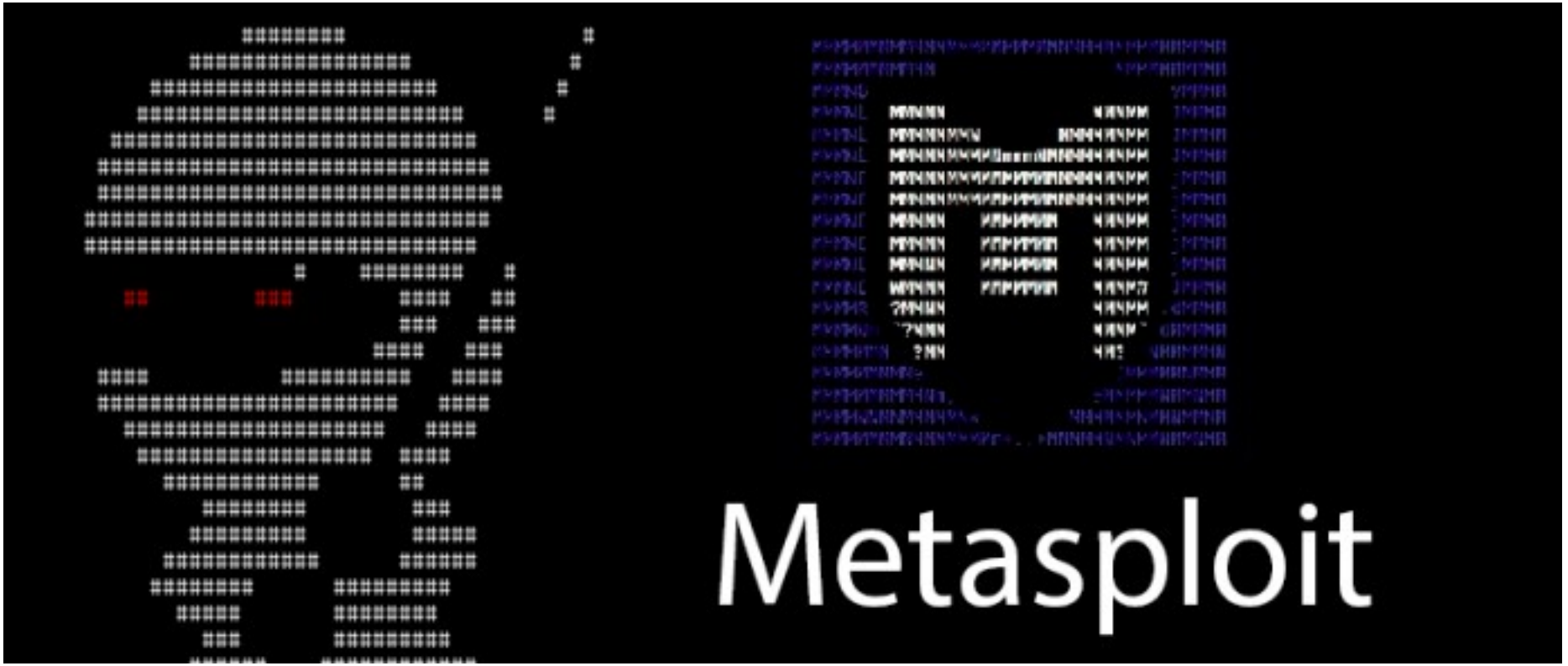
Metasploit



Índice de contenido

Introducción

- Proyecto Open Source (Metasploit Framework)
- Asistencia en pruebas de pentesting:
  - escaneo, enumeración, explotación, organización de la información, post-explotación, ...
- Desarrollado y mantenido por Rapid 7.
  - Metasploit Framework (versión libre).
  - Metasploit Pro (versión comercial).
- Escrito en Ruby.

https://www.rapid7.com/products/metasploit/

https://www.metasploit.com

Introducción

- Metasploit Framework vs Metasploit Pro

All Features	Pro	Framework
Collect		
De-facto standard for penetration testing with more than 1,500 exploits	✓	✓
Import of network data scan	✓	✓
Network discovery	✓	✗
Basic exploitation	✓	✗
MetaModules for discrete tasks such as network segmentation testing	✓	✗
Integrations via Remote API	✓	✗

https://www.rapid7.com/products/metasploit/download/editions/

Introducción

- Metasploit Framework vs Metasploit Pro

All Features	Pro	Framework
Automate		
Simple web interface	✓	✗
Smart Exploitation	✓	✗
Automated credentials brute forcing	✓	✗
Baseline penetration testing reports	✓	✗
Wizards for standard baseline audits	✓	✗
Task chains for automated custom workflows	✓	✗
Closed-Loop vulnerability validation to prioritize remediation	✓	✗

https://www.rapid7.com/products/metasploit/download/editions/

Introducción

- Metasploit Framework vs Metasploit Pro

All Features	Pro	Framework
Infiltrate		
Basic command-line interface	✗	✓
Manual exploitation	✗	✓
Manual credentials brute forcing	✗	✓
Dynamic payloads to evade leading anti-virus solutions	✓	✗
Phishing awareness management and spear phishing	✓	✗
Web app testing for OWASP Top 10 vulnerabilities	✓	✗
Choice of advance command-line (Pro Console) and web interface	✓	✗

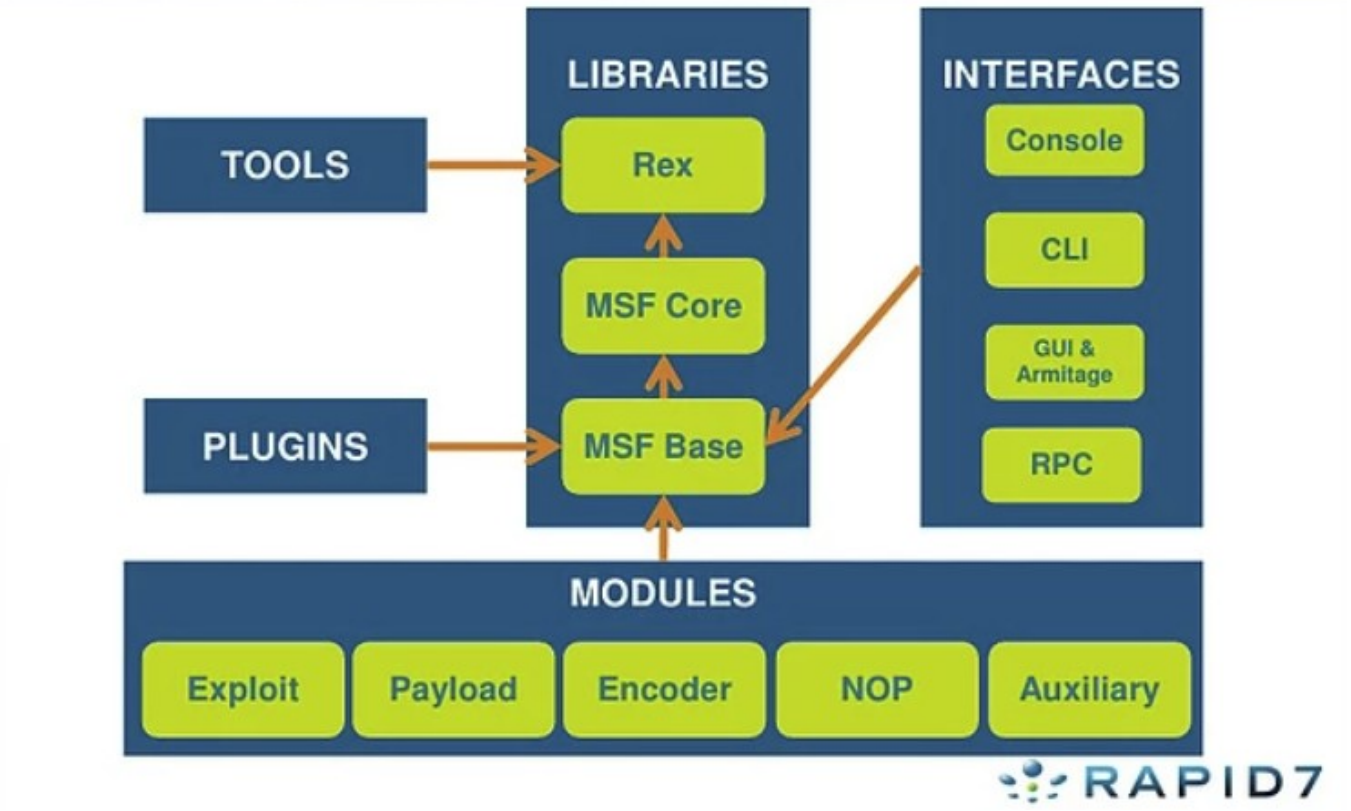
https://www.rapid7.com/products/metasploit/download/editions/

Introducción

- **Interfaces de usuario**.
  - ***msfconsole***. Interfaz en línea de comandos con numerosas opciones. Es la herramienta principal que usaremos. La versión pro dispone de una consola con opciones más avanzadas.
  - **Armitage**. Interfaz gráfica desarrollada en Java que permite interaccionar con el framework de forma sencilla e intuitiva.
  - **Web UI**. Solo Disponible en la versión pro.
  - ***msfcli***. Desaparecido, alternativa con la opción -x de msfconsole.

Arquitectura de Metasploit

- La parte principal de Metasploit son los **módulos**
- A través de ellos, interactuamos con el framework



https://docs.rapid7.com/metasploit/

Arquitectura de Metasploit

- Los módulos se clasifican según su utilidad:
  - **Exploits**: Se encuentran los exploits, organizados por categorías, por SO o tecnología.
  - **Auxiliary**: Para la integración y utilización de herramientas externas del framework. Hay erramientas como escáner de puertos, sniffers, DoS, fuzzers, …
  - **Payloads**: códigos que se ejecutan en la máquina atacada. Se organizan por categorías (*single* o *staged* y por tecnología (windows, linux, java, php…).
    - ***Single*** (_ : *windows/shell_reverse_tcp*). De un solo uso.
    - ***Staged*** (/ : *windows/shell/reverse_tcp*). Dispone de un handler para hacerlo persistente.

Arquitectura de Metasploit

- Los módulos se clasifican según su utilidad:
  - **Encoders**: proporciona codificadores para ofuscar los payloads y evitar que sean detectados por los antivirus.
  - **Post**: almacena las acciones a realizar en la fase de postexplotación: escalada de privilegios, impersonalización de tokens, captura de pruebas sobre la máquina remota, etc.
  - **Nop generators**: Contiene código capaz de generar instrucciones NOP (No Operation) para mantener el tamaño de los payloads consistentes.

Gestión de BD con Metasploit

- Una función de Metasploit es **ayudarnos a organizar la información recopilada** de la organización objetivo.
- Para ello, ofrece una **integración con PostgreSql**
- A través de **workspaces** nos permite guardar todas las máquinas, servicios, vulnerabilidades, credenciales recopiladas, …
  - Así como **consultarlos de manera ágil**

Gestión de BD con Metasploit

Comandos de inicio

- ***msfdb init***: Arranca e inicializa la BD
- ***msfdb reinit***: Borra y reinicia la BD
- ***msfdb delete***: Borra la BD
- ***msfdb start***: Arranca la BD
- ***msfdb stop***: Para la BD
- ***msfdb status***: Estado de la BD
- ***msfdb run***: Arranca la BD y luego metasploit

Gestión de BD con Metasploit

Gestión de workspaces

- ***workspace***: lista los workspaces
- ***workspace -a nombre***: crea un workspace
- ***workspace nombre***: entra en un workspace
- ***workspace -d nombre***: borra un workspace
- ***workspace -h***: ayuda

Gestión de BD con Metasploit

Principales comandos

- ***db_status***: comprueba el estado de la BD.
- ***db_nmap***: permite hacer escáneres nmap e incorporar el resultado directamente al workspace.
- ***hosts***: nos devuelve todos los hosts registrados en nuestro workspace.
- ***services***: lista de todos los servicios de todos los hosts de la base de datos.
- ***vulns***: listado de todas las vulnerabilidades.
- ***creds***: lista de credenciales recopiladas.
- ***loot***: ver dump de hashes recopilados.

Gestión de BD con Metasploit

hosts

- ***hosts -h***: ayuda sobre el comando
- ***hosts -c <columnas separadas por coma>***: permite seleccionar la información a mostrar
- ***hosts -S <palabras clave>***: filtrado de resultados

Gestión de BD con Metasploit

services

- ***services -h***: ayuda sobre el comando
- ***services IP***: servicios del host con cierta IP
- ***services -c <columnas separadas por coma>***: permite seleccionar la información a mostrar
- ***services -S <palabras clave>***: filtrado de resultados
- ***services -p <listado de puertos separados por coma>***: sólo servicios asociados a los puertos indicados.

Gestión de BD con Metasploit

creds

- ***creds -h***: ayuda sobre el comando
- ***creds IP***: credenciales recopiladas en esa IP

Búsquedas en Metasploit

Búsqueda de módulos en metasploit

- Comando **search**

```
search [<options>] [<keywords>:<value>]
```

- **Keywords** o palabras clave:
  - *arch*: arquitectura
  - *bid*: Bugtrack Id
  - cve: CVE ID
  - description
  - name
  - platform
  - port
  - ...

Otros comandos básicos

- ***info***: Muestra información del módulo que se haya seleccionado (comando use), o bien indicando como parámetro la ruta al módulo que se desea consultar.
- ***show***: Muestra las diferentes opciones de configuración de cualquier módulo. Si un módulo se encuentra seleccionado el comando tiene otras opciones disponibles como mostrar las variables de configuración (*show options*), los sistemas operativos vulnerables (*show targets*), entre otros.
- ***use***: Permite seleccionar el módulo que se desea usar para ejecutarlo contra el objetivo (ej: *use exploit/multi/handler*).

Otros comandos básicos

- ***back***: Permite salir del módulo seleccionado y volver al prompt de la consola de Metasploit.
- ***set, setg***: Permite asignar valores a las variables de configuración de un módulo. El comando *set* lo hace para el módulo seleccionado y *setg* para todo el contexto (global).
- ***unset, unsetg***: Hace lo opuesto que los anteriores, desasigna el valor de la variable indicada.
- ***connect***. Permite conectarnos a otras máquinas, similar a netcat.
- ***irb***. Permite ejecutar un intérprete de Ruby.

Otros comandos básicos

- ***load, unload y loadpath***: Permite cargar y descargar los plugins del framework en la carpeta por defecto. Si se encuentra en otro directorio se utiliza *loadpath*.
- ***check, exploit, sessions***: Se emplean en la fase de explotación una vez está seleccionado y configurado un exploit.
  - ***check*** permite comprobar si el sistema es vulnerable o no antes de lanzar el exploit.
  - ***exploit*** ejecuta el código malicioso y normalmente devolerá una shell de la máquina atacada.
  - ***sessions*** permite visualizar las sesiones abiertas a equipos comprometidos.

Otros comandos básicos

- ***resource***: Permite cargar un fichero con acciones especificas a realizar, útil para automatizar tareas.
- ***makerc***: Almacena en un fichero los comandos y acciones realizadas durante la sesión. Por defecto, el fichero se almacena en el home del usuario en una carpeta oculta ***msfX*** (*X*: número de versión).
- ***save***. Aporta persistencia a la configuración del entorno. Muy útil cuando el test de intrusión es largo y con muchos elementos. El fichero se almacena con el nombre *config* en la carpeta oculta y se comprueba si existe cuando se arranca msfconsole.
- ***jobs***. Gestiona los módulos en segundo plano.

Metasploit

Fin