Resumen Teórica 13 : Redes

Tomás F. Melli

September 2025

Índice

1 Redes Wi-Fi

Las redes Wi-Fi se han convertido en el estándar para la conectividad inalámbrica en entornos domésticos, empresariales y públicos. Si bien facilitan la movilidad y el acceso a recursos de red sin necesidad de cableado físico, también introducen importantes riesgos de seguridad y privacidad debido a la naturaleza del medio inalámbrico. El diseño seguro de una red Wi-Fi requiere considerar múltiples aspectos técnicos y operativos.

Cifrado

Los protocolos de cifrado, como WEP, WPA1, WPA2 y WPA3, son fundamentales para proteger la confidencialidad de la comunicación. Cada evolución ha mejorado la seguridad frente a ataques pasivos y activos, siendo WPA3 la versión más moderna y robusta.

- **WEP (Wired Equivalent Privacy):** Protocolo antiguo basado en RC4 con IV corta (24 bits). Vulnerable a ataques pasivos y activos; no recomendado actualmente.
- **WPA1 (Wi-Fi Protected Access):** Utiliza TKIP para mejorar WEP. Mitiga algunas vulnerabilidades de WEP, pero TKIP es también inseguro hoy.
- **WPA2:** Basado en AES-CCMP. Dos modos:
  - **Personal (PSK):** Clave compartida; seguridad depende de la calidad de la passphrase.
  - **Enterprise (802.1X/EAP):** Autenticación mediante RADIUS y credenciales por usuario o certificados. Recomendado para entornos corporativos.
- **WPA3:** Proporciona SAE (Simultaneous Authentication of Equals) para resistencia a ataques offline, mejor protección en redes abiertas y perfiles para IoT.

Control de acceso

Técnicas como ocultar el SSID o filtrar por MAC ofrecen cierto grado de control, pero no deben considerarse medidas de seguridad sólidas por sí mismas, ya que pueden ser fácilmente eludidas.

- **Ocultar SSID:** Solo evita la difusión del nombre de la red; no protege frente a escuchas pasivas.
- **Filtrado por MAC:** Permite restringir dispositivos por dirección MAC; puede ser fácilmente evadido mediante *MAC spoofing*.

Detección y monitoreo

Sistemas Wireless IDS/WIPS permiten monitorear la red para identificar APs no autorizados, ataques de deauth, jammers y otros eventos anómalos, aumentando la visibilidad y la capacidad de respuesta frente a incidentes.

- **Wireless IDS/WIPS:** Sensores que detectan anomalías como Rogue APs, ataques de deauth o jammers. WIPS añade capacidad de prevención (bloqueo de intrusos).

Autenticación

Los métodos de autenticación incluyen WPS, modos personal (PSK) y enterprise (802.1X/EAP). La autenticación enterprise basada en certificados (EAP-TLS) ofrece un nivel alto de seguridad y control de acceso granular.

- **WPS (Wi-Fi Protected Setup):** Facilita la configuración mediante PIN o push-button. Vulnerable en modo PIN; se recomienda deshabilitar si no es necesario.
- **Personal (PSK):** Clave compartida; sencillo pero menos seguro si se requiere control por usuario o auditoría.
- **Enterprise (802.1X/EAP):** Uso de RADIUS y métodos EAP (PEAP, EAP-TLS, etc.). EAP-TLS (certificados mutuos) ofrece alta seguridad y soporte para PFS.

Amenazas y herramientas

Herramientas como *Wifiphisher* permiten simular APs maliciosos para capturar credenciales, mientras que los Wi-Fi jammers pueden interrumpir el servicio mediante interferencia. La detección y mitigación de estas amenazas es esencial.

- **Rogue AP / Evil Twin:** AP malicioso que imita una red legítima para capturar credenciales (ej. herramientas como *Wifiphisher*).
- **Wi-Fi Jammers:** Interferencia deliberada para degradar o bloquear el servicio; legalmente regulados en muchos países.

Privacidad

La exposición de SSIDs históricos, direcciones MAC y metadatos de conexión puede comprometer la privacidad del usuario. Medidas como la randomización de MAC, la minimización de probes activos y el uso de autenticación segura ayudan a reducir estos riesgos.

- **Probe requests y SSID históricos:** Pueden revelar redes previamente conectadas y facilitar tracking de dispositivos.
- **Direcciones MAC:** Únicas por dispositivo; suplantables y usadas para seguimiento. La randomización de MAC mejora la privacidad.
- **Metadatos de conexión:** Logs de AP/RADIUS pueden exponer hábitos y localización de usuarios.
- **Localización por triangulación:** Medidas de RSSI permiten determinar la posición de dispositivos.
- **Mitigaciones:** Habilitar randomización de MAC, minimizar probes activos, usar autenticación fuerte (EAP-TLS), gestionar logs con políticas claras y educar usuarios.

2 Dark Web y Anonimato en Línea

La **Dark Web** es una parte de Internet que no es indexada por motores de búsqueda tradicionales y a la que se accede únicamente mediante herramientas especializadas. Se utiliza tanto para proteger la privacidad y libertad de expresión en entornos hostiles como para actividades ilegales.

2.1 TOR - The Onion Router

- **Concepto:** TOR es un sistema diseñado para permitir la navegación anónima en Internet, protegiendo la privacidad de los usuarios y dificultando la trazabilidad de su actividad en línea.
- **Funcionamiento:** TOR utiliza un *encaminamiento en cebolla* (onion routing), donde el tráfico se cifra en múltiples capas y se enruta a través de una serie de nodos (relays) antes de alcanzar su destino. Cada nodo solo conoce el siguiente y el anterior, lo que garantiza anonimato y dificulta la correlación de tráfico.
- **Usos:** Navegación privada, protección frente a censura, comunicación segura, y acceso a servicios ocultos (*onion services*).
- **Referencias oficiales:**
  - TOR Project
  - TOR Design Paper
  - TOR 2004 Design Paper - Part 1
  - TOR 2004 Design Paper - Part 2
  - TOR Overview
  - Tor: A Layman's Guide

DNS en TOR

El manejo del DNS en TOR es crítico para mantener el anonimato de los usuarios y prevenir fugas de información que puedan revelar la identidad o la ubicación del cliente.

- **Resolución de nombres:** En TOR, las consultas DNS no se resuelven directamente mediante el proveedor de servicios de Internet del usuario. En lugar de eso, TOR enruta las solicitudes DNS a través de la misma red de nodos (*onion routing*) que el tráfico normal, evitando la exposición de los nombres de dominio solicitados.
- **Prevención de fugas (DNS leaks):** Si un cliente TOR resolviera nombres mediante el DNS local o externo, podría filtrar la actividad de navegación y comprometer el anonimato. Por eso, todos los resolvers de nombres deben estar integrados dentro de la red TOR.
- **Servicios .onion:** Las direcciones *.onion* no existen en el DNS público y son resueltas exclusivamente dentro de la red TOR. Esto significa que no hay consultas externas que puedan revelar la existencia o acceso a estos servicios ocultos.
- **Configuración segura:** Navegadores y aplicaciones TOR (como el *TOR Browser*) están diseñados para forzar todas las solicitudes DNS a través de TOR y bloquear cualquier intento de resolución directa fuera de la red. Esto es fundamental para mantener la privacidad y el anonimato.

2.2 Tails Live System

- **Descripción:** Tails es un sistema operativo en modo live (no requiere instalación) diseñado para garantizar privacidad y anonimato completos. Todo el tráfico de red pasa por TOR, y no deja rastros en el equipo host.
- **Objetivo:** Permitir a usuarios comunicarse y navegar de manera segura y anónima, protegiendo identidad, ubicación y datos sensibles.
- **Referencia:** Tails Live System

Consideraciones de seguridad

- La Dark Web protege el anonimato, pero no garantiza la legalidad de los contenidos ni la seguridad frente a malware o ataques.
- TOR reduce riesgos de seguimiento y censura, pero es vulnerable a ataques de correlación de tráfico si no se configura correctamente.
- Tails, al ejecutarse en modo live y enrutar todo por TOR, minimiza riesgos de fugas de información, aunque el comportamiento del usuario y la seguridad física del equipo siguen siendo críticos.

3 Nmap (Network Mapper)

Nmap (*Network Mapper*) es una utilidad open-source diseñada para explorar y auditar redes de forma rápida y flexible. Está orientada tanto a administradores de red (auditoría, inventario, verificación de políticas) como a analistas de seguridad (evaluación de superficie de ataque). Nmap puede descubrir hosts, detectar puertos abiertos y servicios asociados, identificar sistemas operativos remotos por *fingerprinting*, detectar dispositivos de filtrado/intermediarios y ejecutar chequeos programables mediante su potente *Nmap Scripting Engine* (NSE).

3.1 Capacidades principales

- **Descubrimiento de hosts:** ping sweeps (ICMP, TCP/UDP ping) para determinar qué hosts están activos.
- **Escaneo de puertos:** determina qué puertos TCP/UDP están abiertos, cerrados o filtrados.
- **Detección de servicios y versiones:** identifica el servicio que escucha en un puerto y su versión (banner/version detection).
- **Identificación de sistema operativo (OS fingerprinting):** análisis remotos de respuestas TCP/IP para inferir el sistema operativo y su versión.
- **Detección de firewalls / filtros:** técnicas que permiten inferir si hay packet filters o NAT en el camino.
- **Nmap Scripting Engine (NSE):** framework de scripting (Lua) para automatizar tareas: detección de vulnerabilidades, brute force, enumeración, exploits, etc.
- **Múltiples modos de salida:** texto, XML, grepable (legacy), y `--script` outputs; utilidades para procesar resultados (e.g. `ndiff`, `Nmap XML` parsing).

3.2 Técnicas y tipos de escaneo

Nmap soporta una amplia variedad de técnicas de escaneo, cada una con objetivos, ventajas, limitaciones y señales de detección distintas.

- **Vanilla TCP connect() scanning (`-sT`):**
  - **Descripción:** utiliza la llamada de sistema `connect()` del sistema operativo para completar el handshake TCP completo con el host objetivo.
  - **Comportamiento:** si el puerto está abierto el destino responderá con `SYN/ACK` y se completa el `3-way handshake`; Nmap entonces cierra la conexión (envía `RST`).
- **TCP SYN (half-open) scanning (`-sS`):**
  - **Descripción:** envía un paquete SYN y observa la respuesta, pero no completa el handshake (no envía el ACK final), por eso se le llama *half-open*.

  - **Comportamiento:** `SYN/ACK` indica puerto abierto; `RST` indica cerrado; ausencia de respuesta o ICMP puede indicar filtrado.

- **Stealth scans: FIN, Xmas, NULL (`-sF`, `-sX`, `-sN`):**
  - **Descripción:** envían paquetes TCP con flags inusuales (solo FIN; FIN+PSH+URG = Xmas; sin flags = NULL) para explotar la forma en que stacks TCP reaccionan ante paquetes inesperados.
  - **Comportamiento:** muchos stacks responden con `RST` cuando el puerto está cerrado, y no responden cuando está abierto (heurística usada por Nmap).

- **TCP ACK y Window scanning (`-sA`, `--window`):**
  - **Descripción:** envía paquetes ACK (o usa el tamaño de ventana) para determinar si un firewall es *stateful* y qué puertos están filtrados o no.
  - **Comportamiento:** un RST indica que el puerto no está filtrado (host responde normalmente); ausencia de respuesta o ICMP indica filtrado por firewall.

- **UDP scanning (`-sU`):**
  - **Descripción:** envía datagramas UDP a puertos objetivo y espera respuestas (o ICMP `port unreachable`).
  - **Comportamiento:** ausencia de respuesta puede significar abierto—filtrado; ICMP `port unreachable` indica cerrado.

- **ICMP / Ping sweeps:**
  - **Descripción:** sondeos para descubrir hosts vivos usando ICMP echo, TCP SYN a determinados puertos, o otras sondas.
  - **Comportamiento:** rápido para inventarios; sin embargo, muchos hosts bloquean ICMP o responden solo a sondas específicas.

- **Fragmentación IP (`--mtu` o `-f`):**
  - **Descripción:** fragmenta los paquetes de sonda en múltiples fragmentos IP pequeños intentando evadir firmas simples de IDS/IPS que inspeccionan cabeceras completas.

- **FTP proxy / Bounce scan (histórico):**
  - **Descripción:** emplea servidores que implementan el comando FTP `PORT` para redirigir el escaneo a través de ellos (efecto de *bounce*), ocultando el origen.

- **Idle scan (`-sI`):**
  - **Descripción:** técnica avanzada que aprovecha el campo IP ID predecible de un host *zombie* para inferir el estado de puertos en el objetivo sin que el escáner directo sea la fuente de las sondas.
  - **Mecanismo:** el escáner induce al objetivo (vía el zombie) a enviar paquetes al objetivo y observa el cambio en el IP ID del zombie para deducir si el puerto objetivo respondió (cambio en contador IP ID).

- **Reverse-ident scanning y RPC directo:**
  - **Descripción:** técnicas para identificar servicios específicos (por ejemplo, RPC) o para consultar servicios de identificación remota (IDENT) para obtener información adicional sobre conexiones.

3.3 Detección de Sistema Operativo (OS fingerprinting)

Nmap implementa técnicas de *TCP/IP fingerprinting* que analizan pequeñas variaciones en la construcción y respuesta de paquetes TCP/IP (opciones TCP, valores de TTL, manejo de flags, respuestas a paquetes no estándar, etc.). El artículo técnico sobre fingerprinting de Nmap explica los métodos y heurísticas utilizadas:

- Nmap OS fingerprinting (artículo)

La precisión depende de la red (NAT, proxies), el sistema objetivo y las versiones del stack TCP/IP; por ello Nmap reporta probabilidades y coincidencias.

3.4 Nmap Scripting Engine (NSE)

- NSE permite extender Nmap con scripts escritos en `Lua`. Los scripts se clasifican por categorías: `auth`, `broadcast`, `brute`, `default`, `discovery`, `dos`, `exploit`, `external`, `fuzzer`, `intrusive`, `malware`, `safe`, `version`, `vuln`, etc.
- Ejemplos de uso:

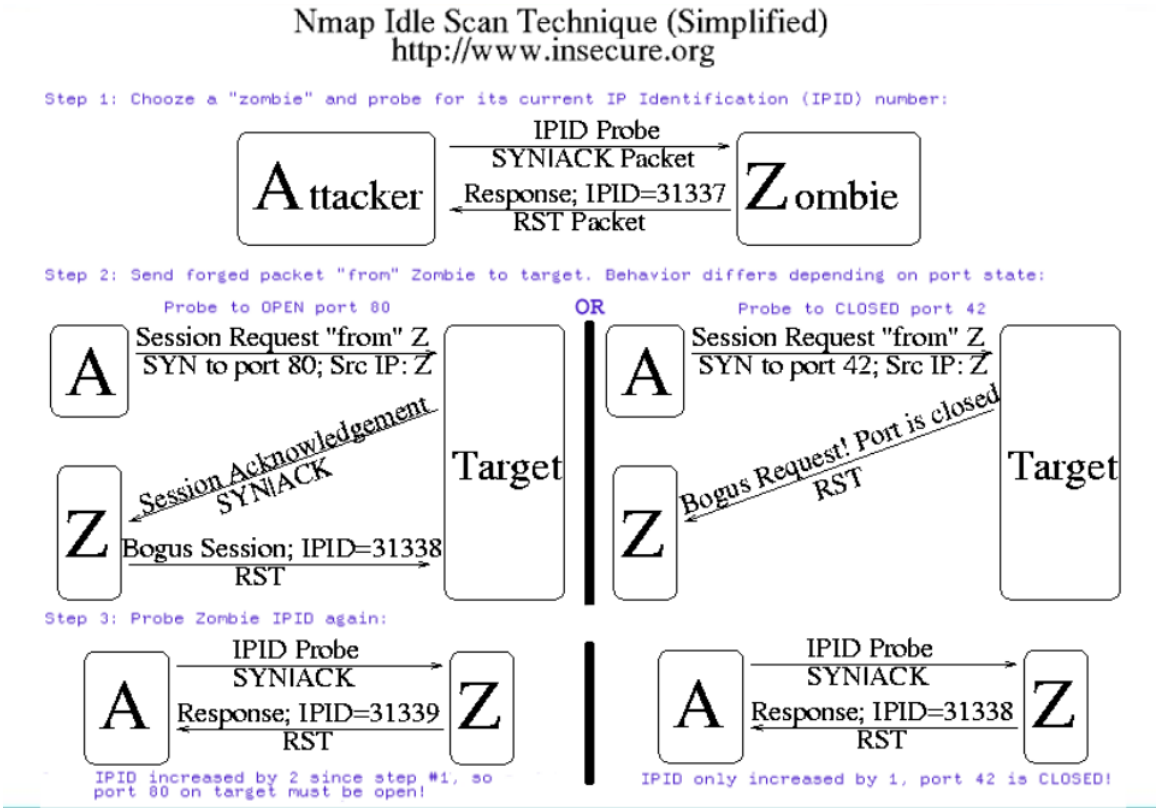
```
# Escaneo con scripts por defecto
nmap -sC example.com

# Ejecutar un script específico
nmap --script smb-os-discovery example.com

# Enumeración HTTP
nmap --script http-enum example.com

# Ejecutar todos los scripts de la categoría 'auth'
nmap --script "auth" example.com

# Buscar vulnerabilidades
nmap --script vuln example.com

# Ayuda sobre un script concreto
nmap --script-help ssl-enum-ciphers
```

- NSE permite detectar vulnerabilidades conocidas, enumerar servicios, bruteforce, comprobar configuraciones inseguras, y automatizar pruebas a gran escala.
- Para desarrollar scripts NSE se utiliza Lua y las APIs que Nmap expone (biblioteca de red, manipulación de paquetes, output), lo que permite tareas muy flexibles.

3.5 Ejemplos prácticos y sintaxis útil

- `nmap -sS -p 1-65535 -T4 target`
  Realiza SYN scan en todos los puertos con timing `T4` (rápido).
- `nmap -sU -p 53,123 target`
  Escaneo UDP de puertos DNS y NTP.
- `nmap -A target`
  Realiza detección de OS, servicio/versión y scripts de NSE por defecto (opción `-A` agrega agresividad).
- `nmap -O target`
  Intento de identificación del sistema operativo.
- `nmap --script "safe and (discovery or version)" target`
  Ejecuta un subconjunto de scripts combinando categorías.
- `nmap -oX salida.xml target`
  Genera salida en XML para post-procesado.

3.5.1 Idle Scan (escaneo sigiloso)



El *Idle Scan* (`-sI`) permite que el escáner no revele su IP al objetivo: se usa un host *zombie* con una pila IP que usa un IP ID predecible. El escáner induce al zombie a enviar/recibir paquetes y deduce el estado de puertos del objetivo midiendo cambios en el IP ID del zombie. Técnica potente pero con requisitos específicos (zombie apropiado).

La idea central es que el escáner (attacker) quiere averiguar si un puerto en el *target* está abierto sin que el target vea la IP del escáner. Para ello aprovecha un *host tercero* (el *zombie*) cuyo contador de **IP ID** es predecible y global. El escáner induce, mediante paquetes falsificados (spoofing), que el target envíe respuestas al zombie; el comportamiento del zombie frente a esas respuestas (responder o no) modifica su contador IP ID, y midiendo ese contador antes y después se puede inferir el estado del puerto del target.

Requisitos para que funcione

1. **Zombie adecuado:**
   - Debe utilizar un esquema de IP ID *predictable* (por ejemplo, valor global que incrementa por cada paquete enviado). Muchos sistemas antiguos/unix usaban esto; muchos SO modernos usan valores aleatorios o IDs por-flujo, lo que rompe la técnica.
   - Debe estar "idle": poco o nada de tráfico saliente propio (si el zombie envía paquetes por su cuenta, el IP ID se mueve y corrompe las medidas).
2. **Capacidad de spoofing:** el escáner debe poder enviar paquetes con `src IP = zombie` hacia el target (requiere raw sockets/privilegios y que la red permita paquetes con origen falsificado — muchos routers/firewalls bloquean esto por políticas anti-spoofing BCP38).
3. **Target que responda a sondas TCP/UDP** según comportamiento estándar del stack.
4. **Ausencia de NAT/filtrado** que modifique las respuestas entre target y zombie de forma inesperada.

Secuencia paso a paso (conteo del IP ID)

Sea X el valor del IP ID inicial del zombie (antes de cualquier acción). El proceso típico usado por Nmap sigue este patrón:

1. **Sondeo inicial del zombie (Probe A):**
   - El escáner envía al zombie un paquete que hace que el zombie responda (por ejemplo, un paquete que provoque RST). Ese intercambio provoca que el zombie envíe una respuesta que incrementa su IP ID.

- El escáner observa y registra el IP ID devuelto; llamémoslo `id1`. (En la práctica `id1 = X+1`, porque la respuesta causada por esta sonda incrementó el contador.)

2. **Sonda falsa al target (Spoofed probe):**
   - El escáner envía un `SYN` al *target* con la dirección origen falsificada igual a la IP del *zombie*.
   - El *target* reacciona dependiendo del estado del puerto:
     - **Si el puerto del target está abierto:** el target enviará un `SYN/ACK` a la IP del zombie. Al recibir el `SYN/ACK`, el zombie, sin sesión, responderá con un `RST`, lo que provoca que el zombie envíe un paquete y aumente su IP ID en 1.
     - **Si el puerto del target está cerrado:** el target enviará un `RST` al zombie. Dado que el zombie recibe un `RST` para una conexión inexistente, normalmente no responde, por lo que su IP ID no se incrementa por esta interacción.

3. **Sondeo final del zombie (Probe B):**
   - El escáner vuelve a provocar que el zombie envíe una respuesta (misma técnica que Probe A) y lee el nuevo valor de IP ID, `id2`. Esta sonda también provoca que el IP ID del zombie aumente un valor adicional por la propia sonda.

4. **Interpretación de la diferencia `id2 - id1`:**
   - Si `id2 == id1 + 2` $\Rightarrow$ **puerto abierto** en el target. (Probe A incrementó a $X+1$; la interacción abierta provocó $+1$ más $\Rightarrow X+2$; Probe B provoca otro incremento observado $\Rightarrow X+3$, por lo que `id2 - id1 = 2`.)
   - Si `id2 == id1 + 1` $\Rightarrow$ **puerto cerrado**. (La interacción con un puerto cerrado no hizo que el zombie respondiera, por lo que solo los probes A y B incrementan: `id2 - id1 = 1`.)
   - Valores mayores o inconsistentes indican tráfico adicional en el zombie, pérdida de paquetes, reordenamientos o que el zombie no es confiable.

3.6 Evasión, timing y consideraciones operativas

- **Timing templates:** `-T0` a `-T5` controlan la agresividad y velocidad; `-T3` por defecto, `-T4` para redes rápidas, `-T0`/`-T1` para stealth.
- **Evasión de IDS/IPS:** fragmentación de paquetes, alteración de fuentes/puertos, pausas aleatorias, uso de `-D` (decoy hosts) o `-S` spoofed source (requiere raw privileges y puede romper legalidad). Estas técnicas aumentan la complejidad y el riesgo legal.
- **Ruido y detección:** muchos escaneos son ruidosos (logs en firewalls/IDS). Use métodos autorizados y registro de pruebas en entornos de auditoría controlados.