

# **EJERCICIO FEEDBACK. MUTACIÓN DE DICCIONARIOS. GENERACIÓN DE CONTRASEÑAS VÁLIDAS CON CUPP.**

**Presentado por:  
Santiago Peñaranda Mejia**

## Descarga de Cupp

Vamos a iniciar con la descarga de Cupp para ello vamos a escribir el comando **git clone** seguido de la URL de la herramienta para clonarla

```
(root@kali)-[/home/kali]
# git clone https://github.com/Mebus/cupp.git
```

Una vez ya la tengamos descargada vamos a entrar en la carpeta de la siguiente manera

```
(root@kali)-[/home/kali]
# ls
AndroRAT  cupp      Desktop  Documents  Impulse  Pictures  PyPhisher  Templates  Videos
arp-scan  CVE-2018-15473  diccionario  Downloads  Music    Public    SecLists   thc-hydra

(root@kali)-[/home/kali]
# cd cupp

(root@kali)-[/home/kali/cupp]
# ls
CHANGELOG.md  cupp.cfg  cupp.py  hydra.restore  LICENSE  README.md  screenshots  test_cupp.py

(root@kali)-[/home/kali/cupp]
```

Ahora vamos a ver los parámetros de la herramienta, para ello escribimos el comando **python3 cupp.py -h**

```
(root@kali)-[/home/kali/cupp]
# python3 cupp.py -h
usage: cupp.py [-h] [-i] [-w FILENAME] [-l] [-a] [-v] [-q]

Common User Passwords Profiler

options:
  -h, --help            show this help message and exit
  -i, --interactive      Interactive questions for user password profiling
  -w FILENAME            Use this option to improve existing dictionary, or WyD.pl output to make some pwnsauc
  -l                    Download huge wordlists from repository
  -a                    Parse default usernames and passwords directly from Alecto DB. Project Alecto uses purified databases of
                        Phenoelit and CIRT which were merged and enhanced
  -v, --version          Show the version of this program.
  -q, --quiet            Quiet mode (don't print banner)
```

Ahora vamos a ejecutar la herramienta con el parámetro **-i** así como vez en la imagen

```
(root@kali) [/home/kali/cupp]
# python3 cupp.py -i

cupp.py!

# Common
# User
# Passwords
# Profiler

[ Muris Kurgas | j0rgan@remote-exploit.org ]
[ Mebus | https://github.com/Mebus/ ]

[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: 
```

## Mutación del diccionario

Ahora vamos a rellenar el cuestionario con información de la víctima, en este caso a máquina metasploitable 2

```
(root@kali) [/home/kali/cupp]
# python3 cupp.py -i

cupp.py!

# Common
# User
# Passwords
# Profiler

[ Muris Kurgas | j0rgan@remote-exploit.org ]
[ Mebus | https://github.com/Mebus/ ]

[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: msf
> Surname: admin
> Nickname: metasploitable
> Birthdate (DDMMYYYY): 10/02/2000

[-] You must enter 8 digits for birthday!
> Birthdate (DDMMYYYY): 1002200
[-] You must enter 8 digits for birthday!
> Birthdate (DDMMYYYY): 10022000

> Partners) name: msfdev
> Partners) nickname:
> Partners) birthdate (DDMMYYYY):
```

```

> Partners) name: msfdev
> Partners) nickname:
> Partners) birthdate (DDMMYYYY):

[+] Now make the dictionary...

> Child's name:
> Child's nickname:
> Child's birthdate (DDMMYYYY):

[+] Now make the dictionary...

> Pet's name: Scott
> Company name: metasploit.com [i.e. from profile]

[+] Now make the dictionary...

> Do you want to add some key words about the victim? Y/[N]: y
> Please enter the words, separated by comma. [i.e. hacker,juice,black], spaces will be removed: msf admin
> Do you want to add special chars at the end of words? Y/[N]:
> Do you want to add some random numbers at the end of words? Y/[N]:
> Leet mode? (i.e. leet = 1337) Y/[N]:

[+] Now making a dictionary... [i.e. in interactive]
[+] Sorting list and removing duplicates...
[+] Saving dictionary to msf.txt, counting 2510 words.
> Hyperspeed Print? (Y/n) : n
[+] Now load your pistolero with msf.txt and shoot! Good luck!

(root@kali)-[/home/kali/cupp]
#

```

Una vez terminado de llenar el formulario con la información de la víctima en este caso la máquina **metasploitable 2**, nos va a dejar un diccionario de la información proporcionada y nos dice la cantidad de palabras que contiene, en este caso contiene **2510** palabras

```

[+] Now making a dictionary... [i.e. in interactive]
[+] Sorting list and removing duplicates...
[+] Saving dictionary to msf.txt, counting 2510 words.
> Hyperspeed Print? (Y/n) : n
[+] Now load your pistolero with msf.txt and shoot! Good luck!

```

## Verificación del diccionario

Ahora vamos a comprobar si la herramienta ha sido capaz de generar la contraseña válida que necesitamos, para ello escribimos el siguiente comando **cat diccionario generado | grep "contraseña"**

```

# cat msf.txt | grep msfadmin
msfadmin01sm_2015
msfadmin0 fsm_2016
msfadmin00 sm_2017
msfadmin000 m_2018
msfadmin002 m_2019
msfadmin010 m_2020
msfadmin02
msfadmin020 root@kali: [/home/kali/cupp]
msfadmin022 /home/cupp.py
msfadmin10 e: cupp.py [-h] [-i] [-w FILENAME]
msfadmin100

```

Y si efectivamente la comprobación fue válida, la herramienta fue capaz de generar la contraseña que necesitábamos que es **"msfadmin"** y así podemos ver la **efectividad** de Cupp.

