

## SERVEUR FAILLE

Machine virtuelle qui est capable de faire tout cela :

- Tester les failles SQL
- Tester les failles XSS
- Tester des attaques brute force
- Gestionner des droits d'accès et élévation de privilèges
- Tester les failles local et remote file inclusion (RFI et LFI)
- Tester les failles full path disclosure
- Tester les failles CRLF.

Pour cela je dois installer des logiciels que l'on me donne et que je dois rechercher. Voici les logiciels :

- THC-HYDRA
- Burpsuite
- Médusa
- Metasploit

J'ai trouvé ces différents logiciels pour tester les différentes failles :

- Nmap/Ncat/Ndiff
- Nikto
- Dirbuster

### **THC-HYDRA :**

- Sudo apt install hydra
- Sudo apt install hydra-gtk

Pour les commandes :

Doc THC-HYDRA : <https://www.kali.org/tools/hydra/#hydra-1>

<https://www.securiteinfo.com/attaques/hacking/outils/thc-hydra.shtml>

<https://hackertarget.com/brute-forcing-passwords-with-ncrack-hydra-and-medusa/>

Info sur THC-HYDRA :

Cet outil est un code de validation technique, qui permet aux chercheurs et aux consultants en sécurité de montrer à quel point il est facile d'obtenir un accès non autorisé depuis un système distant.

```
(user@VmFailles)-[~]
$ sudo apt install hydra
[sudo] Mot de passe de user :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
hydra est déjà la version la plus récente (9.4-1).
hydra passé en « installé manuellement ».
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
```

```

(user@VmFailles)~$ dpl4hydra -h
dpl4hydra v0.9.9 (c) 2012 by Roland Kessler (@rokkessler)

Syntax: dpl4hydra [help] | [refresh] | [BRAND] | [all]

This script depends on a local (d)efault (p)assword (l)ist called
/home/user/.dpl4hydra/dpl4hydra_full.csv. If it is not available, regenerate it with
'dpl4hydra refresh'. Source of the default password list is
http://open-sez.me

Options:
  help      Help: Show this message
  refresh   Refresh list: Download the full (d)efault (p)assword (l)ist
            and generate a new local /home/user/.dpl4hydra/dpl4hydra_full.csv file. Takes time!
  BRAND     Generates a (d)efault (p)assword (l)ist from the local file
            /home/user/.dpl4hydra/dpl4hydra_full.csv, limiting the output to BRAND systems, using
            the format username:password (as required by THC hydra).
            The output file is called dpl4hydra_BRAND.lst.
  all       Dump list of all systems credentials into dpl4hydra_all.lst.

Example:
# dpl4hydra linksys
File dpl4hydra_linksys.lst was created with 20 entries.
# hydra -C ./dpl4hydra_linksys.lst -t 1 192.168.1.1 http-get /index.asp

(user@VmFailles)~$ hydra -h
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Syntax: hydra [[-l LOGIN]-L FILE] [-p PASS]-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE] [-T TASKS]
] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuVd46] [-m MODULE_OPT] [service://server[:
PORT]/[OPT]]

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

Supported services: adam500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[s]{-head|get|post} ht
tp[s]{-get|post}-form http-proxy http-proxy-urldenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] memcached m
ongodb mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin
rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)
These services were not compiled in: afp ncp oracle sapr3 smb2.

Use HYDRA_PROXY_HTTP or HYDRA_PROXY environment variables for a proxy setup.
E.g. % export HYDRA_PROXY=socks5://l.p@127.0.0.1:9150 (or: socks4:// connect://)
% export HYDRA_PROXY=connect_and_socks_proxylist.txt (up to 64 entries)
% export HYDRA_PROXY_HTTP=http://login:pass@proxy:8080

```

## Metasploit :

- sudo apt install metasploit-framework
- sudo /etc/init.d/postgresql start
- sudo /etc/init.d/postgresql status
- curl <https://raw.githubusercontent.com/rapid7/metasploit-omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb>> msfinstall && chmod 755 && msfinstall && ./msfinstall

- msfconsole -q

Doc pour les commandes :

<https://www.fossilinux.com/48112/install-metasploit-kali-linux.html>

<https://www.it-connect.fr/chapitres/utilisation-de-metasploit/>

Info sur Metasploit : Metasploit est un framework libre d'exploitation de vulnérabilités facilitant la pré-exploitation (recherche de bugs, écriture d'exploits ou de shellcodes, ...), l'exploitation (envoi de l'exploit) et la post-exploitation (exécution de code arbitraire, accès à des fichiers, injection de serveur VNC, ...).

### **Nikto :**

- Sudo apt install nikto

( Faire la commande nikto -h pour voir les commandes qui peuvent être utilisé )

Doc : <https://memo-linux.com/nikto-outil-scanner-de-securite-serveur-web/>

<https://github.com/sullo/nikto/wiki/Basic-Testing>

Commande : <https://github.com/sullo/nikto/wiki/Annotated-Option-List>

Infos : Nikto: Un scanner de vulnérabilités web qui peut détecter les failles de sécurité courantes, y compris les failles de script, les inclusions de fichiers, les erreurs CRLF, etc.

### **Nmap/Ncat/Ndiff/nmap-common :**

- Sudo apt install nmap
- Sudo apt install ncat
- Sudo apt install ndiff
- Sudo apt install nmap-common

Pour savoir les commandes il faut faire n... -h

Doc :

<https://www.kali.org/tools/nmap/>

<https://nmap.org/man/fr/>

<https://geekflare.com/fr/nmap-on-linux/>

<https://hackertarget.com/brute-forcing-passwords-with-ncrack-hydra-and-medusa/>

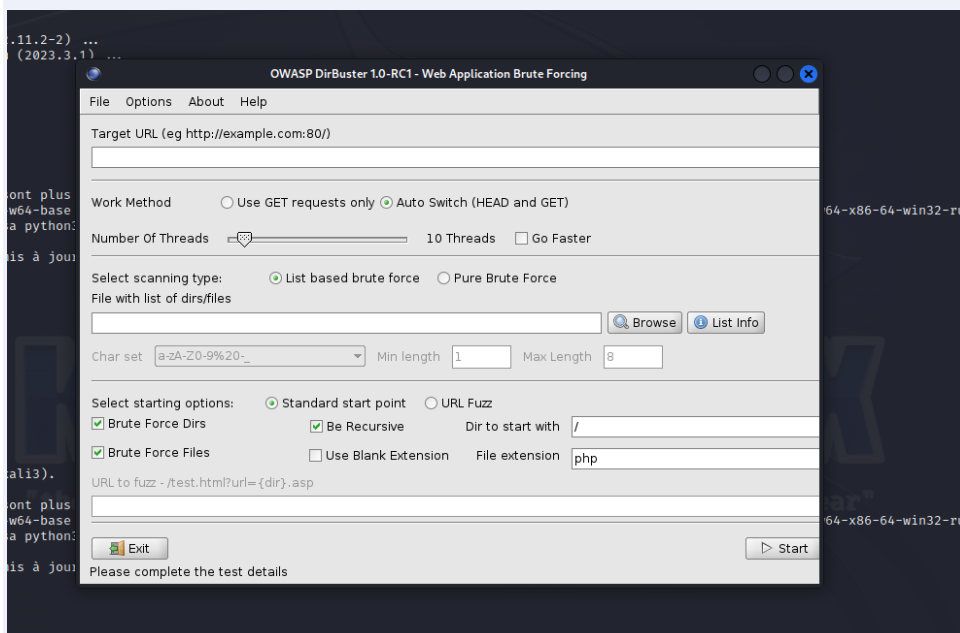
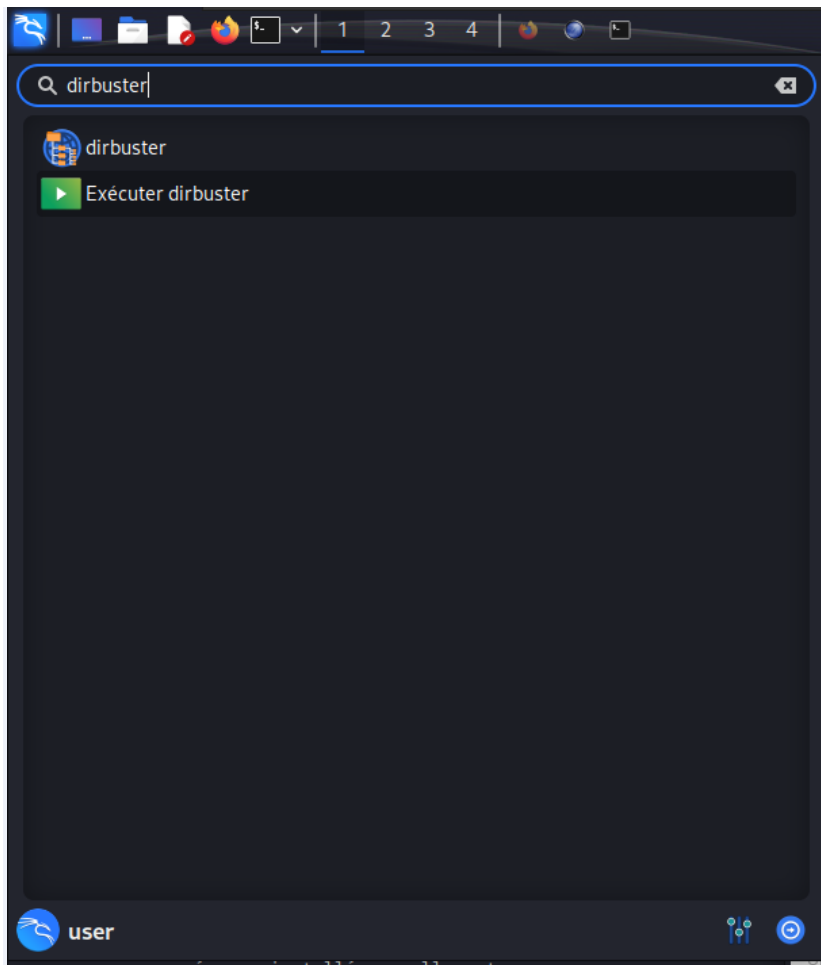
Infos : Un scanner de ports et de vulnérabilités qui peut également aider à identifier les failles de sécurité.

### **Dirbuster :**

- Sudo apt-get install dirbuster

Doc et infos : <https://ourcodeworld.com/articles/read/417/how-to-list-directories-and-files-of-a-website-using-dirbuster-in-kali-linux>

<https://hackfest.ca/blog/2012/backtrack-101-2-dirbuster>



## Medusa :

- Sudo apt install medusa

Doc Medusa : <https://www.kali.org/tools/medusa/>  
<https://en.kali.tools/?p=200>  
<https://hackertarget.com/brute-forcing-passwords-with-ncrack-hydra-and-medusa/>

Pour toute les commandes voir ici : <http://foofus.net/goons/jmk/medusa/medusa.html>

Info sur Medusa :

C'est un outil qui peut fonctionner à grande vitesse en tant que force brute des éléments de connexion dans un système.

Le but de Medusa est de prendre en charge autant de protocoles et de services que possible qui prennent en charge l'authentification à distance (par exemple ssh). Certains des avantages de cette application sont résumés ci-dessous:

- **Utilisation parallèle:** Le forçage brutal peut être effectué sur plusieurs hôtes, utilisateurs ou mots de passe en même temps.
- **Ela flexibilité:** Les informations sur la cible (hôte / utilisateur / mot de passe) peuvent être identifiées de différentes manières.
- **Prise en charge multiprotocole:** Medusa peut prendre en charge différents services et protocoles (par exemple SMP, [HTTP](#), [POP3](#), [MS-SQL](#), [SSHv2](#) etc.)

```
(user@VmFailles)-[~]
$ medusa -h
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

medusa: option requires an argument -- 'h'
CRITICAL: Unknown error processing command-line options.
ALERT: Host information must be supplied.

Syntax: Medusa [-h host|-H file] [-u username|-U file] [-p password|-P file] [-C file] -M module [OPT]
-h [TEXT]      : Target hostname or IP address
-H [FILE]      : File containing target hostnames or IP addresses
-u [TEXT]      : Username to test
-U [FILE]      : File containing usernames to test
-p [TEXT]      : Password to test
-P [FILE]      : File containing passwords to test
-C [FILE]      : File containing combo entries. See README for more information.
-O [FILE]      : File to append log information to
-e [n/s/ns]    : Additional password checks ([n] No Password, [s] Password = Username)
-M [TEXT]      : Name of the module to execute (without the .mod extension)
-m [TEXT]      : Parameter to pass to the module. This can be passed multiple times with a
                  different parameter each time and they will all be sent to the module (i.e.
                  -m Param1 -m Param2, etc.)
-d             : Dump all known modules
-n [NUM]       : Use for non-default TCP port number
-s            : Enable SSL
-g [NUM]       : Give up after trying to connect for NUM seconds (default 3)
-r [NUM]       : Sleep NUM seconds between retry attempts (default 3)
-R [NUM]       : Attempt NUM retries before giving up. The total number of attempts will be NUM + 1.
-c [NUM]       : Time to wait in usec to verify socket is available (default 500 usec).
-t [NUM]       : Total number of logins to be tested concurrently
-T [NUM]       : Total number of hosts to be tested concurrently
-L            : Parallelize logins using one username per thread. The default is to process
                  the entire username before proceeding.
-f            : Stop scanning host after first valid username/password found.
-F            : Stop audit after first valid username/password found on any host.
-b            : Suppress startup banner
-q            : Display module's usage information
-v [NUM]       : Verbose level [0 - 6 (more)]
-w [NUM]       : Error debug level [0 - 10 (more)]
-V            : Display version
-Z [TEXT]      : Resume scan based on map of previous scan

(user@VmFailles)-[~]
$
```

## **Burpsuite :**

- Sudo apt install burpsuite
- Pour lancer l'application tapez : burpsuite

Doc Burpsuite : <https://www.kali.org/tools/burpsuite/>  
<https://docs.bluekeys.org/guide/cyber-securite/outils/burpsuite-introduction>

### Info sur Burpsuite :

Cet outil est le logiciel incontournable pour auditer une application web, car il répond au besoin premier d'un professionnel de l'audit : accéder aux échanges entre le navigateur et le serveur web, afin de comprendre l'architecture et le fonctionnement de la solution à auditer. Grâce à ses différentes fonctionnalités configurables facilement, il est le couteau suisse d'un pentester.

```
(user@VmFailles)-[~]
$ burpsuite --help
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Usage:
--help                Print this message
--version             Print version details
--disable-extensions  Prevent loading of extensions on startup
--diagnostics         Print diagnostic information
--use-defaults        Start with default settings
--collaborator-server Run in Collaborator server mode
--collaborator-config Specify Collaborator server configuration file; defaults to collaborator.config
--data-dir            Specify data directory
--project-file        Open the specified project file; this will be created as a new project if the file does not exist
--developer-extension-class-name Fully qualified name of locally-developed extension class; extension will be loaded from the classpath
--config-file         Load the specified project configuration file(s); this option may be repeated to load multiple files
--user-config-file    Load the specified user configuration file(s); this option may be repeated to load multiple files
--auto-repair         Automatically repair a corrupted project file specified by the --project-file option
--unpause-spider-and-scanner Do not pause the Spider and Scanner when opening an existing project
--disable-auto-update Suppress auto update behavior

(user@VmFailles)-[~]
```