🔍 Ejercicio de Análisis: Caso Real de Vulnerabilidad

🎯 Objetivo del ejercicio

Aplicar los principios, metodologías y normas del hacking ético en el análisis crítico de un caso real de vulnerabilidad informática, identificando fallas, riesgos y posibles medidas de mitigación.

📄 Instrucciones Generales

1. Lee detenidamente el caso proporcionado por el docente (puede basarse en un incidente real como el caso de Equifax, SolarWinds, Heartbleed, Log4Shell, etc.).
2. Analiza el incidente usando las metodologías y principios del hacking ético aprendidos.
3. Responde la guía de análisis entregando un informe estructurado.

🧠 Aspectos a analizar

Área de análisis	Preguntas orientadoras
Identificación del caso	¿Cuál fue la vulnerabilidad? ¿A quién afectó? ¿Cuándo ocurrió?
Vector de ataque	¿Cómo fue explotada la vulnerabilidad? ¿Qué técnica se utilizó?
Impacto	¿Qué consecuencias tuvo? (Pérdida de datos, reputación, dinero, etc.)
Fallas detectadas	¿Qué errores de seguridad permitieron el ataque?
Medidas preventivas ausentes	¿Qué controles pudieron evitarlo? ¿Qué buenas prácticas no se aplicaron?

Recomendaciones de mitigación	¿Qué soluciones propondrías como hacker ético?
Ética y legalidad	¿Qué rol habría tenido un hacker ético en este caso? ¿Hubo negligencia?

📑 Formato del informe (sugerido)

- **Título del caso**
- **Resumen del incidente**
- **Descripción técnica de la vulnerabilidad**
- **Evaluación del impacto**
- **Análisis de causas**
- **Recomendaciones de seguridad**
- **Conclusión ética**
- **Referencias (si corresponde)**

📚 CASO 1: Equifax (2017) – Falla en Apache Struts

📌 Descripción

En 2017, la empresa crediticia Equifax sufrió una de las mayores filtraciones de datos de la historia. Más de **147 millones de registros personales** fueron expuestos, incluyendo nombres, fechas de nacimiento, números de seguro social y licencias de conducir.

🛠️ Vulnerabilidad

Una vulnerabilidad en el framework **Apache Struts** (CVE-2017-5638) no fue parcheada a tiempo, lo que permitió a los atacantes ejecutar comandos remotos (RCE – Remote Code Execution).

💥 Consecuencias

- Robo masivo de información personal.

- Multas por más de **$700 millones**.
- Daño irreparable a la reputación de la empresa.

❗ Lecciones

- La **falta de actualización oportuna** es una falla crítica.
- El monitoreo proactivo y las pruebas de penetración podrían haber evitado el desastre.

📚 CASO 2: SolarWinds (2020) – Ataque a la cadena de suministro

📌 Descripción

En diciembre de 2020 se descubrió que hackers habían comprometido el software de monitoreo **Orion** de SolarWinds, afectando a más de **18,000 organizaciones**, incluidas agencias del gobierno de EE.UU.

🛠️ Vulnerabilidad

Los atacantes lograron insertar código malicioso (Sunburst) en actualizaciones legítimas del software, distribuyéndolo a través de los canales oficiales (ataque a la cadena de suministro).

💥 Consecuencias

- Acceso encubierto a redes gubernamentales.
- Robo de información confidencial de agencias como el Tesoro, Defensa y Seguridad Nacional.

❗ Lecciones

- Incluso los proveedores confiables pueden ser vectores de riesgo.
- La **verificación de integridad del software** es esencial.

📚 CASO 3: Heartbleed (2014) – Vulnerabilidad en OpenSSL

📌 Descripción

Heartbleed fue una falla crítica descubierta en la biblioteca de cifrado **OpenSSL**, ampliamente usada en internet. Permitía a los atacantes leer la memoria de los servidores sin autenticación.

🛠️ Vulnerabilidad

CVE-2014-0160: Error en la extensión **Heartbeat** del protocolo TLS que permitía la lectura de hasta 64KB de memoria por petición.

💥 Consecuencias

- Filtración de claves privadas, contraseñas y datos sensibles.
- Millones de servidores web afectados.
- Alto impacto en bancos, servicios en la nube y sitios populares.

❗ Lecciones

- Incluso una pequeña falla en código abierto puede tener impacto **global**.
- La **auditoría constante** y la respuesta rápida son vitales.

📚 CASO 4: Log4Shell (2021) – Fallo crítico en Log4j

📌 Descripción

Una de las vulnerabilidades más severas descubiertas en 2021 afectó a **Log4j**, una popular biblioteca de registro en Java. Afectó a servidores web, juegos, aplicaciones empresariales y más.

🛠️ Vulnerabilidad

CVE-2021-44228: Permitía la **ejecución remota de código (RCE)** al procesar cadenas de texto especialmente diseñadas, como las que llegan por cabeceras HTTP o campos de usuario.

💥 Consecuencias

- Cientos de millones de sistemas expuestos.
- Ataques masivos automatizados alrededor del mundo.
- Intervención urgente de equipos de seguridad globales.

❗ Lecciones

- Las bibliotecas reutilizadas deben auditarse rigurosamente.
- La actualización automática y el monitoreo continuo son prácticas críticas.

📋 Instrumento de Evaluación – Ejercicio Práctico

Análisis de Caso Real de Vulnerabilidad
Puntaje total: 10 puntos
Nota mínima para aprobar: 6 puntos

Criterio Evaluado	Puntaje
Selección clara del caso real y resumen general	1 pt
Descripción técnica precisa de la vulnerabilidad analizada	2 pts
Explicación clara del vector de ataque y su técnica	1 pt
Evaluación del impacto del incidente (datos, reputación, consecuencias)	2 pts

Identificación de fallas de seguridad y medidas preventivas ausentes	2 pts
Propuesta de recomendaciones de mitigación y reflexión ética	2 pts

---