28-2-2025

Alumna: Andrea Suñer Expósito

PROFESORES: jORGE gaRCÍA Y RAÚL mONCADA

Proyecto fin de Master Ciberseguridad

Máquina debian hackeada

# Introducción

En un entorno empresarial ya sea privado o público, los servidores representan la “columna vertebral” de las operaciones tecnológicas de la empresa. Garantizar la integridad, disponibilidad y confidencialidad es esencial para proteger la información sensible. Este proyecto final de 4Geeks Academy pone a prueba las competencias fundamentales de ciberseguridad en un escenario práctico y realista: la restauración y protección de un servidor comprometido.

El proyecto se divide en tres fases clave, cada una diseñada para abordar aspectos críticos de la ciberseguridad: análisis forense, detección y remediación de vulnerabilidades, y planificación de la respuesta a incidentes. A través de estas etapas, buscamos desarrollar habilidades técnicas avanzadas, una mentalidad analítica y un enfoque proactivo frente a los desafíos de seguridad que se pueden ver en un día a día trabajando de ello.

**Fase 1 - Corrección de un Hackeo**  
Realizaremos un análisis forense detallado del incidente para identificar las vulnerabilidades explotadas por los atacantes. Posteriormente, se implementarán las medidas necesarias para bloquear el exploit y evitar una mayor escalada del ataque. Esta etapa sienta las bases para una respuesta efectiva ante incidentes mediante la recopilación de evidencias y la solución inmediata.

**Fase 2 - Detección y Corrección de una Nueva Vulnerabilidad**  
Se pondrá a prueba las capacidades de detección proactiva al escanear el sistema en busca de una vulnerabilidad adicional. Una vez identificada, se explotará de manera controlada para evaluar su impacto y documentar el proceso. Corrigiendo la vulnerabilidad y presentando un informe que detalle el análisis realizado, las acciones tomadas y los resultados obtenidos.

**Fase 3 - Plan de Respuesta a Incidentes y Certificación**  
Aquí abordaremos la creación de un plan integral de respuesta a incidentes basado en las mejores prácticas, como la guía NIST. Incluyendo un diseño de un Sistema de Gestión de Seguridad de la Información (SGSI) en concordancia a la norma ISO 27001, que contemple políticas de prevención de pérdida de datos (DLP) para evitar fugas de información y fortalecer la postura de seguridad organizacional.

Este proyecto no solo es un ejercicio técnico, sino también una oportunidad para demostrar la capacidad de gestionar incidentes de seguridad de manera profesional, fortalecer sistemas críticos y garantizar su funcionamiento óptimo en un entorno empresarial. Al completar estas tres fases, se consolidarán las habilidades necesarias para abordar amenazas complejas en el ámbito de la ciberseguridad y garantizar la resiliencia tecnológica de cualquier empresa.

# Reconocimiento y Recolección de Evidencias

## ÁNALISIS FORENSE

Primeramente, desconectaremos el servidor de la red para evitar la propagación de amenazas. Analizaremos los servicios comprometidos, recopilando evidencias y evaluando el alcance del daño causado. Se utilizó Nmap para realizar un escaneo de puertos y determinar los servicios que se ejecutan en la máquina y nslookup para obtener información sobre los registros DNS, dominios, subdominios y propietarios del sitio web. Ayudando a mapear la infraestructura y descubrir posibles activos expuestos.

## Identificación y Documentación de Vulnerabilidades

### Configuración de MySQL

**Hallazgo:** Un usuario de base de datos con una contraseña débil fue identificado durante la auditoría.

**Pruebas realizadas:** Se realizó una consulta directa en MySQL para listar los usuarios:

SELECT user, host, authentication\_string FROM mysql.user;

**Impacto:** El atacante podría acceder a la base de datos, modificar o extraer información sensible.

**Solución aplicada:**

1. Contraseñas actualizadas con claves complejas:

ALTER USER 'usuario'@'localhost' IDENTIFIED BY 'ContraseñaSegura123!';

1. Activación de un plugin de validación de contraseñas:

INSTALL PLUGIN validate\_password SONAME 'validate\_password.so';

Cambiando las contraseñas del usuario con credenciales débiles y deshabilitamos conexiones remotas innecesarias editando bind-address en el archivo de configuración.

### Configuración del Servidor FTP

**Hallazgo:** El servidor FTP permitía acceso anónimo y tenía configuraciones inseguras.

**Pruebas realizadas:** Se intentó acceder con credenciales anónimas:

ftp <IP\_DEL\_SERVIDOR>

Usuario: anonymous

Contraseña: (vacío)

**Impacto:** El acceso anónimo podría exponer archivos críticos a atacantes.

**Solución aplicada:**

1. Deshabilitación de acceso anónimo en vsftpd.conf:

anonymous\_enable=NO

1. Reinicio del servicio:

sudo systemctl restart vsftpd

1. Configuración de permisos restrictivos en directorios FTP:

chmod 750 /ruta/directorio

Revisamos la configuración, deshabilitando el acceso anónimo. Ajustando los permisos de los directorios expuestos.

### Acceso SSH Inseguro

**Hallazgo:** El servicio SSH permitía autenticación con contraseñas débiles y acceso de root.

**Pruebas realizadas:** Se intentó un ataque de fuerza bruta.

**Impacto:** Un atacante podría obtener acceso privilegiado al sistema.

**Solución aplicada:**

1. Deshabilitación del acceso de root en /etc/ssh/sshd\_config:

PermitRootLogin no

1. Habilitación de autenticación basada en claves.
2. Reinicio del servicio SSH:

sudo systemctl restart sshd

Deshabilitamos la autenticación con contraseña y configuramos claves SSH. Modifiqué el archivo /etc/ssh/sshd\_config para permitir solo autenticación mediante clave pública.

### Puertos Abiertos Innecesarios

**Hallazgo:** El escaneo con Nmap reveló puertos abiertos no esenciales (por ejemplo, puerto 21, 22, 80, 8080).

**Pruebas realizadas:** Comando ejecutado:

nmap -A -p- <IP\_DEL\_SERVIDOR>

**Impacto:** Los servicios en puertos innecesarios incrementan la superficie de ataque.

**Solución aplicada:**

1. Cierre de puertos innecesarios con ufw:

sudo ufw deny 8080

1. Verificación de puertos activos:

sudo ufw status

### Permisos en wp-config.php

**Hallazgo:** El archivo wp-config.php tenía permisos demasiado permisivos (644).

**Impacto:** Cualquier usuario en el sistema podría leer credenciales de la base de datos.

**Solución aplicada:**

1. Ajuste de permisos:

chmod 600 /var/www/html/wp-config.php

1. Cambio de propietario:

chown www-data:www-data /var/www/html/wp-config.php

Cambiamos los permisos del archivo para restringir el acceso a usuarios no autorizados.

Validación de la Máquina Debian

Se realizaron pruebas exhaustivas para validar que las soluciones aplicadas reforzaron la seguridad del servidor:

* Acceso a MySQL restringido correctamente.
* Conexiones FTP verificadas con configuraciones seguras.
* Puertos innecesarios cerrados y servicios no esenciales deshabilitados.
* Permisos de archivos sensibles ajustados correctamente.

Actualizamos los paquetes del sistema y el kernel con apt update y apt upgrade.

### Escaneo del Sistema con Nmap

**Herramienta:**  
Utilizaré **Nmap** para identificar puertos abiertos y servicios en ejecución en el sistema comprometido.

**Comando de Escaneo:**

nmap -A -p- <IP\_DEL\_SERVIDOR>

* -A: Detecta el sistema operativo y servicios en ejecución.
* -p-: Escanea todos los puertos disponibles.

| **Puerto** | **Servicio** | **Versión/Observación** |
| --- | --- | --- |
| 22 | SSH | OpenSSH 7.4 |
| 80 | HTTP | Apache 2.4.18 |
| 443 | HTTPS | Certificado autofirmado |
| 3306 | MySQL | Contraseña débil detectada |
| 8080 | HTTP | Jenkins sin autenticación |

Identificación de Vulnerabilidad

**Vulnerabilidad Detectada:**

* Apache 2.4.18 tiene habilitados módulos inseguros, que permite a los atacantes listar directorios y archivos confidenciales.

**Prueba de Concepto:**

1. Acceso al servidor web:

http://<IP\_DEL\_SERVIDOR>/directorio/

Se observa una lista de archivos que incluye información sensible como:

* + **backup.sql**: Copia de seguridad con datos de la base de datos.
  + **config.json**: Contiene claves de API.

1. Descarga del archivo:

wget http://<IP\_DEL\_SERVIDOR>/directorio/backup.sql

**Impacto:**  
Esto permite a un atacante acceder a datos críticos sin autenticación, comprometiendo la confidencialidad y la integridad.

Proceso de Explotación

**Pasos Realizados:**

1. **Enumeración del contenido del directorio:**
   * Acceso al directorio listable para identificar archivos potencialmente sensibles.
   * Descarga del archivo backup.sql.
2. **Acceso a datos sensibles:**

Se inspeccionó backup.sql y se descubrieron credenciales en texto claro:

usuario\_admin: admin123

1. **Escalamiento de privilegios:**
   * Estas credenciales se usaron para acceder al servicio MySQL y ejecutar consultas maliciosas.

**Comando para conexión a MySQL:**

mysql -u usuario\_admin -padmin123

**Resultados:**

* Acceso total a la base de datos.
* Posibilidad de modificar, eliminar o extraer datos.

Corrección de la Vulnerabilidad

Medidas Aplicadas:

Deshabilitar la indexación de directorios:

* + Modificar la configuración de Apache para deshabilitar la lista de directorios.
  + Archivo de configuración:

sudo nano /etc/apache2/apache2.conf

* + Configuración añadida:

<Directory /var/www/html/>

Options -Indexes

</Directory>

* + Reiniciar Apache:

sudo systemctl restart apache2

1. **Proteger archivos sensibles:**
   * Mover archivos como backup.sql y config.json fuera del directorio accesible públicamente.
   * Ajustar permisos de acceso:

chmod 600 /ruta/backup.sql

chown root:root /ruta/backup.sql

1. **Implementar controles adicionales:**
   * Configurar reglas de firewall para restringir el acceso a puertos no necesarios:

sudo ufw deny 8080

sudo ufw reload

* + Forzar HTTPS con certificados válidos.

### Informe Detallado

**Vulnerabilidad Detectada:**

* **Servicio Afectado:** Apache 2.4.18
* **Causa:** Configuración insegura que permite la indexación de directorios.

**Impacto:** Acceso no autorizado a archivos sensibles que contienen datos confidenciales. Los atacantes podrían listar y descargar estos.

**Hallazgo:** Los directorios web estaban configurados para permitir indexación.

**Proceso de Explotación:**

1. Escaneo del sistema con Nmap reveló un servidor Apache en el puerto 80.
2. La exploración manual mostró directorios listables y archivos expuestos.
3. El archivo backup.sql fue descargado y analizado, exponiendo credenciales en texto claro.
4. Las credenciales permitieron acceso a la base de datos MySQL.

**Medidas Aplicadas:**

1. Deshabilitar la lista de directorios en Apache mediante la configuración Options -Indexes.
2. Proteger archivos sensibles moviéndolos fuera del alcance público y ajustando sus permisos.
3. Cerrar puertos innecesarios para reducir la superficie de ataque.
4. Configurar HTTPS con un certificado válido.

Recomendaciones para Prevenir Futuros Ataques

1. **Fortalecimiento de credenciales:**

Establecer políticas de contraseñas complicadas (Añadiendo mayúsculas, números y minúsculas) y obligar el uso de claves SSH para el acceso remoto.

Configurar autenticación multifactor (MFA).

1. **Monitoreo continuo y alertas:**

Configurar herramientas de monitoreo como Wazuh para detectar actividades sospechosas en tiempo real.

Configurar alertas para eventos críticos.

1. **Configuración segura:**

Revisar y aplicar configuraciones de seguridad estándar en servicios críticos como MySQL, FTP y Apache.

Mantener los permisos mínimos necesarios en archivos sensibles como wp-config.php.

1. **Auditorías periódicas:**

Realizar auditorías regulares del sistema con herramientas de escaneo de vulnerabilidades como OpenVAS.

1. **Educación y capacitación:**

Proveer capacitación al personal para reconocer intentos de phishing y otras técnicas comunes de ataque. Implementando un programa de concienciación sobre seguridad.

1. **Pruebas de penetración**:

Realizar simulaciones de ataques para validar la seguridad del sistema.

## Plan de Respuesta a Incidentes (PRI) Basado en NIST

El objetivo principal de este PRI es minimizar el impacto de los incidentes de seguridad en la confidencialidad, integridad y disponibilidad de los activos de la organización. El plan es aplicable a todos los sistemas de información, redes, aplicaciones y datos gestionados por la organización.

El NIST propone un enfoque en cuatro fases:

* 1. **Preparación.**
  2. **Detección y Análisis.**
  3. **Contención/Erradicación/Recuperación.**
  4. **Actividades Posteriores**.

**1. Preparación**

* **Definir roles y responsabilidades**:
  + Establecer un Equipo de Respuesta a Incidentes (IRT).
  + Asignar tareas específicas para la gestión de incidentes.
* **Documentar procedimientos**:
  + Crear manuales para enfrentar diferentes tipos de incidentes.
  + Mantener un registro de contactos clave (internos y externos).
* **Proveer herramientas y recursos**:
  + Implementar sistemas de monitoreo y herramientas SIEM (Security Information and Event Management).
  + Configurar firewalls, IDS/IPS y soluciones antivirus.

**2. Detección y Análisis**

* **Establecer métodos de detección**:
  + Usar soluciones de monitoreo para detectar anomalías en el tráfico o el comportamiento de los sistemas.
  + Configurar alertas en caso de cambios inesperados en la configuración.
* **Clasificar incidentes**:
  + Definir niveles de severidad (bajo, medio, crítico).
  + Registrar cada incidente con detalles como fecha, hora, sistemas afectados y descripción.

**3. Contención, Erradicación y Recuperación**

* **Contención**:
  + Aislar sistemas comprometidos para evitar propagación.
  + Bloquear direcciones IP sospechosas y desconectar sesiones activas.
* **Erradicación**:
  + Remover malware, puertas traseras o configuraciones inseguras.
  + Validar la limpieza mediante análisis forense.
* **Recuperación**:
  + Restaurar sistemas a su estado operativo.
  + Verificar la integridad de los datos.

**4. Actividades Posteriores**

* **Lecciones aprendidas**:
  + Reunir al equipo para analizar el incidente y su gestión.
  + Actualizar el PRI basado en hallazgos.
* **Reporte del incidente**:
  + Elaborar informes detallados para compartir con las partes interesadas.

## Sistema de Gestión de Seguridad de la Información (SGSI) conforme a ISO 27001

ISO 27001 se basa en un ciclo de mejora continua (PDCA: Plan-Do-Check-Act) para gestionar la seguridad de la información.

* + 1. **Planificación (Plan)**
* **Definir el alcance del SGSI**:
  + Incluir activos críticos, servicios, procesos y personal.
* **Evaluación de riesgos**:
  + Identificar amenazas y vulnerabilidades.
  + Evaluar el impacto y la probabilidad de cada riesgo.
* **Definir controles de seguridad**:
  + Implementar los controles del Anexo A de ISO 27001, como control de accesos, seguridad de la red y cifrado.
    1. **Implementación (Do)**
* **Establecer políticas y procedimientos**:
  + Crear políticas de seguridad de la información y de control de acceso.
  + Capacitar al personal en mejores prácticas de seguridad.
* **Implementar medidas técnicas**:
  + Cifrado de datos en tránsito y en reposo.
  + Protección contra malware y parches regulares.
* **Adoptar políticas DLP**:
  + Restringir transferencias de datos sensibles fuera de la red.
  + Implementar etiquetado de datos y monitoreo de acceso.

**2.3. Monitoreo y Evaluación (Check)**

* **Auditorías internas**:
  + Realizar auditorías regulares para verificar el cumplimiento.
* **Seguimiento de indicadores clave**:
  + Usar métricas como el tiempo promedio de respuesta a incidentes o la cantidad de accesos no autorizados detectados.

**2.4. Mejora Continua (Act)**

* **Revisar y actualizar el SGSI**:
  + Realizar revisiones anuales o trimestrales.
* **Adaptar controles**:
  + Incorporar nuevas tecnologías o procesos.

### Políticas de Prevención de Pérdida de Datos (DLP)

Estas políticas deben abordar los riesgos de fuga de datos mediante controles específicos:

**1. Identificación de Datos Críticos**

* **Clasificación de datos**:

Identificar y etiquetar datos sensibles (confidenciales, personales, regulatorios).

* **Mapeo de flujos de datos**:

Monitorear cómo y dónde se almacenan, procesan y transmiten los datos.

**2. Implementación de Controles Técnicos**

* **Protección de puntos finales**:

Bloquear dispositivos no autorizados (USB, discos externos).

* **Cifrado de datos**:

Exigir cifrado en archivos enviados fuera de la organización.

* **Restricción de transferencias**:

Limitar el uso de plataformas de almacenamiento en la nube no aprobadas.

**3. Supervisión y Auditoría**

* **Soluciones DLP**:

Configurar herramientas DLP para monitorear la transmisión de datos sensibles.

* **Registros y alertas**:

Generar alertas en tiempo real ante intentos de fuga de información.

**4. Capacitación y Concienciación**

* **Formación al personal**:

Sensibilizar a los empleados sobre las políticas de manejo de datos sensibles.

* **Simulaciones de fuga de datos**:

Realizar ejercicios para reforzar la preparación.

## Conclusión

Este proyecto ha permitido aplicar un enfoque integral total para responder a incidentes de seguridad en un entorno simulado pero representativo de situaciones reales en el ámbito de la ciberseguridad. A lo largo de todo el proyecto, se abordaron tareas críticas que fortalecieron las competencias técnicas y estratégicas necesarias para proteger sistemas clave en una empresa.

**Fase 1: Reconocimiento y Recolección de Evidencias**

En esta primera etapa, se realizó un análisis forense completo para identificar las vulnerabilidades explotadas, como configuraciones inseguras en SSH, FTP, y MySQL, junto con exposiciones en archivos críticos y servicios mal configurados. Las medidas tomadas incluyeron la detención de servicios comprometidos, la eliminación de backdoors, la corrección de permisos en archivos sensibles, y la implementación de configuraciones seguras. Este proceso permitió bloquear el exploit, mitigar el impacto del ataque y prevenir una posible escalación.

**Fase 2: Detección y Corrección de una Nueva Vulnerabilidad**

La segunda fase fue clave para afianzar las capacidades de detección proactiva. Se identificó y explotó de manera controlada una vulnerabilidad adicional, lo que ayudó a entender su impacto potencial. Posteriormente, se corrigieron estas debilidades mediante actualizaciones, configuraciones de seguridad más estrictas, y el cierre de servicios y puertos no esenciales. Esta experiencia permitió desarrollar un enfoque metódico y documentado para abordar y mitigar vulnerabilidades en el entorno del servidor.

**Fase 3: Plan de Respuesta a Incidentes y Certificación**

En la fase final, se diseñó un plan de respuesta a incidentes basado en estándares reconocidos como el NIST, complementado por la creación de un Sistema de Gestión de Seguridad de la Información (SGSI) alineado con la norma ISO 27001. También se implementaron políticas de Prevención de Pérdida de Datos (DLP) para proteger la información crítica de la organización y evitar fugas. Este plan refuerza la resiliencia organizacional al establecer un marco robusto para manejar incidentes futuros y garantizar la continuidad del negocio.

Resultados Clave

1. **Fortalecimiento del sistema:** Se mejoraron las configuraciones de seguridad, reduciendo la superficie de ataque y fortaleciendo las defensas contra amenazas internas y externas.
2. **Habilidades desarrolladas:** Se consolidaron competencias en análisis forense, detección de vulnerabilidades, manejo de incidentes, y creación de estrategias de ciberseguridad proactivas.
3. **Capacitación estratégica:** La creación de un SGSI y un plan de respuesta a incidentes proporciona un marco práctico y profesional para gestionar incidentes de seguridad en entornos reales.

Reflexión Final

Este proyecto demuestra la importancia de adoptar un enfoque estructurado para responder a incidentes y prevenir futuras amenazas. Desde la identificación y mitigación de vulnerabilidades hasta la planificación estratégica, se abarcó un ciclo completo de gestión de seguridad que refuerza la protección de los activos críticos.

El aprendizaje en este ejercicio sienta una base sólida para enfrentar desafíos reales en el ámbito de la ciberseguridad, permitiendo aplicar conocimientos técnicos y estratégicos para proteger entornos digitales.Final del formulario