|  |  |
| --- | --- |
| man | Muestra el manual de usuarios de un comando específico. Por lo tanto, debes escribirlo antes de otro comando para que te explique para lo que sirve. |
| clear | El comando clear limpia el terminal y deja la ventana en blanco eliminando todo lo que había antes. |
| sudo | Es uno de los comandos más poderosos del terminal, y te da privilegios de administrador para ejecutar acciones. Tendrás que escribirlo antes de un comando que requiera permisos de superusuario, y necesitarás escribir tu nombre y contraseña. |
| history | Te muestra un historial de los comandos que has utilizado en el pasado |
| pwd | Te dice cuál es la dirección completa del directorio en el que estás en este momento. |
| ls | Te dice cuál es el contenido de la dirección en la que estás. Si estás en una cartera, te dice el resto de elementos que hay. |
| ls -l | Te da una lista más detallada del contenido que hay en el directorio en el que te encuentras. |
| ls -al | Te muestra todo el contenido del directorio, incluyendo los archivos y carpetas ocultos. |
| cd | Cuando lo escribes así y sin nada más, vas al directorio de raíz, a Home. |
| cd ~ | También sirve para ir al directorio de raíz, a Home. |
| cd nombre\_directorio | Te permite moverte a un directorio o archivo particular, como por ejemplo, "cd Downloads" para ir a las descargas. |
| cd .. | Subir al directorio principal, como ir un paso hacia atrás en el árbol de directorios. |
| cd - | Volver al directorio o carpeta anterior en el que estabas. |
| cd / | Ir al directorio raíz |
| touch nombre\_archivo | Te permite crear un fichero con el nombre de archivo que quieras. |
| nano nombre\_archivo | Te permite crear y editar un fichero con el nombre que le digas, abriéndolo luego en el editor de texto Nano.  Puedes cambiar el nombre de este editor de texto por el de otro que prefieras. |
| file nombre\_archivo | Te permite ver el tipo de archivo que es un archivo concreto que tengas en el Mac. |
| cp nombre\_archivo nombre\_directorio | Te permite copiar el archivo que indiques en el comando que haya dentro de la carpeta donde estás dentro de la carpeta o directorio que indiques en el comando. Por ejemplo: cp miarchivo ~/Desktop/ |
| cp nombre\_archivo nuevo\_nombre\_archivo | Te permite cambiar el nombre de un archivo por otro diferente. |
| mv nombre\_archivo nombre\_directorio | En vez de copiar, este comando te sirve para mover el archivo del que pongas el nombre al directorio que  indiques. |
| mv nombre\_archivo nuevo\_nombre\_archivo | Este comando también te permite cambiar el nombre de un archivo por otro diferente. |
| rm nombre\_archivo | Te permite borrar y eliminar un archivo cuyo nombre indiques. |
| brew update | Actualiza el repositorio de Homebrew. |
| brew upgrade | Actualiza todos los paquetes instalados en tu sistema. |
| brew install nombre\_programa | Instala un programa desde la línea de comandos, aunque debe ser uno sin interfaz gráfica. |
| brew install --cask nombre\_programa | Te permite utilizar Cask para instalar un programa que tenga interfaz gráfica. Por ejemplo: brew install --cask vlc |
| brew uninstall nombre\_programa | Te permite borrar un programa que tengas instalado. |
| softwareupdate -i -a | Te permite instalar tus programas instalados en el Mac, aunque los hayas instalado desde la tienda de apps. |
| ps -ax | Identifica los procesos que hay ejecutándose en tu sistema. |
| top | Te muestra el estado de los principales procesos que hay ejecutándose en tu sistema y más están consumiendo. Lo puedes parar pulsando q o control + C. |
| ps -ax | grep nombre\_programa | Si detectas un proceso que está consumiendo muchos recursos, puedes usar este comando para obtener  más información sobre él. Por ejemplo: ps -ax | grep Telegram |
| sudo killall nombre\_programa | Te permite cerrar por completo un programa para que deje de ejecutarse. Necesitarás escribir  tu nombre de usuario y contraseña para usarlo, porque has utilizado el sudo que ejecuta comandos  con permisos de admin. |
| ping nombre\_host | Te permite hacer un ping a una web o IP para comprobar su conectividad. Por ejemplo: ping xataka.com |
| ifconfig en0 | Te permite ver las direcciones IP y MAC de tu dispositivo. |
| arp -a | Te permite ver todas las direcciones IP y MAC de todos los dispositivos conectados a tu red. |
| netstat | Te muestra la información sobre las conexiones entrantes y salientes de tu dispositivo. |
| lsof | te permite encontrar todos los procesos en ejecución que tienen una conexión activa a Internet. |
| whois nombre\_dominio | Te permite obtener información sobre un dominio concreto, como por ejemplo: whois xataka.com |
| traceroute nombre\_host | Te permite identificar la ruta que recorren los paquetes de datos desde tu dispositivo hasta el destino que indiques. Por ejemplo: traceroute xataka.com |

sudo arp-scan eth0 192.168.1.0/24

iwconfig wlan0 essid MOVISTAR\_PLUS\_14DA

ifconfig wlan0 down

iwconfig wlan0 mode ad-hoc

ifconfig wlan0 up

iwconfig wlan0 essid MOVISTAR\_PLUS\_14DA

service NetworkManager start

python3 -m http.server 80

WIFITE

Esta es una evolución del <wifite>, es un instrumento de las keys wifi de una red de internet.

Este solo debe ser usado de modo de prueba de seguridad para auditar tu wifi. Wifite, ejecuta herramientas de auditoria inalámbrica de manera automática, y lo cual la hace diferente al resto. Además de que requiere la intervención del usuario para alcanzar su fin. Esta se creó para ser usada en Linux o en alguna de sus distribuciones Linux especializadas en seguridad, por ejemplo la podras encontrar en Kali Linux.

Para este proceso debes tener instalado aircrack-ng

LISTADO DE SUS FUNCIONES PRINCIPALES:

REAVER (o –bully) se activa de manera predeterminada, fuerza con: –wps- only.

Captura de handshake de forma preestablecida, con –no-wps.

Valida el handshake con pyrit, tshark, cowpatty y aircrack-ng.

Pruebas de penetración con: replay, chopchop, fragment, hirte, p0841, caffe-latte.

Desbloquea los puntos de acceso ocultos.

Soporte para 5 Ghz.

Guarda las contraseñas y handshake, extraidos en el directorio local – cracked

Comandos que desifran los handshake, WPA capturados ( –crack)

APLICACIONES NECESARIAS PARA UN CORRECTO FUNCIONAMIENTO:

IWCONFIG: identifica los dispositivos inalámbricos en modo monitor.

IFCONFIG: iniciar o detener dispositivos inalámbricos

AIRCRACK-NGSUITE:

AIRCRACK-NG: Descifrar archivos Cap. WEP y capturas del handshake WPA

AIREPLAY- NG: reproducir archivos de capturas, ataques WEP, entre otros.

AIRMON-NG: enumera y habilita el modo monitor en dispositivos inalámbricos.

AIRODUMP-NG: genera los archivos de exploración y la captura en el destino final.

PACKETFORGE-NG: falsificar archivos de captura

OTRAS APLICACIONES OPCIONALES SON:

TSHARK: detecta redes WPS e inspecciona handshake.

REAYER: ataques WPS Pixie-Dust.

BULLY: Ataques wps Pixie\_Dust (alternativa a reayer).

COWPATTY: detecta las capturas del handshake.

Pyrit: detecta las capturas del handshake.

Obtenidos los requisitos podrás instalar wifilite 2.1.0

Instalar es muy sencillo pero para esto debes tener instalado git.

1 git clone https://github.com/derv82/wifite2.git

Abrimos el directorio de la aplicación.

1 cd wifite2

Luego ejecutamos:

1 ./wifite.py

Una vez ejecutado estos pasos solo debes seguir las instrucciones que te da la herramienta.

No hay texto alternativo para esta imagen

**Espionage** is a network packet sniffer that intercepts large amounts of data being passed through an interface. The tool allows users to to run normal and verbose traffic analysis that shows a live feed of traffic, revealing packet direction, protocols, flags, etc. Espionage can also spoof ARP so, all data sent by the target gets redirected through the attacker (MiTM). Espionage supports IPv4, TCP/UDP, ICMP, and HTTP. Espionage was written in Python 3.8 but it also supports version 3.6. This is the first version of the tool so please contact the developer if you want to help contribute and add more to Espionage. Note: This is not a Scapy wrapper, scapylib only assists with HTTP requests and ARP.

**Installation**

* **git clone https://www.github.com/MandConsultingGroup/Espionage.git**
* **cd Espionage**
* **sudo python3 -m pip install -r requirments.txt**
* **sudo python3 espionage.py --help**

**Usage**

* **sudo python3 espionage.py --normal --iface wlan0 -f capture\_output.pcap**  
  Command 1 will execute a clean packet sniff and save the output to the pcap file provided. Replace wlan0 with whatever your network interface is.
* **sudo python3 espionage.py --verbose --iface wlan0 -f capture\_output.pcap**  
  Command 2 will execute a more detailed (verbose) packet sniff and save the output to the pcap file provided.
* **sudo python3 espionage.py --normal --iface wlan0**  
  Command 3 will still execute a clean packet sniff however, it will not save the data to a pcap file. Saving the sniff is recommended.
* **sudo python3 espionage.py --verbose --httpraw --iface wlan0**  
  Command 4 will execute a verbose packet sniff and will also show raw http/tcp packet data in bytes.
* **sudo python3 espionage.py --target <target-ip-address> --iface wlan0**  
  Command 5 will ARP spoof the target ip address and all data being sent will be routed back to the attackers machine (you/localhost).
* **sudo python3 espionage.py --iface wlan0 --onlyhttp**  
  Command 6 will only display sniffed packets on port 80 utilizing the HTTP protocol.
* **sudo python3 espionage.py --iface wlan0 --onlyhttpsecure**  
  Command 7 will only display sniffed packets on port 443 utilizing the HTTPS (secured) protocol.
* **sudo python3 espionage.py --iface wlan0 --urlonly**  
  Command 8 will only sniff and return sniffed urls visited by the victum. (works best with sslstrip).
* Press Ctrl+C in-order to stop the packet interception and write the output to file.

**Menu**

**usage: espionage.py [-h] [–version] [-n] [-v] [-url] [-o] [-ohs] [-hr] [-f FILENAME] -i IFACE  
[-t TARGET]  
optional arguments:  
-h, –help show this help message and exit  
–version returns the packet sniffers version.  
-n, –normal executes a cleaner interception, less sophisticated.  
-v, –verbose (recommended) executes a more in-depth packet interception/sniff.  
-url, –urlonly only sniffs visited urls using http/https.  
-o, –onlyhttp sniffs only tcp/http data, returns urls visited.  
-ohs, –onlyhttpsecure  
sniffs only https data, (port 443).  
-hr, –httpraw displays raw packet data (byte order) recieved or sent on port 80.  
(Recommended) arguments for data output (.pcap):  
-f FILENAME, –filename FILENAME  
name of file to store the output (make extension ‘.pcap’).  
(Required) arguments required for execution:  
-i IFACE, –iface IFACE  
specify network interface (ie. wlan0, eth0, wlan1, etc.)  
(ARP Spoofing) required arguments in-order to use the ARP Spoofing utility:  
-t TARGET, –target TARGET**

Archivos ocultos

defaults write com.apple.Finder AppleShowAllFiles TRUE

presiona la tecla ‘Alt’ y haz clic derecho sobre el icono de ‘Finder’. Selecciona la opción ‘Forzar reinicio’ para que a partir de ahora los archivos ocultos dejen de estarlo. Podrás identificarlos por su icono algo más translúcido.

Con CMD + Shift + . Se muestran los ocultos en Finder

CHATGPT

PASSSWORD CHATGPT Ho$SJqSs073CFt4uzvR6

openai01Nolase

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

VPN´s

protonvpn.com/es.

privadovpn.com/es

Cuando un programa pone que no arranca

sudo xattr -r -d file “nombre del programa arrastrando”

xattr -cr Aplicación dañada en cuestión

Librerías de gestion de redes

   
Scapy: Scapy es una herramienta de manipulación de paquetes basada en Python para el análisis de seguridad de redes.

 PyCrypto: PyCrypto es una biblioteca de algoritmos criptográficos escrita en Python.

 Nmap: Nmap es una herramienta popular de exploración de redes y auditoría de seguridad.

 Requests: Requests es una biblioteca de Python popular utilizada para hacer solicitudes HTTP y manejar respuestas.

 Paramiko: Paramiko es una implementación de Python del protocolo SSH, utilizado para el acceso remoto seguro a servidores.

 Python Cryptography Toolkit (PyCT): PyCT es una colección de algoritmos y protocolos criptográficos escritos en Python.

 Cryptography: Cryptography es una biblioteca de Python que proporciona una interfaz de alto nivel a varios algoritmos criptográficos.

 NetworkX: NetworkX es una biblioteca de Python utilizada para analizar y visualizar redes.

 Pyshark: Pyshark es un envoltorio para la popular herramienta de análisis de paquetes Wireshark, lo que permite usarla programáticamente en Python.

 PyOpenSSL: PyOpenSSL es una biblioteca que proporciona una interfaz de Python a la biblioteca criptográfica OpenSSL.

**REDES**

Ettercap

Nmap

ANÁLISIS DE VULNERABILIDADES / OBTENER INFORMACIÓN

a. Consultas Whois DNS Inversas / Transferencia de zona no autorizada:

arsys.com

cqcounter.com/whois/

cmd: dig (local)

b. Barridos de ping:

nmap / zenmap (nmap automatizado)

arp-scan / airsweep (local)

c. Escaneo vulnerabilidades:

Nessus básico + configuración / instalación

Openvass básico + configuración / instalación

ANNEXO: Encubrir IP

Tor: Instalación + configuración

Proxychains: Instalación + configuración

TOR en navegador

Proxychains + programas (cmd)

ANNEXO: Crear túnel SSH con Nessus + Tor (nivel avanzado)

ANNEXO: SSH teórico + Putty

ANNEXO: robtex.com -> Mapeado DNS

ANNEXO: metagoofil -> Extracción de metadatos

ANNEXO: Obtener subdominios -> theharvester

EXPLOTACIÓN BÁSICA

a. Searchsploit -> BBDD exploit / cmd: searchsploit / guía básica

b. Metasploit

Teoría: Nociones (payload, msfvenom, meterpreter?), estructura módulos

Uso básico: Iniciar (postgresql), comandos básicos (info, run, set?)

Escaneos -> db\_nmap

Combinación módulos básico (exploit + payload)

Proxychains + Metasploit

Al final ejemplo teórico de enviar payload + multi/handler

MAN IN THE MIDDLE (LOCAL)

a. Nociones: arp spoofing / https (problema encriptación paquetes ssl)

b. Exploración -> cmd: route / ifconfig (local) / nmap gateway

c. Configuración: ip\_forward / sysctl

d. Wireshark -

e. Combinaciones de ataques:

Simple: arpspoof / bettercap + wireshark + urlsnarf

Antiguo: iptables + ssltrip + taif

Teoria HSTS -> cmd: curl

mitmf -> básico + preparación python

Otros: ssltrip2 + dns2proxy / squid proxy / mitmproxy / bettercap

Ataque Delorean

ANNEXO: Teoría ampliada protocol ssl / certificados / proxys

PETAR WIFIS Y CREACIÓN DE DICCIONARIOS

Faltaría Geminis auditor / Linset -> Wifislax

a. Aircrack -> airodump-ng / aireplay-ng

b. Diccionarios:

crunch

john

Diccionarios en la web (falta diccionario personalizable con datos persona)

Ataque reaver (wps)

Wifiphisher -> Explicado a nivel teórico require 2 wifi target

ATAQUES A CONTRASEÑAS SISTEMAS WINDOWS (SAM)

a. Teoría -> HASH / Fichero SAM / Ubicación en Windows

b. pwdump7 -> Extraer desde Windows para practicar (instalación)

c. samdump2 -> descifrar SAM

d. Descifrados con fuerza bruta diccionario:

john (nota: lista rockyou.txt)

web ophcrack.sourceforge para diccionarios

crackstation.com

ophcrack (programa de pago en Linux)

e. Extraer SAM desde meterpreter:

Teoría: escalación de privilegios

Escalar privilegios con meterpreter -> commandos / ps + migrate / multishell / bypassing con metasploit

Extracción -> metasploit / hasdump / filtrar pwdump7

ANNEXO: Comandos útiles para Metepreter básico

ATAQUES A CONTRASEÑAS LINUX

a. Teoria -> Fichero shadow

b. Descifrar fichero:

John fuerza bruta

John diccionario

ATAQUES A WEBS

a. Nikto -> info del servidor web + escaneo vulnerabilidades (blog.osvdb.org)

Nota: Sublist3r -> proporciona subdominios

b. Zaproxy (ZAP) -> web browser análisis

Instalación / Certificado

Privoxy + Tor + ZAP

Básico + ataques: spider, fuzzer, fuerza bruta

c. XSS ? Cross Site Scripting

Introducción: Nociones, XSS permanente / reflejo

ANNEXO: Nociones HTLM, PHP/URL, PETICIONES GET, POST, CLIENTE-SERVIDOR

Nociones, practicas, ejemplos teóricos? básico / avanzado

d. Beef XSS: Zombificar web browser

Configuración + nociones

Crear código php / infraestructura HTLM

NOIP (dynamic DNS server) + BEEF

Nota: Uso e introducciones de apache + portforwarding

ANNEXO: Tor Vidalia instalación y uso

ANNEXO: botnet ZEUS

e. Ataques de fuerza bruta:

Medusa (SSH, Logins, etc.)

f. Clickjacking: Nociones

g. Remote file inclusion / Local file inclusion

Nociones: concepto, dorks, detectar vulnerabilidad

DVWA: Entorno de pruebas ataques web -> Instalación

Ejemplo 1/2: shell + netcat

Ejemplo 3: Local file + shell c99

h. SQL injection:

Teoría

Muestra bypassing login -> detección

Muestra sql injection contenidos web -> detección

Blind SQL -> Obtención de datos table manual ejemplos (avanzado)

Burpsuite -> Configuración (con Tor) / Ataque snipper / Ataque obtención de tablas / uso básico

Sqlmap

i. WPSCAN -> Vulnerabilidades gestor de contenidos wordpress

EXPLOTACIÓN NIVEL MEDIO ? VARIOS

a. Armitage / Metasploit

Armitage: Preparación + Nociones básicas (chequeos, fijar/configurar ataques + Hail Maria attack

Ataques en la WAN -> Teoría / Ampliación porforwarding / Avance multi/handler + payload msfvenom combinado con NOIP2 + IP?s teoría

Nociones Armitage pivoting (vnc)

Setear proxy en Metasploit -> socks5 / privoxy / Proxychains

b. Metasploit web attack

Web delivery exploit (avanzado)

Scanner HTTP auxiliary modules

Wmap

Sqlmap + metasploit

ANNEXO: Ataques directos a web servers / dorks: skatewinner00, etc?

c. Fuzzing:

powerfuzzer

cmd: sfuzz

d. Keyloggers (y más)

SETOOLKIT (herramienta ingeniería social muchas opciones) -> código powershell en .bat

SETOOLKIT -> Clonación de páginas (credenciales) ¿DNS?

Meterpreter -> keylogger\_recorder

Beelogger -> en archivos varios (envía correos personales)

TROJANS / MSFVENOM

a. Introducción msfvenom -> multi/handler base

b. Hacking Android:

Con persistencia

Bind payload en apliación .apk -> apktool / zipalign

Firma + antideteccion malware en Android

c. Guía comandos msfvenom / Tipos de payloads / ejemplos

d. Noción dll injection (avanzado)

e. msfvenom + netcat

ANNEXO: Comandos específicos msfvenom

f. Payload powershell en MS office macros con SETOOLKIT

g. Falsear imagen.exe (largo: exige portforwarding, gcc)

h. Veil -> Insertar payload en office

i. Payload en pdf con metasploit

ANNEXO: Enviar mails con ?falso correo? -> sendEmail / smtp2go.com

j. Php payload (web):

File inclusion en web + persistencia

Falsear imagen con php para subir

Alojar código malicioso php en servidor

File inclusions direct

Alojar fichero con burpsuite

POSTEXPLOTACIÓN / BACKDOOR / PERSISTENCIA

a. Meterpreter -> run metsvc / run persistence

b. The backdoor factory -> Insertar malware en ejectuable de Windows

ANNEXO: Post explotación Linux

c. Persistencias con meterpreter PLUS (remember bypass UAC)

d. Netcat persistence

ANNEXO: Amplicación módulos meterpreter

ESCRITORIOS REMOTOS / WINDOWS USERS

a. cmd: rdesktop

b. Crear usuario en Windows con meterpreter

c. Abrir puertos de la víctima con meterpreter

ANNEXO: Monitorización -> run getgui / run vnc

VIRUS (Ejecutables -> Windows)

a. Generación de virus:

Virus con notepad.bat -> Bat to Exe / webs / DELmE?s

Virus Pharming -> Alterar fichero host (redireccionar webs)

Virus código fuente C

b. ?Trabajar? con el virus (vale payloads)

Meterpreter -> Inserción en windows32 + autoejecutar

?Embedding? en .exe -> pyJoiner / shell7er

ANNEXO: Evasión ?no se puede comprobar el editor?

c. Evasión AV (vale payloads):

Firma con openssl + osslsigncode

Cheat protectors -> themida / winlicense

Comentarios modificación de firma

d. Cifrado avanzado de ejecutable -> añadir basura con metasm.rb + peencode.rb

Notas a cifrar con ASLR (muy muy avanzado)

MISCELÁNEOS FINALES:

a. Paylod python to .exe (indetectable)

b. Ngrok -> Como NOIP pero más básico (opciones PLUS de pago)

c. ATSCAN -> Buscador de dorks

d. Maltego -> Buscar datos de corporaciones / empresas (limitado)

e. Meterpreter:

Desactivar AV + Firewall

Limpiar logs y rastro (varias técnicas)

f. Sodan

APPS desde terminal

Esteganografía

steghide extract -sf /Users/jacervi/Downloads/sample.jpg -p perseverance