying its IP address and port

### $ tftp [server\_ip] [port]

Challenge Names:

* Enubox Mattermost Vulnhub Walkthrough
* Hack The Box Challenge Joker Walkthrough

### **umask**

General Description:

Used to set the default permissions for newly created files and directories. It defines the permissions that are automatically subtracted or masked from the maximum permissions when a new file or directory is created

Default Commands:

* Allow all permissions for newly created files and directories

### $ umask 000

Challenge Names:

Hack The Box Wall Walkthrough

### **xxd**

General Description:

Used to create a hex dump of a given file or standard input. It can also convert a hex dump back to its original binary form.

General Syntax:

### xxd [options] [infile [outfile]]

Syntax Options:

* -r, --revert 🡪 Reverse operation: convert (or patch) hexdump into binary
* -p, --ps, --postscript, --plain 🡪 Output in postscript continuous hexdump style

Default Commands:

* Revert a plaintext hexdump back into binary, and save it as a binary file

### $ xxd -r -p [input\_file] [output\_file]

Challenge Names:

* Hack The Box Curling Walkthrough
* Mission Pumpkin V1 0 Pumpkinfestival Vulnhub Walkthrough

### **tac**

General Description:

Write each file to standard output, last line first. This is useful for tasks that involve analyzing or manipulating data in a reverse order like f.e. reverse engineering, detection of patterns

General Syntax:

### tac [OPTION]... [FILE]...

Default Commands:

* Concatenate specific files in reversed order

### $ tac [path/to/file1 path/to/file2 ...]

Challenge Names:

* Hack The Box Waldo Walkthrough

### **nslookup**

General Description:

Used to query Internet domain name servers. Interactive mode allows to query name servers for information about various hosts and domains or to print a list of hosts in a domain. Non-interactive mode prints just the name and requested information for a host or domain

General Syntax:

### nslookup [-option] [domain\_name | -] [server]

Default Commands:

* Query a DNS server for a specified query type record of a specific domain name

### nslookup -q=[query\_type] [domain\_name] [IP\_address]

Challenge Names:

* Hack Game Thrones Vm Ctf Challenge

### **mawk**

General Description:

Interpreter for the AWK programming language. It is used to process text files by applying patterns and actions to extract or transform data based on specified rules

Default Commands:

* Search for lines containing a search string / keyword, printing the ones that match the pattern

### $ mawk '/[search\_string]/' [example.txt]

Challenge Names:

* Hack The Toppo1 Vm Ctf Challenges

### **snmp-check**

General Description:

Used to identify the configuration and capabilities of SNMP agents running on the target system

General Syntax:

### snmp-check [options] [target]

Default Commands:

* Check for used SNMP version

### $ snmp-check [IP\_address]

Challenge Names:

* Pandora Hackthebox Walkthrough

### **snmpwalk**

General Description:

Used to explore and query SNMP-supported devices to gather information about network configuration, system statistics, and more

Default Commands:

* Query the system information of a remote host using SNMPv1 and a community string

### $ snmpwalk -v1 -c [community\_string] [ip]

* Query system information on a remote host using SNMPv2 on a specified port

### $ snmpwalk -v2c -c [community\_string] [ip]:[port]

Challenge Names:

* Pandora Hackthebox Walkthrough
* Pit Hackthebox Walkthrough

### **svn**

General Description:

Used to retrieve the entire repository or a specific directory and its contents to your local machine, creating a working copy that can be used to change, commit revisions, and synchronize with the remote repository. Useful for analyzing code changes or identifying modifications that may have introduced vulnerabilities or hidden information

General Syntax:

### svn command [options] [args]

Default Commands:

* Check out a Subversion (SVN) repository located

### $ svn checkout svn://[SVN\_IP\_address]

* Command compares the changes made between two revisions in an SVN repository

### $ svn diff -r2

Challenge Names:

* Worker Hackthebox Walkthrough

### **sqlitebrowser**

General Description:

Visual tool used to create, design and edit database files compatible with SQLite used to f.e. look for tables and entries stored in a database file

General Syntax:

### sqlitebrowser [file]

Default Commands:

* Open and browse the contents of a specified SQLite database file

### $ sqlitebrowser [filename]

Challenge Names:

* Mustacchio Tryhackme Walkthrough

### **cewl**

General Description:

Custom wordlist generator which spiders a given URL, up to a specified depth, and returns a list of words which can then be used for password bruteforce attacks

General Syntax:

### cewl [OPTION] ... URL

Syntax Options:

* -w, --write 🡪 Write the output to the specified file
* -d, --depth 🡪 Depth to spider (default = 2)
* -m, --min\_word\_length 🡪 Minimum word length (default = 3)
* --with-numbers 🡪 Accept words with numbers in as well as just letters

Default Commands:

* Create a wordlist file from the given URL with a specified links depth

### $ cewl --depth [spider\_depth] - [path/to/wordlist.txt] [url]

* Create a wordlist file from the given URL accepting words with numbers

### $ cewl -w [path/to/wordlist] --with-numbers [url]

* Create a wordlist file from the given URL through a proxy with specified links depth, minimum word length

### $ cewl -d [spider\_length] -m [minimum\_word\_length] --proxy\_host [host] --proxy\_port [port] -w [path/to/wordlist] [url]

Challenge Names:

* Ha Vedas Vulnhub Walkthrough
* Zion 1 1 Vulnhub Walkthrough
* Hack Usv Vm Ctf Challenge

smbpasswd

General Description:

When run as a normal user it allows the user to change the password used for their SMB sessions on any machines that store SMB passwords.

Default Commands:

* Change the password for a specified user on a specified remote server

### $ smbpasswd -r [IP\_address\_remote\_server] -U [username]

Challenge Names:

* Fuse Hackthebox Walkthrough

### **host**

General Description:

Used to perform DNS (Domain Name System) lookups and retrieve information about a specific host or domain

Default Commands:

* Request a full DNS zone transfer from a specified domain

### $ host -l [domain\_name] [domain\_IP\_address]

Challenge Names:

* Hack The Box Friendzone Walkthrough

### **ltrace**

General Description:

Used to trace and analyze the library calls made by an executable. It helps in understanding the behavior of the application, including any function calls, arguments, and return values, which can be useful in identifying the password being used by the application.

Default Commands:

* Trace and display the library calls made by an executable

### $ ltrace [executable]

Challenge Names:

* Hack The Box Dab Walkthrough

### **ss**

General Description:

Used to dump socket statistics. It allows showing information similar to netstat

General Syntax:

### ss [options] [FILTER]

Default Commands:

* List all listening TCP network sockets on a system

### $ ss -tnl

Challenge Names:

* Cache Hackthebox Walkthrough

### **volatility**

General Description:

Memory forensics framework used for the extraction of digital artifacts from volatile memory (RAM) samples. It is useful in forensics analysis. This can be used to find potential relevant information in the memory dump.

Default Commands:

* Analyze a memory dump file and extract information about the memory image

### $ volatility -f [path/to/memory\_dump\_file] imageinfo

Challenge Names:

* Ha Sherlock Vulnhub Walkthrough

### **wireshark**

General Description:

GUI network protocol analyzer used to interactively browse and analyse packet data from a live network or from a previously saved capture file (the native capture file formats are pcapng and pcap)

Default Commands:

* Launche the Wireshark network protocol analyzer and open a specified file for analysis

### $ wireshark [filename]

Challenge Names:

* Ha Chanakya Vulnhub Walkthrough

### **groupadd**

General Description:

Used to create a new group on a Linux or Unix system. A group is a collection of user accounts that share certain permissions and privileges. When you create a group, you can assign multiple users to it, allowing them to access shared resources and files. By creating a new group with the same Group ID (GID) as a group that had access to the certain directory, and adding a new user to that group, the added user gains the same group membership and access permissions as the original group. This allows the new user to access the specified directory and its contents, bypassing permission rights.

General Syntax:

### groupadd [options] group\_name

Syntax Options:

* -g, --gid GID 🡪 The numerical value of the groups ID (Group ID)

Default Commands:

* Create a new group with a specific GID

### $ sudo groupadd --gid [GID] [group\_name]

Challenge Names:

* Happycorp1 Vulnhub Walkthrough

### **useradd**

General Description:

Used to add a new user to the system, providing them with a unique user ID (UID) and the option to assign them to one or more existing groups, granting them access to specific resources and files.

Usage: see General Description of the preceding command groupadd

Default Commands:

* Create a new user (with specified UID) belonging to an existing group

### $ useradd uid [UID] --group [group\_name] [new\_username]

Challenge Names:

* Happycorp1 Vulnhub Walkthrough

### **swaks**

General Description:

Tool for sending and analyzing email messages providing extensive options for configuring SMTP parameters, message content, and delivery options. In CTF challenges, the command is commonly used to test email delivery and explore SMTP server configurations

Default Commands:

* Send a test email to a specified SMTP server with a defined mail body and subject header

### $ swaks --to [recipient\_email\_address] --from [sender\_email\_address] --server [SMTP\_IP\_address]:[port] – body ["default\_text”]  --header [“default\_text”]

Challenge Names:

* Hack Billy Madison Vm Ctf Challenge

### **fdisk**

General Description:

examine the partition table of a disk image or device, providing information about the disk's layout, partition types, and sizes to potentially identify specific information of interest. Often used to later perform disk partition-related operations (f.e. the following fls command)

General Syntax:

### fdisk [options] disk\_image\_file

Syntax Options:

* -l, --list 🡪 List the partition tables for the specified devices and then exit
* -u, --units[=unit] 🡪 When listing partition tables, show sizes (default = sectors)

Default Commands:

* Display the partition table of the specified disk image file

### $ fdisk -lu [disk\_image\_file]

Challenge Names:

* Hack Gibson Vm Ctf Challenge
* Hack The Teuchter Vm Ctf Challenge

### **fls**

General Description:

As part of the Sleuth Kit (TSK) suite, which is a collection of forensic analysis tools, it is commonly used to list information about the file and directory entries within a forensic image or a disk partition allwing to examine the file system structure, retrieve file metadata, and identify deleted or hidden files.

General Syntax:

### fls [options] image [inode]

Syntax Options:

* -f fstype 🡪 The type of file system
* -o imgoffset 🡪 The sector offset where the file system starts in the image

Default Commands:

* List the files and directories within a given file system forensic image or disk partition

### $ fls -f [file\_system\_type] -o [offset] [forensic\_image]

Challenge Names:

* Hack Gibson Vm Ctf Challenge

### **7z**

General Description:

Used for compressing and decompressing files

Default Commands:

* Extract the contents of a compressed file

### $ 7z e [path/to/compressed\_file]

* Extract the contents with full path of a compressed file

### $ 7z x [path/to/compressed\_file]

Challenge Names:

* Hack The Box Access Walkthrough
* Hack Game Thrones Vm Ctf Challenge

### **sockstat**

General Description:

List open Internet or UNIX domain sockets

Default Commands:

* View which users/processes are listening to which ports.

### $ sockstat -l

Challenge Names:

* Hack The Box Poison Walkthrough

### **wc**

General Description:

Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified

General Syntax:

### wc [OPTION]... [FILE]...

Syntax Options:

* -l, --lines 🡪 Print the newline counts

Default Commands:

* Count all lines in a file

### $ wc -l [path/to/file]

Challenge Names:

* Hack The Box Curling Walkthrough

### **df**

General Description:

Used to display the amount of space available on the file system containing each file name argument

Default Commands:

* Display all filesystems and their disk usage

### $ df

Challenge Names:

* Hack The Box Challenge Falafel Walkthrough

### **more**

General Description:

A filter used for paging through text one screenful at a time

Default Commands:

* Open a file

### $ more [path/to/file]

* Display the contents of the $DATA stream associated with a specified file

### $ more < [path/to/file]:$DATA

Challenge Names:

* Hack The Box Challenge Jeeves Walkthrough

### **psql**

General Description:

Command-line tool used to interact with PostgreSQL databases.

General Syntax:

### psql [option...] [dbname [username]]

Syntax Options:

* -h hostname, --host=hostname 🡪 Specifies the host name of the machine on which the server is running
* -U username, --username=username 🡪 Connect to the database as the user username instead of the default
* -d dbname, --dbname=dbname 🡪 Specifies the name of the database to connect to

Default Commands:

* Connect to a known database on given server host with given username

### $ psql -h [host] -U [username] -d [database\_name]

Challenge Names:

* Hack Game Thrones Vm Ctf Challenge

### **iptables**

General Description:

Used to set up, maintain, and inspect the tables of IPv4 and IPv6 packet filter rules in the Linux kernel. Can be utilized to gather information about the target network by examining its firewall rules and filtering configurations, granting insights into the network's security posture to identify open ports and services, and understand the network traffic flow.

Default Commands:

* List the current firewall rules configured in the iptables firewall

### $ iptables -L

Challenge Names:

* Hack Super Mario Ctf Challenge

### **chkrootkit**

General Description:

Command-line utility used for checking various system files, directories, and processes to look for common rootkit signatures, suspicious network connections, hidden processes and vulnerabilities. Some of its version are exploitable.

Default Commands:

* Display the version information of the chkrootkit tool

### $ chkrootkit -V

Challenge Names:

* Hack Sickos 1.2 Vm Ctf Challenge

### **unrar**

General Description:

Extract files from rar archives

Default Commands:

* Extract files with original directory structure

### $ unrar x [compressed.rar]

* Extract files into current directory, losing directory structure in the archive

### $ unrar e [compressed.rar]

Challenge Names:

* Hack Acid Reloaded Vm Ctf Challenge

### **foremost**

General Description:

Recover files from a disk image based on headers and footers specified by the user.

Default Commands:

* Recover files embedded within or associated with a specified file

### $ foremost [filename]

Challenge Names:

* Hack Acid Reloaded Vm Ctf Challenge

# Exploitation

### **smbclient**

General Description:

The command allows you to access and interact with SMB/CIFS resources. It enables you to access and manipulate files, folders, and resources on remote Windows or Samba servers.

General Syntax:

### smbclient [*servicename*] [*password*] [*OPTIONS*]

Servicename = Name of the service to use on the server

password = password required to access the specified service on the specified server

Syntax Options:

* - I IP number 🡪 represents the IP number of the server to connect to
* -L 🡪 lists the services that are available on the server
* -N 🡪 this parameter suppresses the normal password prompt from the client to the user. This is useful when accessing a service that does not require a password.
* -U username 🡪 Username that will be used by the client to make a connection

Default Commands:

* Connect with a username and password

### $ smbclient [//server/share] --user [username%password]

### **vi**

General Description:

Opens a text editor that allows to view and modify a file

Default Commands:

* Open a specified file in a text editor in write mode

### $ vi [path/to/file]

### **git**

General Description:

Version control system that helps manage and track changes to files and projects.

General Syntax:

### git <command> [<args>]

Syntax Options:

Command:

* <clone> 🡪 clone a repository into a new directory
* <clean> 🡪 remove untracked files from the working tree
* <log> 🡪 show commit logs
* <show> 🡪 show various types of objects
* <ls-tree> 🡪 list the contents of a tree object

Default Commands:

* Execute a Git subcommand

### $ git [subcommand]

* Clone a Git-Repository

### $ git clone https://github.com/repository.git

### **pip3**

General Description:

A tool for installing and managing Python packages

Default Commands:

* Install a package

### $ pip3 install [package\_name]

* Install packages from a file

### $ pip3 install --requirement [requirements.txt]

### **hashcat**

General Description:

Specialized password cracking tool designed to recover lost or forgotten passwords by utilizing the computational power of a system's GPU or CPU

Default Commands:

* Performing a password cracking attack on a hashfile using given hashtype and wordlist

### $ hashcat -m [hash-type-id] [hashfile] [wordlist.txt]

### **ldapsearch**

General Description:

Used to search and retrieve information from an LDAP (Lightweight Directory Access Protocol) server. It allows you to query an LDAP directory by specifying search criteria and search filters

Default Commands:

* Perform a simple authentication LDAP Search with given Distinguished Name, Password and Search Base

### $ ldapsearch -H [LDAP\_Server\_URL] -x -W -D [Authentication\_Name] -b [Search\_Base]

### **apt**

General Description:

The command provides a command-line interface to handle package-related operations, including installing, updating, and removing software packages on the system.

Default Commands:

* Install a package, or update it to the latest available version

### $ sudo apt install [package]

### **apt-get**

General Description:

This command provides another older, but still widely used, command-line interface for managing software packages on the system

Default Commands:

* Update the list of available packages and versions (run this before other apt-get commands)

### $ apt-get update

* Install a package, or update it to the latest available version

### $ apt-get install [package]

### **stegsnow**

General Description:

Used for concealing and extracting messages in text files encoded as tabs and spaces.

Default Commands:

* Extract [C]ompressed message from file and save to result file

### $ stegsnow -C [path/to/message.txt] > [result.txt]

### **nc / netcat**

General Description:

Used for reading from and writing to network connections. Can be used to establish network connections, interact with services, and perform various network-related tasks. It is particularly useful for tasks such as port scanning, service enumeration, exploiting vulnerable services, and creating reverse or bind shell connections

General Syntax:

### nc [-options] hostname port[s] [ports] ...

Syntax Options:

* -l 🡪 enables listen mode (listen for incoming connections)
* -v 🡪 enables verbose output (providing more detailed information about the connections and data transfer
* -p Port🡪 specifies the port number to listen for

Default Commands:

* Listen on a specified TCP port and print any data received

### $ nc -l [Port]

* Listen on a specified TCP port with verbose output and print any data received

### $ nc -lvp [Port]

### **socat**

General Description:

Establishes a bidirectional data relay between a source and a destination. It listens for incoming connections on a specified port and facilitates the transfer of data between the connected client and the designated endpoint

Default Commands:

* Listen to a port, wait for an incoming connection and transfer data to STDIO

### $ socat - TCP-LISTEN:8080,fork

Challenge Names:

* Maskcrafter 1 1 Vulnhub Walkthrough
* Hack the Box Wall Walkthrough

### **xfreedrp**

General Description:

Used to launch the FreeRDP client, which is an open-source remote desktop protocol (RDP) client application

General Syntax:

### xfreerdp [file] [options] [/v:server[:port]]

Default Commands:

* Connect to a FreeRDP server

### $ xfreerdp /u:[username] /p:[password] /v:[ip\_address]

### **lsb\_release**

General Description:

Used to ge tinformation about the Linux version on the target machine

General Syntax:

### lsb\_release [OPTION]...

Default Commands:

* Print all available information

### $ lsb\_release -a

Challenge Names:

* Hack The Lampiao 1 Ctf Challenge
* Grimtheripper 1 Vulnhub Walkthrough

### **msfconsole**

General Description:

Launch the interactive console, where you can interact with Metasploit's functionalities, including selecting and configuring exploits, setting up listeners, crafting payloads, and executing various security testing techniques

General Syntax:

### msfconsole [options]

Syntax Options:

* -r, --resource FILE 🡪 Execute the specified resource file

Default Commands:

* Entry the interactive console of the Metasploit

### $ msfconsole

* Run the commands and execute the specified resource file

### $ msfconsole -r [resource\_file]

### **set**

General Description:

Set or unset options and positional parameters

General Syntax:

### set [Optional \_Flags] [-o option] [argument...]

Default Commands:

* Setting the value of a environmental variable

### $ set [value] [variable]

### **hydra**

General Description:

Starts a parallelized login cracker which supports numerous protocols to attack able to perform password auditing, identify weak credentials and potentially exploit vulnerabilities in the target system. It provides options for specifying the target service, usernames, passwords, and other parameters required for the attack.

Default Commands:

* Guess SSH credentials using a given list of usernames and a list of passwords

### $ hydra -L [path/to/usernames.txt] -P [path/to/passwords.txt] [host\_ip] [ssh]

* Guess SSH credentials using a given list of usernames and a list of passwords using a specified port

### $ hydra -L [path/to/usernames.txt] -P [path/to/passwords.txt] [host\_ip] [ssh] -s [port]

### **tar**

General Description:

Used for creating, manipulating, and extracting files from tar archive.

Default Commands:

* E[x]tract a (compressed) archive [f]ile into the current directory [v]erbosely

### $ tar xvf [path/to/source.tar[.gz|.bz2|.xz]]

* E[x]tract a (compressed) archive [f]ile into the target directory

### $ tar xf [path/to/source.tar[.gz|.bz2|.xz]] --directory=[path/to/directory]

* [c]reate a g[z]ipped archive from a directory using relative paths with detailed output during the process

### $ tar cvzf [path/to/target.tar.gz] --directory=[path/to/directory]

### **gcc**

General Description:

Used for compiling C or C++ source code files into executable binaries. The gcc command allows to compile code and generate executable files that can be executed or tested against the challenge requirements.

Default Commands:

* Compile multiple source files into an executable

### $ gcc [path/to/source1.c path/to/source2.c ...] -o [path/to/output\_executable]

### **john**

General Description:

Used for password cracking as part of the John the Ripper password cracker tool. It is primarily used to perform offline password cracking by using various attack techniques.

Default commands:

* Crack password hashes, using a custom wordlist

### $ john --wordlist=[path/to/wordlist.txt] [path/to/password\_hashes]

### **zip2john**

General Description:

Used to extract password hashes from zip files into a format that is suitable for use with John the Ripper password cracker

Default Commands:

* Extract the password hash from a ZIP file to a specific file (for use with John the Ripper)

### $ zip2john [path/to/file.zip] > [file.hash]

### **ftp**

General Description:

Used to establish an FTP (File Transfer Protocol) connection with a remote server. Once connected, it allows to perform various file transfer operations between the local system and the remote server, such as uploading, downloading, listing directories, and managing files.

General Syntax:

### ftp [OPTION...] [HOST [PORT]]

Default Commands:

* Connect to an FTP server

### $ ftp [example\_IP\_address]

### **curl**

General Description:

Sends a HTTP requests, such as GET, POST, or PUT, to web servers. This enables them to retrieve web content, submit form data, or manipulate requests for specific purposes.

General Syntax:

### curl [options] [URL]

Default Commands:

* Download the contents of a URL and display the response in the command-line interface

### $ curl [http://example.com]

* Download the contents of a URL to a file

### $ curl [http://example.com] --output [path/to/file]

* Download specified contents of a URL bypassing certification verification

### $ curl -d "[parameter\_name]=[parameter\_value]" [URL] -k

Challenge Names:

* Mumbai:1 Vulnhub Walkthrough
* Casino Royale 1

### **steghide**

General Description:

Provides functionality to embed and extract hidden data from various types of files, particularly images

General Syntax:

### steghide command [arguments]

Syntax Options:

* -sf, --stegofile filename 🡪 Specify the stego file (the file that contains embedded data
* -xf, --extractfile filename 🡪 Create a file with the name filename and write the data that is embedded in the stego file to it

Default Commands:

* Extract hidden data from the stegofile save it as an extractfile

### $ steghide extract -sf [/path/to/input\_image\_file.jpg] -xf [/path/to/generated\_output\_file.txt]

### **wfuzz**

General Description:

Allows to automate the process of sending different requests with various payloads to different parts of a web application, such as URLs, parameters, headers, or cookies. It replaces specific parts of the request with payloads from wordlists or predefined patterns and analyzes the response received from the server to identify vulnerabilities, such as injection points or sensitive information disclosure.

Default Commands:

* Use a specified wordlist to fuzz the target URL

### $ wfuzz -w [path/to/wordlist] -u [target\_URL]

* Use a specified wordlist to fuzz subdomains while hiding specific response codes and word counts. Use increased threads and include the target ip/domain

### $ wfuzz -w [path/to/wordlist] -H ["Host: FUZZ.example.com"] --hc [301] --hw [222] -t [100] [example.com]

### **mysql**

General Description:

Used to connect to a MySQL database server and perform various database operations

General Syntax:

### mysql [options] db\_name

Default Commands:

* Connect to a specified MySQL server using a specific username and entering via a known password

### $ mysql -u [username] -h [Server\_IP\_address] -p

### **mysqldump**

General Description:

Used to create backups of MySQL databases or to extract data from MySQL databases into a file

General Syntax:

### mysqldump [options] [db\_name [tbl\_name ...]]

Default Commands:

* Create a backup (user will be prompted for a password)

### $ mysqldump --user [username] --password [database\_name] --backupfile=[path/to/file.sql]

### **crunch**

General Description:

Generates a wordlist or dictionary from a character set containing all possible combinations of characters within a specified length range

General Syntax:

### crunch <min-len> <max-len> [<charset string>] [options]

Default Commands:

* Generate a wordlist based on a predefined pattern/combination of characters in between minimum and maximum length. Append the wordlist to the wordlist file

### $ crunch [min\_len] [max\_len] -t [pattern] >> [wordlist.txt]

### **knock**

General Description:

Uses the knock tool to send a sequence of connection attempts to specific ports on a target host. This sequence is often used as a form of port knocking, which is a technique to secure network services by hiding them behind closed ports until a specific sequence of connection attempts is made. Once the correct knock sequence is received, the hidden services or ports are made accessible

Default Commands:

* Reveal open ports on a target system using specified port numbers

### $ knock -u [target\_IP\_address] [Port…]

### **pkttyagent**

General Description:

Used when dealing with challenges that require interaction with graphical applications that may require manual input. The command allows to interact with the application through the terminal, making it easier to analyze, provide input, or retrieve important information needed to solve the challenge

General Syntax:

### pkttyagent [--process { pid | pid,pid-start-time } | --system-bus-name busname] [--notify-fd fd] [--fallback]

Default Commands:

* Enables process interaction for graphical applications

### pkttyagent –process [process\_id]

### **strings**

General Description:

Checks for any human-readable strings inside the machine code and prints the printable character sequences that are at least 4 characters long (or the number given with the options)

Default Commands:

* Print all strings in a binary

### $ strings [path/to/file]

* Limit results to strings at least \*length\* characters long

### $ strings -n [length] [path/to/file]

* Prefix each result with its offset within the file in hexadecimal

### $ strings -t x [path/to/file]

### **msfvenom**

General Description:

Manually generate payloads for Metasploit that can be used to exploit specific vulnerabilities in target systems. These payloads can be crafted to deliver malicious code, gain unauthorized access, or take control of the compromised system

Default Commands:

* Generates a specific payload with the given payload type, local host IP, local port number, output format, and saves it to the specified output file

### msfvenom -p [Payload] lhost=[local\_host\_IP\_address] lport=[local\_Port] -f [format\_safe\_file] > [path/to/storage/file]

### **scp**

General Description:

Used for securely copying files between different hosts on a network. It utilizes the SSH (Secure Shell) protocol for data transfer, ensuring that the data remains encrypted and protected during transmission

Default Commands:

* Copy a file from a remote host to a local directory

### $ scp [remote\_host]:[path/to/remote\_file] [path/to/local\_directory]

Challenge Names:

* Hack The Box Challenge Bitlab Walkthrough
* Jigsaw1 Vulnhub Walkthrough

### **ldd**

General Description:

Prints the shared objects (shared libraries) required by each program or shared object specified on the command line. Analyzing the dependencies aids in reverse engineering, vulnerability identification, exploitation, and runtime behavior analysis

General Syntax:

### ldd [option]... file...

Default Commands:

* Display shared library dependencies of a executable file

### $ ldd [path/to/executable\_file]

Challenge Names:

* Jigsaw 1 Vulnhub Walkthrough
* Hack the Box Dab Walkthrough

### **fcrackzip**

General Description:

Searches each zipfile given for encrypted files and tries to guess the password

Default Commands:

* Crack the password of a ZIP file named "winrm\_backup.zip" using a dictionary attack with given wordlist

### $ fcrackzip -D -u [zip\_filename.zip] -p [path/to/wordlists.txt]

Challenge Names:

* Timelapse Hackthebox Walkthrough
* Ha Forensics Vulnhub Walkthrough

### **unix2dos**

General Description:

Used to convert plain text files in DOS or Mac format to Unix format and vice versa. Used when the analysis or manipulation of text files that have been formatted with Unix line endings is needed. However, some tools or systems expect files with Windows line endings. In such cases, unix2dos is used to convert the file's line endings to the appropriate format

Default Commands:

* Change the line endings of a file

### $ unix2dos [filename]

Challenge Names:

* Blackfield Hackthebox Walkthrough

### **ncat**

General Description:

Used to establish connections, send and receive data, and interact with remote systems over the network

General Syntax:

### ncat [OPTIONS...] [hostname] [port]

Default Commands:

* Establish a network connection between a (local) port and an (remote) IP address

### ncat -p [local\_port] [remote\_IP\_address]

Challenge Names:

* Hack the Box Vault Walkthrough

### **aircrack-ng**

General Description:

Used for analyzing and cracking Wi-Fi passwords

General Syntax:

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

Default Commands:

* Crack key from capture file using [w]ordlist

### $ aircrack-ng -w [path/to/wordlist.txt] [path/to/capture.cap]

Challenge Names:

* Tr0Ll 3 Vulnhub Walkthrough
* Hack The Box Olympus Walkthrough

### **stty**

General Description:

Print or set the options for a terminal. This helps to prevent interference or modification of input and output streams by disabling any special interpretation or processing of characters. This can be advantageous when dealing with binary data, interactive programs, or when precise control over input and output is required

Default Commands:

* Configure the terminal to operate in raw mode with echo disabled

### $ stty raw -echo

Challenge Names:

* Cengbox 1 Vulnhub Walkthrough

### **readelf**

General Description:

Display information about ELF files. Often used in combination with grep to extract certain information (indicated via a keyword) from specified files.

Default Commands:

* Extract information from a specified file (often from the specified libc library: /lib/i386-linux-gnu/libc.so.6) using a keyword to filter the output (containing the keyword)

### $ readelf -s [path/to/file] | grep [keyword]

Challenge Names:

* Jigsaw1 Vulnhub Walkthrough
* Hack the Box Frolic Walkthrough

### **mount**

General Description:

Mount a filesystem. Used to connect storage devices or network shares to specific directories in the file system, allowing access to their contents

General Syntax:

### mount [options] [source] [mount-point]

Syntax Options:

* -t, --types fstype 🡪 Indicate the filesystem type
* -r, --read-only 🡪 Mount the filesystem read-only

Default Commands:

* Mount a device to a directoy in read-onyl

### $ mount -r [path/to/device\_file] [path/to/target\_directory]

* Mount a device with specified filesystem type to a directory

### $ mount -t [filesystem\_type] [path/to/device\_file] [path/to/target\_directory]

Challenge Names:

* Kenobi Tryhackme Walkthrough
* Remote Hackthebox Walkthrough
* Hack The Box Challenge Bastion Walkthrough

### **chisel**

General Description:

Creates TCP tunnels allowing for remote access and port forwarding. It is employed to establish a reverse connection from the target machine to the attacker's machine, providing a way to bypass firewall restrictions and gain network access

Default Commands:

* Run a Chisel server listening to a specific port

### $ chisel server -p [server\_port]

Challenge Names:

* Anubis Hackthebox Walkthrough

### **evil-winrm**

General Description:

Commonly used as a tool to gain remote access to Windows machines. It provides an interactive shell session over WinRM, which f.e. allows for executing commands or exploring the file system.

Default Commands:

* Connect to a host with specified username and password

### $ evil-winrm -i [IP\_address] -u [username] -p [password]

* Connect to a host with specified NTLM hash of the user's password for authentication.

### $ evil-winrm -i [IP\_address] -u [username]-H '[ntml\_password\_hash]'

* Connect to a host using SSL/TLS encryption using a provided certification and private key file for secure communication

### $ evil-winrm -i [IP\_address] -c [path/to/certifcation\_file] -k [path/to/private\_key\_file] -S -r timelapse

Challenge Names:

* Worker Hackthebox Walkthrough
* Blackfield Hackthebox Walkthrough
* Timelapse Hackthebox Walkthrough

### **msfpc**

General Description:

Used to generate various types of basic Meterpreter payloads based on users choice via msfvenom

Default Commands:

* Generate a Metasploit payload for a target operating system

### $ msfpc [operating\_system] [IP\_address]

Challenge Names:

* Sunset Sunrise Vulnhub Walkthrough

### **drush**

General Description:

Command-line utility that is used to communicate with drupal CMS. Can be used to change/modify contents of a Drupal site to access information.

Default Commands:

* Change the password of a specified user in a Drupal site

### $ drush user-password [username] --password=[new\_password]

Challenge Names:

* Dc7 Vulnhub Walkthrough

### **crontab**

General Description:

Used to manage and schedule cron jobs (scheduled tasks that are executed automatically at specific time intervals or dates). Crontab can be used to schedule or modify cron jobs to perform specific actions at predetermined times, such as executing scripts, allowing to identify scheduled tasks that can be potentially abused or manipulated to gain unauthorized access or perform other malicious activitie

Default Commands:

* View a list of existing cron jobs for current user

### $ crontab -l

Challenge Names:

* Hack The Box Carrier Walkthrough

### **gdb**

General Description:

GDB is a debugger that enables to debug, analyze, and exploit binary executables by providing insights into their internal workings. This allows to understand the behavior of binary executables, identify vulnerabilities, and craft exploits

Default Commands:

* Debug an executable in quiete mode

### $ gdb -q [executable]

Challenge Names:

* Hack The Box Challenge Enterprises Walkthrough
* Hack The Box Frolic Walkthrough

### **cc**

General Description:

This command is an alias of gcc. It can be used as alternative to gcc

Default Commands:

* Compile a C source file into an executable file

### $ cc [c\_source\_filename] -o [output\_filename

Challenge Names:

* Hack The Golden Eye1 Ctf Challenge

### **vncviewer**

General Description:

Used to launch a VNC (Virtual Network Computing) viewer application, which allows you to connect to and interact with remote desktops using the VNC protocol

Default Command:

* Launch a VNC client, connecting to a specified VNC server using a known password

### $ vncviewer -passwd [example\_password] [VNC\_server\_IP\_address]:[port]

Challenge Names:

* Hack The Box Poison Walkthrough

### **medusa**

General Description:

Command-line tool used to launch brute-force attacks to discover valid credentials (such as usernames and passwords).

General Syntax:

### medusa [-h host|-H file] [-u username|-U file] [-p password|-P file] [-C file] -M module [OPTIONS]

Default Commands:

* Execute a brute force attempt against a specified server using a file containing usernames and a file containing passwords

### $ medusa -M [module\_name] -h [server\_IP\_address] -U [path/to/username\_file] -P [path/to/password\_file]

Challenge Names:

* Hack The Lampsecurity Ctf8 Ctf Challenge 2

# Privilege Escalation

### **adb**

General Description:

The adb(Android Debug Bridge) command is a versatile command-line tool that is primarily used for communicating with Android devices from a computer

Default Commands:

* Start a remote shell in the target emulator/device instance

### $ adb shell

* connect to a device via TCP/IP (default port=5555)

### $ adb connect [Host\_IP\_Address[:PORT]]

### **su**

General Description:

Allows commands to be run with a substitute user and group ID

General Syntax:

### su [options] [-] [*user* [*argument...*]]

Default Commands:

* Switch to superuser (requires the root password)

### $ su

* Execute a command as another user

### $ su - [username] -c "[command]"

### **pkexec**

General Description:

Execute a command as another user

Default Commands:

* Switch to the root user with a login shell (environment is set up as if the root user had logged in directly)

### pkexec su -l root

### **bash**

General Description:

Command language interpreter that executes commands read from the standard input or from a file. These commands are often utilized to gather information, manipulate files, interact with services, exploit vulnerabilities, and perform various tasks required to solve the given challenges.

General Syntax:

### bash [options] [command\_string | file]

Syntax Options:

* -i 🡪 if present, the shell is interactive
* -c 🡪 if present, then commands are read from the first non-option argument command\_string
* - l 🡪 Make bash act as if it had been invoked as a login shell

Default Commands:

* Execute exemplary commands (“ls -la”) in the bash shell

### bash -c [ls -la]

* Fetching contents from a file and searching for keywords in them

### bash -c [curl http://example.com/secret-data.txt | grep "password"]

### **chmod**

General Description:

Used to modify the permissions of files and directories in a Unix-like operating system. It stands for "change mode" and allows the user to specify the permissions for reading, writing, and executing a file or directory

General Syntax:

chmod [OPTION]... MODE[MODE]... FILE...

Default Commands:

* Set file permissions

### $ chmod [Octal\_Notation] [Filename]

### **getcap**

General Description:

Displays the name and capabilities (permission levels) of each specified file

General Syntax:

getcap [-v] [-n] [-r] [-h] filename [ ... ]

Syntax Options:

* -h 🡪 prints quick usage.
* -n 🡪 prints any non-zero user namespace root user ID value found to be associated with a file's capabilities.
* -r 🡪 enables recursive search.
* -v 🡪 display all searched entries, even if the have no file-capabilities.

Default Commands:

* Get capabilities for the given files

### getcap [path/to/example\_file1 path/to/example\_file2 ...]

* Recursively search and display the capabilities set on files in the root directory

### getcap -r / 2>/dev/null

### **export**

General Description:

Used to set environment variables. Environment variables are dynamic values that can affect the behavior and configuration of programs and the operating system. In CTF challenges, you may encounter situations where you need to set environment variables to control the execution of certain scripts or programs

Default Commands:

* Set a new environment variable

### $ export [VARIABLE]=[value]

* Search for executable files in the current directory when entering a command

### $ export PATH=.:$PATH

### **openssl**

General Description:

provides a wide range of functionalities related to SSL/TLS protocols, encryption, decryption, certificate management, and more.

Default Commands:

* Generate a hashed password using the MD5-based algorithm

### openssl passwd –1 –salt [salt-value] [example\_password]

### **ln**

General Description:

Creates symbolic links or hard links between files, commonly used for creating aliases, redirecting file paths, or hiding important files within different locations

Default Commands:

* Create a symbolic link to a file or directory

### $ ln -s [/path/to/file\_or\_directory] [path/to/symlink]

### **lxc**

General Description:

Used to manage (create, start, stop, and manage) Linux Containers

Default Commands:

* List images matching a string. Omit the string to list all images

### $ lxc image list [[remote]:][match\_string]

### **tail**

General Description:

Print the last 10 lines of each FILE to standard output. With more than one FILE, precede each with a header giving the file name.

General Syntax:

### tail [OPTION]... [FILE]...

Syntax Options:

* -n, --lines=[+]NUM 🡪 output the last NUM lines, instead of the last 10; or use -n +NUM to output starting with line NUM
* -f, --follow[={name|descriptor}] 🡪 output appended data as the file grows; an absent option argument means 'descriptor'

Default Commands:

* Show the last 10 lines of a given file

### tail [path/to/file]

* Display a selected number of last lines of a given file

### tail -n [NUM] [path/to/file]

* Continuously monitor and display new lines appended to a file

### tail -f [path/to/file]

### **docker**

General Description:

Provides a controlled and isolated environment to create, manipulate, and execute container instances, enabling efficient and portable deployment of applications across different systems and environments

General Syntax:

### docker [OPTIONS] COMMAND [ARG...]

Default Commands:

* Display the list of already downloaded images

### $ docker images

* Runs an interactive Docker container based in a given image, mounting source directory to target directory

### $ docker run -v [source\_directory]:[target\_directory] -i -t [image\_name]

### **which**

General Description:

Used to locate the path of an executable program or command within the operating system's directories. It helps participants determine the location of specific programs or commands, which can be valuable for solving challenges or exploiting vulnerabilities.

General Syntax:

### which [options] [program\_name]

Default Commands:

* Search the PATH environment variable and display the location of any matching executables

### $ which [executable]

### **redis**

General Description:

Provides a simple command-line interface to a Redis server allowing to perform operations and access the necessary information within the Redis infrastructure

General Syntax:

### redis-cli [options]

Default Commands:

* Connect to the local server

### $ redis-cli

* Connect to a remote server on the default port (6379) with a specified password

### $ redis-cli -h [host] -a [password]

### **touch**

General Description:

Used to create a new file or update the timestamp of an existing file. It is commonly used to create placeholder files or mark a specific event or action. The created files can be used as inputs for further Escalation techniques

General Syntax:

### touch [OPTION]... FILE...

Default Commands:

* Create a new placeholder or marker file named in the current directory

### $ touch [filename]

Challenge Names:

* DC 1 Vulnhub Walkthrough
* Hack the Box Networked Walkthrough

### **cpulimit**

General Description:

Limit the CPU usage of a process. This is used to observe and exploit timing-based vulnerabilities in order to extract sensitive information or gain unauthorized access.

General Syntax:

### cpulimit [TARGET] [OPTIONS...] [ -- PROGRAM]

Syntax Options:

* -l, --limit=N 🡪 percentage of CPU allowed from 1 up (usually 1 – 100)
* -f, --foreground 🡪 run cpulimit in foreground while waiting for launched process to finish

Default Commands:

* Launch a given program and limit it to only use a certain percentage of the CPU

### $ cpulimit --limit [specified\_percentage\_limitation] -- [program arg1 arg2 ...]

Challenge Names:

* Shelldredd 1 Hannah Vulnhub Walkthrough

### **ssh-keygen**

General Description:

Generates, manages, and converts authentication keys for ssh. Used to for Key-based attacks or to authenticate securely without relying on passwords. The generated public key can be provided to the challenge server, enabling passwordless access and making authentication more convenient.

Default Commands:

* Generate a key interactively

### $ ssh-keygen

* Specify file in which to save the key

### $ ssh-keygen -f [filename]

* Generate a key with specified type of the key

### $ ssh-keygen -t [key\_type] -f [filename]

Challenge Names:

* Matrix 3 Vulnhub Walkthrough
* Pandora Hack the Box
* Hack the Box Postman Walkthrough

### **perl**

General Description:

Used to run Perl scripts or execute Perl code directly from the command line

Default Commands:

* Parse and execute a Perl statement

### $ perl -e [perl\_statement]

Challenge Names:

* Hack The Box Challenge Shocker Walkthrough

### **nohup**

General Description:

U to run a command or script in the background and keep it running even after the current shell session has ended. This allows long-running processes or tasks to continue execution uninterrupted, even when the user logs out or closes the terminal.

Default Commands:

* Run a process and write the output to a specific file

### $ nohup [command] [argument1 argument2 ...] > [path/to/output\_file]

Challenge Names:

* Victim1 Vulnhub Walkthrough

### **gdbus**

General Description:

Allows to interact with D-Bus (Desktop Bus) systems, which are message bus systems used for interprocess communication on Linux-based systems. It enables you to perform various operations related to D-Bus, such as listing available services, querying or monitoring bus objects, invoking methods, and more

Default Commands:

* Invokes a specified method on the destinated D-Bus service with given object path, using a source file and copying its contents to a destination file

### $ gdbus call [--system | --session] --dest [bus\_name] --object-path [path/to/object] --method [method\_name] [source\_file] [destination\_file]

Challenge Names:

* Passage Hackthebox Walkthrough

### **tmux**

General Description:

Enables a number of terminals to be created, accessed, and controlled from a single screen. Used to attach to an existing tmux session and gain root access, allowing privileged operations within the session, such as accessing sensitive files

Default Commands:

* Attach to a tmux session using a specific socket file

### $ tmux -S [path/to/socket\_file] attach

Challenge Names:

* Panabee 1 Vulnhub Walkthrough
* Hack The Box Valentine Walkthrough

### **mkfifo**

General Description:

Used to create a special type of file called a FIFO (First-In-First-Out) or named pipe that allows interprocess communication between different processes or programs. Acting as a bridge, this is often used to establish communication channels or exploit specific vulnerabilities that involve the exchange of data between processes.

General Syntax:

### mkfifo [OPTION]... NAME...

Default Commands:

* Create a named pipe for data transfer between processes

### $ mkfifo [path/to/created\_fifo\_file]; nc [IP\_address] [port] 0<[path/to/created\_fifo\_file]

Challenge Names:

* Dc:7 Vulnhub Walkthrough

### **ldconfig**

General Description:

Used to update the cache of shared libraries on the system. This ensures that newly created shared library are recognized and accessible by other programs. This is important when the shared library needs to be loaded and used by other binaries or processes.

Default Commands:

* Update the system's cache of shared libraries

### $ ldconfig

Challenge Names:

* Hack The Box Dab Walkthrough

### **keepass2**

General Description:

Used to open and access KeePass password database files in order to manage and retrieve stored passwords and other sensitive information

Default Commands:

* Open a specified KeePass database file (Masterkey must be known)

### $ keepass2 [keepass\_filename]

Challenge Names:

* Hack The Box Challenge Jeeves Walkthrough
* Hack The Box Challenge Tally Walkthrough

### **chown**

General Description:

* Change the file ownership. This is often used to elevate privileges and gain root access

General Syntax:

### chown [OPTION] [OWNER][:[GROUP]] FILE...

Default Commands:

* Change the ownership of a specified file to the user "root" and the group "root"

### $ chown root:root [filename]

Challenge Names:

* Goodgames Hackthebox Walkthrough
* Hack Orcus Vm Ctf Challenge

### **exec**

General Description:

Execute commands and open, close, or copy file descriptors. Used to replace the current running process with a new command or program. It terminates the current process and starts the specified command in its place

General Syntax:

### exec [command [argument...]]

Default Commands:

* Replace with the specified command, clearing environment variables

### $ exec -c [command -with -flags]

Challenge Names:

* Antique Hackthebox Walkthrough
* Knife Hackthebox Walkthrough

### **doas**

General Description:

Execute a command as another user. Allows running a command with different privileges

Default Commands:

* Run a command as root

### $ doas [command]

* Run a command as another [u]ser on a specified file

### $ doas -u [user] [command] [path/to/file]

Challenge Names:

* Luanne Hackthebox Walkthrough
* Fourandsix 2 Vulnhub Walkthrough

### **gem**

General Description:

Package manager for Ruby libraries and programs used to install, manage, and interact with RubyGems, allowing users to easily install and manage Ruby dependencies and software packages.

Default Commands:

* Install the latest version of a gem (package or library in Ruby programming language)

### $ gem install [gemname]

Challenge Names:

* Dailybugle Tryhackme Walkthrough
* Remote Hackthebox Walkthrough

### **screen**

General Description:

Allows multiple virtual terminal sessions within a single terminal window or SSH session. This provides a way to create, manage, and switch between multiple shell sessions, known as "screens." Commonly used to keep processes running in the background even after disconnecting from a remote server

General Syntax:

### screen [ options ] [ cmd [ args ] ]

Syntax Options:

* -r 🡪 reattach to a session
* -D -m 🡪 starts screen in detached mode (allows it to run in the background)
* -L 🡪 turn on automatic output logging for the windows

Default Commands:

* Reattach to an open screen

### $ screen -r [session\_name

* Show open screen sessions

### $ screen -ls

* Execute a command within a detached screen session

### $ screen -D -m -L [command]

Challenge Names:

* Backdoor Hackthebox Walkthrough
* Hack The Wintermute 1 Ctf Challenge

### **timeout**

General Description:

Run a command with a time limit. This is useful in scenarios where you need to control the execution time of potentially malicious or resource-intensive commands, preventing them from running indefinitely and potentially causing harm

Default Commands:

* Run a command with a given time limit

### $ timeout [time\_limit] [command]

Challenge Names:

* Five86 2 Vulnhub Walkthrough

### **debugfs**

General Description:

Used to examine and change the state of an ext2, ext3, or ext4 file system. This provides low-level access to the file system and allows to examine and modify file system structures, such as inodes and directories. The command is likely used to access and retrieve data from the file system that may not be accessible through regular mean

Default Commands:

* Open the filesystem in read only mode

### $ debugfs [/dev/sdXN]

Challenge Names:

* Hack The Box Challenge Falafel Walkthrough

### **gpg**

General Description:

Used for encryption, decryption, signing, and verification of files in the OpenPGP format

General Syntax:

### gpg [--homedir dir] [--options file] [options] command [args]

Syntax Options:

* -d, --decrypt 🡪 Decrypt the given file and write it to standard output
* --batch 🡪 Use batch mode (suppresses most interactive prompts and assumes default or predefined options for key operations)
* --passphrase string 🡪 Use string as the passphrase (= a secret passphrase or password which is required to unlock the private key)

Default Commands:

* Decrypt a specified file (output to stdout)

### $ gpg - [filename.txt.gpg]

* Decrypt a specified file using a known passphrase in batch mode

### $ gpg --batch –passphrase [passphrase] -d [filename.txt.gpg]

Challenge Names:

* Hack The Box Vault Walkthrough
* Hack The Bob 1 0 1 Vm Ctf Challenge

### **expose**

General Description:

Often used in the context of disk shadow copy manipulation. Thereby the command is used to mount or expose a disk shadow copy as a separate drive, allowing access and interaction with its contents as if it were a distinct storage location

Default Commands:

* Expose / mounts a disk shadow copy volume with an alias as a new drive letter

### $ expose %[alias]% [drive\_letter]:

Challenge Names:

* Blackfield Hackthebox Walkthrough

### **run-parts**

General Description:

Runs all the executable files named (within constraints), found in the specified directory. Other files and directories are silently ignored

General Syntax:

### run-parts [options] directory

Default Commands:

* Execute all scripts located in the specified directory and store the output in a file

### $ run-parts --lsbsysinit [path/to/directory] > [path/to/storage\_file]

Challenge Names:

* Hack The Box Writeup Walkthrough

# Additional found Commands

### **crackmapexec (Exploitation)**

General Description:

The command is a powerful penetration testing tool used for security assessments and password cracking in Windows networks. It allows you to perform various network-related tasks, such as credential harvesting, privilege escalation, and lateral movement within a network.

General Syntax:

### crackmapexec [*OPTIONS*] <*target(s)>*

Syntax Options:

* -u: Specify a username or a list of usernames to use for authentication.
* -p: Specify a password or a list of passwords to use for authentication.
* -H: Specify a hash or a list of hashes to use for authentication.
* -M: Specify the module or modules to execute.
* -o: Specify an output file to save the results.
* -d: Specify the domain name to target.
* -R: Use a random username or password from the specified list.
* -L: Load usernames or passwords from a file.
* -x: Specify the target protocol (e.g., SMB, LDAP).
* -s: Specify the target port.
* -v: Enable verbose output.
* -h: Display the help menu with a list of available options.