

GUÍA DE ESTUDIO

Computer Security

Seguridad Informática — Examen a Título de Suficiencia

Periodo Escolar 2026/2 · Semestre VI

Profesor: M.B.A. Rodrigo Ojeda Santillán

Cobertura: los cuatro bloques temáticos del curso — hardening de sistemas operativos (Linux y Windows), seguridad y enumeración de red, acceso inicial y escalamiento de privilegios, y gestión segura de identidades y servicios.

Valor del examen: 100 puntos · **Duración:** 120 minutos · **Mínima aprobatoria:** 60

Cómo aprovechar esta guía. No es un resumen para leer una vez: es la lista de todo lo que el examen puede preguntar. Estudia cada bloque, memoriza la **hoja de comandos** (§5) y la **tabla MITRE** (§3.4), y al final resuelve la **autoevaluación** (§6) sin ver las respuestas. Presta atención especial a la sección de **errores comunes** (§7): están tomados de fallas reales.

Contenido

1. Estructura del examen y administración del tiempo
2. Bloque 1 — Hardening de Sistemas Operativos (Linux y Windows)
3. Bloque 2 — Seguridad de Red y Enumeración de Servicios
4. Bloque 3 — Acceso Inicial y Escalamiento de Privilegios (+ tabla MITRE ATT&CK)
5. Bloque 4 — Gestión Segura de Identidades y Servicios
6. Hoja de comandos esenciales (cheat sheet)
7. Autoevaluación con respuestas
8. Estrategia de examen y errores comunes que debes evitar

§1 Estructura del examen

El examen tiene cinco secciones que evalúan los cuatro bloques del semestre. Conoce el peso de cada una para administrar tu tiempo:

Sección	Contenido	Tipo	Puntos	Tiempo sugerido
I	Conceptos fundamentales y terminología	Opción múltiple	20	20 min
II	Hardening de sistemas (Linux / Windows)	Respuesta breve	20	25 min
III	Seguridad de red y enumeración de servicios	Mixto	25	30 min
IV	Acceso inicial y escalamiento de privilegios	Análisis / MITRE	20	25 min
V	Caso práctico integrador	Desarrollo	15	20 min

REGLA DE ORO

Se evalúa la **precisión técnica**, no la extensión. En las preguntas de comandos, **la sintaxis cuenta**: un comando conceptualmente correcto con errores graves recibe puntaje parcial. Y lo más importante — **contesta todas las preguntas**: en las abiertas hay puntaje parcial y dejar en blanco es garantizar cero.

§2 Bloque 1 — Hardening de Sistemas Operativos

Hardening = reducir la superficie de ataque de un sistema: desactivar lo innecesario, endurecer la configuración y activar controles de defensa.

2.1 Linux — Endurecimiento de SSH

SSH es el punto de entrada remoto más atacado. Medidas concretas de hardening (memoriza al menos tres distintas entre sí):

- **Deshabilitar el login de root:** `PermitRootLogin no` en `/etc/ssh/sshd_config`. Obliga a entrar con un usuario normal y luego escalar con `sudo`.
- **Deshabilitar autenticación por contraseña:** `PasswordAuthentication no` → forzar **autenticación por clave pública**.
- **Fail2Ban:** detecta y bloquea automáticamente intentos de fuerza bruta (banea la IP tras N intentos fallidos).
- **Algoritmos post-cuánticos:** usar intercambios de claves resistentes a cómputo cuántico, p. ej. `mlkem768x25519-sha256`.
- Complementos: cambiar el puerto por defecto, restringir con `AllowUsers`, limitar reintentos con `MaxAuthTries`.

PREGUNTA TÍPICA DE EXAMEN

¿Qué es `mlkem768x25519-sha256`? Un algoritmo de **intercambio de claves resistente a computación cuántica (post-cuántico)** — combina ML-KEM-768 con X25519. NO es un cifrador obsoleto ni un método de compresión.

POR QUÉ LA CLAVE PÚBLICA DERROTA LA FUERZA BRUTA

Con contraseñas, un atacante puede probar millones de combinaciones. Con clave pública, tendría que **adivinar la clave privada**, lo cual es **computacionalmente inviable**. Al deshabilitar contraseñas, el vector de fuerza bruta simplemente deja de existir.

2.2 Linux — Integridad y auditoría

Herramienta	Qué es / para qué sirve	Comando clave
AIDE Advanced Intrusion Detection Environment	Verificador de integridad de archivos. Crea una base de datos de hashes de archivos críticos y luego detecta si alguno fue modificado (posible intrusión). No es un auditor de vulnerabilidades.	<code>aide --init</code> (inicializa la base de integridad)
Lynis	Herramienta de auditoría de hardening . Revisa el sistema y produce un índice de hardening (0–100) que puedes reportar como métrica cuantitativa.	<code>lynis audit system</code>
systemd-analyze security	Asigna un puntaje de exposición (0–10) a un servicio y evalúa su nivel de hardening (cuanto más bajo, más seguro).	<code>systemd-analyze security <servicio></code>

2.3 Linux — systemd y firewall

- **DynamicUser=yes:** hace que systemd asigne un **UID/GID efímero** al servicio, que **se libera al detenerlo, sin dejar rastro** en `/etc/passwd`. Reduce la superficie: no hay cuenta permanente que abusar.

- **nftables**: el firewall moderno de Linux, con familia `inet` (maneja IPv4 e IPv6 juntos). Reemplaza a `iptables`.

2.4 Windows 11 — Controles clave

Control	Qué protege
Credential Guard	Aísla y protege las credenciales (secretos de LSA) mediante seguridad basada en virtualización. Evita el robo de credenciales y ataques tipo <i>pass-the-hash</i> .
BitLocker	Cifrado de disco completo (usa AES). Protege los datos en reposo: si roban el equipo o el disco, la información es ilegible sin la clave.
ASR Attack Surface Reduction	Conjunto de reglas de Microsoft Defender que bloquean comportamientos riesgosos: macros de Office maliciosas, ejecutables lanzados desde correo, scripts ofuscados, etc. No es un firewall ni protege navegadores.

2.5 Seguridad de contenedores — SBOM

CONCEPTO CLAVE

Un **SBOM (Software Bill of Materials)** es un **inventario de todos los componentes y dependencias** que contiene una imagen o software. Es relevante porque permite saber exactamente qué hay dentro de un contenedor y **correlacionarlo con CVEs conocidos** (detectar vulnerabilidades). **Herramientas que lo generan**: Syft, Trivy, Gype. (No confundir: el SBOM no "protege datos", es un inventario.)

§3 Bloque 2 — Seguridad de Red y Enumeración de Servicios

3.1 Herramientas y su función

Herramienta	Función principal
ss	Reemplazó a <code>netstat</code> como estándar moderno para inspeccionar sockets en escucha en Linux. Ej.: <code>ss -tlnp</code> .
Nmap	Escáner de referencia con detección de versión (<code>-sV</code>) y scripts NSE.
RustScan / Naabu	Escáner asíncrono (RustScan escrito en Rust) que descubre los 65 535 puertos en segundos.
Nuclei	Motor de plantillas que correlaciona servicios con CVEs conocidos.
nftables	Firewall moderno de Linux con familia <code>inet</code> (IPv4 + IPv6).
Fail2Ban	Detección y bloqueo automático de intentos de fuerza bruta.

3.2 Estados de puertos y superficie de ataque

DIFERENCIA ESENCIAL: 127.0.0.1 VS 0.0.0.0

Un puerto en **LISTEN** sobre `127.0.0.1` solo es accesible **localmente** (interfaz loopback). Sobre `0.0.0.0` escucha en **todas las interfaces** y por lo tanto **sí constituye superficie de ataque remota**. Regla de hardening: los servicios internos (bases de datos, Redis) deben escuchar solo en `127.0.0.1`.

3.3 Tipos de escaneo TCP

Escaneo	Cómo funciona	¿Requiere privilegios?
TCP SYN <code>-sS</code>	"Half-open": envía SYN, recibe SYN-ACK y NO completa el handshake. Más sigiloso y rápido.	SÍ (root/admin) , porque construye paquetes crudos (raw sockets).
TCP Connect <code>-sT</code>	Completa el handshake de 3 vías usando la llamada <code>connect()</code> del sistema operativo.	NO , cualquier usuario puede ejecutarlo.

Comando de detección de versión + sistema operativo

```
$ nmap -sV -O 192.168.1.10
# -sV → detección de versión de servicios
# -O  → detección de sistema operativo (OS fingerprinting)
# 192.168.1.10 → IP del host objetivo
```

NO CONFUNDAS LA BANDERA

-O (letra O mayúscula) = detección de **sistema operativo**. -f = **fragmentar** paquetes (evasión), NO detecta versiones ni SO. Usar -f en esta pregunta es un error frecuente.

3.4 UDP y sus riesgos

- **¿Por qué open|filtered?** UDP es **sin conexión**: si no llega respuesta, puede significar que el puerto **está abierto** (y el servicio simplemente no respondió) **o** que un **firewall descartó** el paquete. Esa ambigüedad es inherente al protocolo.
- **Riesgo de servicios UDP expuestos (DNS, NTP): ataques de amplificación / reflexión (DDoS)**. El atacante falsifica la IP de la víctima y envía consultas pequeñas que generan respuestas grandes, amplificando el tráfico contra el objetivo.

3.5 Flujo de enumeración moderno (dos etapas)

ORDEN CORRECTO

Etapas: Etapa 1: escáner rápido asíncrono (**RustScan / Naabu**) para **descubrir puertos abiertos**. → Etapa 2: Nmap detallado (-sV , NSE) **solo sobre los puertos abiertos**, para profundidad. Se combina velocidad (etapa 1) con profundidad (etapa 2).

3.6 Reconocimiento pasivo

- **Shodan / Censys**: bases de datos que **ya escanearon todo internet**. Obtienes información del objetivo **sin enviarle tráfico** (recon **pasivo**, sigiloso, no genera registros en el objetivo).
- **Punto ciego de TLS 1.3 / QUIC**: estos protocolos **cifran más del handshake** (TLS 1.3 oculta datos como el SNI con Encrypted Client Hello; QUIC va sobre UDP y cifra metadatos de transporte). Esto **oculta información** (banners, SNI) que el escaneo tradicional usaba para identificar servicios → el escáner ve menos.

§4 Bloque 3 — Acceso Inicial y Escalamiento de Privilegios

4.1 Enumeración local tras obtener acceso inicial

Tras entrar con un usuario sin privilegios, el atacante busca rutas para escalar a root. Qué revela cada comando:

Comando	Qué busca / ruta de escalamiento
<code>sudo -l</code>	Lista qué comandos puede ejecutar el usuario con sudo (como root u otro usuario). Revela rutas directas de escalamiento.
<code>find / -perm -4000 -type f 2>/dev/null</code>	Busca binarios con bit SUID (se ejecutan con los permisos del propietario, normalmente root). Vectores potenciales de escalamiento.
<code>getcap -r / 2>/dev/null</code>	Lista archivos con capabilities de Linux asignadas (p. ej. <code>cap_setuid</code>). Otro vector de escalamiento.

4.2 GTFOBins y grupo docker

- **GTFOBins**: catálogo que documenta **cómo binarios Unix legítimos pueden abusarse** para escalar privilegios o escapar de restricciones (p. ej. un binario con SUID que permite lanzar una shell como root).
- **Pertenecer al grupo `docker` = ruta directa a root**: permite **montar el sistema de archivos del host dentro de un contenedor** y así obtener control total (p. ej. `chroot` al host). No requiere exploit: es abuso de configuración.

4.3 MITRE ATT&CK — técnicas clave (memorizar)

Acción del atacante	Técnica MITRE	ID	Control defensivo
Fuerza bruta contra SSH	Brute Force	T1110	Fail2Ban, MFA, clave pública
Login con credenciales robadas	Valid Accounts	T1078	MFA, detección de anomalías de acceso
Abuso de binario SUID vía GTFOBins	Abuse Elevation Control Mechanism: Setuid and Setgid	T1548.001	Quitar el bit SUID, auditoría de binarios
Enumeración de grupos y permisos locales	Permission Groups Discovery	T1069	Mínimo privilegio, monitoreo de comandos
Robo de sesión con proxy (AiTM)	Adversary-in-the-Middle	T1557	Passkeys/FIDO2 ligados al origen

POR QUÉ T1078 (VALID ACCOUNTS) ES PELIGROSA

Permite el acceso **usando credenciales legítimas**, evadiendo muchas defensas perimetrales: para el sistema, el atacante "es" un usuario válido. No dispara alarmas de intrusión clásicas.

§5 Bloque 4 — Gestión Segura de Identidades y Servicios

5.1 MFA y su límite: ataques AiTM

CONCEPTO CLAVE

Un ataque **adversary-in-the-middle (AiTM)** es peligroso frente a MFA porque **intercepta el token de sesión mediante un proxy inverso**, permitiendo **reutilizar la sesión ya autenticada**. Derrota al MFA porque no ataca el segundo factor: **roba la sesión después** de que el usuario ya se autenticó. **Mitigación:** passkeys / FIDO2 resistentes a phishing (ligan la credencial al origen legítimo).

5.2 Servicios seguros con systemd

- Ejecutar servicios con el **mínimo privilegio**: `DynamicUser=yes`, evitar correr como root.
- *Sandboxing*: `ProtectSystem=strict`, `PrivateTmp=yes`, `NoNewPrivileges=yes`.
- Evaluar el resultado con `systemd-analyze security <servicio>` (puntaje 0–10).

§6 Hoja de comandos esenciales

Objetivo	Comando
Ver sockets/puertos en escucha (moderno)	<code>ss -tlnp</code>
Inicializar base de integridad de AIDE	<code>aide --init</code>
Auditar hardening y obtener índice 0–100	<code>lynis audit system</code>
Evaluar exposición de un servicio (0–10)	<code>systemd-analyze security <servicio></code>
Detección de versión + sistema operativo	<code>nmap -sV -O 192.168.1.10</code>
Escaneo sigiloso half-open (requiere root)	<code>nmap -sS 192.168.1.10</code>
Escaneo TCP connect (sin privilegios)	<code>nmap -sT 192.168.1.10</code>
Descubrir puertos abiertos (rápido)	<code>rustscan -a 192.168.1.10</code>
Ver permisos sudo del usuario actual	<code>sudo -l</code>
Buscar binarios SUID	<code>find / -perm -4000 -type f 2>/dev/null</code>
Listar capabilities de archivos	<code>getcap -r / 2>/dev/null</code>
Deshabilitar login de root en SSH	<code>PermitRootLogin no</code> (en sshd_config)
Deshabilitar contraseñas en SSH	<code>PasswordAuthentication no</code> (en sshd_config)

§7 Autoevaluación

Resuélvela sin ver las respuestas (al final de la sección). Simula el estilo del examen.

1. ¿Qué herramienta reemplazó a `netstat` para inspeccionar sockets en escucha?

- a) ifconfig b) ss c) route d) tcpdump

2. En `systemd`, `DynamicUser=yes` provoca que...

- a) el servicio corra como root b) se cree una cuenta permanente en `/etc/passwd` c) `systemd` asigne un UID/GID efímero que se libera al detener el servicio d) el servicio no pueda escribir en ningún directorio

3. ¿Cuál es la diferencia esencial entre un puerto `LISTEN` en `127.0.0.1` y uno en `0.0.0.0`?

4. Explica qué es AIDE y cuál es el comando para inicializar su base de datos de integridad.

5. Escribe el comando de Nmap que detecta versión de servicios y sistema operativo del host `10.0.0.5`, y justifica cada bandera.

6. Explica la diferencia entre un escaneo TCP SYN (`-ss`) y TCP Connect (`-sT`). ¿Cuál requiere privilegios de administrador y por qué?

7. Un servicio UDP se reporta como open|filtered. ¿Por qué ocurre esa ambigüedad, y qué riesgo tiene un servicio UDP expuesto como DNS o NTP?
8. Relaciona cada acción con su técnica MITRE ATT&CK: (a) fuerza bruta contra SSH, (b) login con credenciales robadas, (c) abuso de binario SUID, (d) enumeración de grupos y permisos.
9. ¿Por qué pertenecer al grupo `docker` se considera una ruta directa a root?
10. Menciona DOS medidas de hardening contra fuerza bruta por SSH y explica por qué la autenticación por clave pública es efectiva contra ese vector.
11. Define brevemente qué protege Credential Guard, BitLocker y ASR en Windows 11.
12. ¿Qué herramienta usarías para verificar de forma cuantitativa que un servicio quedó más seguro tras el hardening, y qué métrica reportarías?

Respuestas

#	Respuesta
1	b) ss.
2	c) systemd asigna un UID/GID efímero que se libera al detener el servicio, sin rastro en /etc/passwd.
3	127.0.0.1 solo es accesible localmente (loopback); 0.0.0.0 escucha en todas las interfaces y sí constituye superficie de ataque remota.
4	AIDE (Advanced Intrusion Detection Environment) es un verificador de integridad de archivos: crea una base de hashes y detecta modificaciones no autorizadas. Comando: <code>aide --init</code> .
5	<code>nmap -sV -O 10.0.0.5</code> . -sV = detección de versión de servicios; -O = detección de sistema operativo; 10.0.0.5 = IP objetivo.
6	SYN (-sS) es "half-open": no completa el handshake y requiere privilegios de root porque construye paquetes crudos. Connect (-sT) completa el handshake de 3 vías con connect() y NO requiere privilegios. El que requiere admin es SYN.
7	Ambigüedad: UDP es sin conexión; la falta de respuesta puede significar puerto abierto (servicio no respondió) o paquete descartado por firewall. Riesgo: ataques de amplificación/reflexión DDoS.
8	(a) T1110 Brute Force · (b) T1078 Valid Accounts · (c) T1548.001 Setuid/Setgid · (d) T1069 Permission Groups Discovery.
9	Permite montar el sistema de archivos del host dentro de un contenedor y obtener control total (p. ej. chroot al host).
10	Dos medidas: deshabilitar login de root + Fail2Ban (o deshabilitar contraseñas). La clave pública es efectiva porque el atacante tendría que adivinar la clave privada, algo computacionalmente inviable; al quitar contraseñas, el vector de fuerza bruta desaparece.
11	Credential Guard: aísla las credenciales (LSA) con virtualización. BitLocker: cifrado de disco completo con AES. ASR: reglas de Defender que bloquean comportamientos riesgosos (macros, ejecutables desde correo).
12	Lynis; métrica = índice de hardening (0–100). También se puede reportar el puntaje de <code>systemd-analyze security</code> (0–10).

§8 Estrategia de examen y errores comunes

8.1 Cómo presentar el examen

- **Administra tu tiempo** según el peso de cada sección (§1). No te atores en una pregunta difícil: márcala y regresa.
- **Contesta absolutamente todo.** En las preguntas abiertas hay puntaje parcial — escribir lo que recuerdes casi siempre suma. Dejar en blanco es cero seguro.
- **Precisión, no extensión.** Una respuesta corta y exacta vale más que un párrafo vago. Nombra la herramienta, el comando o el ID exacto.
- **En comandos, cuida la sintaxis.** Banderas correctas, orden correcto. Un error grave de sintaxis baja el puntaje.

8.2 Errores comunes que debes evitar

ERROR 1 — CONFUNDIR AIDE CON UN AUDITOR DE VULNERABILIDADES

AIDE verifica **integridad de archivos** (detecta cambios). El que audita el hardening y da un índice es **Lynis**. No los mezcles.

ERROR 2 — INVERTIR SYN Y CONNECT

El que **requiere privilegios** es **SYN (-sS)** (paquetes crudos), no Connect. Connect (-sT) es el que **NO** los requiere.

ERROR 3 — MEZCLAR WINDOWS CON LINUX

En un escenario Linux (Ubuntu) **NO** apliquen BitLocker ni contraseñas de BIOS. Las remediaciones deben ser del sistema correcto: usuario sin privilegios, `PermitRootLogin no`, `nftables`, quitar bit SUID, etc.

ERROR 4 — DEJAR LA TABLA MITRE VACÍA O CON IDS INVENTADOS

Memoricen los cuatro IDs base: **T1110, T1078, T1548.001, T1069**. Y llenen SIEMPRE la columna de controles defensivos con controles concretos (Fail2Ban, MFA, quitar SUID), no genéricos.

ERROR 5 — CONFUNDIR LA BANDERA -O CON -F EN NMAP

`-O` = sistema operativo. `-f` = fragmentar paquetes. El comando de detección es `nmap -sV -O`.

CIERRE

Si dominas los cuatro bloques, memorizas la hoja de comandos y evitas los cinco errores de arriba, tienes todo lo necesario para aprobar. Estudia entendiendo el **porqué** de cada control, no solo el nombre — el examen premia justamente eso. ¡Éxito!