Parrot OS Tolls

Miguel Ángel Roldán de Haro

1. Escaneo de Red y Enumeración:

- Nmap
- Zenmap (Interfaz gráfica para Nmap)
- Netdiscover
- Hping3
- Nessus (versión de prueba)

2. Análisis de Tráfico y Captura de Paquetes:

- Wireshark
- Tcpdump
- Dsniff
- Ettercap

3. Ataques y Explotación:

- Metasploit Framework
- Armitage (Interfaz gráfica para Metasploit)
- Social Engineering Toolkit (SET)
- Burp Suite
- SQLmap
- Aircrack-ng
- John the Ripper
- Hydra
- Hashcat

4. Análisis de Vulnerabilidades y Escaneo de Aplicaciones Web:

- OWASP Zap
- Nikto
- w3af (Web Application Attack and Audit Framework)
- WPScan (Escáner de seguridad para WordPress)
- SSLScan

5. Forense Digital:

- Autopsy
- Volatility
- Foremost
- Sleuth Kit
- Guymager

6. Anonimato y Privacidad:

- Tor
- JonDo
- BleachBit
- MACChanger

7. Otros:

- OpenVAS (escáner de vulnerabilidades)
- Maltego (herramienta de código abierto)
- Hash-Identifier
- Crunch (generador de listas de palabras)
- BeEF (Framework para explotar web)
- Hydra-GTK (Interfaz gráfica para Hydra)

1. Escaneo de Red y Enumeración:

Nmap

Nmap es una herramienta de escaneo de redes que permite descubrir hosts y servicios, así como realizar auditorías de seguridad. Aquí están algunos de los comandos más utilizados de acuerdo a la documentación oficial:

- `**nmap <target>**`: Realiza un escaneo de puertos en el objetivo especificado.

- `**nmap -sV <target>**`: Realiza un escaneo de puertos e intenta determinar las versiones de los servicios.

- `**nmap -p <ports> <target>**`: Especifica los puertos a escanear.

- `**nmap -A <target>**`: Realiza un escaneo agresivo que incluye detección de versiones, detección de sistemas operativos y scripts de automatización.

- `**nmap -O <target>**`: Intenta determinar el sistema operativo del objetivo.

- `**nmap -sS <target>**`: Realiza un escaneo de puertos TCP SYN stealth.

- `**nmap -sU <target>**`: Realiza un escaneo de puertos UDP.

- `**nmap -v <target>**`: Muestra una salida detallada durante el escaneo.

- `**nmap -Pn <target>**`: Ignora la detección de hosts y realiza un escaneo directo del objetivo.

- `**nmap -T<0-5> <target>**`: Especifica el perfil de velocidad/tiempo de escaneo, siendo 0 el más lento y 5 el más rápido.

- `**nmap --script <script> <target>**`: Ejecuta un script específico durante el escaneo.

- `**nmap -sC <target>**`: Ejecuta scripts de detección de vulnerabilidades y pruebas de seguridad en los puertos y servicios encontrados.

- `**nmap -sN <target>**`: Realiza un escaneo de puertos TCP NULL.

- `**nmap -sF <target>**`: Realiza un escaneo de puertos TCP FIN.

- `**nmap -sX <target>**`: Realiza un escaneo de puertos TCP Xmas.

- `**nmap -sP <target>**`: Realiza un escaneo de hosts en la red sin escanear puertos.

- `**nmap -sn <target>**`: Realiza un escaneo de hosts en la red sin enviar paquetes de solicitud de ICMP.

- `**nmap -sS -sV <target>**`: Realiza un escaneo de puertos TCP SYN stealth e intenta determinar las versiones de los servicios.

- `**nmap -p- <target>**`: Escanea todos los puertos en el objetivo.

- `**nmap -F <target>**`: Realiza un escaneo rápido de puertos más comunes.

- `**nmap -oN <output_file> <target>**`: Guarda la salida del escaneo en un archivo en formato normal.

- `**nmap -oX <output_file> <target>**`: Guarda la salida del escaneo en un archivo específico en formato XML.

- `**nmap -oG <output_file> <target>**`: Guarda la salida del escaneo en un archivo específico en formato Grepable.

- `**nmap -sL <target>**`: Lista los hosts en una red, sin enviar paquetes de solicitud a los hosts.

- `**nmap -6 <target>**`: Realiza un escaneo de puertos utilizando IPv6.

- `**nmap -sA <target>**`: Realiza un escaneo de puertos TCP ACK.

- `**nmap -sW <target>**`: Realiza un escaneo de puertos TCP Window.

- `**nmap -sM <target>**`: Realiza un escaneo de puertos TCP Maimon.

- `**nmap -sI <zombie host> <target>**`: Escaneo de puertos utilizando un host zombie como intermediario.

- `**nmap -sR <target>**`: Realiza un escaneo de puertos TCP RPC.

- `**nmap -sY <target>**`: Realiza un escaneo de puertos SCTP INIT.

- `**nmap -sZ <target>**`: Realiza un escaneo de puertos SCTP COOKIE-ECHO.

- `**nmap -sE <target>**`: Realiza un escaneo de puertos de capa 2 utilizando marcos Ethernet.

- `**nmap -sN -sF -sX <target>**`: Realiza un escaneo de puertos TCP NULL, FIN y Xmas combinados.

- `**nmap -sC -sV <target>**`: Ejecuta scripts de detección de vulnerabilidades y pruebas de seguridad, e intenta determinar las versiones de los servicios.

- `**nmap -p- -sV <target>**`: Escanea todos los puertos en el objetivo e intenta determinar las versiones de los servicios.

- `**nmap -p <ports> --top-ports <number> <target>**`: Escanea los puertos especificados y los primeros N puertos más comunes.

- `**nmap -iL <input_file>**`: Lee los objetivos de escaneo desde un archivo en lugar de especificarlos en el comando.

- `**nmap -d <target>**`: Muestra una salida de depuración detallada durante el escaneo.

- `**nmap -sT -p <port> <target>**`: Escaneo de puertos TCP utilizando el método de conexión estándar.

Nessus

Nessus es una herramienta de seguridad de red que se utiliza para realizar escaneos de seguridad y evaluaciones de vulnerabilidades en sistemas y redes. Aquí tienes una lista de comandos y funcionalidades disponibles en la versión de prueba de Nessus:

- **`nessuscli`:** Inicia la interfaz de línea de comandos de Nessus.

- **`nessuscli help`:** Muestra una lista de comandos y opciones disponibles.

- **`nessuscli update`:** Actualiza las definiciones de vulnerabilidad y plugins de Nessus.

- **`nessuscli policy list`:** Lista las políticas de escaneo disponibles.

- **`nessuscli policy add <name> <template>`:** Crea una nueva política de escaneo basada en una plantilla especificada.

- **`nessuscli policy delete <name>`:** Elimina una política de escaneo existente.

- **`nessuscli scan new <policy> <target>`:** Crea un nuevo escaneo utilizando una política y un objetivo especificados.

- **`nessuscli scan launch <scan_id>`:** Inicia un escaneo previamente creado con el ID del escaneo.

- **`nessuscli scan status <scan_id>`:** Muestra el estado y el progreso de un escaneo en curso.

- **`nessuscli scan pause <scan_id>`:** Pausa un escaneo en curso.

- **`nessuscli scan resume <scan_id>`:** Reanuda un escaneo pausado.

- **`nessuscli scan stop <scan_id>`:** Detiene un escaneo en curso.

- **`nessuscli scan export <scan_id> <file>`:** Exporta los resultados de un escaneo a un archivo específico.

- **`nessuscli report list`:** Lista los informes disponibles.

- **`nessuscli report export <report_id> <file>`:** Exporta un informe específico a un archivo.

- `nessuscli logout`: Cierra la sesión actual en Nessus.

- `**nessuscli user add <username> <password>**`: Crea un nuevo usuario en Nessus con el nombre de usuario y la contraseña especificados.

- `**nessuscli user delete <username>**`: Elimina un usuario existente de Nessus.

- `**nessuscli user list**`: Lista los usuarios configurados en Nessus.

- `**nessuscli plugin list**`: Lista los plugins de vulnerabilidad disponibles en Nessus.

- `**nessuscli plugin details <plugin_id>**`: Muestra detalles específicos de un plugin de vulnerabilidad.

- `**nessuscli scan resume-paused <scan_id>**`: Reanuda un escaneo que se haya pausado debido a la falta de tokens de escaneo disponibles.

- `**nessuscli scan reschedule <scan_id> <time>**`: Reprograma un escaneo para que se ejecute en un momento específico en el futuro.

- `**nessuscli plugin update**`: Actualiza los plugins de vulnerabilidad de Nessus.

- `**nessuscli feed update**`: Actualiza los feeds de seguridad de Nessus.

- `**nessuscli fetch <file_id> <output_file>**`: Descarga un archivo específico desde Nessus y lo guarda en la ubicación especificada.

- `**nessuscli prefs set <preference> <value>**`: Configura una preferencia específica de Nessus con el valor especificado.

Zenmap

Los comandos disponibles en Zenmap, la interfaz gráfica para Nmap:

1. File Menu (Menú Archivo):

- **New Scan** (Nuevo escaneo): Inicia un nuevo escaneo.

- **Open** (Abrir): Abre un archivo de escaneo guardado previamente.

- **Save** (Guardar): Guarda el resultado del escaneo actual en un archivo.

- **Save As** (Guardar como): Guarda el resultado del escaneo actual en un archivo con un nombre específico.

- **Print** (Imprimir): Imprime el resultado del escaneo actual.

- **Exit** (Salir): Cierra Zenmap.

2. Profile Menu (Menú Perfil):

- **Load Profile** (Cargar perfil): Carga un perfil de escaneo previamente guardado.

- **Save Profile** (Guardar perfil): Guarda la configuración actual como un perfil de escaneo.

- **Save Profile As** (Guardar perfil como): Guarda la configuración actual como un nuevo perfil con un nombre específico.

3. Scan Menu (Menú Escanear):

- **Quick Scan** (Escaneo rápido): Realiza un escaneo rápido de los puertos más comunes.

- **Intense Scan** (Escanear intensivo): Realiza un escaneo más exhaustivo que incluye detección de versiones, detección de sistemas operativos y scripts.

- **Intense Scan Plus UDP** (Escanear intensivo + UDP): Realiza un escaneo intensivo que incluye escaneo de puertos TCP y UDP.

- **Ping Scan** (Escanear ping): Realiza un escaneo para determinar qué hosts están activos en la red.

- **Full Scan** (Escaneo completo): Realiza un escaneo completo de todos los puertos.

- **Custom Scan** (Escanear personalizado): Permite personalizar los parámetros del escaneo según tus necesidades.

4. Tools Menu (Menú Herramientas):

- N**SLookup**: Realiza consultas de DNS para resolver nombres de dominio.

- **Traceroute**: Realiza un seguimiento de la ruta que sigue un paquete desde tu dispositivo hasta el objetivo.

- **Ping**: Envía paquetes de solicitud de ICMP para verificar la conectividad con un host.

- **Service Scan** (Escanear servicios): Escanea un host en busca de servicios y muestra información detallada sobre ellos.

- **Host Scan** (Escanear hosts): Escanea una red en busca de hosts activos.

5. View Menu (Menú Ver):

- **Topology** (Topología): Muestra una representación gráfica de la topología de red descubierta durante el escaneo.

- **Ports/Hosts** (Puertos/Hosts): Lista de los puertos y hosts escaneados y su estado

6. Help Menu (Menú Ayuda):

- **Contents** (Contenidos): Abre la documentación de ayuda de Zenmap.

- **About** (Acerca de): Muestra información sobre la versión y los creadores de Zenmap.

Netdiscover

Netdiscover es una herramienta de código abierto que permite realizar escaneos de red para descubrir hosts activos en una red local. Proporciona información como direcciones IP, direcciones MAC y proveedores de servicios de Internet (ISP), lo que resulta útil para detectar dispositivos no autorizados y resolver problemas de red. Su interfaz de línea de comandos permite personalizar el escaneo y mostrar los resultados en diferentes formatos.

Lista de los comandos disponibles en la herramienta Netdiscover:

- `netdiscover`: Ejecuta un escaneo de la red local para descubrir hosts activos.

- `netdiscover -i <interface>`: Especifica la interfaz de red a utilizar para el escaneo.

- `netdiscover -r <IP_range>`: Especifica un rango de direcciones IP a escanear en lugar de la red local.

- `netdiscover -p`: Muestra información adicional sobre los hosts descubiertos, como las direcciones MAC.

- `netdiscover -D`: Realiza el escaneo en modo pasivo, sin enviar paquetes de descubrimiento.

- `netdiscover -F`: Muestra el resultado del escaneo en formato de lista fija.

- `netdiscover -c`: Muestra el resultado del escaneo en formato de tabla compacta.

- `netdiscover -s`: Muestra el resultado del escaneo en formato de tabla estándar.

- `netdiscover -n`: No resuelve los nombres de host durante el escaneo.

- `netdiscover -P`: Muestra información adicional sobre el proveedor de servicios de Internet (ISP) de los hosts descubiertos.

- `netdiscover -v`: Muestra información detallada durante el escaneo.

- `netdiscover -h`: Muestra la ayuda y la lista de opciones disponibles.

- `netdiscover -L`: Muestra el resultado del escaneo en formato de lista larga, que incluye información detallada sobre cada host descubierto.

- `netdiscover -o <output_file>`: Guarda el resultado del escaneo en un archivo específico en lugar de mostrarlo en la salida estándar.

- `netdiscover -t <timeout>`: Especifica el tiempo de espera (timeout) para recibir respuestas de los hosts durante el escaneo.

- `netdiscover -f`: Activa el modo de escucha continua, que monitorea la red en busca de nuevos hosts activos en tiempo real.

- `netdiscover -N`: No muestra la dirección IP de la interfaz de red utilizada para el escaneo.

- `netdiscover -B`: Muestra el resultado del escaneo en formato de tabla HTML.

- `netdiscover -j <json_output_file>`: Guarda el resultado del escaneo en un archivo JSON en lugar de mostrarlo en la salida estándar.

- `netdiscover -x <exclude_ip_list>`: Excluye las direcciones IP especificadas de ser escaneadas.

- `netdiscover -T <threads>`: Especifica el número de hilos de escaneo simultáneos a utilizar para acelerar el proceso.

- `netdiscover -m <mac_vendor_file>`: Especifica un archivo personalizado que mapea direcciones MAC a fabricantes (vendors) para mostrar información adicional.

Hping3

Hping3 es una poderosa herramienta de prueba y análisis de redes que permite realizar diversas acciones en redes IP. A continuación, se presenta una lista de los comandos disponibles en Hping3:

- `**hping3 <target>**`: Envía paquetes ICMP al objetivo especificado.

- `**hping3 -S <target>**`: Envía paquetes TCP SYN al objetivo para realizar un escaneo de puertos.

- `**hping3 -s <source_ip> <target>**`: Especifica una dirección IP de origen para los paquetes enviados.

- `**hping3 -p <port> <target>**`: Especifica un puerto destino para los paquetes enviados.

- `**hping3 -c <count> <target>**`: Especifica el número de paquetes a enviar.

- `**hping3 -i <interval> <target>**`: Especifica el intervalo de tiempo entre el envío de paquetes.

- `**hping3 -t <ttl> <target>**`: Especifica el valor del campo TTL (Time To Live) en los paquetes enviados.

- `**hping3 -n <target>**`: No resuelve los nombres de host durante el escaneo.

- `**hping3 -R <target>**`: Envía paquetes TCP RST (Reset) al objetivo.

- `**hping3 -F <target>**`: Envía paquetes TCP FIN al objetivo.

- `**hping3 -U <target>**`: Envía paquetes UDP al objetivo.

- `**hping3 -A <target>**`: Envía paquetes TCP ACK al objetivo.

- `**hping3 -M <mtu> <target>**`: Especifica el tamaño máximo de unidad de transmisión (MTU) en los paquetes enviados.

- `**hping3 -V**`: Muestra información detallada sobre la versión de Hping3.

- `**hping3 -h**`: Muestra la ayuda y la lista de opciones disponibles.

- `**hping3 -Q**`: Activa el modo "quiet", lo que significa que Hping3 no mostrará información detallada durante la ejecución.

- `**hping3 -z**`: Realiza un escaneo de puertos utilizando un método de escaneo aleatorio.

- `**hping3 -E <data> <target>**`: Envía datos personalizados en los paquetes enviados.

- `**hping3 -X <data> <target>**`: Envía paquetes TCP con el indicador Xmas (todos los indicadores de control establecidos).

- `**hping3 -Y <data> <target>**`: Envía paquetes TCP con el indicador Yanqui (todos los indicadores de control excepto ACK).

- `**hping3 -w <wait> <target>**`: Especifica el tiempo de espera (wait) para recibir respuestas de los hosts durante el escaneo.

- `**hping3 -r <count> <target>**`: Envía paquetes con la bandera de reinicio activada, lo que puede ayudar a identificar sistemas vulnerables.

- `**hping3 -D <target>**`: Envía paquetes TCP con la bandera de no fragmentación activada.

- `**hping3 -b <size> <target>**`: Especifica el tamaño del búfer en los paquetes enviados.

- `**hping3 -G <filename> <target>**`: Lee los datos a enviar desde un archivo en lugar de especificarlos en el comando.

- `hping3 -A <data> <target>`: Envía paquetes TCP con la bandera de urgencia activada.

- `**hping3 -T <type> <target>**`: Especifica el tipo de servicio en el campo ToS (Type of Service) de los paquetes enviados.

- `**hping3 -L**`: Activa el modo "linger", lo que hace que Hping3 espere un tiempo antes de finalizar la ejecución.

Nessucli

Nessus es una herramienta de seguridad de red que se utiliza para realizar escaneos de seguridad y evaluaciones de vulnerabilidades en sistemas y redes. Aquí tienes una lista de comandos y funcionalidades disponibles en la versión de prueba de Nessus:

- `**nessuscli**`: Inicia la interfaz de línea de comandos de Nessus.

- `**nessuscli help**`: Muestra una lista de comandos y opciones disponibles.

- `**nessuscli update**`: Actualiza las definiciones de vulnerabilidad y plugins de Nessus.

- `**nessuscli policy list**`: Lista las políticas de escaneo disponibles.

- `**nessuscli policy add <name> <template>**`: Crea una nueva política de escaneo basada en una plantilla especificada.

- `nessuscli policy delete <name>`: Elimina una política de escaneo existente.

- `**nessuscli scan new <policy> <target>**`: Crea un nuevo escaneo utilizando una política y un objetivo especificados.

- `**nessuscli scan launch <scan_id>**`: Inicia un escaneo previamente creado con el ID del escaneo.

- `**nessuscli scan status <scan_id>**`: Muestra el estado y el progreso de un escaneo en curso.

- `**nessuscli scan pause <scan_id>**`: Pausa un escaneo en curso.

- `**nessuscli scan resume <scan_id>**`: Reanuda un escaneo pausado.

- `**nessuscli scan stop <scan_id>**`: Detiene un escaneo en curso.

- `**nessuscli scan export <scan_id> <file>**`: Exporta los resultados de un escaneo a un archivo específico.

- `**nessuscli report list**`: Lista los informes disponibles.

- `**nessuscli report export <report_id> <file>**`: Exporta un informe específico a un archivo

- `**nessuscli user add <username> <password>**`: Crea un nuevo usuario en Nessus con el nombre de usuario y la contraseña especificados.

- `**nessuscli user delete <username>**`: Elimina un usuario existente de Nessus.

- `**nessuscli user list**`: Lista los usuarios configurados en Nessus.

- `**nessuscli plugin list**`: Lista los plugins de vulnerabilidad disponibles en Nessus.

- `**nessuscli plugin details <plugin_id>**`: Muestra detalles específicos de un plugin de vulnerabilidad.

- `**nessuscli scan resume-paused <scan_id>**`: Reanuda un escaneo que se haya pausado debido a la falta de tokens de escaneo disponibles.

- `**nessuscli scan reschedule <scan_id> <time>**`: Reprograma un escaneo para que se ejecute en un momento específico en el futuro.

- `**nessuscli plugin update**`: Actualiza los plugins de vulnerabilidad de Nessus.

- `**nessuscli feed update**`: Actualiza los feeds de seguridad de Nessus.

- `**nessuscli fetch <file_id> <output_file>**`: Descarga un archivo específico desde Nessus y lo guarda en la ubicación especificada.

- `**nessuscli prefs set <preference> <value>**`: Configura una preferencia específica de Nessus con el valor especificado.

- `**nessuscli logout**`: Cierra la sesión actual en Nessus.

- `**nessuscli user add <username> <password>**`: Crea un nuevo usuario en Nessus con el nombre de usuario y la contraseña especificados.

- `**nessuscli user delete <username>**`: Elimina un usuario existente de Nessus.

- `**nessuscli user list**`: Lista los usuarios configurados en Nessus.

- `**nessuscli plugin list**`: Lista los plugins de vulnerabilidad disponibles en Nessus.

- `**nessuscli plugin details <plugin_id>**`: Muestra detalles específicos de un plugin de vulnerabilidad.

- `**nessuscli user add <username> <password>**`: Crea un nuevo usuario en Nessus con el nombre de usuario y la contraseña especificados.

- `**nessuscli user delete <username>**`: Elimina un usuario existente de Nessus.

- `**nessuscli user list**`: Lista los usuarios configurados en Nessus.

- `**nessuscli plugin list**`: Lista los plugins de vulnerabilidad disponibles en Nessus.

- `**nessuscli plugin details <plugin_id>**`: Muestra detalles específicos de un plugin de vulnerabilidad.

- `**nessuscli scan resume-paused <scan_id>**`: Reanuda un escaneo que se haya pausado debido a la falta de tokens de escaneo disponibles.

- `**nessuscli scan reschedule <scan_id> <time>**`: Reprograma un escaneo para que se ejecute en un momento específico en el futuro.

- `**nessuscli plugin update**`: Actualiza los plugins de vulnerabilidad de Nessus.

- `**nessuscli feed update**`: Actualiza los feeds de seguridad de Nessus.

- `**nessuscli fetch <file_id> <output_file>**`: Descarga un archivo específico desde Nessus y lo guarda en la ubicación especificada.

- `**nessuscli prefs set <preference> <value>**`: Configura una preferencia específica de Nessus con el valor especificado.

2. Análisis de Tráfico y Captura de Paquetes: Wireshark

Wireshark es una popular herramienta de análisis de protocolos de red que permite capturar y examinar el tráfico de red en tiempo real. A continuación, se presenta una lista de comandos y funcionalidades disponibles en Wireshark:

- `**wireshark**`: Inicia la interfaz gráfica de Wireshark para capturar y analizar el tráfico de red.

- `**wireshark -i <interface>**`: Especifica la interfaz de red desde la cual capturar el tráfico.

- `**wireshark -r <file>**`: Abre un archivo de captura previamente guardado en Wireshark para su análisis.

- `**wireshark -f <capture_filter>**`: Aplica un filtro de captura para limitar los paquetes mostrados según criterios específicos.

- `**wireshark -Y <display_filter>**`: Aplica un filtro de visualización para mostrar solo los paquetes que coincidan con el filtro especificado.

- `**wireshark -z <statistics_type>**`: Muestra estadísticas específicas sobre la captura actual, como conversaciones, flujos o resumen de protocolos.

- `**wireshark -d <dissector>**`: Muestra información detallada sobre un campo específico en un paquete.

- `**wireshark -w <output_file>**`: Guarda el tráfico capturado en un archivo para su posterior análisis.

- `**wireshark -E <name=value>**`: Establece una variable de entorno específica para controlar el comportamiento de Wireshark.

- `**wireshark -h**`: Muestra la ayuda y la lista de opciones disponibles.

Aquí tienes algunos comandos adicionales de Wireshark:

- `**wireshark -K**`: Inicia Wireshark en modo promiscuo, lo que permite capturar todos los paquetes en la red, incluidos los que no están destinados a la interfaz de red en uso.

- `**wireshark -n**`: Deshabilita la resolución de nombres de host durante la captura y muestra direcciones IP en su lugar.

- `**wireshark -t ad**`: Habilita la resolución de direcciones MAC a nombres de fabricantes durante la captura.

- `**wireshark -N**`: Deshabilita el cálculo automático de resumen de protocolos en la vista de resumen.

- `**wireshark -o <protocol>:<pref>=<value>**`: Establece opciones personalizadas para un protocolo específico.

- `**wireshark -X lua_script:<script>**`: Ejecuta un script Lua durante la captura para personalizar el análisis y procesamiento de paquetes.

- `wireshark -G`: Muestra una lista de todos los dispositivos de red disponibles en el sistema.

- `**wireshark -C <file_size>**`: Limita el tamaño de los archivos de captura, creando nuevos archivos de captura cuando se alcanza el tamaño especificado.

- `**wireshark -I**`: Habilita la actualización automática de la lista de capturas disponibles en tiempo real.

- `**wireshark -D**`: Muestra una lista de las interfaces de red disponibles en el sistema.

- `**wireshark -o <preference>:<value>**`: Establece una preferencia específica de Wireshark con el valor especificado.

- `**wireshark -R <display_filter>**`: Aplica un filtro de visualización inicial a la captura cargada desde un archivo.

- `**wireshark -s <snaplen>**`: Especifica la longitud máxima de los paquetes a capturar.

Wireshark (Interfaz Gráfica)

1. Menú Archivo:

- Nuevo escaneo: Inicia un nuevo escaneo de red para capturar y analizar el tráfico en tiempo real.

- Abrir: Permite abrir un archivo de captura previamente guardado para su análisis.

- Guardar: Guarda el resultado del escaneo actual en un archivo.

- Guardar como: Permite guardar el resultado del escaneo actual en un archivo con un nombre específico.

- Imprimir: Imprime el resultado del escaneo actual o partes seleccionadas de él.

- Exportar objetos: Permite exportar objetos incrustados en los paquetes capturados, como imágenes o archivos.

- Salir: Cierra la aplicación Wireshark.

2. Botones de control de captura:

- Iniciar: Comienza la captura del tráfico de red.

- Detener: Detiene la captura del tráfico de red.

- Reiniciar: Borra la ventana de captura y reinicia la captura.

- Opciones de captura: Abre un diálogo para configurar las opciones de captura, como la interfaz de red, el filtro de captura, etc.

3. Paneles de visualización:

- Lista de paquetes: Muestra una lista de los paquetes capturados, con información resumida sobre cada uno.

- Detalles del paquete: Muestra información detallada sobre el paquete seleccionado en la lista de paquetes.

- Bytes en crudo: Muestra la representación en formato hexadecimal y ASCII de los datos contenidos en el paquete seleccionado.

- Árbol de protocolos: Muestra una vista jerárquica de los protocolos utilizados en el paquete seleccionado.

- Seguimiento de secuencia: Muestra el flujo de datos para un par de direcciones IP y puertos.

4. Opciones de filtro y búsqueda:

- Filtro de captura: Permite especificar un filtro de captura para mostrar solo los paquetes que coincidan con los criterios específicos.

- Búsqueda: Permite buscar paquetes específicos según palabras clave o patrones en el contenido del paquete.

- Filtro de visualización rápida: Permite aplicar rápidamente filtros predefinidos para mostrar solo ciertos tipos de paquetes.

5. Menú Estadísticas:

- Estadísticas de conversación: Muestra información estadística sobre las conversaciones de red detectadas.

- Estadísticas de protocolo: Muestra información estadística sobre los protocolos utilizados en la captura.

- Estadísticas de flujo: Proporciona estadísticas detalladas sobre el flujo de datos entre direcciones IP y puertos.

- **Estadísticas IO:** Muestra información sobre el tráfico de entrada y salida durante la captura.

6. Menú Ver:

- **Opciones de visualización:** Permite personalizar la apariencia y el diseño de la ventana de Wireshark.

- **Actualizar:** Actualiza la visualización de la captura actual.

- **Zoom:** Permite acercar o alejar la vista de los paquetes capturados.

- **Ajustar ancho de columna:** Ajusta automáticamente el ancho de las columnas en la lista de paquetes para mostrar toda la información.

7. Menú Telephony:

- Opciones de VoIP: Proporciona herramientas y opciones para el análisis de llamadas VoIP.

8. Menú Statistics (Estadísticas):

- **Conversations** (Conversaciones): Muestra información estadística sobre las conversaciones de red.

- **Endpoints** (Puntos finales): Muestra información estadística sobre los puntos finales de la comunicación.

- **Protocol Hierarchy** (Jerarquía de protocolos): Muestra una vista jerárquica de los protocolos utilizados en la captura.

9. Menú Tools (Herramientas):

- Resolución de nombres: Realiza la resolución de nombres de host y direcciones IP.

- Asistentes: Proporciona asistentes para tareas específicas, como la configuración de filtros avanzados o la configuración de la captura remota.

- Plugins: Permite administrar los plugins y extensiones instalados en Wireshark.

- Preferences (Preferencias): Permite configurar las preferencias y opciones de Wireshark.

10. Menú Help (Ayuda):

- Documentación en línea: Abre la documentación oficial de Wireshark en el navegador web predeterminado.

- About Wireshark: Muestra información sobre la versión de Wireshark instalada.

TcpDump

Tcpdump es una herramienta de captura y análisis de tráfico de red en línea de comandos. Permite examinar y filtrar paquetes en tiempo real, proporcionando una visión detallada de las comunicaciones en una red. Es ampliamente utilizado en administración de redes, resolución de problemas y seguridad. Tcpdump ofrece funcionalidades avanzadas para capturar, filtrar y analizar el tráfico de red, lo que permite comprender mejor el comportamiento de la red y detectar posibles problemas o amenazas.

Aquí tienes una lista de los comandos de Tcpdump con una breve descripción de lo que hacen:

- `**tcpdump**`: Inicia la captura de paquetes en tiempo real.

- `**tcpdump -A**`: Muestra el contenido completo de los paquetes en formato ASCII.

- `**tcpdump -D**`: Muestra las interfaces de red disponibles para capturar.

- `**tcpdump -c <cantidad>**`: Limita el número de paquetes a capturar.

- `**tcpdump -d**`: Muestra el filtro de captura en formato legible por Tcpdump.

- `**tcpdump -dd**`: Muestra el filtro de captura en formato legible por Libpcap.

- `**tcpdump -e**`: Muestra información detallada sobre la estructura de los paquetes.

- `**tcpdump -E**`: Muestra las técnicas de encriptación disponibles en el sistema.

- `**tcpdump -F**`: Lee el filtro de captura desde un archivo.

- `**tcpdump -G**`: Especifica un intervalo de tiempo para dividir automáticamente los archivos de captura.

- `**tcpdump -h**`: Muestra una ayuda rápida sobre los comandos de Tcpdump.

- `**tcpdump -i <interfaz>**`: Especifica la interfaz de red a utilizar para la captura.

- `**tcpdump -l**`: Activa el modo de salida en línea.

- `**tcpdump -m**`: Muestra los archivos de impresión disponibles.

- `**tcpdump -M**`: Muestra los métodos de impresión disponibles.

- `**tcpdump -n**`: No resuelve nombres de host ni de puertos.

- `**tcpdump -N**`: Deshabilita la resolución de nombres de puerto.

- `**tcpdump -nn**`: Muestra direcciones IP y números de puerto en lugar de nombres.

- `**tcpdump -o**`: Muestra opciones de captura disponibles.

- `**tcpdump -O**`: Desactiva la optimización del filtro de captura.

- `**tcpdump -p**`: Pone la interfaz de red en modo promiscuo.

- `**tcpdump -q**`: Modo silencioso, muestra menos información adicional.

- `**tcpdump -r <archivo>**`: Lee los paquetes capturados desde un archivo previamente guardado en lugar de capturarlos en tiempo real.

- `**tcpdump -R**`: Especifica un filtro de captura en tiempo real.

- `**tcpdump -s <tamaño>**`: Especifica el tamaño máximo de los paquetes a capturar.

- `**tcpdump -S**`: Muestra la secuencia numérica de los paquetes en lugar de resolver los números de secuencia TCP en nombres simbólicos.

- `**tcpdump -t**`: Omite la impresión de la marca de tiempo.

- `**tcpdump -tt**`: Muestra la marca de tiempo relativa de cada paquete.

- `**tcpdump -ttt**`: Muestra la marca de tiempo completa de cada paquete.

- `**tcpdump -U**`: Guarda los paquetes capturados en archivos separados por usuario.

- `**tcpdump -v**`: Aumenta el nivel de verbosidad, mostrando más información detallada durante la captura.

- `**tcpdump -V**`: Muestra la versión de Tcpdump.

- `**tcpdump -w <archivo>**`: Guarda los paquetes capturados en un archivo en lugar de mostrarlos en la salida estándar.

- `**tcpdump -W**`: Especifica el número máximo de archivos de captura a generar.

- `**tcpdump -x**`: Muestra los datos del paquete en formato hexadecimal.

- `**tcpdump -X**`: Muestra los datos del paquete en formato hexadecimal y ASCII.

- `**tcpdump -y**`: Muestra los enlaces de capa 2 disponibles en el sistema.

- `**tcpdump -z**`: Especifica un comando para ejecutar sobre los archivos de captura.

Dsniff

Dsniff es una suite de herramientas de seguridad que proporciona funcionalidades para la vigilancia y auditoría de redes. Permite interceptar y analizar el tráfico de red en tiempo real, capturando información sensible como contraseñas, correos electrónicos, archivos descargados y más. Dsniff es ampliamente utilizado en pruebas de penetración y evaluaciones de seguridad para identificar posibles vulnerabilidades en una red. Sin embargo, es importante utilizar esta herramienta de manera responsable y ética, respetando las leyes y regulaciones aplicables, ya que su mal uso puede tener consecuencias negativas y violar la privacidad de las personas.

Lista de todos los comandos disponibles en la suite Dsniff:

- `**arpspoof**`: Interceptar y redirigir el tráfico en una red local mediante respuestas ARP falsificadas.

- `**dnsspoof**`: Envenenar la caché DNS de una red local para redirigir consultas DNS legítimas.

- `**filesnarf**`: Intercepta y guarda archivos descargados a través de conexiones FTP no cifradas.

- `**macof**`: Generar tráfico de red falso enviando tramas Ethernet con direcciones MAC aleatorias.

- `**mailsnarf**`: Capturar correos electrónicos en texto sin formato transmitidos a través de conexiones POP3.

- `**msgsnarf**`: Capturar mensajes de chat en tiempo real transmitidos a través del protocolo IRC.

- `**sshmitm**`: Interceptar y registrar conexiones SSH realizadas en la red.

- `**sshow**`: Mostrar información sobre las sesiones SSH activas en una red.

- `**tcpkill**`: Finalizar conexiones TCP seleccionadas enviando paquetes RST (Reset) a ambas partes.

- `**tcpnice**`: Limitar el ancho de banda de las conexiones TCP seleccionadas.

- `**urlsnarf**`: Capturar información sensible, como contraseñas, al interceptar y registrar solicitudes HTTP.

- `**webmitm**`: Interceptar y registrar conexiones HTTP realizadas en la red.

- `**webspy**`: Capturar imágenes en tiempo real de las páginas web visitadas en la red.

Ettercap

Ettercap es una herramienta de seguridad de red que se utiliza para llevar a cabo ataques de tipo "man-in-the-middle" (hombre en el medio) en redes locales. Algunos de los comandos y características principales de Ettercap son:

- `**ettercap**`: Inicia la herramienta Ettercap y muestra la interfaz interactiva.

- `**ettercap -T -i <interfaz>**`: Inicia Ettercap en modo texto y especifica la interfaz de red a utilizar.

- `**ettercap -G**`: Inicia Ettercap en modo gráfico y muestra la interfaz gráfica de usuario.

- `**ettercap -C**`: Carga un archivo de configuración personalizado al iniciar Ettercap.

- `**ettercap -M**`: Activa el modo de ataque "man-in-the-middle" (MITM) para interceptar y modificar el tráfico en la red.

- `**ettercap -L**`: Guarda los registros de actividad de Ettercap en un archivo.

- `**ettercap -w <archivo>**`: Guarda los paquetes capturados en un archivo pcap.

- `**ettercap -r <archivo>**`: Lee y reproduce un archivo pcap previamente guardado en lugar de capturar en tiempo real.

- `**ettercap -F <filtro>**`: Especifica un filtro para capturar solo paquetes que coincidan con ciertos criterios.

- `**ettercap -P**`: Muestra información sobre los plugins disponibles en Ettercap.

- `**ettercap -M ARP**`: Activa el modo de ataque "man-in-the-middle" utilizando envenenamiento ARP.

- `**ettercap -M DHCP**`: Activa el modo de ataque "man-in-the-middle" utilizando envenenamiento DHCP.

- `**ettercap -M ICMP**`: Activa el modo de ataque "man-in-the-middle" utilizando

envenenamiento ICMP.

- `**ettercap -M DNS**`: Activa el modo de ataque "man-in-the-middle" utilizando envenenamiento DNS.

- `**ettercap -M RIPv1**`: Activa el modo de ataque "man-in-the-middle" utilizando envenenamiento RIPv1.

- `**ettercap -M RIPv2**`: Activa el modo de ataque "man-in-the-middle" utilizando envenenamiento RIPv2.

- `**ettercap -h**`: Muestra la ayuda y una descripción de los comandos disponibles en Ettercap.

3. Ataques y explotación

Metasploit Framework

El Metasploit Framework es una poderosa plataforma de pruebas de penetración y desarrollo de exploits que permite a los profesionales de seguridad evaluar la seguridad de sistemas y redes. Ofrece una amplia gama de herramientas y funcionalidades para realizar pruebas de penetración, investigación de vulnerabilidades y desarrollo de exploits.

Algunos de los comandos y características clave de Metasploit Framework son:

- `**msfconsole**`: Inicia el intérprete de comandos de Metasploit Framework, proporcionando una interfaz interactiva para ejecutar comandos y módulos.

- `**msfvenom**`: Genera payloads (cargas útiles) maliciosos para explotar vulnerabilidades en sistemas y dispositivos.

- `**msfcli**`: Permite ejecutar comandos de módulos de Metasploit Framework desde la línea de comandos.

- `**msfencode**`: Ofrece capacidades de codificación y ofuscación para evitar la detección de malware.

- `**msfdb**`: Gestiona la base de datos de Metasploit Framework, que almacena información sobre hosts, servicios, exploits, credenciales, entre otros.

- `**msfupdate**`: Actualiza el Metasploit Framework con las últimas actualizaciones y módulos disponibles.

- `**search**`: Permite buscar módulos y exploits en la base de datos de Metasploit Framework.

- `**use**`: Se utiliza para cargar un módulo específico y prepararlo para su uso.

- `**exploit**`: Ejecuta un exploit específico contra un objetivo previamente seleccionado.

- `**set**`: Establece valores de opciones para un módulo específico.

- `**show**`: Muestra información detallada sobre módulos, exploits, hosts, sesiones, entre otros.

- `**sessions**`: Proporciona comandos para gestionar sesiones de acceso remoto después de una explotación exitosa.

- `**db_nmap**`: Ejecuta un escaneo de puertos utilizando Nmap y guarda los resultados en la base de datos de Metasploit Framework.

- `**msfconsole**`: Inicia el intérprete de comandos de Metasploit Framework.

- `**msfcli**`: Permite ejecutar comandos de módulos de Metasploit Framework desde la línea de comandos.

- `**msfvenom**`: Genera payloads (cargas útiles) maliciosos.

- `**msfupdate**`: Actualiza el Metasploit Framework con las últimas actualizaciones y módulos.

- `**msfdb**`: Gestiona la base de datos de Metasploit Framework.

- `**msfconsole -r <script>**`: Ejecuta un script de Metasploit Framework al iniciar la consola.

- `**search <keyword>**`: Busca módulos y exploits por palabra clave.

- `**use <module>**`: Carga un módulo específico y lo prepara para su uso.

- `**show options**`: Muestra las opciones disponibles para el módulo cargado.

- `**set <option> <value>**`: Establece el valor de una opción para el módulo cargado.

- `**show payloads**`: Muestra los payloads disponibles para el módulo cargado.

- `**show exploits**`: Muestra los exploits disponibles.

- `**show auxiliary**`: Muestra los módulos auxiliares disponibles.

- `**show post**`: Muestra los módulos de post-explotación disponibles.

- `**show encoders**`: Muestra los encoders disponibles.

- `**show nops**`: Muestra los nops (instrucciones sin operación) disponibles.

- `**exploit**`: Ejecuta el exploit cargado contra el objetivo seleccionado.

- `**setg <option> <value>**`: Establece el valor de una opción global.

- `**unsetg <option>**`: Restablece el valor de una opción global.

- `**sessions -l**`: Lista las sesiones activas.

- `**sessions -i <id>**`: Interactúa con una sesión específica.

- `**jobs**`: Muestra los trabajos en ejecución.

- `**bg <job>**`: Ejecuta un trabajo en segundo plano.

- `**fg <job>**`: Trae un trabajo al primer plano.

- `**exit**`: Sale de la consola de Metasploit Framework.

Nikto

Nikto es una herramienta de escaneo de vulnerabilidades de código abierto diseñada para realizar un análisis exhaustivo de la seguridad de aplicaciones web. Utiliza una amplia base de datos de firmas y pruebas de seguridad para detectar posibles vulnerabilidades en servidores web y aplicaciones en funcionamiento. Nikto busca activamente diferentes tipos de problemas de seguridad, como configuraciones incorrectas del servidor, archivos y directorios confidenciales expuestos, versiones obsoletas de software y otras vulnerabilidades conocidas. Proporciona un informe detallado de los hallazgos, lo que permite a los administradores de sistemas y auditores de seguridad tomar medidas para fortalecer la seguridad y proteger los sistemas web contra posibles ataques. Nikto es una herramienta valiosa para realizar auditorías de seguridad y asegurar que los sistemas web estén protegidos de manera adecuada.

Lista de comandos existentes para la herramienta Nikto:

- `**nikto -h <host>**`: Realiza un escaneo en el host especificado.

- `**nikto -h <host> -p <puerto>**`: Realiza un escaneo en el host y puerto especificados.

- `**nikto -h <host> -ssl**`: Realiza un escaneo en el host utilizando una conexión SSL.

- `**nikto -h <host> -p <puerto> -ssl**`: Realiza un escaneo en el host y puerto especificados utilizando una conexión SSL.

- `**nikto -h <host> -C <config-file>**`: Utiliza un archivo de configuración personalizado para el escaneo en el host especificado.

- `**nikto -list-plugins**`: Muestra una lista de todos los plugins disponibles en Nikto.

- `**nikto -Display <option>**`: Especifica las opciones de visualización, como "Cert" para mostrar información del certificado SSL.

- `**nikto -Format <output-format>**`: Especifica el formato de salida para los resultados del escaneo, como "csv", "xml" o "html".

- `**nikto -output <output-file>**`: Guarda los resultados del escaneo en un archivo de salida especificado.

- `**nikto -dbcheck**`: Realiza una verificación de la base de datos de Nikto para asegurarse de que está actualizada.

- `**nikto -update**`: Actualiza la base de datos de Nikto con las últimas firmas de vulnerabilidades.

- `**nikto -Plugins <plugin(s)>**`: Especifica los plugins a ejecutar durante el escaneo.

- `**nikto -id <custom-header>**`: Especifica un encabezado personalizado para enviar con las solicitudes del escaneo.

- `**nikto -evasion <technique>**`: Especifica una técnica de evasión para evitar la detección durante el escaneo.

- `**nikto -Tuning <tuning-option>**`: Especifica una opción de ajuste para personalizar el escaneo, como "paranoid", "stealth" o "aggressive".

- `**nikto -list-plugins**`: Muestra una lista de todos los plugins disponibles en Nikto.

- `**nikto -help**`: Muestra la ayuda y los comandos disponibles en Nikto.

Armitage

Armitage es una interfaz gráfica de usuario (GUI) diseñada para trabajar con el Metasploit Framework. Proporciona una forma más intuitiva y visual de interactuar con las funcionalidades de Metasploit, lo que facilita la ejecución de pruebas de penetración y el aprovechamiento de las capacidades del framework. Algunos de los comandos y características clave de Armitage son:

- `**armitage**`: Inicia la interfaz gráfica de Armitage.

- `**msfconsole**`: Abre una consola de Metasploit Framework dentro de Armitage para ejecutar comandos y módulos.

- `**connect**`: Establece una conexión con un servidor de Metasploit Framework para acceder a sus funcionalidades.

- `**db_connect**`: Conecta Armitage a una base de datos de Metasploit Framework.

- `**Hosts**`: Muestra una lista de hosts descubiertos durante las pruebas de penetración.

- `**Services**`: Muestra los servicios en ejecución en los hosts y las vulnerabilidades asociadas a ellos.

- `**Attacks**`: Permite seleccionar y ejecutar ataques específicos contra los hosts y servicios seleccionados.

- `**Exploits**`: Proporciona una lista de exploits disponibles para explotar las vulnerabilidades detectadas.

- `**Post Exploitation**`: Ofrece opciones para realizar actividades de post-explotación después de comprometer un host.

- `**Hail Mary**`: Ejecuta una serie de exploits y ataques contra todos los hosts objetivo al mismo tiempo.

- `**AutoPwn**`: Realiza un escaneo de vulnerabilidades y ejecuta automáticamente exploits contra los hosts objetivo.

- `**Team Server**`: Permite configurar un servidor de Metasploit Framework para compartir colaborativamente sesiones y datos con otros usuarios de Armitage.

- `**armitage**`: Inicia la interfaz gráfica de Armitage.

- `**msfconsole**`: Abre una consola de Metasploit Framework dentro de Armitage.

- `**connect**`: Establece una conexión con un servidor de Metasploit Framework.

- `**db_connect**`: Conecta Armitage a una base de datos de Metasploit Framework.

- `**Hosts**`: Muestra una lista de hosts descubiertos durante las pruebas de penetración.

- `**Services**`: Muestra los servicios en ejecución en los hosts y las vulnerabilidades asociadas.

- `**Attacks**`: Permite seleccionar y ejecutar ataques contra los hosts y servicios seleccionados.

- `**Exploits**`: Proporciona una lista de exploits disponibles.

- `Payloads`: Muestra los payloads (cargas útiles) disponibles para su uso en los exploits.

- `**Listeners**`: Configura y administra los listeners para capturar conexiones de shell inversas.

- `**Post Exploitation**`: Ofrece opciones para realizar actividades de post-explotación en los hosts comprometidos.

- `**Scans**`: Permite realizar escaneos de vulnerabilidades en los hosts seleccionados.

- `**Hail Mary**`: Ejecuta una serie de exploits y ataques contra todos los hosts objetivo.

- `**AutoPwn**`: Realiza un escaneo de vulnerabilidades y ejecuta automáticamente exploits contra los hosts objetivo.

- `**Loot**`: Administra los datos y archivos capturados durante las pruebas de penetración.

- `**Search**`: Busca módulos y exploits en la base de datos de Metasploit Framework.

- `**Modules**`: Muestra una lista de módulos disponibles para su uso.

- `**Console**`: Abre una consola interactiva para ejecutar comandos de Metasploit Framework.

- `**Team Server**`: Permite configurar un servidor de Metasploit Framework para compartir sesiones y datos con otros usuarios de Armitage.

- `**Help**`: Proporciona información sobre los comandos y opciones disponibles en Armitage.

Social Engineering Toolkit (SET)

Social Engineering Toolkit (SET) es una poderosa herramienta diseñada para realizar ataques de ingeniería social. Permite a los profesionales de seguridad simular ataques de phishing, spear phishing, captura de contraseñas, generación de archivos maliciosos y diversas técnicas de manipulación psicológica. SET automatiza muchos de los procesos involucrados en los ataques de ingeniería social, lo que facilita la creación y ejecución de escenarios realistas.

Algunos de los comandos y opciones disponibles en SET son:

Aquí tienes la lista de todos los comandos disponibles en Social Engineering Toolkit (SET) junto con una breve descripción de cada uno:

1. Social Engineering Attacks:

- `1. Spear-Phishing Attack Vectors`: Permite realizar ataques de phishing dirigidos.

- `2. Website Attack Vectors`: Proporciona vectores de ataque específicos para sitios web.

- `3. Infectious Media Generator`: Genera archivos maliciosos para propagar infecciones.

- `4. Create a Payload and Listener`: Crea un payload y un escuchador para obtener acceso remoto a un sistema.

- `5. Mass Mailer Attack`: Envía correos masivos con contenido malicioso.

- `6. Arduino-Based Attack Vector`: Utiliza dispositivos Arduino para realizar ataques físicos.

- `7. Wireless Access Point Attack Vector`: Configura un punto de acceso inalámbrico malicioso.

- `8. QRCode Generator Attack Vector`: Genera códigos QR maliciosos.

- **`9. Powershell Attack Vectors`:** Utiliza scripts de PowerShell para llevar a cabo ataques.

- **`10. Third Party Modules`:** Importa módulos de terceros para ampliar las funcionalidades de SET.

2. Credential Harvester:

- **`1. Site Cloner`:** Clona un sitio web específico para realizar un ataque de recolección de credenciales.

- **`2. Site Importer`:** Importa un sitio web para realizar un ataque de recolección de credenciales.

- **`3. Web Templates`:** Utiliza plantillas web para crear páginas de phishing personalizadas.

3. Metasploit Browser Exploit Method:

- **`1. Microsoft Browser Autopwn`:** Utiliza exploits en navegadores de Microsoft.

- **`2. BEEF Injection Method`:** Inyecta scripts maliciosos utilizando BEEF (Browser Exploitation Framework).

4. Credential Harvester Method:

- **`1. Web Jacking Attack Method`:** Realiza un ataque de web jacking para robar credenciales.

- **`2. Tabnabbing Attack Method`:** Realiza un ataque de tabnabbing para robar credenciales.

- **`3. Multi-Attack Web Method`:** Combina varios métodos de ataque web.

5. Tabnabbing Attack Method:

- **`1. Clone Website`:** Clona un sitio web para realizar un ataque de tabnabbing.

- **`2. Web Jacking`:** Realiza un ataque de web jacking durante un ataque de tabnabbing.

6. Web Jacking Attack Method:

- **`1. Credential Harvester Attack Method`:** Crea una página de phishing para robar credenciales durante un ataque de web jacking.

- **`2. Web Jacking Attack Method`:** Realiza un ataque de web jacking durante un ataque de web jacking.

- **`3. Multi-Attack Web Method`:** Combina varios métodos de ataque web.

7. Infectious Media Generator:

- **`1. PDF Attack`:** Genera un archivo PDF malicioso para infectar sistemas.

- **`2. Infectious PDF Embedded Executable`:** Genera un archivo PDF con un ejecutable incrustado para propagar infecciones.

- **`3. Microsoft Office Excel Formula Injection`:** Genera un archivo de Excel con una inyección de fórmula maliciosa.

- **`4. Microsoft Office Word RTF Exploit`:** Genera un archivo de Word con un exploit RTF para explotar vulnerabilidades.

8. Teensy USB HID Attack Vector:

- **`1. Teensy HID Attack Vector`:** Utiliza el dispositivo Teensy para realizar ataques USB HID.

9. Wireless Access Point Attack Vector:

- **`1. Wireless Access Point Attack Vector`:** Configura un punto de acceso inalámbrico malicioso para realizar ataques.

10. QRCode Generator Attack Vector:

- **`1. QRCode Generator`:** Genera códigos QR maliciosos para realizar ataques.

11. Powershell Attack Vectors:

- **`1. Powershell Attack Vector`:** Utiliza scripts de PowerShell para llevar a cabo ataques.

12. Third Party Modules:

- **`1. Import a Third Party Module`:** Importa un módulo de terceros para ampliar las funcionalidades de SET.

13. Update the Social Engineering Toolkit:

- **`1. Update the Social Engineering Toolkit`:** Actualiza la versión de SET a la más reciente.

14. Update the Metasploit Framework:

- **`1. Update the Metasploit Framework`:** Actualiza la versión de Metasploit Framework a la más reciente.

15. Report a Bug:

- **`1. Report a Bug`:** Informa de errores o problemas encontrados en SET.

16. Exit the Social Engineering Toolkit: Sale de Social Engineering Toolkit.

- `1.Exit.

Burp Suite

Burp Suite es una herramienta de prueba de seguridad líder en el mercado, utilizada para evaluar la seguridad de las aplicaciones web. Proporciona una interfaz gráfica fácil de usar y está diseñada para ser utilizada por profesionales de seguridad y probadores de penetración. Burp Suite consta de varias herramientas que permiten realizar diversas actividades relacionadas con la seguridad de las aplicaciones web, como la identificación de vulnerabilidades, la manipulación de solicitudes y respuestas, la interceptación y modificación del tráfico, y la generación de informes detallados. Con Burp Suite, los profesionales de seguridad pueden detectar y aprovechar las vulnerabilidades de las aplicaciones web para mejorar su seguridad.

A continuación se enumeran algunos de los componentes y características clave de Burp Suite:

1. Proxy: Permite interceptar y modificar el tráfico entre el navegador y la aplicación web.

2. Scanner: Realiza un escaneo automático en busca de vulnerabilidades conocidas en la aplicación web.

3. Intruder: Automatiza ataques de fuerza bruta, fuzzing y otros ataques personalizados.

4. Repeater: Permite realizar solicitudes HTTP/HTTPS repetidas y modificar parámetros para probar diferentes escenarios.

5. Sequencer: Analiza la aleatoriedad de los tokens generados por la aplicación web.

6. Decoder: Facilita la decodificación y codificación de datos en diferentes formatos.

7. Comparer: Compara dos solicitudes o respuestas para detectar diferencias sutiles.

8. Extender: Permite integrar complementos personalizados para ampliar la funcionalidad de Burp Suite.

9. Target: Proporciona herramientas para configurar y gestionar objetivos de prueba.

10. Spider: Explora automáticamente la aplicación web para descubrir nuevas páginas y funcionalidades.

11. Scanner de contenido activo: Analiza el contenido en busca de vulnerabilidades y errores de configuración.

12. Intruder de secuencia de comandos: Automatiza ataques utilizando secuencias de comandos personalizadas.

13. Buscador de contraseñas: Encuentra contraseñas débiles y vulnerabilidades de autenticación.

14. Mapper: Genera un mapa visual de la aplicación web y sus recursos.

15. Comparador de respuesta diferencial: Compara las respuestas de diferentes variantes de una solicitud.

16. Sequencer de token CSRF: Analiza la aleatoriedad de los tokens de protección contra CSRF.

17. Hinchador de cookies: Genera y prueba diferentes combinaciones de cookies.

18. Control de contenido: Manipula y prueba diferentes combinaciones de tipos de contenido.

19. Suite de colaboración: Permite a los equipos colaborar y compartir información en tiempo real.

SQLmap

SQLmap es una herramienta de prueba de penetración de código abierto que automatiza la detección y explotación de vulnerabilidades de inyección SQL en aplicaciones web. Está diseñada para ayudar a los profesionales de seguridad a evaluar la seguridad de las aplicaciones que interactúan con bases de datos SQL. SQLmap utiliza técnicas avanzadas para identificar y aprovechar las vulnerabilidades de inyección SQL, lo que permite extraer información confidencial de la base de datos, obtener acceso no autorizado o incluso ejecutar comandos arbitrarios en el sistema.

A continuación se presentan algunos de los comandos más comunes utilizados en SQLmap:

- `**sqlmap -u <URL>**`: Especifica la URL del objetivo para realizar la prueba de inyección SQL.

- `**sqlmap -r <ruta>**`: Especifica la ruta del archivo que contiene una solicitud HTTP capturada para realizar la prueba de inyección SQL.

- `**sqlmap -p <parámetro>**`: Especifica el parámetro vulnerable a la inyección SQL.

- `**sqlmap -b**`: Realiza una prueba de fuerza bruta en los campos de autenticación.

- `**sqlmap --dbs**`: Enumera las bases de datos disponibles en el servidor.

- `**sqlmap --current-db**`: Muestra la base de datos actual.

- `**sqlmap --tables**`: Enumera las tablas de una base de datos específica.

- `**sqlmap --columns**`: Enumera las columnas de una tabla específica.

- `**sqlmap --dump**`: Extrae los datos de una tabla específica.

- `**sqlmap --dump-all**`: Extrae los datos de todas las tablas en la base de datos.

- `**sqlmap --dump-format <formato>**`: Especifica el formato de salida para la extracción de datos.

- `**sqlmap --os-shell**`: Abre una shell interactiva en el sistema operativo subyacente.

- `**sqlmap --os-pwn**`: Intenta obtener acceso interactivo a través de comandos del sistema operativo.

- `**sqlmap --os-cmd <comando>**`: Ejecuta un comando en el sistema operativo subyacente.

- `**sqlmap --technique <técnica>**`: Especifica la técnica de inyección SQL a utilizar.

- `**sqlmap --level <nivel>**`: Especifica el nivel de intensidad de la prueba de inyección SQL.

- `**sqlmap --risk <riesgo>**`: Especifica el nivel de riesgo de la prueba de inyección SQL.

- `**sqlmap --tamper <script>**`: Especifica un script de manipulación de payloads para evadir filtros de seguridad.

- `**sqlmap --batch**`: Ejecuta SQLmap en modo batch sin interacción del usuario.

- `**sqlmap --flush-session**`: Limpia todas las sesiones y datos almacenados en cache de SQLmap.

- `**sqlmap --purge-output**`: Elimina todos los archivos de salida generados por SQLmap.

- `**sqlmap --wizard**`: Inicia el asistente interactivo de SQLmap para guiar en la configuración de los parámetros.

- `**sqlmap --check-waf**`: Detecta y verifica si hay un Web Application Firewall (WAF) en el objetivo.

Aircrack-ng

Aircrack-ng es una suite de herramientas de seguridad informática diseñada para evaluar y analizar la seguridad de redes inalámbricas. Esta potente herramienta proporciona funcionalidades para capturar paquetes de redes inalámbricas, realizar ataques de inyección de paquetes, desencriptar claves de cifrado WEP y WPA/WPA2, y llevar a cabo diferentes pruebas de seguridad en entornos WiFi. Aircrack-ng se utiliza comúnmente en auditorías de seguridad, pruebas de penetración y actividades relacionadas con la seguridad en redes inalámbricas. Su amplia gama de funcionalidades y su capacidad para analizar y evaluar la seguridad de redes inalámbricas lo convierten en una herramienta popular en el campo de la seguridad de la información.

Aquí tienes la lista completa de comandos existentes en Aircrack-ng:

- `**airmon-ng**`: Configura y controla las interfaces de red inalámbricas en modo monitor.

- `**airodump-ng <interfaz>**`: Captura y muestra información detallada de las redes inalámbricas.

- `**airodump-ng --channel <canal> <interfaz>**`: Captura y muestra información detallada de una red inalámbrica específica en un canal específico.

- `**airodump-ng --bssid <BSSID> <interfaz>**`: Captura y muestra información detallada de una red inalámbrica específica con la dirección BSSID especificada.

- `**aireplay-ng**`: Realiza ataques de inyección de paquetes en redes inalámbricas.

- `**aireplay-ng --deauth <número> -a <BSSID> -c <MAC>**`: Envía paquetes de desautenticación a un dispositivo específico en una red inalámbrica específica.

- `**aireplay-ng --arpreplay -b <BSSID> -h <MAC> <interfaz>**`: Realiza un ataque de reinyección ARP para desencriptar tráfico en una red WEP.

- `**aircrack-ng <archivo.cap>**`: Utiliza técnicas de fuerza bruta o diccionario para recuperar claves de cifrado WEP y WPA/WPA2.

- `**airdecap-ng -e <ESSID> -p <clave> <archivo.cap>**`: Desencripta capturas de tráfico cifrado utilizando la clave especificada.

- `**airbase-ng**`: Configura una estación base falsa para llevar a cabo ataques de redireccionamiento y captura de contraseñas.

- `**airbase-ng --essid <ESSID> -c <canal> <interfaz>**`: Configura una estación base falsa con un ESSID y canal específicos.

- `**airdecloak-ng <archivo.cap>**`: Elimina el ocultamiento de caracteres especiales en la captura de tráfico cifrado.

- `**airgraph-ng <archivo.cap>**`: Genera gráficos y visualizaciones a partir de archivos de captura de tráfico.

- `**airserv-ng**`: Permite controlar y manipular varios aspectos de Aircrack-ng a través de una interfaz web.

- `**airtun-ng**`: Crea interfaces virtuales para el enrutamiento de tráfico a través de redes inalámbricas.

- `**packetforge-ng**`: Genera paquetes personalizados para su uso en ataques de inyección.

- `**ivstools**`: Herramientas para la manipulación y análisis de capturas de tráfico cifrado.

- `**wpaclean <archivo.cap>**`: Limpia archivos de captura de tráfico WPA/WPA2 eliminando paquetes innecesarios.

- `**wesside-ng**`: Realiza ataques de fuerza bruta a contraseñas WEP en redes inalámbricas.

- `**besside-ng**`: Captura los handshakes WPA/WPA2 necesarios para realizar ataques de fuerza bruta.

John the Ripper

John the Ripper es una herramienta de cracking de contraseñas que se utiliza para auditar y recuperar contraseñas olvidadas o perdidas. Es una herramienta de línea de comandos altamente versátil que admite una amplia gama de formatos de contraseñas y técnicas de ataque. John the Ripper puede realizar ataques de fuerza bruta, ataques de diccionario y ataques de combinación, utilizando reglas y transformaciones personalizables para optimizar la velocidad y la efectividad del cracking. Es ampliamente utilizada en pruebas de seguridad y auditorías para evaluar la fortaleza de las contraseñas y garantizar la seguridad de los sistemas.

Aquí tienes la lista completa de comandos existentes en John the Ripper:

- `**john <archivo>**`: Inicia el proceso de cracking utilizando el archivo de contraseñas especificado.

- `**john --wordlist=<archivo>**`: Realiza un ataque de fuerza bruta utilizando una lista de palabras predefinida.

- `**john --incremental[=modo]**`: Realiza un ataque de fuerza bruta incremental, probando todas las combinaciones posibles.

- `**john --rules**`: Aplica reglas de transformación a las palabras de la lista de contraseñas.

- `**john --stdout**`: Imprime las contraseñas descifradas en la salida estándar.

- `**john --show <archivo>**`: Muestra las contraseñas descifradas encontradas en el archivo.

- `**john --test**`: Realiza una prueba de rendimiento para medir la velocidad de cracking.

- `**john --make-charset=<charset>**`: Crea un archivo de conjunto de caracteres personalizado para realizar ataques de fuerza bruta.

- `**john --mask=<máscara>**`: Utiliza una máscara de caracteres para generar todas las combinaciones posibles.

- `**john --session=<nombre>**`: Guarda el estado actual del cracking en una sesión específica.

- `**john --status[=intervalo]**`: Muestra el estado actual del cracking.

- `**john --single**`: Realiza un ataque de fuerza bruta en una sola contraseña.

- `**john --fork=<número>**`: Especifica el número de procesos en paralelo para acelerar el cracking.

- `**john --format=NT**`: Especifica el formato de las contraseñas de Windows NTLM.

- `**john --format=md5crypt**`: Especifica el formato de las contraseñas en formato md5crypt.

- `**john --format=sha512crypt**`: Especifica el formato de las contraseñas en formato sha512crypt.

- `**john --format=bcrypt**`: Especifica el formato de las contraseñas en formato bcrypt.

- `**john --format=zip**`: Especifica el formato de las contraseñas en archivos ZIP.

- `**john --format=PDF**`: Especifica el formato de las contraseñas en archivos PDF.

- `**john --format=SSH**`: Especifica el formato de las contraseñas en claves SSH.

- `**john --format=bitcoin**`: Especifica el formato de las contraseñas en carteras de Bitcoin.

- `**john --format=raw-md5**`: Especifica el formato de las contraseñas en formato MD5 sin sal.

Hydra

Hydra es una herramienta de cracking de contraseñas que se utiliza para realizar ataques de fuerza bruta o diccionario en servicios remotos. Puede ser utilizada para probar la seguridad de contraseñas débiles y ayudar en la auditoría de sistemas. Hydra es capaz de realizar ataques en una amplia variedad de protocolos y servicios, incluyendo FTP, SSH, HTTP, SMTP, entre otros. Proporciona flexibilidad y personalización a través de una amplia gama de opciones y parámetros para adaptarse a diferentes escenarios de cracking. Hydra es una herramienta poderosa, pero debe ser utilizada de manera responsable y legal, con el permiso del propietario del sistema objetivo.

Lista de comandos existentes en Hydra:

- `**hydra -l <usuario> -P <archivo> <servicio>://<objetivo>**`: Realiza un ataque de fuerza bruta utilizando una lista de contraseñas específica para el usuario y el servicio especificado en el objetivo.

- `**hydra -L <archivo> -P <archivo> <servicio>://<objetivo>**`: Realiza un ataque de fuerza bruta utilizando listas de usuarios y contraseñas específicas para el servicio especificado en el objetivo.

- `**hydra -C <archivo> <servicio>://<objetivo>**`: Realiza un ataque de fuerza bruta utilizando un archivo que contiene combinaciones de usuario:contraseña para el servicio especificado en el objetivo.

- `**hydra -M <archivo> <servicio>://<objetivo>**`: Realiza un ataque de fuerza bruta utilizando múltiples objetivos especificados en un archivo.

- `**hydra -t <número> <servicio>://<objetivo>**`: Especifica el número de hilos simultáneos para acelerar el ataque de fuerza bruta.

- `**hydra -e <ns>**`: Establece un tiempo de espera en segundos entre intentos de autenticación.

- `**hydra -F**`: Utiliza una estrategia de fuerza bruta más inteligente para probar las contraseñas más probables primero.

- `**hydra -x <min>:<max>:<paso>**`: Realiza un ataque de fuerza bruta utilizando una longitud de contraseña variable, especificando el rango mínimo, máximo y el paso.

- `**hydra -o <archivo>**`: Guarda los resultados del ataque en un archivo.

- `**hydra -U**`: Utiliza un archivo de usuarios predefinido que contiene nombres de usuario comunes.

- `**hydra -p <contraseña>**`: Especifica una única contraseña para probar contra los usuarios o servicios objetivo.

- `**hydra -s <puerto>**`: Especifica un puerto de destino para el servicio objetivo.

- `**hydra -v**`: Habilita el modo verboso y muestra información detallada durante el ataque.

- `**hydra -w <tiempo>**`: Establece un tiempo de espera global en segundos para todas las solicitudes de autenticación.

- `**hydra -I**`: Ignora los mensajes de advertencia durante el ataque de fuerza bruta.

- `**hydra -R**`: Restringe el acceso a través de un proxy especificado.

- `**hydra -S**`: Utiliza conexiones seguras SSL/TLS para el ataque.

- `**hydra -M <archivo> -o <archivo>**`: Guarda los resultados del ataque en un archivo específico para cada objetivo.

- `**hydra -x <min>:<max>:<paso> <servicio>://<objetivo>**`: Realiza un ataque de fuerza bruta utilizando una longitud de contraseña variable, especificando el rango mínimo, máximo y el paso para el servicio objetivo.

Hashcat

Hashcat es una poderosa herramienta de recuperación de contraseñas que permite realizar ataques de fuerza bruta o diccionario para descifrar contraseñas encriptadas. Es capaz de manejar una amplia variedad de algoritmos de hash y métodos de cifrado, lo que la convierte en una opción popular para probar la seguridad de contraseñas y sistemas. Hashcat utiliza el poder de procesamiento de la GPU y la CPU para acelerar el proceso de cracking de contraseñas y ofrece una amplia gama de opciones y modos de ataque para adaptarse a diferentes escenarios. Es importante utilizar Hashcat de manera legal y ética, respetando siempre los derechos y la privacidad de los demás.

Lista de comandos existentes en Hashcat:

- `**hashcat -m <modo> <archivo-hash> <archivo-diccionario>**`: Realiza un ataque de fuerza bruta utilizando el archivo hash y un archivo diccionario.

- `**hashcat -m <modo> <archivo-hash> -a <tipo-ataque>**`: Realiza un ataque utilizando el archivo hash y un tipo específico de ataque.

- `**hashcat -m <modo> <archivo-hash> -a <tipo-ataque> <archivo-máscara>**`: Realiza un ataque utilizando el archivo hash y una máscara específica.

- `**hashcat -b**`: Realiza una prueba de benchmark para medir el rendimiento de Hashcat en el sistema.

- `**hashcat -m <modo> --show**`: Muestra las contraseñas descifradas encontradas para el modo de hash especificado.

- `**hashcat --stdout**`: Imprime las contraseñas descifradas en la salida estándar.

- `**hashcat --keyspace**`: Calcula el espacio de claves para el archivo hash y el tipo de hash especificados.

- `**hashcat --speed-only**`: Muestra solo la velocidad de cracking sin otro resultado.

- `**hashcat --increment**`: Realiza un ataque de fuerza bruta incremental, probando todas las combinaciones posibles.

- `**hashcat --attack-mode=<modo-ataque>**`: Especifica el modo de ataque a utilizar, como ataque de diccionario, ataque de máscara, etc.

- `**hashcat --optimized-kernel-enable**`: Habilita el uso de núcleos de hash optimizados para mejorar el rendimiento.

- `**hashcat --username**`: Habilita la opción de búsqueda de nombres de usuario en el archivo de contraseñas.

- `**hashcat --session=<nombre>**`: Guarda el estado actual del cracking en una sesión específica.

- `**hashcat --restore**`: Restaura una sesión guardada anteriormente.

- `**hashcat --status**`: Muestra el estado actual del cracking.

- `**hashcat --remove**`: Elimina hashes descifrados y no descifrados del archivo hash especificado.

- `**hashcat --outfile=<archivo-salida>**`: Especifica un archivo de salida para guardar las contraseñas descifradas encontradas.

- `**hashcat --potfile-path=<archivo-pot>**`: Especifica la ubicación del archivo pot para guardar los hashes descifrados y las contraseñas correspondientes.

4. Análisis de Vulnerabilidades y Escaneo de Aplicaciones Web

OWASP Zap

OWASP ZAP (Zed Attack Proxy) es una herramienta de seguridad de aplicaciones web de código abierto que permite encontrar y mitigar vulnerabilidades en aplicaciones web. Ofrece un conjunto de funcionalidades avanzadas para el escaneo, análisis y explotación de aplicaciones web, ayudando a fortalecer la seguridad de los sistemas y proteger contra posibles ataques. OWASP ZAP es ampliamente utilizado y recomendado en la industria de la seguridad de aplicaciones como una herramienta esencial para evaluar y mejorar la seguridad de las aplicaciones web.

Lista de comandos para la herramienta OWASP ZAP (Zed Attack Proxy):

- `**zap.sh**`: Inicia OWASP ZAP en modo gráfico.

- `**zap.sh -cmd**`: Inicia OWASP ZAP en modo de línea de comandos.

- `**zap.sh -daemon**`: Inicia OWASP ZAP en modo de demonio (headless) sin interfaz gráfica.

- `**zap.sh -quickurl <URL>**`: Inicia OWASP ZAP y realiza un escaneo rápido de la URL especificada.

- `**zap.sh -quickscan <URL>**`: Realiza un escaneo rápido de la URL especificada sin iniciar la interfaz gráfica.

- `**zap.sh -scan <URL>**`: Realiza un escaneo completo de la URL especificada.

- `**zap.sh -spider <URL>**`: Realiza un rastreo (spidering) de la URL especificada para descubrir enlaces y contenido relacionado.

- `**zap.sh -ajaxspider <URL>**`: Realiza un rastreo (spidering) AJAX de la URL especificada para descubrir contenido generado dinámicamente.

- `**zap.sh -port <puerto>**`: Especifica el puerto en el que se ejecutará OWASP ZAP.

- `**zap.sh -importsession <archivo-sesión>**`: Importa una sesión guardada previamente.

- `**zap.sh -exportreport <archivo-reporte> -format <formato>**`: Exporta un informe en el formato especificado (por ejemplo, HTML, XML, JSON).

- `**zap.sh -cmd -cmdf <archivo-comandos>**`: Ejecuta comandos desde un archivo de comandos.

- `**zap-cli**`: Interfaz de línea de comandos (CLI) para OWASP ZAP.

- `**zap-cli -p <puerto>**`: Especifica el puerto en el que se ejecuta OWASP ZAP.

- `**zap-cli -cmd <comando>**`: Ejecuta un comando específico de OWASP ZAP a través de la línea de comandos.

- `**zap-cli -addoninstall <nombre-addon>**`: Instala un complemento (addon) para OWASP ZAP.

Nikto

Nikto es una herramienta de escaneo de vulnerabilidades de código abierto diseñada para realizar un análisis exhaustivo de la seguridad de aplicaciones web. Utiliza una amplia base de datos de firmas y pruebas de seguridad para detectar posibles vulnerabilidades en servidores web y aplicaciones en funcionamiento. Nikto busca activamente diferentes tipos de problemas de seguridad, como configuraciones incorrectas del servidor, archivos y directorios confidenciales expuestos, versiones obsoletas de software y otras vulnerabilidades conocidas. Proporciona un informe detallado de los hallazgos, lo que permite a los administradores de sistemas y auditores de seguridad tomar medidas para fortalecer la seguridad y proteger los sistemas web contra posibles ataques. Nikto es una herramienta valiosa para realizar auditorías de seguridad y asegurar que los sistemas web estén protegidos de manera adecuada.

Lista de comandos existentes para la herramienta Nikto:

- `**nikto -h <host>**`: Realiza un escaneo en el host especificado.

- `**nikto -h <host> -p <puerto>**`: Realiza un escaneo en el host y puerto especificados.

- `**nikto -h <host> -ssl**`: Realiza un escaneo en el host utilizando una conexión SSL.

- `**nikto -h <host> -p <puerto> -ssl**`: Realiza un escaneo en el host y puerto especificados utilizando una conexión SSL.

- `**nikto -h <host> -C <config-file>**`: Utiliza un archivo de configuración personalizado para el escaneo en el host especificado.

- `**nikto -list-plugins**`: Muestra una lista de todos los plugins disponibles en Nikto.

- `**nikto -Display <option>**`: Especifica las opciones de visualización, como "Cert" para mostrar información del certificado SSL.

- `**nikto -Format <output-format>**`: Especifica el formato de salida para los resultados del escaneo, como "csv", "xml" o "html".

- `**nikto -output <output-file>**`: Guarda los resultados del escaneo en un archivo de salida especificado.

- `**nikto -dbcheck**`: Realiza una verificación de la base de datos de Nikto para asegurarse de que está actualizada.

- `**nikto -update**`: Actualiza la base de datos de Nikto con las últimas firmas de vulnerabilidades.

- `**nikto -Plugins <plugin(s)>**`: Especifica los plugins a ejecutar durante el escaneo.

- `**nikto -id <custom-header>**`: Especifica un encabezado personalizado para enviar con las solicitudes del escaneo.

- `**nikto -evasion <technique>**`: Especifica una técnica de evasión para evitar la detección durante el escaneo.

- `**nikto -Tuning <tuning-option>**`: Especifica una opción de ajuste para personalizar el escaneo, como "paranoid", "stealth" o "aggressive".

- `**nikto -list-plugins**`: Muestra una lista de todos los plugins disponibles en Nikto.

- `**nikto -help**`: Muestra la ayuda y los comandos disponibles en Nikto.

w3af (Web Application Attack and Audit Framework)

W3af (Web Application Attack and Audit Framework) es una herramienta de código abierto diseñada para realizar pruebas de seguridad y auditorías en aplicaciones web. Proporciona un conjunto de funciones y plugins que permiten identificar y explotar vulnerabilidades en aplicaciones web, como inyecciones de SQL, ataques de cross-site scripting (XSS), inclusión de archivos locales y muchas otras. W3af se destaca por su enfoque modular y flexible, lo que permite personalizar el escaneo de acuerdo con las necesidades del proyecto. Además, ofrece una interfaz de consola interactiva que facilita la configuración de los parámetros del escaneo y la generación de informes detallados de los hallazgos. W3af es una herramienta valiosa para evaluar la seguridad de las aplicaciones web y ayudar a los profesionales de seguridad a identificar y solucionar las vulnerabilidades antes de que sean explotadas por atacantes.

Lista de comandos existentes para la herramienta W3af (Web Application Attack and Audit Framework):

- `**w3af_console**`: Inicia la consola interactiva de W3af.

- `**w3af_console -s <script>**`: Ejecuta un script específico en la consola de W3af.

- `**w3af_console -g <profile>**`: Ejecuta un perfil predefinido en la consola de W3af.

- `**w3af_console -y**`: Ejecuta en modo silencioso, sin mensajes de advertencia.

- `**w3af_console -l <log-file>**`: Especifica un archivo de registro para guardar los resultados del escaneo.

- `**w3af_console -D**`: Activa el modo de depuración para obtener información detallada durante el escaneo.

- `**w3af_console -B <batch-file>**`: Ejecuta un archivo por lotes que contiene comandos para la consola de W3af.

- `**w3af_console -P <profile-file>**`: Ejecuta un archivo de perfil personalizado en la consola de W3af.

- `**w3af_console -x**`: Salir después de ejecutar el script o perfil especificado.

- `**w3af_console -c "<command>**"`: Ejecuta un comando específico en la consola de W3af.

- `**w3af_console -p <plugin>[:<option>]**`: Activa un plugin específico con opciones personalizadas.

- `**w3af_console -P <plugin>[:<option>]**`: Desactiva un plugin específico.

- `**w3af_console -u <target-url>**`: Establece el objetivo de escaneo como una URL específica.

- `**w3af_console -A**`: Inicia un escaneo automático sin interacción del usuario.

- `**w3af_console -N**`: No inicia el navegador web después del escaneo.

- `**w3af_console -R**`: No muestra el progreso del escaneo en tiempo real.

- `**w3af_console -H**`: Muestra información sobre el uso de comandos de la consola de W3af.

WPScan (Escáner de seguridad para WordPress)

WPScan es un escáner de seguridad diseñado específicamente para WordPress. Permite realizar escaneos exhaustivos en sitios web de WordPress para detectar posibles vulnerabilidades y fortalecer la seguridad. WPScan ofrece una amplia gama de comandos y opciones para enumerar usuarios, plugins, temas, plantillas de página y vulnerabilidades conocidas. También permite personalizar la ruta de los directorios "wp-content" y "wp-plugins" y utilizar un token de API para el escaneo. Con WPScan, puedes realizar ataques de fuerza bruta en nombres de usuario y contraseñas utilizando una lista de palabras personalizada. Además, ofrece opciones para utilizar un servidor proxy, especificar un agente de usuario personalizado y seguir las redirecciones durante el escaneo. Recuerda utilizar WPScan de manera ética y con el consentimiento del propietario del sitio web de WordPress, con el objetivo de mejorar la seguridad y proteger contra posibles vulnerabilidades.

Lista de comandos existentes para la herramienta WPScan (Escáner de seguridad para WordPress):

- `**wpscan --url <url>**`: Realiza un escaneo de seguridad en la URL especificada.

- `**wpscan --url <url> --enumerate**`: Enumera las diferentes partes de un sitio web de WordPress, como usuarios, plugins, temas, etc.

- `**wpscan --url <url> --enumerate u**`: Enumera los usuarios registrados en el sitio de WordPress.

- `**wpscan --url <url> --enumerate p**`: Enumera los plugins instalados en el sitio de WordPress.

- `**wpscan --url <url> --enumerate t**`: Enumera los temas instalados en el sitio de WordPress.

- `**wpscan --url <url> --enumerate tt**`: Enumera las plantillas de página instaladas en el sitio de WordPress.

- `**wpscan --url <url> --enumerate vt**`: Enumera las vulnerabilidades conocidas en los temas instalados en el sitio de WordPress.

- `**wpscan --url <url> --enumerate vp**`: Enumera las vulnerabilidades conocidas en los plugins instalados en el sitio de WordPress.

- `**wpscan --url <url> --enumerate vt**`: Enumera las vulnerabilidades conocidas en los temas instalados en el sitio de WordPress.

- `**wpscan --url <url> --enumerate vt,tt,u,p**`: Enumera usuarios, plugins, temas y plantillas de página instalados en el sitio de WordPress.

- `**wpscan --url <url> --api-token <token>**`: Utiliza un token de API para realizar el escaneo. Requiere una cuenta de WPScan.

- `**wpscan --url <url> --wordlist <wordlist>**`: Utiliza un archivo de lista de palabras personalizado para realizar ataques de fuerza bruta en nombres de usuario y contraseñas.

- `**wpscan --url <url> --wp-content-dir <wp-content-dir>**`: Especifica la ruta personalizada al directorio "wp-content" en el sitio de WordPress.

- `**wpscan --url <url> --wp-plugins-dir <wp-plugins-dir>**`: Especifica la ruta personalizada al directorio "wp-plugins" en el sitio de WordPress.

- `**wpscan --url <url> --proxy <proxy>**`: Especifica un servidor proxy para el escaneo.

- `**wpscan --url <url> --user-agent <user-agent>**`: Especifica un agente de usuario personalizado para las solicitudes HTTP.

- `**wpscan --url <url> --follow-redirection**`: Sigue las redirecciones en las solicitudes HTTP durante el escaneo.

- `**wpscan --url <url> --disable-tls-checks**`: Desactiva la verificación de certificados TLS/SSL durante el escaneo.

SSLScan

SSLScan es una herramienta de línea de comandos que se utiliza para realizar escaneos de seguridad en servidores que utilizan el protocolo SSL/TLS. Su objetivo principal es evaluar la configuración y la seguridad de los servicios SSL/TLS, como servidores web, servidores de correo electrónico y otros servicios que utilizan cifrado SSL/TLS. SSLScan realiza un escaneo exhaustivo de los protocolos SSL/TLS compatibles y sus configuraciones, y muestra información detallada sobre los protocolos, cifrados, certificados y vulnerabilidades potenciales encontradas. Esto ayuda a identificar posibles debilidades en la implementación de SSL/TLS y a tomar medidas para fortalecer la seguridad de los servicios en línea.

Lista de todos los comandos existentes para el programa SSLScan:

- `**sslscan <host>[:<port>]**`: Realiza un escaneo SSL en el host y puerto especificados.

- `**sslscan --no-failed <host>[:<port>]**`: Realiza un escaneo SSL y solo muestra las conexiones exitosas.

- `**sslscan --no-color <host>[:<port>]**`: Realiza un escaneo SSL y muestra los resultados sin colores.

- `**sslscan --xml=<file> <host>[:<port>]**`: Realiza un escaneo SSL y guarda los resultados en formato XML en el archivo especificado.

- `**sslscan --csv=<file> <host>[:<port>]**`: Realiza un escaneo SSL y guarda los resultados en formato CSV en el archivo especificado.

- `**sslscan --show-certificate <host>[:<port>]**`: Realiza un escaneo SSL y muestra los detalles del certificado SSL.

- `**sslscan --show-ciphers <host>[:<port>]**`: Realiza un escaneo SSL y muestra los detalles de los ciphers (cifrados) soportados.

- `**sslscan --show-heartbleed <host>[:<port>]**`: Realiza un escaneo SSL y muestra si el servidor es vulnerable a la vulnerabilidad de Heartbleed.

- `**sslscan --version**`: Muestra la versión de SSLScan instalada.

- `**sslscan --help**`: Muestra la ayuda y la lista de opciones disponibles.

5.Forense Digital

Autopsy

Autopsy es una herramienta forense digital de código abierto que proporciona un conjunto completo de capacidades para realizar investigaciones forenses en sistemas informáticos. Con su interfaz gráfica intuitiva, Autopsy permite a los investigadores examinar imágenes de disco, archivos y metadatos para descubrir y analizar evidencias digitales relevantes. Proporciona funciones de búsqueda, filtrado y visualización avanzadas, así como análisis automatizados para identificar archivos sospechosos, extraer información oculta y generar informes detallados. Autopsy es ampliamente utilizado por profesionales de la seguridad y forenses para llevar a cabo investigaciones forenses digitales y ayudar en la resolución de delitos cibernéticos.

Lista de todos los comandos existentes para la herramienta Autopsy:

- `**autopsy**`: Inicia la interfaz gráfica de Autopsy.

- `**autopsy --version**`: Muestra la versión de Autopsy instalada.

- `**autopsy --help**`: Muestra la ayuda y la lista de opciones disponibles.

- `**autopsy --profile <profile>**`: Inicia Autopsy utilizando un perfil específico.

- `**autopsy --port <port>**`: Especifica el puerto en el que se ejecutará Autopsy.

- `**autopsy --start <module>**`: Inicia Autopsy y abre directamente el módulo especificado.

- `**autopsy --export**`: Exporta el informe actual en formato HTML.

- `**autopsy --import <report_file>**`: Importa un informe previamente exportado en Autopsy.

- `**autopsy --add-image <image_file>**`: Agrega una imagen de disco para su análisis en Autopsy.

- `**autopsy --add-image <image_file> --password <password>**`: Agrega una imagen de disco protegida con contraseña para su análisis en Autopsy.

- `**autopsy --add-image <image_file> --password-file <password_file>**`: Agrega una imagen de disco protegida con contraseña almacenada en un archivo para su análisis en Autopsy.

- `**autopsy --delete-image <image_id>**`: Elimina una imagen de disco del caso actual en Autopsy.

- `**autopsy --import-keyword <keyword_file>**`: Importa una lista de palabras clave para su uso en la función de búsqueda en Autopsy.

- `**autopsy --export-keyword <export_file>**`: Exporta la lista de palabras clave actual en un archivo.

- `**autopsy --import-extension <extension_file>**`: Importa una lista de extensiones de archivo para su uso en el análisis de archivos en Autopsy.

- `**autopsy --export-extension <export_file>**`: Exporta la lista de extensiones de archivo actual en un archivo.

Volatility

Volatility es una herramienta de análisis de memoria ampliamente utilizada en investigaciones forenses digitales. Permite examinar imágenes de memoria y extraer información valiosa sobre procesos, conexiones de red, registros y otros artefactos relacionados. Con una amplia gama de comandos, Volatility brinda la capacidad de identificar procesos ocultos, analizar la actividad de red, buscar archivos abiertos y analizar el historial de navegación web, entre otras funciones. Además, se pueden aplicar reglas YARA para detectar patrones de malware en la memoria.

Lista de todos los comandos existentes para la herramienta Volatility:

- `**volatility -f <memory_image>**`: Analiza la imagen de memoria especificada.

- `**volatility -f <memory_image> imageinfo**`: Muestra información básica sobre la imagen de memoria.

- `**volatility -f <memory_image> kdbgscan**`: Escanea la imagen de memoria en busca del valor de depuración del kernel (KDBG).

- `**volatility -f <memory_image> kpcrscan**`: Escanea la imagen de memoria en busca del Registro de control de procesos (PCR).

- `**volatility -f <memory_image> pslist**`: Enumera todos los procesos en la imagen de memoria.

- `**volatility -f <memory_image> pstree**`: Muestra los procesos en forma de árbol en la imagen de memoria.

- `**volatility -f <memory_image> psscan**`: Escanea la imagen de memoria en busca de objetos de proceso ocultos.

- `**volatility -f <memory_image> cmdline**`: Muestra los comandos utilizados para ejecutar procesos en la imagen de memoria.

- `**volatility -f <memory_image> consoles**`: Muestra información sobre las consolas utilizadas por los procesos en la imagen de memoria.

- `**volatility -f <memory_image> getsids**`: Enumera los identificadores de seguridad (SIDs) asociados a los procesos en la imagen de memoria.

- `**volatility -f <memory_image> dlllist**`: Muestra las DLL cargadas en cada proceso en la imagen de memoria.

- `**volatility -f <memory_image> handles**`: Enumera los descriptores de archivos y registros asociados a los procesos en la imagen de memoria.

- `**volatility -f <memory_image> filescan**`: Escanea la imagen de memoria en busca de archivos abiertos por los procesos.

- `**volatility -f <memory_image> iehistory**`: Muestra el historial de navegación web en Internet Explorer en la imagen de memoria.

- `**volatility -f <memory_image> connections**`: Muestra las conexiones de red activas en la imagen de memoria.

- `**volatility -f <memory_image> netscan**`: Escanea la imagen de memoria en busca de conexiones de red.

- `**volatility -f <memory_image> sockets**`: Enumera los sockets de red abiertos por los procesos en la imagen de memoria.

- `**volatility -f <memory_image> malfind**`: Escanea la imagen de memoria en busca de procesos sospechosos o malware.

- `**volatility -f <memory_image> yarascan --yara-file=<yara_rule_file>**`: Escanea la imagen de memoria utilizando reglas YARA para detectar patrones de malware.

- `**volatility -f <memory_image> printkey --key=<registry_key>**`: Muestra los valores de un registro específico en la imagen de memoria.

Foremost

Foremost es una herramienta de recuperación de archivos diseñada para buscar y extraer datos eliminados o perdidos de dispositivos de almacenamiento. Utiliza firmas y patrones predefinidos para identificar y recuperar una amplia variedad de tipos de archivos, como imágenes, documentos, videos, audios y más. Foremost es especialmente útil en investigaciones forenses digitales, ya que puede ayudar a recuperar información importante de dispositivos dañados o comprometidos. Es una herramienta de línea de comandos fácil de usar y altamente configurable, que ofrece opciones para especificar el directorio de salida, utilizar archivos de configuración personalizados y mucho más. Con su capacidad para realizar una recuperación de archivos completa, Foremost se ha convertido en una herramienta popular en el campo de la recuperación de datos y la investigación forense digital.

Lista de todos los comandos existentes para la herramienta Foremost:

- `**foremost -h**`: Muestra la ayuda y la lista de opciones disponibles.

- `**foremost -V**`: Muestra la versión de Foremost instalada.

- `**foremost -a**`: Realiza una recuperación de archivos completa.

- `**foremost -v**`: Habilita la salida detallada durante la ejecución.

- `**foremost -Q**`: Desactiva la salida en la consola, solo muestra los resultados en el archivo log.

- `**foremost -o <output_directory>**`: Especifica el directorio de salida para almacenar los archivos recuperados.

- `**foremost -c <config_file>**`: Especifica un archivo de configuración personalizado.

- `**foremost -i <input_file>**`: Especifica el archivo o dispositivo de entrada para realizar la recuperación de archivos.

- `**foremost -T**`: Muestra una lista de los tipos de archivo soportados por Foremost.

- `**foremost -X**`: Muestra una lista de las extensiones de archivo soportadas por Foremost.

- `**foremost -Qb**`: Desactiva la creación de archivos de resumen.

- `**foremost -w**`: Habilita la búsqueda en archivos binarios.

- `**foremost -s <sector_size>**`: Especifica el tamaño del sector en bytes.

- `**foremost -k <keyword_file>**`: Especifica un archivo de palabras clave personalizado para la búsqueda de firmas.

- `**foremost -r <file_range>**`: Especifica un rango de bytes para la recuperación de archivos.

- `**foremost -M <memory_size>**`: Especifica el tamaño máximo de memoria a utilizar.

Sleuth Kit

Sleuth Kit es una herramienta forense de código abierto utilizada para el análisis de imágenes de disco y sistemas de archivos. Proporciona un conjunto de comandos que permiten examinar metadatos, extraer archivos y realizar análisis forenses en una amplia gama de sistemas de archivos, como NTFS, FAT, HFS, HFS+, Ext, entre otros. Con Sleuth Kit, los investigadores forenses pueden obtener información valiosa de las imágenes de disco, como la lista de archivos, metadatos, registros de actividad y archivos eliminados o perdidos. La herramienta también incluye funcionalidades para la recuperación de datos y la búsqueda de información oculta en los sistemas de archivos. Sleuth Kit es ampliamente utilizado en la comunidad forense digital debido a su poder y flexibilidad para investigaciones forenses.

Lista de todos los comandos existentes para la herramienta Sleuth Kit:

- `**fsstat <image>**`: Muestra información del sistema de archivos en la imagen.

- `**fls <image>**`: Lista los archivos y directorios en el sistema de archivos.

- `**ils <image>**`: Lista los archivos y directorios junto con sus metadatos.

- `**istat <image> <inode_number>**`: Muestra los metadatos de un archivo o directorio específico.

- `**icat <image> <inode_number>**`: Extrae el contenido de un archivo específico.

- `**ifind <image> <file_name>**`: Encuentra inodos asociados a un archivo por nombre.

- `**img_stat <image>**`: Muestra información sobre la imagen de disco.

- `**img_cat <image> <sector_number>**`: Muestra el contenido de un sector específico en la imagen.

- `**mmls <image>**`: Muestra la disposición de las particiones en la imagen.

- `**mmstat <image> <partition_number>**`: Muestra información del sistema de archivos en una partición específica.

- `**mmcat <image> <partition_number> <file_path>**`: Muestra el contenido de un archivo en una partición específica.

- `**mmfind <image> <partition_number> <file_name>**`: Encuentra archivos por nombre en una partición específica.

- `**mmstat <image> <partition_number> <inode_number>**`: Muestra los metadatos de un archivo o directorio en una partición específica.

- `**hfind <image>**`: Encuentra archivos ocultos en el sistema de archivos.

- `**hfsstat <image>**`: Muestra información del sistema de archivos HFS en la imagen.

- **`hfsplus`, `ntfs`, `ext4` (y otros)**: Muestra información del sistema de archivos específico.

- `**tsk_loaddb -d <database_file>**`: Carga una base de datos de Sleuth Kit.

- `**tsk_recover -a <output_directory> <image>**`: Recupera archivos eliminados o perdidos en la imagen.

Guymager

Guymager es una herramienta de código abierto utilizada en investigaciones forenses digitales para la adquisición y análisis de imágenes de dispositivos de almacenamiento. Proporciona una interfaz gráfica intuitiva que permite a los investigadores realizar copias forenses de manera eficiente y segura, preservando la integridad de la evidencia digital. Guymager ofrece diversas funciones, como la gestión de proyectos, exploración de dispositivos, análisis de archivos y generación de informes. Es una herramienta confiable y ampliamente utilizada en el campo de la informática forense para respaldar la investigación y el análisis de evidencia digital.

Lista completa de todos los comandos existentes para la herramienta Guymager:

- `**guymager**`: Inicia la interfaz gráfica de Guymager.

- `**guymager -h**`: Muestra la ayuda y la lista de opciones disponibles.

- `**guymager -v**`: Muestra la versión de Guymager instalada.

- `**guymager -s <source_device>**`: Especifica el dispositivo de origen para crear una imagen forense.

- `**guymager -r <raw_image_file>**`: Especifica el archivo de imagen RAW para abrirlo en Guymager.

- `**guymager -p <project_file>**`: Abre un archivo de proyecto previamente guardado en Guymager.

- `**guymager -c <config_file>**`: Especifica un archivo de configuración personalizado.

- `**guymager -e**`: Habilita el modo experto con más opciones y configuraciones avanzadas.

- `**guymager -m**`: Muestra información sobre los módulos de Guymager.

- `**guymager -w**`: Abre la ventana de configuración para personalizar las opciones de Guymager.

- `**guymager -L**`: Muestra los registros de eventos y actividades en la ventana de registros.

- `**guymager -R**`: Muestra los informes generados por Guymager en la ventana de informes.

- `**guymager -D**`: Muestra información detallada sobre los dispositivos en la ventana de dispositivos.

- `**guymager -F**`: Muestra información detallada sobre los archivos(ventana de archivos)

- `**guymager -M**`: Muestra información detallada sobre los mapas de memoria en la ventana de mapas de memoria.

- `**guymager -B**`: Muestra información detallada sobre los bloques en la ventana de bloques.

- `**guymager -P**`: Muestra información detallada sobre los sectores en la ventana de sectores.

- `**guymager -E**`: Muestra información detallada sobre los errores en la ventana de errores.

- `**guymager -I**`: Muestra información detallada sobre las imágenes en la ventana de imágenes.

- `**guymager -T**`: Muestra información detallada sobre las tareas en la ventana de tareas.

- `**guymager -G**`: Muestra información detallada sobre las opciones de GUI en la ventana de GUI.

- `**guymager -O**`: Muestra información detallada sobre las opciones de línea de comandos en la ventana de opciones.

- `**guymager -U**`: Muestra información detallada sobre las actualizaciones en la ventana de actualizaciones.

- `**guymager -A**`: Muestra información detallada sobre los ajustes avanzados en la ventana de ajustes avanzados.

- `**guymager -X**`: Muestra información detallada sobre las extensiones en la ventana de extensiones.

6. Anonimato y Privacidad

TOR

Tor es una herramienta de software libre y de código abierto diseñada para garantizar la privacidad y el anonimato en línea. Utiliza una red de servidores distribuidos en todo el mundo para enrutar las comunicaciones de manera segura, ocultando la identidad y la ubicación del usuario. Tor proporciona una capa adicional de protección al cifrar el tráfico y hacerlo rebotar a través de múltiples nodos antes de llegar a su destino final, lo que dificulta el seguimiento y la vigilancia. Se utiliza ampliamente para acceder a contenido bloqueado, evitar la censura en línea y proteger la privacidad personal en Internet. Tor también permite el acceso a servicios ocultos en la red oscura, proporcionando una capa adicional de anonimato. Además de su funcionalidad básica, Tor ofrece una variedad de comandos y opciones que permiten personalizar su configuración y adaptarlo a las necesidades individuales de privacidad y seguridad.

Lista completa de todos los comandos existentes en la herramienta Tor:

- `**tor**`: Inicia el servicio Tor.

- `**tor -f <config_file>**`: Inicia el servicio Tor utilizando un archivo de configuración específico.

- `**tor --verify-config**`: Verifica la validez de la configuración de Tor sin iniciar el servicio.

- `**tor --list-fingerprint**`: Muestra la huella digital del nodo Tor actual.

- `**tor --hash-password <password>**`: Genera un hash de contraseña para usar en la configuración de autenticación.

- `**tor --keygen**`: Genera una nueva clave de encriptación de circuitos.

- `**tor --quiet**`: Ejecuta Tor en modo silencioso sin mostrar mensajes de registro.

- `**tor --hush**`: Ejecuta Tor en modo silencioso y oculta incluso los mensajes de advertencia.

- `**tor --version**`: Muestra la versión de Tor instalada.

- `**tor --help**`: Muestra la ayuda y la lista de opciones disponibles.

- `**tor-resolve <hostname>**`: Resuelve un nombre de host a través del servicio Tor.

- `**torify <command>**`: Ejecuta un comando específico a través del servicio Tor.

- `**torsocks <command>**`: Envuelve un comando específico en una conexión a través del servicio Tor.

- `**tor-gencert**`: Genera un certificado para usar en la autenticación de directorios.

- `**tor-gencert --create-identity-key**`: Crea una clave de identidad para usar en la autenticación de directorios.

- `**tor-gencert --sign-identity-key <keyfile>**`: Firma una clave de identidad para usar en la autenticación de directorios.

- `**tor-gencert --create-microdescriptor**`: Crea un microdescriptor para usar en la autenticación de directorios.

- `**tor-gencert --sign-microdescriptor <descriptor>**`: Firma un microdescriptor para usar en la autenticación de directorios.

- `**tor-gencert --create-link-certificate**`: Crea un certificado de enlace para usar en la autenticación de directorios.

- `**tor-gencert --sign-link-certificate <certfile>**`: Firma un certificado de enlace para usar en la autenticación de directorios.

- `**tor-gencert --create-auth-certificate**`: Crea un certificado de autenticación para usar en la autenticación de directorios.

- `**tor-gencert --sign-auth-certificate <certfile>**`: Firma un certificado de autenticación para usar en la autenticación de directorios.

JonDo

JonDo es una herramienta diseñada para brindar privacidad en línea al permitir a los usuarios mantener su anonimato y ocultar su dirección IP. Proporciona una interfaz gráfica y de línea de comandos para controlar el servicio JonDo, iniciar el navegador JonDoFox y configurar opciones como cadenas de mezcla, saltos y países de salida. También ofrece funciones de depuración, limpieza de archivos temporales y caché, actualización y visualización del estado. JonDo es una solución útil para aquellos que buscan proteger su privacidad y anonimato mientras navegan por Internet.

Lista completa de todos los comandos existentes en la herramienta JonDo:

- `**jondo**`: Inicia la interfaz gráfica de JonDo.
- `**jondoconsole**`: Inicia la interfaz de línea de comandos de JonDo.
- `**jondoctl**`: Controla el servicio JonDo.
- `**jondofox**`: Inicia el navegador JonDoFox.
- `**jondonym**`: Inicia el cliente JonDo.
- `**jondonym-console**`: Inicia el cliente JonDo en modo consola.
- `**jondonym-anonstart**`: Inicia el cliente JonDo en modo anónimo.
- `**jondonym-anonstop**`: Detiene el modo anónimo del cliente JonDo.
- `**jondonym-proxyon**`: Activa el proxy JonDo.
- `**jondonym-proxyoff**`: Desactiva el proxy JonDo.
- `**jondonym-status**`: Muestra el estado actual del cliente JonDo.
- `**jondonym-update**`: Actualiza el cliente JonDo.
- `**jondonym-chains**`: Muestra la lista de cadenas de mezcla disponibles.
- `**jondonym-chain <chainname>**`: Configura la cadena de mezcla deseada.

- `**jondonym-jump <jumpindex>**`: Configura el salto de mezcla deseado.

- `**jondonym-country <countrycode>**`: Configura el país de salida deseado.

- `**jondonym-reset**`: Restablece la configuración de JonDo a los valores predeterminados.

- `**jondonym-loglevel <level>**`: Configura el nivel de registro deseado.

- `**jondonym-debugon**`: Activa el modo de depuración.

- `**jondonym-debugoff**`: Desactiva el modo de depuración.

- `**jondonym-clean**`: Limpia los archivos temporales y la caché de JonDo.

- `**jondonym-version**`: Muestra la versión actual de JonDo.

- `**jondonym-help**`: Muestra la ayuda y la lista de comandos disponibles en JonDo.

BleachBit

BleachBit es una herramienta de software de código abierto diseñada para limpiar y mejorar la privacidad en sistemas informáticos. Permite eliminar de forma segura archivos temporales, cachés, historiales de navegación, cookies y otros datos no deseados generados por aplicaciones y navegadores web. Además de liberar espacio en disco, BleachBit ayuda a proteger la privacidad del usuario al eliminar rastros de actividad en línea y datos sensibles. También proporciona opciones avanzadas de limpieza, como la sobrescritura segura de archivos eliminados. Con su interfaz gráfica de usuario intuitiva y su potente capacidad de limpieza, BleachBit es una herramienta confiable para mantener la privacidad y el rendimiento óptimo de los sistemas informáticos.

Lista completa de todos los comandos existentes de la herramienta BleachBit:

- `**bleachbit**`: Inicia la interfaz gráfica de BleachBit.

- `**bleachbit-cli**`: Inicia la interfaz de línea de comandos de BleachBit.

- `**bleachbit -c <config_file>**`: Especifica un archivo de configuración personalizado.

- `**bleachbit -h` o `bleachbit --help**`: Muestra la ayuda y la lista de comandos disponibles.

- `**bleachbit -l` o `bleachbit --list**`: Muestra la lista de limpieza disponible.

- `**bleachbit -s` o `bleachbit --sysinfo**`: Muestra información del sistema.

- `**bleachbit -p` o `bleachbit --preview**`: Muestra una vista previa de los elementos que se eliminarán.

- `**bleachbit -o` o `bleachbit --overwrite**`: Sobrescribe archivos eliminados de forma segura.

- `**bleachbit -f` o `bleachbit --force**`: Ejecuta la limpieza sin confirmación.

- `**bleachbit -v` o `bleachbit --version**`: Muestra la versión actual de BleachBit.

- `**bleachbit -d**` o `**bleachbit --debug**`: Activa el modo de depuración.

MACChanger

MACChanger es una herramienta de línea de comandos que te permite cambiar la dirección MAC de tus interfaces de red en sistemas operativos basados en Unix. La dirección MAC es un identificador único asignado a cada dispositivo de red, y cambiarla puede ayudarte a mejorar la privacidad y evitar el seguimiento en redes. MACChanger te ofrece varias opciones para cambiar la dirección MAC, como generar una dirección aleatoria, restaurar la dirección original o establecer una dirección MAC permanente. Es una herramienta útil para usuarios que desean modificar su dirección MAC y proteger su identidad en entornos de red.

Lista completa de todos los comandos existentes en la herramienta MACChanger:

- `**macchanger**`: Muestra la ayuda y la lista de comandos disponibles.

- `**macchanger -s <interface>**`: Muestra la dirección MAC actual de la interfaz especificada.

- `**macchanger -r <interface>**`: Cambia la dirección MAC de forma aleatoria en la interfaz especificada.

- `**macchanger -e <interface>**`: Restaura la dirección MAC original en la interfaz especificada.

- `**macchanger -p <interface>**`: Establece una dirección MAC permanente en la interfaz especificada.

- `**macchanger -a <interface>**`: Genera una dirección MAC diferente en la interfaz especificada.

- `**macchanger -l**`: Muestra la lista de direcciones MAC disponibles.

- `**macchanger -m <MAC_address> <interface>**`: Cambia la dirección MAC a la dirección especificada en la interfaz especificada.

- `**macchanger -b <vendor> <interface>**`: Cambia la dirección MAC a una dirección MAC conocida del fabricante especificado en la interfaz especificada.

- `**macchanger -A <interface>**`: Genera una dirección MAC diferente y la establece en la interfaz especificada.

- `**macchanger -P <interface>**`: Muestra la dirección MAC permanente de la interfaz especificada.

- `**macchanger -s <interface> -m <MAC_address>**`: Cambia la dirección MAC a la dirección especificada en la interfaz especificada sin mostrar ninguna salida adicional.

7. Otros:

OpenVAS (escáner de vulnerabilidades)

OpenVAS (Open Vulnerability Assessment System) es una herramienta de escaneo de vulnerabilidades de código abierto que se utiliza para identificar y evaluar las debilidades de seguridad en sistemas y redes. Proporciona una plataforma completa para realizar análisis de seguridad y pruebas de penetración en busca de vulnerabilidades conocidas y potenciales. OpenVAS utiliza una amplia variedad de pruebas de seguridad y técnicas de escaneo para detectar riesgos de seguridad, como vulnerabilidades de software, configuraciones inseguras y fallos de cumplimiento. Además, permite la gestión centralizada de escaneos, generación de informes detallados y seguimiento de la mitigación de vulnerabilidades. OpenVAS es una herramienta fundamental para garantizar la seguridad y protección de sistemas y redes contra posibles amenazas y ataques.

Lista de los comandos existentes en la herramienta OpenVAS (escáner de vulnerabilidades):

- `**openvas-start**`: Inicia el servicio OpenVAS.

- `**openvas-stop**`: Detiene el servicio OpenVAS.

- `**openvas-check-setup**`: Verifica la configuración de OpenVAS y muestra posibles problemas.

- `**openvas-setup**`: Configura OpenVAS y descarga las últimas bases de datos de vulnerabilidades.

- `**openvasmd**`: Gestiona el escáner de vulnerabilidades OpenVAS Manager.

- `**omp**`: Interfaz de línea de comandos para OpenVAS Management Protocol (OMP).

- `**openvas-cli**`: Interfaz de línea de comandos para ejecutar escaneos de vulnerabilidades en OpenVAS.

- `**openvas-adduser**`: Agrega un nuevo usuario al sistema OpenVAS.

- `**openvas-rmuser**`: Elimina un usuario existente del sistema OpenVAS.

- `**openvas-mkcert**`: Genera certificados SSL para el servidor OpenVAS.

- `**openvas-nvt-sync**`: Sincroniza las bases de datos de pruebas de vulnerabilidad (NVT) con los servidores de OpenVAS.

- `**openvasmd --update**`: Actualiza la base de datos de vulnerabilidades de OpenVAS.

- `**openvasmd --rebuild**`: Reconstruye la base de datos de OpenVAS.

- `**greenbone-nvt-sync**`: Sincroniza las bases de datos de pruebas de vulnerabilidad (NVT) con los servidores de OpenVAS.

- `**greenbone-certdata-sync**`: Sincroniza la base de datos de certificados con los servidores de OpenVAS.

- `**greenbone-scapdata-sync**`: Sincroniza la base de datos de datos SCAP con los servidores de OpenVAS.

Maltego

Maltego es una herramienta de inteligencia y análisis de enlaces que ofrece una variedad de comandos para facilitar la exploración y visualización de relaciones entre entidades. Con Maltego, puedes iniciar la interfaz gráfica principal o acceder a versiones específicas, como la Community Edition o la versión XL. Además, puedes utilizar la interfaz de línea de comandos Maltego Teeth para ejecutar transformaciones y realizar operaciones avanzadas desde la terminal. La herramienta también te permite exportar datos y gráficos generados en Maltego, importar datos desde archivos externos y actualizar la configuración y transformaciones. Puedes acceder a la documentación de Maltego directamente desde la herramienta y obtener información sobre la licencia y la versión instalada. Además, Maltego cuenta con un servidor integrado que permite la colaboración en tiempo real. Si encuentras problemas, puedes iniciar Maltego en modo de depuración para analizar y solucionar posibles errores. En resumen, Maltego es una herramienta poderosa y versátil que te brinda capacidades avanzadas de inteligencia y análisis para investigaciones y análisis de seguridad.

Lista de los comandos existentes en la herramienta Maltego:

- `**maltego**`: Inicia la interfaz gráfica de Maltego.

- `**maltegoce**`: Inicia la versión Community Edition de Maltego.

- `**maltegoxl**`: Inicia la versión XL de Maltego.

- `**maltego-teeth**`: Inicia Maltego Teeth, una interfaz de línea de comandos para Maltego.

- `**maltego-trx**`: Permite ejecutar transformaciones desde la línea de comandos.

- `**maltego-export**`: Permite exportar datos y gráficos generados en Maltego.

- `**maltego-import**`: Permite importar datos en Maltego desde archivos externos.

- `**maltego-update**`: Actualiza la configuración y transformaciones de Maltego.

- `**maltego-docs**`: Abre la documentación de Maltego en el navegador web.

- `**maltego-license**`: Muestra la información de la licencia de Maltego.

- `**maltego-version**`: Muestra la versión actual de Maltego instalada.

- `**maltego-server**`: Inicia el servidor de Maltego para la colaboración en tiempo real.

- `**maltego-debug**`: Inicia Maltego en modo de depuración para el análisis de problemas.

Hash-Identifier

Hash-Identifier es una herramienta de línea de comandos que permite identificar diferentes tipos de hash. Su objetivo principal es ayudar en el análisis y la seguridad de hash en diversos contextos. Puede analizar archivos y mostrar los tipos de hash contenidos en ellos, o puede recibir un hash específico y determinar su tipo. Además, ofrece una amplia variedad de funciones, como benchmarking, actualización de la base de datos de hash y limpieza de caché. Con su interfaz sencilla y sus capacidades de identificación de hash, Hash-Identifier es una herramienta útil para profesionales de la seguridad y la criptografía en su trabajo diario.

Lista completa de todos los comandos existentes en la herramienta Hash-Identifier:

- `**hash-identifier**`: Inicia la herramienta Hash-Identifier.

- `**hash-identifier --help**`: Muestra la ayuda y la lista de comandos disponibles.

- `**hash-identifier --version**`: Muestra la versión actual de Hash-Identifier.

- `**hash-identifier --file <ruta_archivo>**`: Identifica los tipos de hash contenidos en un archivo.

- `**hash-identifier --hash <hash>**`: Identifica el tipo de un hash específico.

- `**hash-identifier --benchmark**`: Ejecuta un benchmark para medir la velocidad de identificación de hash.

- `**hash-identifier --list**`: Muestra la lista de todos los tipos de hash soportados por Hash-Identifier.

- `**hash-identifier --example <tipo_hash>**`: Muestra un ejemplo de hash para un tipo específico.

- `**hash-identifier --update**`: Actualiza la base de datos de tipos de hash de Hash-Identifier.

- `**hash-identifier --clear-cache**`: Limpia la caché de Hash-Identifier.

Crunch

(generador de listas de palabras)

Crunch es una herramienta de generación de listas de palabras que permite crear diccionarios personalizados para pruebas de penetración y auditorías de seguridad. Con Crunch, puedes generar todas las combinaciones posibles de caracteres dentro de un rango de longitud específico, siguiendo un patrón personalizado o utilizando un conjunto de caracteres predefinidos. También puedes agregar cadenas de inicio y final, controlar el tamaño en bytes de las palabras generadas y guardar los resultados en un archivo. Crunch es una herramienta útil y flexible que te permite crear listas de palabras personalizadas para realizar ataques de fuerza bruta o probar la fortaleza de contraseñas y claves de cifrado.

Lista de todos los comandos existentes en la herramienta Crunch:

- `**crunch <min> <max>**`: Genera todas las combinaciones posibles de caracteres de longitud mínima a máxima especificada.

- `**crunch <min> <max> -f <charset.lst>**`: Genera todas las combinaciones posibles utilizando los caracteres de un archivo de lista de caracteres.

- `**crunch <min> <max> -t <pattern>**`: Genera todas las combinaciones posibles siguiendo un patrón específico.

- `**crunch <min> <max> -s <start>**`: Genera todas las combinaciones posibles a partir de una cadena de inicio específica.

- `**crunch <min> <max> -e <end>**`: Genera todas las combinaciones posibles terminando en una cadena específica.

- `**crunch <min> <max> -o <output.txt>**`: Guarda los resultados en un archivo de salida especificado.

- `**crunch <min> <max> -b <byte>**`: Genera todas las combinaciones posibles de un tamaño de bytes específico.

- `**crunch <min> <max> -l <logfile>**`: Guarda el progreso del generador de palabras en un archivo de registro especificado.

- `**crunch -h**`: Muestra la ayuda y la lista de comandos disponibles.

- `**crunch -v**`: Muestra la versión actual de Crunch.

BeEF (Framework para explotar web)

BeEF (Browser Exploitation Framework) es una herramienta de código abierto diseñada para explotar y comprometer navegadores web con el objetivo de evaluar la seguridad de las aplicaciones web y probar la efectividad de las defensas implementadas. BeEF proporciona una plataforma integral para interactuar con los navegadores de los usuarios y aprovechar las vulnerabilidades presentes en ellos. Permite lanzar ataques sofisticados como la inyección de código en páginas web, la ejecución remota de comandos y la manipulación de sesiones de usuario. Con BeEF, los profesionales de seguridad y los investigadores pueden simular ataques reales en entornos controlados, identificar debilidades y mejorar las medidas de protección.

Lista de todos los comandos existentes en la herramienta BeEF (Browser Exploitation Framework):

- `beef`: Inicia el servidor BeEF.

- `**beef -h**`: Muestra la ayuda y la lista de comandos disponibles.

- `**beef -v**`: Muestra la versión actual de BeEF.

- `**beef -x <path/to/config.yaml>**`: Inicia el servidor BeEF utilizando un archivo de configuración específico.

- `**beef -c <command>**`: Ejecuta un comando específico en el servidor BeEF.

- `**beef -s <script.js>**`: Carga un script JavaScript personalizado en el servidor BeEF.

- `**beef -a <module>**`: Activa un módulo específico en el servidor BeEF.

- `**beef -d <module>**`: Desactiva un módulo específico en el servidor BeEF.

- `**beef -e <command>**`: Ejecuta un comando en todos los clientes comprometidos.

- `**beef -l**`: Muestra la lista de clientes comprometidos.

- `**beef -r <report.html>**`: Genera un informe HTML con la información del servidor BeEF y los clientes comprometidos.

- `**beef -k**`: Detiene el servidor BeEF.

Hydra-GTK (Interfaz gráfica para Hydra)

Hydra-GTK es una interfaz gráfica de usuario (GUI) para la herramienta de hacking Hydra. Proporciona una forma intuitiva y fácil de utilizar Hydra para realizar ataques de fuerza bruta en protocolos de autenticación. Con Hydra-GTK, los usuarios pueden seleccionar el objetivo, especificar el protocolo de autenticación, configurar los diccionarios de contraseñas y ajustar las opciones avanzadas para personalizar el ataque. Una vez configurado, pueden iniciar el ataque y monitorear los resultados en tiempo real. Hydra-GTK simplifica el proceso de realizar ataques de fuerza bruta, brindando a los usuarios una herramienta efectiva y accesible para evaluar la seguridad de los sistemas de autenticación.

Lista de todos los botones en la interfaz de Hydra-GTK:

- **Seleccionar objetivo**: Permite ingresar la dirección IP o el nombre de host del objetivo y especificar el puerto de destino.

- **Seleccionar protocolo**: Permite elegir el protocolo de autenticación que se utilizará en el ataque.

- **Configurar diccionarios**: Permite agregar diccionarios de contraseñas que se utilizarán en el ataque de fuerza bruta.

- **Configurar opciones avanzadas**: Permite acceder a opciones adicionales para personalizar el ataque.

- **Iniciar ataque**: Inicia el ataque de fuerza bruta utilizando los parámetros y configuraciones especificadas.

- **Detener ataque**: Detiene el ataque en curso y muestra los resultados obtenidos hasta ese momento.

- **Ver registro**: Muestra el registro de eventos y actividades durante el ataque.

- **Limpiar registro**: Limpia el registro de eventos y actividades.

- **Ver estadísticas**: Muestra estadísticas y métricas relacionadas con el ataque en curso.

- A**brir archivo de configuración**: Permite cargar un archivo de configuración previamente guardado.

- **Guardar archivo de configuración**: Guarda la configuración actual en un archivo para su uso posterior.

- **Ayuda**: Proporciona información y documentación sobre el uso de Hydra-GTK.

- **Acerca de**: Muestra detalles sobre la versión y los créditos de Hydra-GTK.