Daniel Pérez **Máster Hacking Ético**

Framework Pentesting

Para esta práctica he decidido basarme en el Modelo de **Pentesting PTES**, el cual se compone de 7 fases, y puede revisarse desde este enlace.

A continuación, mostraré un Framework a seguir basado en este modelo, explicando a su vez en qué consiste cada una de las fases, y qué herramientas usaría en cada una de ellas.

Fase 1: Pre-engagement (pre-acuerdo)

El objetivo de esta fase consiste en definir el alcance del Pentest, qué reglas deben respetarse, qué objetivos se pretenden cumplir y qué pruebas están permitidas. Durante esta fase hay que reunirse con el cliente, y según el tipo de Pentesting (Caja Blanca, Caja Gris o Caja Negra), deberá proporcionar los datos importantes de prueba, como direcciones IP, dominios y credenciales de algún (o varios) usuario/s (aunque en el caso de realizar un Pentest de caja negra, no conoceremos ningún dato). También habrá que plantear hacer copias de seguridad y proponer una manera segura de comunicar incidencias con el cliente durante el Pentesting.

Como herramientas a utilizar en esta fase, habrá que emplear documentación proporcionada por el cliente, una checklist de riesgos, y emplear un canal seguro para comunicación, como cifrado OpenPGP o algún sistema de OPS Ticketing (como osTicket).

Es muy importante llevar a cabo las pruebas SÓLO CON AUTORIZACIÓN EXPRESA. Por eso hay que pactar con el cliente y registrar todas las pruebas por escrito antes de llevarlas a cabo.

Fase 2: Intelligence Gathering (Enumeración pasiva y activa)

El objetivo de esta fase consiste en recoger información pública y activa sobre el objetivo (hosts, dominios, empleados, etc), para crear un esquema de ataque.

¿Qué información se puede recopilar en esta fase? Pues por ejemplo dominios DNS, subdominios, hosts, puertos abiertos, tecnologías web, servicios expuestos, usuarios en Linkedin, correos, etc.

Como herramientas a utilizar en esta fase, habrá dos tipos, **pasivas y activas**, y habrá que usar unas u otras en función de la situación (las activas por ejemplo hacen más ruido que las pasivas, por lo que según lo que se dictamine con el cliente, hay que plantearse si las usarían unos atacantes reales o si por lo contrario preferían pasar más desapercibidos).

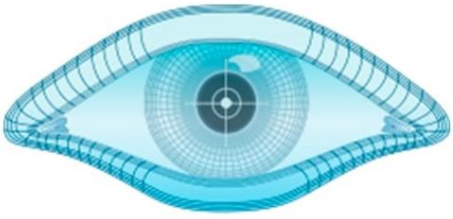
Como herramientas pasivas podemos usar varias que ya conocemos de OSINT, como **Maltego, Harvester** o servicios online como WhatsMyName.

Y como herramientas activas tenemos **Nmap** y **Masscan** para escaneo de puertos a gran escala, **Nuclei** para plantillas de reconocimiento rápido, o **Nikto** para detección de configuraciones web básicas.

También disponemos de herramientas de fingerprinting para aplicaciones Web, tales como **Burp Suite** (Escáner pasivo/proxy) y **OWASP ZAP** (proxy y spidering).

Todo esto nos permitirá por ejemplo recoger una lista de hosts, puertos y servicios, huellas tecnológicas, y hacer un mapa de subdominios.

Es importante mencionar que es de vital importancia **minimizar el impacto**, por lo que hay que evitar escaneos de alta intensidad fuera de las ventanas acordadas.



Fase 3: Threat Modeling & Vulnerability Analysis (Modelado de amenazas y análisis de vulnerabilidades)

El objetivo de esta fase consiste en priorizar vectores de ataque relevantes y mapear vulnerabilidades explotables. Hay que correlacionar activos críticos con amenazas, ejecutar escáneres de vulnerabilidades, validar falsos positivos y clasificar los posibles riesgos.

Como herramientas a utilizar en esta fase, podemos emplear **Nessus** para escaneo de vulnerabilidades de host e infraestructura, las ya mencionadas **Nuclei** para crear plantillas rápidas para CVEs y exposiciones y **Nikto** para configuraciones web paupérrimas, y **Burp Suite o OWASP ZAP** para interacciones de cara a identificar inyecciones, sesiones, etc, a nivel de descubrimiento.

Con esto podremos tener un listado de vulnerabilidades con evidencia, incluso con clasificación CVSS o similar.

Es importante en esta fase clasificar correctamente los falsos positivos, y no explotar vulnerabilidades críticas sin la aprobación expresa del cliente.

Fase 4: Vulnerability Validation / Exploitation (Validación y explotación de vulnerabilidades)

El objetivo de esta fase será validar vulnerabilidades de alto impacto (si nos lo ha permitido el cliente) y demostrar factibilidad. Haremos pruebas controladas para confirmar vectores, pruebas de explotación con medidas de contención, y registraremos las evidencias.

Como herramientas a utilizar en esta fase, podemos emplear **Metasploit** para framework de pruebas y validación de exploits, que es una herramienta bastante potente y adecuada en entornos controlados, **Burp Suite** para pruebas web dirigidas como intruder, repeater o profiler, y también **OWASP ZAP** para pruebas activas de web.

Luego deberemos entregar pruebas reproducibles de vulnerabilidades validadas, como capturas de logs y capturas de pantalla, y el impacto estimado de las pruebas.

Es muy importante no ejecutar exploits destructivos, a fin de preservar la integridad de los datos del cliente. En su lugar, es preferible una validación no intrusiva, o usar entorno de pruebas si hay riesgo de interrupción.

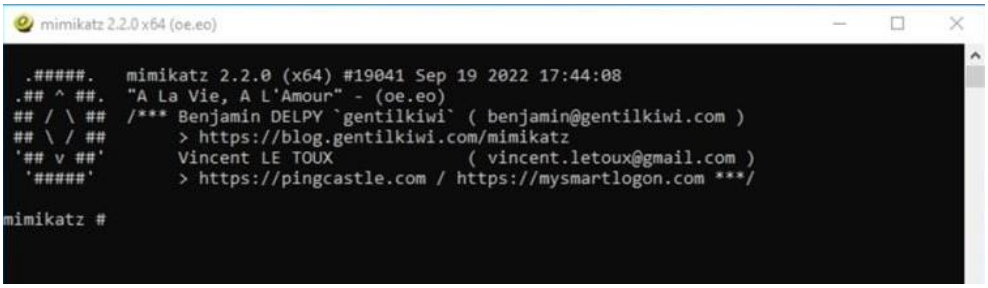
Fase 5: Post-Explotación, Movimiento Lateral y Persistencia

El objetivo de esta fase consiste en entender el alcance **post-compromiso con el cliente**, el acceso a datos sensibles, y si está permitido, moverse lateralmente y persistir en el ejercicio de obtener datos valiosos. Por ejemplo, deberemos realizar una enumeración interna, intentar lograr hacer un **escalado de privilegios**, un **mapeo del Active Directory** (si existiese), e intentar **extraer credenciales**.

Como herramientas a utilizar en esta fase, podemos emplear **BloodHound** para mapear relaciones y caminos de escalado en Active Directory (muy útil para análisis de privilegios), **Mimikatz** para extraer credenciales en memoria, **Sliver C2** como herramienta de adversary simulation/C2 (buena alternativa a CobaltStrike), y **WinPEAS ó LinPEAS** para hacer una enumeración automática de escalado de privilegios Windows/Linux

Así tendremos un mapa de accesos bastante completo, un listado de cuentas y privilegios comprometidos, y podremos indicar recomendaciones de remediación para bloquear caminos de escalado.

Es importante documentar bien todos los accesos para el cliente, y tener en cuenta que las herramientas que extraen credenciales (como Mimikatz) siempre deben usarse con cuidado por el impacto y las huellas que deja.



Fase 6: Reporte (Remediación y comunicación para el cliente)

El objetivo de esta fase es entregar resultados que el cliente pueda entender y con soluciones que pueda llevar a cabo, con recomendaciones concretas por nuestra parte. Habrá que redactar informe ejecutivo y técnico, con pruebas reproducibles (evidencias) de lo que se ha llevado a cabo, un plan de remediación, presentárselo al cliente, y si fuese necesario, hacer un retest.

Para el informe, podríamos utilizar este esquema de secciones:

1. Un resumen ejecutivo (con el impacto y las prioridades).
2. El alcance y la metodología (qué se probó y bajo qué condiciones).
3. Los hallazgos técnicos (ordenados por severidad), incluyendo evidencias (capturas, logs, hashes), y una mitigación propuesta.
4. Un mapa de riesgo y remediación priorizada.
5. Artefactos entregables, como IOCs, scripts seguros, y configuraciones
6. Y una lista de herramientas usadas, con detalles técnicos.

Es importante no publicar vulnerabilidades ni IOCs sin coordinación.

Fase 7: Cleanup y lecciones aprendidas

Por último, el objetivo de esta fase consistirá en eliminar cualquier rastro de prueba, restaurar las configuraciones como estaban y compartir lecciones con el cliente. Tendremos que revocar credenciales de prueba, eliminar las herramientas y cuentas creadas, entregar los logs, y hacer una reunión post-prueba con el cliente.

Las herramientas que usaremos en esta fase serán un checklist de cleanup, para poder llevar a cabo todas las recomendaciones pendientes, y scripts de reversión (aprobados por el cliente.

Será importante conservar las evidencias necesarias para el reporte y su cumplimiento, y eliminar los accesos activos.