MANUAL PRÁCTICO

NMAP

PARA PENTESTERS

DE CERO A PRO

MIGUEL ANGEL VILLALOBOS GARCIA

NMAP PARA PENTESTERS: DE CERO A PRO

Autor: **Miguel Ángel Villalobos García**
https://www.linkedin.com/in/m7villalobos/

Nmap la navaja suiza de reconocimiento y enumeración

Nmap ("Network Mapper") es tu navaja suiza para explorar redes y auditar seguridad. Es gratuita, open-source y fundamental en pentesting. Su función principal es el **reconocimiento (recon)** y la **enumeración**: descubrir hosts activos, puertos abiertos, servicios ejecutándose, sus versiones y el sistema operativo. Esta información es *crítica* porque define tu superficie de ataque. Sin un buen mapeo inicial con Nmap, es fácil pasar por alto vulnerabilidades clave.

En frameworks como MITRE ATT&CK, Nmap se asocia directamente con **T1046 - Network Service Scanning**. Pero gracias al **Nmap Scripting Engine (NSE)**, puede hacer mucho más: detectar vulnerabilidades, realizar fuerza bruta, etc.

Entender Nmap es vital. No solo te permite "fotografiar" la red objetivo, sino que también te ayuda a pensar como un administrador (o identificar dónde fallan). Para un pentester, dominar Nmap no es opcional, es el punto de partida.

Fundamentos Clave para el Día a Día

Para usar Nmap eficientemente, necesitas dominar lo básico:

Sintaxis Básica:

nmap [Tipo(s) de Escaneo] [Opciones] {Objetivo}

- **Tipo(s) de Escaneo:** Cómo Nmap sondea los puertos (e.g., -sS, -sT, -sU). Si no especificas, Nmap elige por defecto (-sS si eres root, -sT si no).
- **Opciones:** Modificadores clave (-p para puertos, -sV para versión, -O para SO, -T para tiempo, -oN/-oX/-oA para salida).
- **Objetivo:** A quién escaneas.

Especificación de Objetivos (Lo más usado):

- **IP Individual:** nmap 192.168.1.1
- **Nombre de Host:** nmap servidor.dominio.com (Nmap usa DNS)
- **Rango CIDR:** nmap 192.168.1.0/24 (Forma estándar para subredes)
- **Desde Archivo (-iL <fichero>):** nmap -iL lista_objetivos.txt (¡Esencial para alcances definidos!)

Exclusión de Objetivos (¡CRÍTICO!):

- **Excluir en línea (--exclude <lista>):** nmap 192.168.1.0/24 --exclude 192.168.1.1,192.168.1.100 (Para no tocar sistemas fuera de alcance o críticos).
- **Excluir desde Archivo (--excludefile <fichero>):** nmap -iL red_completa.txt --excludefile fuera_de_alcance.txt (Para listas de exclusión reutilizables).
    - **¡Atención!** Equivocarse al definir el objetivo puede llevarte a escanear fuera del alcance autorizado. ¡Siempre verifica dos veces tus objetivos y exclusiones!

Selección de Puertos (-p): La Decisión Estratégica

- **Por defecto:** Nmap escanea los 1000 puertos TCP más comunes. A menudo insuficiente.
- **Puerto Único:** -p 80
- **Rango:** -p 1-1024
- **Lista:** -p 22,80,443
- **Todos los Puertos (-p- o -p 1-65535):** nmap -p- 192.168.1.1
    - **Uso:** Exhaustivo, no te pierdes nada. **Contra:** Muy lento y muy ruidoso (fácil de detectar). Úsalo cuando la cobertura total sea más importante que el sigilo o el tiempo.
- **Escaneo Rápido (-F):** nmap -F 192.168.1.1 (Escanea los 100 puertos más comunes).
    - **Uso:** Reconocimiento inicial muy rápido y menos ruidoso. **Contra:** Puedes perder servicios en puertos no comunes.
- **Top N Puertos (--top-ports <n>):** nmap --top-ports 20 192.168.1.1 (Escanea los 'n' más comunes).
    - **Uso:** Buen equilibrio entre velocidad y cobertura.
- **Especificar Protocolo (TCP/UDP):** -p T:22,U:53,T:8080 (Recuerda usar -sU para escanear puertos U:).

Estrategia Práctica: A menudo se empieza con un escaneo rápido (-F o --top-ports) para identificar hosts interesantes, y luego se lanza un escaneo completo (-p-) sobre esos hosts específicos.

Tabla 2.1: Resumen Práctico - Objetivos y Puertos

Tipo	Opción Nmap	Ejemplo Práctico	Caso de Uso Pentest Diario

Objetivo	(ninguna)	nmap 192.168.1.10	Escanear un host específico.
Objetivo	(ninguna)	nmap webserver.cliente.com	Escanear por nombre (DNS debe funcionar).
Objetivo	(ninguna)	nmap 192.168.5.0/24	**Estándar** para definir el alcance de una subred.
Objetivo	-iL <fichero>	nmap -iL scope.txt	**Esencial** para escanear listas de IPs/redes del cliente.
Objetivo	--exclude <lista>	nmap 10.0.0.0/16 --exclude 10.0.1.5	Excluir IPs críticas o fuera de alcance directamente.
Objetivo	--excludefile <fichero>	nmap -iL all.txt --excludefile exclude.txt	Excluir múltiples sistemas definidos en un archivo.
Puerto	-p <lista>	nmap -p 21,22,23,25,80,443,3389 target	Escanear puertos comunes específicos rápidamente.
Puerto	-p-	nmap -p- target	**Exhaustivo**. Lento y ruidoso. Útil para análisis profundo.
Puerto	--top-ports <n>	nmap --top-ports 20 target	Buen compromiso velocidad/cobertura para recon inicial.
Puerto	-F	nmap -F target	**Muy rápido**. Para una primera pasada veloz (100 puertos).

Puerto	-p T:<...>,U:<...>	sudo nmap -sS -sU -p T:1-1000,U:53,161,500,4500 target	Escanear puertos TCP y UDP específicos (requiere -sS/-sT y -sU).

Técnicas de Escaneo Clave

3.1 Descubrimiento de Hosts: ¿Hay alguien ahí?

Antes de escanear puertos, identifica qué hosts responden.

- **Ping Scan (-sn):** nmap -sn 192.168.1.0/24
  - **Qué hace:** Solo descubrimiento de hosts, *no* escanea puertos. Rápido y relativamente sigiloso.
  - **Cómo (Root):** Envía ICMP echo, TCP SYN a 443, TCP ACK a 80, ICMP timestamp.
  - **Cómo (No Root):** Intenta conectar (TCP handshake) a puertos 80 y 443.
  - **Uso:** Primera pasada rápida para listar IPs activas en la red.
- **TCP SYN Ping (-PS<lista_puertos>):** sudo nmap -PS80,443,8080 10.0.0.0/16 (Requiere root)
  - **Qué hace:** Envía paquetes SYN a los puertos dados. Si recibe SYN/ACK (puerto abierto) o RST (puerto cerrado), el host está vivo.
  - **Uso:** Muy útil para encontrar hosts detrás de firewalls que bloquean ICMP pero permiten TCP a puertos comunes (web).
- **TCP ACK Ping (-PA<lista_puertos>):** sudo nmap -PA21,22,80 172.16.0.0/24 (Requiere root)
  - **Qué hace:** Envía paquetes ACK. Los hosts activos deberían responder con RST.
  - **Uso:** Intenta evadir firewalls *sin estado* que bloquean SYN entrantes. Menos efectivo contra firewalls modernos *con estado*.
- **ARP Scan (-PR):** sudo nmap -sn -PR 192.168.1.0/24 (Nmap lo usa por defecto en LAN si eres root y usas -sn)
  - **Qué hace:** Envía peticiones ARP en la red local.
  - **Uso: El método más rápido y fiable para descubrir hosts en tu misma red local (LAN).** No funciona para redes remotas.
- **Saltar Descubrimiento (-Pn):** nmap -Pn 192.168.1.50
  - **Qué hace:** Le dice a Nmap: "No te molestes en descubrir si el host está vivo, asume que lo está y escanea los puertos directamente".
  - **Uso: ¡CRÍTICO!** Úsalo cuando sepas (o sospeches) que los hosts están activos pero no responden a las sondas de descubrimiento (firewalls muy restrictivos). También útil si ya tienes una lista de hosts confirmados (-iL).
  - **Advertencia:** Si usas -Pn contra muchos hosts que *realmente* están caídos, el escaneo será *extremadamente* lento porque Nmap intentará escanear puertos en hosts que nunca responderán.

3.2 Escaneo de Puertos: ¿Qué servicios hay?

- **SYN Scan (-sS) - Stealth Scan:** sudo nmap -sS 192.168.1.10 (Requiere root)
  - **Qué hace:** Envía SYN, espera SYN/ACK (abierto) o RST (cerrado). *No completa la conexión* (envía RST en lugar de ACK final).
  - **Por qué usarlo: Es el escaneo TCP preferido.** Más rápido y mucho más sigiloso que -sT porque no completa conexiones (menos logs). Diferencia bien entre open, closed, filtered.
- **TCP Connect Scan (-sT):** nmap -sT 192.168.1.10
  - **Qué hace:** Usa la llamada connect() del sistema operativo para intentar una conexión TCP completa (handshake de 3 vías).
  - **Por qué usarlo: Tu única opción para TCP si NO tienes privilegios root.**
  - **Desventajas:** Más lento y mucho más ruidoso (fácil de detectar y loguear) que -sS.
- **UDP Scan (-sU):** sudo nmap -sU 192.168.1.10 (Normalmente requiere root para una mayor eficiencia)
  - **Qué hace:** Envía paquetes UDP (vacíos o con payload específico para puertos comunes como 53, 161).
  - **Por qué usarlo: Esencial para encontrar servicios UDP** (DNS, SNMP, DHCP, etc.) que a menudo se pasan por alto y pueden ser vulnerables.
  - **Interpretación:**
    - Respuesta UDP -> open.
    - ICMP Port Unreachable (Type 3, Code 3) -> closed.
    - Otros ICMP Unreachable -> filtered.
    - Sin respuesta -> open|filtered (¡El desafío!).
  - **Desventajas:** Mucho más lento que TCP. El estado open|filtered es ambiguo (¿está abierto o un firewall bloquea?). A menudo necesitas -sV para intentar obtener una respuesta del servicio y confirmar si está open.
- **Otros Escaneos TCP (Menos comunes pero útiles para evasión/análisis):**
  - **FIN (-sF), NULL (-sN), Xmas (-sX):** Envían paquetes con flags TCP inesperados. Algunos sistemas (no Windows) responden con RST si el puerto está cerrado, y no responden si está abierto. Pueden pasar algunos firewalls sin estado. Estado open|filtered si no hay respuesta. Requieren root.
  - **ACK Scan (-sA):** sudo nmap -sA target (Requiere root). Envía ACK. Si recibe RST, el puerto es unfiltered (alcanzable, Nmap no sabe si open/closed). Si no recibe respuesta o recibe ICMP error, es filtered. **No detecta puertos abiertos**, pero es **muy útil para mapear reglas de firewall** (ver qué puertos bloquea un firewall con estado).

3.3 Estados de Puerto (Qué significan para ti):

- **open:** ¡Bingo! Hay una aplicación escuchando. **Objetivo principal.** Investiga más (-sV, NSE).
- **closed:** El host respondió, pero no hay servicio en ese puerto. Confirma que el host está vivo y alcanzable en ese puerto (no hay firewall bloqueando *totalmente*).
- **filtered:** Nmap no pudo determinar el estado. Un firewall, ACL u otro filtro está bloqueando las sondas o las respuestas. Indica defensas activas. Requiere técnicas de evasión o diferentes tipos de escaneo.

- **unfiltered:** (Principalmente con -sA). El puerto es alcanzable, pero Nmap no sabe si está abierto o cerrado. Útil para mapear firewalls. Necesitas otro scan (-sS) para saber el estado real.
- **open|filtered:** (Principalmente con -sU, -sF, -sN, -sX). Nmap no puede distinguir. Podría estar abierto o filtrado. **Necesita más investigación** (prueba -sV).

Tabla 3.1: Comparativa Práctica de Escaneos Clave

Técnica	**Opción**	**Root?**	**Sigilo**	**Velocidad**	**Ventaja Principal (Pentest)**	**Desventaja Principal (Pentest)**
Host Discovery						
Ping Scan	-sn	Rec.	Medio	Muy Rápida	Lista rápida de hosts activos, bajo ruido.	Puede ser bloqueado por firewalls.
TCP SYN Ping	-PS	Sí	Medio+	Rápida	Encuentra hosts tras firewalls que bloquean ICMP (usa puertos).	Requiere root, puede ser bloqueado si filtran esos SYN.
TCP ACK Ping	-PA	Sí	Medio+	Rápida	Puede evadir firewalls *sin estado*.	Ineficaz vs firewalls con estado. Requiere root.
ARP Scan (LAN)	-PR	Sí	N/A	Ext. Rápida	**El mejor método en LAN.**Fiable y rápido.	Solo funciona en la red local.

Skip Discovery	-Pn	No	Bajo	Variable	**Garantiza escaneo** en hosts que no responden a pings.	**Muy lento** si muchos hosts están realmente caídos.
Port Scanning						
SYN Scan	-sS	Sí	Alto	Muy Rápida	**El mejor scan TCP.** Rápido, sigiloso, fiable.	Requiere root.
Connect Scan	-sT	No	Bajo	Media	**Alternativa TCP si no eres root.**	Lento, muy ruidoso (fácil de detectar/loguear).
UDP Scan	-sU	Sí(Rec)	Medio	Lenta	Encuentra servicios UDP olvidados (DNS, SNMP...).	Lento, `open
FIN/NULL/Xmas Scan	-sF/N/X	Sí	Alto	Media	Pueden evadir algunos filtros/IDS.	No fiable en Windows. `open
ACK Scan	-sA	Sí	Medio	Rápida	**Mapea reglas de firewall.**Determina filtered/unfiltered.	No detecta puertos open.

Enumeración Avanzada: ¿Qué hay realmente ahí?

Saber que un puerto está abierto no es suficiente. Necesitas saber *qué* servicio es y *en qué* SO corre.

4.1 Detección de Servicios y Versiones (-sV)

- **Por qué es CRÍTICO:** Te dice la aplicación exacta (Apache, OpenSSH) y su versión. **Esto es lo que usas para buscar vulnerabilidades (CVEs) y exploits específicos.**
- **Cómo funciona:** Envía sondas a puertos open u open|filtered y compara las respuestas con la base de datos nmap-service-probes.
- **Uso Básico:** Combínalo con tu escaneo de puertos: sudo nmap -sS -sV target
- **Control de Intensidad (--version-intensity <0-9>):**
    - Controla cuántas sondas se prueban (0=muy pocas, 9=todas). El **default es 7**.
    - --version-light (es alias de --version-intensity 2): Más rápido, menos preciso. Útil para pasadas rápidas.
    - --version-all (es alias de --version-intensity 9): Más lento, más exhaustivo. Útil si el default no identifica algo.
    - **Práctica:** Empieza con el default (7). Usa 2 si necesitas velocidad, 9 si necesitas profundidad en un objetivo específico.
- **Traza (--version-trace):** Para depurar por qué un servicio no se identifica. Muestra las sondas y respuestas.
- **Salida:** Busca la columna VERSION. Te dará Nombre, Versión, a veces Protocolo, Hostname, tipo de Dispositivo, CPE (Common Platform Enumeration). **La versión es oro.**
- **Ayuda con UDP:** -sV puede convertir un open|filtered UDP en open si consigue una respuesta válida del servicio.

4.2 Detección de Sistema Operativo (-O)

- **Por qué es IMPORTANTE:** Te ayuda a:
    - Entender el entorno (Windows vs Linux).
    - **Seleccionar exploits** (muchos son específicos de SO).
    - Evaluar vulnerabilidades específicas del SO.
- **Cómo funciona:** "TCP/IP Stack Fingerprinting". Envía sondas TCP/UDP/ICMP y analiza detalles de las respuestas (TTL, Window Size, TCP options, etc.). Compara la "huella" con la base de datos nmap-os-db. Requiere **privilegios root** y funciona mejor con **al menos un puerto TCP abierto y uno cerrado** en el objetivo.
- **Uso Básico:** sudo nmap -sS -O target (Necesita un scan de puertos para encontrar los abiertos/cerrados).
- **Opciones Prácticas:**
    - --osscan-limit: Solo intenta la detección de SO en hosts "prometedores" (con puerto TCP abierto y cerrado encontrados). Ahorra tiempo en escaneos grandes.
    - --osscan-guess o --fuzzy: Nmap será más "agresivo" al adivinar el SO si no hay una coincidencia perfecta. Útil para tener una pista, pero tómalo con cautela.
- **Salida:** Busca líneas como OS details:, Device type:, Running:, OS CPE:. Te da el SO, versión aproximada y tipo de dispositivo.

4.3 El Combo Agresivo (-A)

- **Qué es:** Un atajo conveniente para activar varias opciones útiles a la vez:
    - Detección de SO (-O)
    - Detección de Versión (-sV)
    - Escaneo de Scripts por Defecto (-sC)
    - Traceroute (--traceroute)
- **Comando:** sudo nmap -A target
- **Uso Práctico:** Muy popular para obtener una enumeración bastante completa con un solo comando.
- **Desventajas:** Es más **lento**, **ruidoso** e **intrusivo** que escaneos más específicos. No lo uses para sigilo o como primera pasada muy rápida. Úsalo cuando necesites información detallada y el "ruido" sea aceptable.

¡Importante! La precisión de -sV y -O depende de que las bases de datos (nmap-service-probes, nmap-os-db) estén actualizadas. ¡Mantén tu Nmap al día! (sudo apt update && sudo apt upgrade nmap o similar).

Nmap Scripting Engine (NSE) - El Multiplicador de Fuerza

NSE te permite automatizar tareas usando scripts en Lua. Nmap viene con cientos de scripts listos para ser empleados en tus pentests.

5.1 Uso Básico de NSE

- **Scripts por Defecto (-sC):** nmap -sC target
    - Ejecuta scripts de la categoría default (útiles, rápidos, no *demasiado* intrusivos).
    - Equivalente a --script=default. La opción -A también incluye -sC.
    - **Uso:** Una forma rápida y fácil de obtener información adicional más allá de la versión/SO.
- **Selección Específica (--script <script|categoria|directorio|expresión>):**
    - **Por Categoría:** Ejecuta todos los scripts de una categoría. **Las más útiles para pentesters:**
        - discovery: Obtiene más info (shares SMB, subdominios DNS, títulos HTTP...). nmap --script discovery target
        - vuln: **Busca vulnerabilidades conocidas.** nmap --script vuln target (¡Popular!)
        - brute: **Intenta fuerza bruta de credenciales** (FTP, SSH, SMB...). nmap --script brute target(¡Ruidoso y potencialmente bloqueante!)
        - auth: Relacionado con autenticación (login anónimo FTP, sesiones SMB...). nmap --script auth target
        - exploit: Intenta explotar vulnerabilidades (¡**USAR CON EXTREMA PRECAUCIÓN Y SOLO CON PERMISO EXPLÍCITO!**).
    - **Por Nombre de Script:** nmap --script smb-vuln-ms17-010 target
    - **Combinando:** nmap --script "default or vuln" target

- **Actualizar Base de Datos de Scripts (--script-updatedb):** sudo nmap --script-updatedb (Hazlo si añades scripts manualmente).
- **Ayuda sobre Scripts (--script-help <script|categoria>):** nmap --script-help smb-brute (Te dice qué hace el script y qué argumentos (args) acepta).

5.2 Scripts NSE Clave para Pentesters (Ejemplos)

- **Discovery:**
  - dns-brute: Fuerza bruta de subdominios.
  - smb-enum-shares, smb-enum-users: Enumera recursos compartidos y usuarios SMB.
  - smb-os-discovery: Intenta obtener info detallada del SO vía SMB.
  - http-enum: Busca directorios/archivos web comunes.
  - http-title: Obtiene el <title> de páginas web.
  - snmp-enum*: Si encuentras SNMP abierto, estos scripts pueden sacar mucha info.
- **Vulnerabilidades (vuln):**
  - --script vuln: Ejecuta todos los de esta categoría. Buen punto de partida.
  - smb-vuln-ms17-010: Detecta EternalBlue.
  - http-vuln-*: Busca diversas CVEs web.
  - ssl-heartbleed: Detecta Heartbleed.
  - **Externos (requieren instalación/configuración):**
    - vulscan: Compara banners (-sV necesario) con bases de datos *offline* de CVEs. Debes mantener las BBDD.
    - nmap-vulners: Consulta la base de datos *online* Vulners.com (más actualizada, requiere internet). -sV recomendado.
- **Fuerza Bruta (brute):**
  - ftp-brute, ssh-brute, telnet-brute, smb-brute, snmp-brute, mysql-brute, pgsql-brute, rdp-brute (a través de rdp-enum-encryption), etc.
  - **¡Necesitan argumentos (--script-args) con listas de usuarios/contraseñas!**
- **Autenticación (auth):**
  - ftp-anon: Comprueba login anónimo FTP.
  - smb-enum-sessions: Lista sesiones activas SMB.

¡Advertencia NSE! Scripts de categorías brute, intrusive, exploit, dos pueden:

- Ser **muy ruidosos** y detectados fácilmente.
- **Bloquear cuentas** (fuerza bruta).
- **Causar inestabilidad** o caídas en sistemas objetivo.
- Contactar a terceros (external), filtrando información. **¡SIEMPRE entiende qué hace un script antes de ejecutarlo y asegúrate de tener permiso!**

5.3 Argumentos de Scripts (--script-args, --script-args-file)

Muchos scripts necesitan parámetros para funcionar bien.

- **Sintaxis:** --script-args <arg1>=<val1>,<arg2>=<val2>,...

    - Usa comillas si el valor tiene espacios, comas, etc.: smbuser=test,smbpass='P@ss word!'
- **Argumentos Calificados:** Para evitar conflictos, usa nombre_script.argumento=valor.
    - Ej: nmap -p 445 --script smb-brute --script-args 'smb-brute.threads=5,userdb=users.txt,passdb=passes.txt'
- **Desde Archivo (--script-args-file <fichero>):** Carga argumentos desde un archivo (un key=value por línea o separados por comas).
    - Ej: nmap --script ftp-brute --script-args-file ftp-args.txt target

Dominar --script-args es esencial para usar scripts NSE de forma efectiva (especialmente los brute).

Tabla 5.1: Categorías NSE Relevantes (Resumen Práctico)

Categoría	Relevancia Pentest	Riesgo / Consideración
default	Buena base para info extra sin ser (normalmente) demasiado intrusivo.	Algunos pueden ser detectados.
discovery	**Fundamental** para enumeración profunda (usuarios, shares, subdominios...).	Puede generar bastante tráfico.
vuln	**Clave** para encontrar vulnerabilidades conocidas basadas en versión/servicio.	Depende de la base de datos. Scripts externos (vulners, vulscan) requieren gestión.
brute	**Muy útil** para probar credenciales débiles/default. Puede dar acceso directo.	**Alto riesgo de detección y bloqueo de cuentas.**Requiere listas (--script-args).
auth	Útil para encontrar accesos anónimos, sesiones, bypass.	Generalmente menos arriesgado que brute.

exploit	Intenta explotación directa.	**¡EXTREMO RIESGO!** Usar solo con permiso explícito y entendiendo consecuencias.
intrusive	Scripts que pueden crashear, consumir recursos o ser maliciosos.	**Alto riesgo de detección y disrupción.** Evitar si no está permitido en RoE.
external	Contacta a terceros (Whois, Shodan...).	**Puede filtrar información** sobre el objetivo/pentester a terceros.

Evasión de Firewalls/IDS (El Juego del Gato y el Ratón)

Las defensas son comunes. Nmap tiene opciones para intentar sortearlas, pero no hay recetas mágicas. Requiere entender las defensas y combinar técnicas.

- **Fragmentación de Paquetes (-f, --mtu <val>):**
  - **Qué hace:** Divide las cabeceras TCP en fragmentos IP pequeños (-f usa 8 bytes, -ff 16, --mtu permite especificar un tamaño específico múltiplo de 8). Requiere root.
  - **Objetivo:** Confundir IDS/Firewalls simples que no logren reensamblar bien los fragmentos.
  - **Limitaciones: Cada vez menos efectivo.** Muchos sistemas reensamblan bien los paquetes que reciben. Algunos firewalls bloquean fragmentos. **Crucial: NO funciona con -sV, NSE, -sT.** Sacrificas enumeración avanzada.
- **Decoys / Señuelos (-D <decoy1,decoy2,ME,...>):**
  - **Qué hace:** Hace parecer que el escaneo viene de múltiples IPs (los decoys + tu IP real ME). Requiere root.
  - **Objetivo:** Ofuscar tu IP real en los logs/alertas. Dificulta la atribución.
  - **Limitaciones:** Los decoys deben estar activos. Puede ser detectado por análisis avanzado. ISPs pueden filtrar IP spoofing. **NO funciona con -sV, NSE, -sT.** Ralentiza el escaneo.
  - Ej: nmap -sS target -D decoy1,decoy2,ME,decoy4,RND (RND para IP aleatoria).
- **Spoofing de Puerto Fuente (-g <puerto> o --source-port <puerto>):**
  - **Qué hace:** Fija el puerto origen desde el que Nmap envía los paquetes (e.g., 53 o 80). Requiere root.
  - **Objetivo:** Intentar saltar firewalls mal configurados que confían en el puerto origen (permiten todo desde puerto 53 DNS, por ejemplo).

    - **Limitaciones:** Solo afecta escaneos con paquetes raw (-sS, -sU). **NO afecta a -sV, NSE, -O, -sT.**
- **Ajuste de Temporización (-T <0-5>, controles finos):**
    - **Objetivo Principal:** Evadir IDS/IPS basados en umbrales de detección (demasiados paquetes por segundo). Ralentizar para pasar desapercibido.
    - **Plantillas (-T <num>):**
        - -T0 (paranoid), -T1 (sneaky): **Extremadamente lentos.** Para máximo sigilo. Pueden tardar días.
        - -T2 (polite): Más lento que el normal, menos impacto. Buena opción para entornos sensibles si -T0/-T1 son inviables.
        - -T3 (normal): Default. Equilibrio razonable.
        - -T4 (aggressive), -T5 (insane): **Muy rápidos y ruidosos.** Más detectables, riesgo de perder paquetes si la red/host no aguanta. Usar con cuidado.
    - **Controles Finos (Más precisos que -T):**
        - --scan-delay <tiempo>: Espera mínima entre sondas a un mismo host (e.g., 5s, 500ms). Muy útil para evadir rate-limiting.
        - --max-rate <num>: Máximo de paquetes por segundo.
    - **Compromiso:** Sigilo vs Tiempo. Elige según las RoE, la sensibilidad del entorno y tu paciencia.
- **Otras Técnicas:**
    - **ACK Scan (-sA):** Como se vio, no evade directamente pero ayuda a mapear reglas de firewall (filtered vs unfiltered).
    - **Añadir Datos (--data-length <num>):** Añade bytes aleatorios a los paquetes. Puede confundir IDS muy básicos que esperan payloads vacíos.
    - **Idle Scan (-sI <zombie>):** Muy sigiloso (tu IP no toca al objetivo), pero complejo, lento y requiere un host "zombie" adecuado (difícil).

Estrategia de Evasión: No hay una receta única que funcione con todo. A menudo: 1) Reconoce las defensas (e.g., -sA para mapear firewall). 2) Elige/Combina técnicas (e.g., -sS con -T2 y quizás -f si sospechas de filtros simples). Adapta tu enfoque.

Tabla 6.1: Técnicas de Evasión (Resumen Práctico)

Técnica	Opción(es)	Mecanismo / Objetivo Evasión	Limitaciones / Riesgo Detección
Fragmentación	-f, --mtu	Divide cabeceras / Confundir reensamblado simple.	Menos efectivo. **Incompatible con -sV, NSE, -sT.** Puede bloquearse.

Decoys	-D <lista,ME>	Ofusca IP real / Confundir logs/alertas.	Requiere decoys activos. **Incompatible con -sV, -sT.** Ralentiza.
Spoofing Pto. Fuente	-g, --source-port	Fija puerto origen / Saltar filtros basados en puerto fuente.	Solo para scans raw (-sS/-sU). **No afecta -sV, NSE, -O, -sT.**
Temporización Lenta	-T0, -T1, -T2	Reduce velocidad / Evadir IDS por umbral (rate-based).	**Muy lento (T0/T1).** -T2 es un compromiso razonable.
Retardo Manual	--scan-delay <tiempo>	Espera mínima entre sondas / Eludir rate-limiting.	Ralentiza proporcionalmente.
Control de Tasa	--max-rate <num>	Limita paquetes/segundo / Evadir umbrales, reducir impacto.	Puede ralentizar mucho si el límite es bajo.
ACK Scan	-sA	Mapea reglas firewall (detecta filtered/unfiltered).	No detecta puertos open. Ayuda a planificar otros scans.
Añadir Datos	--data-length <num>	Añade payload / Evadir IDS muy simples basados en payload.	Impacto limitado contra IDS modernos.

Gestión de Salida y Reportes (Guárdalo todo, podrías necesitarlo más adelante)

Escanear es solo la mitad. Necesitas guardar y usar los resultados.

Formatos de Salida Clave:

- **Normal (-oN <fichero>):** nmap -sS -A target -oN scan_normal.nmap
  - Similar a la salida en pantalla, legible por humanos. Útil para revisión rápida o logs simples.
- **XML (-oX <fichero>):** nmap -sS -A target -oX scan_completo.xml
  - **¡El formato más importante!** Estructurado, ideal para procesar con scripts (Python, etc.) o importar en otras herramientas (Metasploit, Faraday, Nessus, etc.). Contiene *toda* la información. **Es el estándar de facto.**
- **Grepable (-oG <fichero>):** nmap -F target -oG scan_grep.gnmap
  - Una línea por host, fácil de parsear con grep, awk, cut. Considerado obsoleto por Nmap en favor de XML, pero aún usado para búsquedas rápidas en consola.
- **Todos los Formatos (-oA <basename>):** nmap -sS -A target -oA scan_final
  - **¡Muy práctico!** Guarda en los tres formatos a la vez (scan_final.nmap, scan_final.xml, scan_final.gnmap). Te da flexibilidad inmediata. **Recomendado en pentests.**

Opciones de Salida y Control Útiles:

- **Verbosidad (-v, -vv, -vvv):** nmap -sS -v target
  - **¡Úsalo siempre!** Muestra el progreso del escaneo, estimaciones de tiempo, y te notifica de puertos abiertos *mientras* escanea. -vv da más detalles. Fundamental para monitorizar scans largos.
- **Mostrar Razón (--reason):** nmap -sS --reason target
  - Indica *por qué* Nmap marcó un puerto con un estado (e.g., syn-ack para open, reset para closed). Añade contexto.
- **Mostrar Solo Abiertos (--open):** nmap -p- --open target
  - Filtra la salida para mostrar solo puertos open (o open|filtered). **Muy útil** para reducir el ruido en redes grandes o con muchos filtros.
- **Traza de Paquetes (--packet-trace):** Para depuración de red muy bajo nivel. Muestra todos los paquetes enviados/recibidos.

Procesamiento para Informes (Enfócate en XML):

El XML (-oX) es tu amigo para reportar. Métodos comunes:

1. **Scripts (Python/Perl...):** Usa librerías XML para extraer IPs, puertos abiertos, versiones, resultados NSE relevantes para tu informe.
2. **Transformación XSLT:** xsltproc scan.xml -o scan.html (usa nmap.xsl que viene con Nmap) para crear un informe HTML rápido.
3. **Importar a Herramientas:** Carga el XML en Metasploit (db_import), Faraday, Dradis, etc., para gestionar hallazgos y correlacionar datos.

Saber manejar la salida de Nmap, especialmente XML, es clave para ser eficiente.

Consideraciones Éticas y Legales

Nmap es muy potente. Úsalo con responsabilidad.

- **¡LA REGLA DE ORO: AUTORIZACIÓN EXPLÍCITA Y POR ESCRITO!**
  - Antes de lanzar *cualquier* escaneo, necesitas permiso firmado del dueño de la red/sistemas.
  - El permiso (en SOW o RoE) debe definir CLARAMENTE:
    - **Alcance:** IPs, redes, dominios INCLUIDOS y EXCLUIDOS.
    - **Horarios:** Cuándo puedes escanear.
    - **Herramientas:** Menciona Nmap.
    - **Contactos:** A quién llamar si algo va mal.
  - Escanear sin permiso puede llevar a: quejas a tu ISP, acciones legales (civiles/penales), daño a tu reputación.
- **RESPETA EL ALCANCE ESTRICTAMENTE:**
  - Usa -iL, CIDR, --exclude, --excludefile con precisión. **¡Un error aquí es grave!**
  - Verifica dos veces tus objetivos antes de darle a Enter.
- **USA LA HERRAMIENTA RESPONSABLEMENTE:**
  - **Minimiza Intrusividad:** Prefiere -sS sobre -sT. Usa el sigilo necesario.
  - **Controla Agresividad/Ruido:**
    - Escaneos agresivos (-T4/-T5), exhaustivos (-p-), o con scripts intrusive/brute/exploit son **detectables** y pueden **impactar** la red/sistemas.
    - Usa temporización adecuada (-T2, -T1, --scan-delay) si el sigilo es necesario o el entorno es sensible.
    - **Comunica:** Avisa al cliente antes de scans pesados. Considera hacerlos fuera de horas pico.
  - **Evita Daños:** Nmap raramente causa problemas, pero puede pasar con sistemas inestables. Sé especialmente cuidadoso con sistemas críticos. La responsabilidad es tuya.
- **Intención:** Tú eres un pentester ético. Operas con permiso para encontrar debilidades y mejorar la seguridad, no para causar daño.

Nmap es indispensable para cualquier pentester. Desde el mapeo básico hasta la enumeración avanzada con NSE y la evasión de defensas, dominar esta herramienta es crucial. Este manual te ha dado las claves prácticas para el día a día: los comandos y opciones más usados, cómo interpretar resultados y cómo integrar Nmap en tu flujo de trabajo, siempre recordando la importancia vital de la ética y la autorización. La práctica constante y el uso responsable te convertirán en un experto con Nmap, capaz de descubrir eficazmente la superficie de ataque de tus objetivos.