M2: Reconocimiento y Fingerprinting

1. Introducción al reconocimiento en pentesting

El reconocimiento es la **primera etapa activa en un test de intrusión**. En metodologías como **PTES (Penetration Testing Execution Standard)** o **OSSTMM (Open Source Security Testing Methodology Manual)**, se considera la fase crítica que define el éxito del resto del proceso.

Un error típico de auditores novatos es lanzarse directamente a intentar explotar vulnerabilidades sin haber hecho un mapa detallado del objetivo. El profesional, en cambio, sabe que, **sin una fase de reconocimiento sólida, el ataque será ineficaz o demasiado ruidoso**.

1.1 Objetivos del reconocimiento

- Identificar **activos expuestos**: dominios, subdominios, IPs, servicios, tecnologías.
- Descubrir **vectores de ataque probables**: puertos abiertos, versiones vulnerables, configuraciones inseguras.
- Correlacionar información pública y técnica para construir un **perfil de ataque realista**.
- Detectar **riesgos indirectos**, como empleados expuestos en redes sociales o fugas de información en certificados SSL.

1.2 Tipos de reconocimiento

- **Pasivo (OSINT):**
  Se obtiene información sin interactuar directamente con los sistemas objetivo. Ejemplos: Shodan, Google Dorks, registros WHOIS, leaks en bases de datos.

  Ventaja: *difícil de detectar.*

- **Activo:**
  El auditor interactúa con los sistemas, por ejemplo, escaneando puertos con Nmap.

  Ventaja: *resultados más precisos.*
  Desventaja: *genera tráfico que puede ser registrado por IDS/IPS o firewalls.*

1.3 Importancia en un pentest profesional

En un test de intrusión real, el reconocimiento ocupa fácilmente un **30–40% del tiempo total**. Sin esta fase, el informe final sería incompleto o basado en conjeturas.

La meta no es “ejecutar herramientas”, sino **interpretar la información** y **transformarla en inteligencia de ataque**.

2. OSINT avanzado

2.1 Definición y marco legal

OSINT (*Open Source Intelligence*) consiste en la **recolección de inteligencia a partir de fuentes abiertas y accesibles al público**.

En el marco legal europeo y español, el OSINT es generalmente legal, siempre que no se acceda a datos protegidos o sistemas privados sin autorización.

Ejemplo de legalidad:

- Revisar certificados SSL con `crt.sh` es legal.
- Acceder con credenciales filtradas a un sistema sin autorización es **ilegal** (delito según Código Penal español, art. 197).

2.2 Shodan

Shodan es un motor de búsqueda que indexa banners de servicios de dispositivos conectados a Internet. Mientras Google indexa contenido web, Shodan indexa **puertos, servicios, versiones de software, certificados SSL, ubicaciones geográficas**.

```
apache country:ES
```

Devuelve servidores Apache en España.

```
port:21 vsftpd
```

Busca servidores FTP corriendo vsFTPd.

```
org:"Universidad Complutense" port:25
```

Servidores SMTP asociados a esa organización.

```
ssl.cert.subject.cn:*.empresa.com
```

Subdominios de empresa.com extraídos de certificados.

Uso de la API de Shodan

Ejemplo en Python:

```
import shodan
api = shodan.Shodan('API_KEY')
results = api.search('nginx country:ES')
```

```
for r in results['matches']:
    print(r['ip_str'], r['port'], r['data'])
```

Interpretación: con un script así, podemos generar un inventario de servidores Nginx en España, con IP, puerto y banner.

Ejemplo

Supongamos que buscamos:

```
"apache country:ES city:Madrid"
```

Shodan nos devuelve 20 servidores Apache en Madrid. Entre ellos:

- `IP: 83.45.123.67` Apache/2.4.7 (Ubuntu).
- `IP: 213.99.45.10` Apache/2.2.22 (muy antiguo).

Un auditor marcaría el segundo host como **prioridad alta** por su riesgo de explotación.

2.3 theHarvester

theHarvester es una herramienta de recolección OSINT diseñada específicamente para recopilar **correos electrónicos, subdominios, hosts y nombres de empleados** desde fuentes públicas.

Es muy popular porque permite a un auditor reunir rápidamente información clave antes de realizar pruebas activas.

Funcionamiento

- Realiza consultas en buscadores como Google, Bing, Yahoo, DuckDuckGo.
- Consulta fuentes específicas como LinkedIn, Baidu o incluso Shodan.
- Extrae direcciones de correo, nombres de host y subdominios asociados al dominio objetivo.

Opciones principales

```
theHarvester -d dominio.com -b fuente -l límite -f salida
```

- `-d dominio.com` dominio objetivo.
- `-b fuente` motor de búsqueda o “all” para todos.
- `-l límite` número de resultados.
- `-f salida` exporta resultados en HTML o XML.

Ejemplo práctico

```
theHarvester -d empresa.com -b all -l 500 -f informe_empresa.html
```

Salida esperada (fragmento):

```
Emails found:
admin@empresa.com
soporte@empresa.com
j.garcia@empresa.com

Hosts found:
vpn.empresa.com
mail.empresa.com
intranet.empresa.com
```

Interpretación de resultados

- **Correos electrónicos** sirven para ataques de *phishing* (en fase de ingeniería social), pero también para ataques técnicos (fuerza bruta en OWA, VPN, etc.).
- **Subdominios** revelan infraestructura adicional, muchas veces olvidada o no actualizada.
- **Hosts** ayudan a construir el mapa de la red externa de la empresa.

Nivel avanzado: correlación

- Los correos encontrados pueden comprobarse en **HaveIBeenPwned** o bases de datos de leaks para detectar contraseñas filtradas.
- Los subdominios pueden validarse con Nmap/Masscan para comprobar si exponen servicios inseguros.
- Integración con Maltego: los resultados de theHarvester pueden exportarse y visualizarse en un grafo de relaciones.

2.4 Maltego avanzado

Maltego es una herramienta de **inteligencia y análisis de relaciones**, especialmente potente en la fase de OSINT.

A diferencia de herramientas de línea de comandos, Maltego es visual: construye grafos que muestran cómo se relacionan entidades como dominios, correos, IPs, redes sociales o leaks.

Concepto de *Transforms*

Un *transform* es una consulta predefinida que parte de un dato conocido y genera datos relacionados.

Ejemplo:

- Partimos de `empresa.com`.
- Ejecutamos transform: obtenemos IP pública asociada.
- De la IP obtenemos ASN y rango de red.
- De correos recolectados obtenemos perfiles de LinkedIn o leaks en bases de datos públicas.

Flujo práctico en Maltego

1. **Entidad inicial:** dominio `empresa.com`.
2. **Transform a DNS records:** encuentra `mail.empresa.com`, `vpn.empresa.com`.
3. **Transform a IPs:** asigna 185.45.12.34 a `vpn.empresa.com`.
4. **Transform a ASN:** descubre que pertenece al ASN de Telefónica.
5. **Transform a correos:** recolecta `soporte@empresa.com`.
6. **Transform a redes sociales:** conecta `soporte@empresa.com` con perfil de LinkedIn.

Resultado: un grafo visual que conecta **infraestructura técnica (servidores)** con **infraestructura humana (empleados)**.

Ejemplo real

Un pentester trabaja para una universidad:

- Con Maltego, parte del dominio `universidad.edu`.
- Encuentra 25 correos de personal administrativo.
- Descubre que 3 de esos correos aparecen en leaks de 2019 con contraseñas sin cifrar.
- Correlaciona subdominios y detecta `vpn.universidad.edu`.
- Conecta los datos: empleados filtrados + VPN expuesta = vector de ataque prioritario.

Nivel avanzado

- **Integración con APIs:** Maltego puede conectarse a VirusTotal, Shodan, HaveIBeenPwned, etc.
- **Investigación forense:** útil no solo en pentesting, sino en ciberinteligencia (seguimiento de actores, análisis de cibercrimen).

2.5 Comparación Shodan, theHarvester y Maltego

HERRAMIENTA	TIPO	FORTALEZAS	LIMITACIONES
SHODAN	Buscador de dispositivos	Permite encontrar servicios expuestos en Internet. Ideal para búsqueda por puerto/tecnología.	Limitado a lo que indexa Shodan; no descubre correos ni relaciones humanas.
THEHARVESTER	Recolector de correos y subdominios	Rápido y automatizado para correos, subdominios y hosts.	Resultados dependen de buscadores externos; puede devolver duplicados.
MALTEGO	Análisis visual de relaciones	Potente para correlacionar dominios, IPs, correos y redes sociales.	Requiere más tiempo y conocimiento para interpretar grafos.

3. Enumeración de servicios y puertos

Una vez realizada la fase de OSINT pasivo, el siguiente paso en un pentest es la **enumeración activa de servicios y puertos**.

Esto implica interactuar directamente con los sistemas objetivo, normalmente mediante **escaneadores de red**.

La enumeración avanzada no se limita a detectar si un puerto está abierto o cerrado:

- Permite conocer **qué servicio corre en ese puerto**.
- Identificar la **versión exacta del software**.
- Detectar **configuraciones incorrectas o vulnerabilidades conocidas**.

3.1 Escaneo con Nmap avanzado

Nmap (Network Mapper) es la herramienta estándar de la industria. Se utiliza en auditorías de todo tipo, desde pentesting corporativo hasta investigaciones forenses.

3.1.1 Modos de escaneo

- **SYN Scan (-sS):** el más utilizado, rápido y sigiloso. Envía paquetes SYN sin completar la conexión.
- **Connect Scan (-sT):** más detectable, establece conexiones completas TCP.
- **UDP Scan (-sU):** más lento, pero crítico para detectar servicios como DNS (53), SNMP (161), NTP (123).
- **ACK Scan (-sA):** útil para identificar reglas de firewall.
- **FIN/NULL/XMAS Scan (-sF, -sN, -sX):** técnicas furtivas que envían paquetes anómalos para eludir IDS.

3.1.2 Opciones avanzadas de Nmap

- Escanear todos los puertos:

```
nmap -p- 192.168.1.100
```

- Detectar servicios y versiones:

```
nmap -sV 192.168.1.100
```

- Detección de sistema operativo:

```
nmap -O 192.168.1.100
```

- Combinar técnicas:

```
nmap -sS -sV -O -p- 192.168.1.100
```

3.1.3 NSE (Nmap Scripting Engine)

El NSE permite ejecutar scripts especializados. Se clasifican en categorías:

- `auth` scripts de autenticación.
- `vuln` detección de vulnerabilidades.
- `discovery` descubrimiento de servicios.
- `brute` ataques de fuerza bruta controlados.

Ejemplo:

```
nmap -sV --script=vuln 192.168.1.100
```

Esto puede devolver vulnerabilidades conocidas, como **SSL POODLE** o **Heartbleed**.

3.1.4 Interpretación de resultados

Ejemplo de salida de Nmap:

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu))
443/tcp  open  ssl/https Apache httpd 2.4.29
3306/tcp open  mysql   MySQL 5.7.33-0ubuntu0.18.04.1
```

Interpretación profesional:

- **SSH 7.6p1** revisar si permite autenticación con contraseña.
- **Apache 2.4.29** versión antigua, potencialmente vulnerable a CVE-2019-0211.
- **MySQL 5.7.33** riesgo si expuesto públicamente; comprobar autenticación y permisos.

3.2 Escaneo con Masscan

Masscan es el escáner más rápido disponible.

- Capaz de escanear toda la red IPv4 en cuestión de minutos.
- Utiliza su propia librería de TCP/IP, lo que le da velocidad, pero reduce precisión.

3.2.1 Uso básico

```
masscan 192.168.1.0/24 -p1-65535 --rate=1000
```

- Escanea todos los puertos en un rango de red.
- `--rate` controla la velocidad de envío de paquetes (1000 = 1000 p/s).

3.2.2 Flujo combinado Masscan + Nmap

1. Usar Masscan para descubrir rápidamente puertos abiertos.
2. Pasar los resultados a Nmap para análisis detallado.

Ejemplo:

```
masscan 192.168.1.0/24 -p1-65535 --rate=500 -oG masscan.txt
awk '/open/ {print $4}' masscan.txt > targets.txt
nmap -sV -iL targets.txt -oA nmap_detallado
```

Así logramos lo mejor de ambos mundos: **velocidad + precisión**.

3.3 Técnicas de evasión en escaneos

En auditorías avanzadas, es importante no generar demasiado ruido. Los sistemas defensivos modernos (IDS/IPS, SIEMs) pueden detectar escaneos agresivos.

Opciones de evasión en Nmap:

- **Decoys:**

  ```
  nmap -D RND:10 192.168.1.100
  ```

  Envía tráfico falso desde 10 IPs ficticias.

- **Source port spoofing:**

  ```
  nmap --source-port 53 192.168.1.100
  ```

  Simula que el tráfico viene de un servidor DNS (a veces permitido por firewalls).

- **MAC spoofing:**

  ```
  nmap --spoof-mac Cisco 192.168.1.100
  ```

  Finge que la máquina tiene una MAC de Cisco.

- **Fragmentación:**

  ```
  nmap -f 192.168.1.100
  ```

  Envía paquetes fragmentados para evadir IDS.

Nota: estas técnicas pueden ser vistas como actividad maliciosa si no se tiene autorización. Deben usarse **únicamente en un pentest autorizado**.

3.4 Comparativa de herramientas

HERRAMIENTA	VELOCIDAD	PRECISIÓN	CASOS DE USO
NMAP	Media	Muy alta	Escaneo detallado, detección de versiones, scripts NSE
MASSCAN	Muy alta	Media	Descubrimiento rápido en redes grandes
UNICORNSCAN	Alta	Media	Alternativa menos usada, pero flexible para investigación

4. Fingerprinting avanzado

El **fingerprinting** es el proceso de identificar con precisión las **tecnologías, versiones y configuraciones** que utilizan los servicios y aplicaciones de un sistema.

Mientras la enumeración responde a la pregunta: "¿Qué puertos y servicios están abiertos?",

el fingerprinting responde: "¿Qué software exacto está detrás y en qué versión?".

Esto es crucial porque:

- Permite correlacionar versiones con vulnerabilidades públicas (**CVE, Exploit-DB, NVD**).
- Ayuda a diseñar ataques más precisos y menos ruidosos.
- Evita falsos positivos, al diferenciar tecnologías con comportamientos similares.

4.1 Fingerprinting web

Las aplicaciones web son uno de los principales objetivos en cualquier auditoría.

4.1.1 WhatWeb

Herramienta diseñada para identificar **CMS, frameworks, servidores, librerías** y otras tecnologías web.

Ejemplo básico:

```
whatweb https://www.empresa.com
```

Ejemplo avanzado:

```
whatweb -a 3 -v https://www.empresa.com
```

- `-a 3`: nivel de agresividad máximo (fuerza búsquedas más profundas).
- `-v`: modo verbose, más información en la salida.

Posible salida:

```
  http://empresa.com [200 OK] Country[SPAIN][ES],
Apache[2.4.29],
```

```
OpenSSL[1.1.1],
PHP[7.2.24],
WordPress[5.4],
Google-Analytics,
Bootstrap
```

Interpretación:

- Apache 2.4.29 versión antigua, revisar CVE-2019-0211.
- WordPress 5.4 versión desactualizada, riesgo de exploits públicos.
- PHP 7.2.24 fin de soporte, riesgo elevado.

4.1.2 Wappalyzer

Extensión de navegador y herramienta CLI que detecta tecnologías frontend.
Ejemplo CLI:

```
wappalyzer https://www.empresa.com
```

Resultados típicos:

- Framework: React 16.8.
- JS library: jQuery 3.3.1.
- CMS: WordPress.

El valor de Wappalyzer está en descubrir librerías frontend que puedan estar vulnerables (ej. jQuery < 3.5 vulnerable a XSS).

4.1.3 Fingerprinting con Nmap

Nmap también permite hacer fingerprinting de aplicaciones web.

Ejemplo:

```
nmap -p80,443 --script=http-headers www.empresa.com
```

Salida:

```
Server: Apache/2.4.29 (Ubuntu)
X-Powered-By: PHP/7.2.24
```

Ejemplo con detección de directorios:

```
nmap -p80 --script=http-enum www.empresa.com
```

Posible salida:

```
/admin (200 OK)
/phpmyadmin (403 Forbidden)
/uploads (200 OK)
```

Interpretación: rutas como `/phpmyadmin` o `/admin` son **vectores de ataque de alta prioridad**.

4.2 Fingerprinting de servicios no web

4.2.1 Netcat (banners manuales)

Netcat permite conectarse a un puerto y leer el banner devuelto.

Ejemplo:

```
nc 192.168.1.50 21
```

Salida:

```
220 (vsFTPd 3.0.3)
```

Interpretación: versión concreta de FTP, comprobar si existe un CVE asociado (ej. vsFTPd 2.3.4 tenía backdoor).

4.2.2 Telnet

Aunque en desuso, Telnet sigue siendo útil para fingerprinting.

Ejemplo:

```
telnet 192.168.1.50 25
```

Salida:

```
220 mail.empresa.com ESMTP Postfix 2.11
```

Interpretación: versión de Postfix, revisar en bases de datos CVE.

4.2.3 Nmap NSE para servicios específicos

- SMB:

```
nmap --script smb-os-discovery -p445 192.168.1.50
```

Salida:

```
OS: Windows Server 2012 R2
Name: DC01
Workgroup: EMPRESA
```

- SSL/TLS:

```
nmap --script ssl-enum-ciphers -p443 www.empresa.com
```

Salida:

```
TLSv1.0 enabled (insecure)
TLSv1.2 enabled
Weak cipher: RC4
```

Esto permite detectar configuraciones criptográficas inseguras.

4.3 Fingerprinting en entornos industriales (ICS/SCADA)

Muchos sistemas industriales exponen servicios inseguros, y son especialmente sensibles.

Ejemplo:

```
nmap -sV --script modbus-discover -p502 192.168.1.200
```

Salida:

```
Modbus device information
VendorName: Schneider Electric
ProductCode: M340
```

Interpretación: PLC específico detectado, requiere medidas especiales, ya que explotar vulnerabilidades en un PLC puede interrumpir procesos industriales.

4.4 Ejemplo narrado de fingerprinting completo

Escenario: auditoría de un servidor web corporativo.

1. **Enumeración inicial con Nmap:**

```
22/tcp   open  ssh     OpenSSH 7.6p1
80/tcp   open  http    Apache httpd 2.4.29
443/tcp  open  ssl/http Apache httpd 2.4.29
```

2. **Fingerprinting con WhatWeb:**

   ```
   Apache/2.4.29, PHP/7.2.24, WordPress 5.4
   ```

3. **Cabeceras con Nmap:**

   ```
   Server: Apache/2.4.29 (Ubuntu)
   X-Powered-By: PHP/7.2.24
   ```

4. **Banner SSH con Netcat:**

   ```
   SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
   ```

Conclusión:

- Apache y PHP están obsoletos, buscar CVEs en Exploit-DB.
- WordPress desactualizado, riesgo alto de exploits públicos.
- SSH 7.6p1 revisar si permite contraseñas débiles.

5. Casos prácticos narrados

Caso A: Auditoría de un e-commerce

Escenario:
La empresa "ShopOnline" contrata una auditoría de seguridad sobre su portal de comercio electrónico.

Fase 1 – OSINT

- Se usa **theHarvester**:

  ```
  theHarvester -d shoponline.com -b all -l 200 -f report.html
  ```

  Resultado: 12 correos de empleados, entre ellos `admin@shoponline.com`.

- En **Shodan**, búsqueda:

  ```
  ssl.cert.subject.cn:*.shoponline.com
  ```

  Aparece un subdominio `payments.shoponline.com`.

Fase 2 – Enumeración

- Escaneo con Masscan:

```
masscan shoponline.com -p1-65535 --rate=1000
```

Detecta puertos 22, 80, 443 y 3306.

- Nmap detallado:

```
22/tcp  ssh     OpenSSH 7.4
80/tcp  http    Apache 2.4.29
443/tcp ssl/http Apache 2.4.29
3306/tcp mysql  MySQL 5.5.62
```

Fase 3 – Fingerprinting

- WhatWeb en `shoponline.com`:

```
Apache/2.4.29, PHP/7.2.24, WordPress 5.3.1
```

- En `payments.shoponline.com`:

```
Nginx/1.14, Magento 2.2
```

Análisis final:

- WordPress 5.3.1 y Magento 2.2 son versiones antiguas con CVEs críticos.
- MySQL 5.5 expuesto a Internet es un riesgo grave.
- Vectores de ataque prioritarios: SQL Injection en Magento y explotación de CVEs en Apache/PHP.

Caso B: Red corporativa interna

Escenario:
Una multinacional permite auditar su red interna (rango 10.0.0.0/16).

Fase 1 – Escaneo rápido con Masscan

```
masscan 10.0.0.0/16 -p1-65535 --rate=5000
```

Detecta 1.200 hosts con puertos abiertos.

Fase 2 – Escaneo detallado con Nmap

```
nmap -sS -sV -O -iL hosts.txt -oA corp_scan
```

Servicios comunes: SMB (445), RDP (3389), MSSQL (1433).

Fase 3 – Fingerprinting SMB

```
nmap --script smb-os-discovery -p445 10.0.5.23
```

Resultado:

```
OS: Windows Server 2012 R2
Domain: CORP
Workgroup: CORP-NET
```

Fase 4 – Correlación

- Muchos servidores son Windows 2012, fin de soporte, parcheado incompleto.
- Algunos expuestos con SMBv1, riesgo de EternalBlue.

Análisis final:

- La empresa debe priorizar desactivar SMBv1 y actualizar servidores antiguos.

Caso C: SCADA industrial

Escenario:
Auditoría a una planta de energía que usa PLCs Modbus.

Escaneo Nmap especializado:

```
nmap -sV --script modbus-discover -p502 192.168.100.50
```

Resultado:

```
Modbus device information
Vendor: Schneider Electric
Product: Modicon M340
```

Interpretación:

- El dispositivo revela fabricante y modelo sin autenticación.
- Riesgo: un atacante podría enviar comandos maliciosos para interrumpir procesos industriales.

Nota: en entornos industriales, **el reconocimiento debe hacerse con precaución extrema** para no interrumpir sistemas críticos.

6. Detección y evasión

Un auditor ético no solo debe saber detectar servicios, también debe ser consciente de **cómo lo detectan los defensores**.

6.1 Detección por defensores

- **IDS/IPS (Snort, Suricata):** detectan patrones de escaneo (ej. muchos SYN seguidos).
- **Firewalls con rate-limiting:** bloquean IPs que escanean muchos puertos en poco tiempo.
- **SIEMs (Splunk, ELK):** correlacionan logs de sistemas diferentes para detectar reconocimiento.

6.2 Técnicas de evasión en Nmap

- **Uso de decoys:**

  ```
  nmap -D 192.168.1.10,192.168.1.20,ME 192.168.1.100
  ```

  Mezcla la IP real con señuelos.

- **Fragmentación de paquetes:**

  ```
  nmap -f 192.168.1.100
  ```

  Divide paquetes en fragmentos pequeños, difícil de analizar por IDS.

- **MAC spoofing:**

  ```
  nmap --spoof-mac Cisco 192.168.1.100
  ```

  Suplantar fabricante de tarjeta de red.

- **Timing y velocidad:**

  - `-T0` muy lento y sigiloso.
  - `-T5` muy rápido y ruidoso.

Un pentester avanzado ajusta estas técnicas según los objetivos del contrato (ej. auditoría furtiva vs auditoría abierta).

7. Buenas prácticas y errores comunes

7.1 Buenas prácticas

- Documentar **todo**: fecha, hora, herramienta, parámetros usados, resultados obtenidos.
- Usar un **cuaderno de pentest** digital (ej. CherryTree, Obsidian, Joplin).
- Escanear **progresivamente**: primero Masscan para descubrimiento, luego Nmap para detalle.
- Usar herramientas complementarias (Censys, ZoomEye) para validar resultados.
- Mantener el reconocimiento **dentro del alcance autorizado** del contrato.

7.2 Errores comunes de principiantes

- Escanear solo puertos “comunes” (ej. top 1000 de Nmap) y olvidar puertos altos: muchos backdoors se esconden en puertos no estándar.
- Ignorar servicios UDP: SNMP, DNS o NTP pueden ser puertas de entrada críticas.
- No interpretar resultados: obtener la versión de Apache y no revisarla en CVE Details no sirve de nada.
- Ser demasiado agresivo: lanzar Masscan con `--rate 100000` y tumbar servicios en producción.
- No correlacionar OSINT con escaneos. información dispersa sin análisis integrado pierde valor.