# Informe de Pentest y Evaluación de Seguridad

**4Geeks Academy**

## 1. Introducción

Este documento detalla el proceso integral de evaluación de seguridad realizado sobre el servidor crítico de 4Geeks Academy. Se realizó un análisis forense y de vulnerabilidades con el objetivo de identificar puntos débiles, corregirlos y establecer medidas preventivas para futuros incidentes. La auditoría se estructuró en tres fases:

* **Fase 1:** Análisis forense y mitigación del hackeo.
* **Fase 2:** Detección, explotación controlada y corrección de una vulnerabilidad adicional.
* **Fase 3:** Diseño de un plan de respuesta a incidentes y establecimiento de un Sistema de Gestión de Seguridad de la Información (SGSI) conforme a la norma ISO 27001.

## 2. Alcance y Objetivos

**Alcance:**

* Evaluación de servicios críticos: MySQL, FTP, SSH y servidor web.
* Revisión de configuraciones y permisos en archivos sensibles (por ejemplo, wp-config.php).

**Objetivos:**

* Realizar un análisis forense para identificar vectores de ataque y revertir cambios maliciosos.
* Detectar y explotar de forma controlada vulnerabilidades adicionales.
* Implementar soluciones correctivas y recomendaciones preventivas.
* Desarrollar un plan de respuesta a incidentes y establecer un SGSI.

## 3. Metodología y Herramientas Utilizadas

### Fase 1: Análisis Forense y Mitigación del Hackeo

**Actividades Realizadas:**

**Recolección y Análisis de Logs:**

* + Herramienta: grep
  + Se examinaron logs críticos (por ejemplo, /var/log/auth.log) para identificar accesos SSH sospechosos y patrones anómalos.

**Identificación de Archivos y Procesos Sospechosos:**

* + Herramientas: ps, ls, find
  + Se inspeccionaron procesos en ejecución y se listaron archivos con modificaciones inusuales, detectando posibles backdoors y usuarios fraudulentos.

**Detección de Malware y Rootkits:**

* + Herramienta: chkrootkit
  + Se realizó un escaneo del sistema para detectar la presencia de malware o rootkits.

**Contención y Reversión de Acciones Maliciosas:**

* + Herramienta: systemctl
  + Se detuvieron temporalmente los servicios comprometidos, eliminando usuarios no autorizados, cerrando puertos innecesarios y removiendo archivos maliciosos.

**Actualización y Fortalecimiento de Configuraciones:**

* + Herramientas: Gestores de paquetes (apt/yum), edición manual de archivos de configuración, y comandos como chmod y chown para ajustar permisos.

### Fase 2: Detección y Corrección de Vulnerabilidad Adicional

**Actividades Realizadas:**

**Escaneo del Sistema y Detección de Vulnerabilidades:**

* + Herramienta: Nmap
  + Se efectuó un escaneo completo para identificar puertos abiertos y servicios expuestos no esenciales.

**Explotación Controlada de la Vulnerabilidad:**

* + Se identificó una falla en la configuración del servidor FTP y la exposición de directorios web listables.
  + Herramienta Adicional: Scripts de explotación controlada en ambiente de pruebas para evaluar la magnitud del riesgo.

**Corrección y Mitigación:**

* + Se reconfiguró el servidor FTP para deshabilitar accesos anónimos y se ajustaron los permisos de archivos críticos (como wp-config.php).
  + Se cerraron puertos innecesarios y se reforzó la seguridad de SSH, implementando autenticación basada en claves.

### Fase 3: Plan de Respuesta a Incidentes y Certificación SGSI

**Actividades Realizadas:**

**Desarrollo del Plan de Respuesta a Incidentes:**

* + Se diseñó un plan basado en la guía **NIST SP 800-61**, abarcando las fases de identificación, contención, erradicación y recuperación.

**Implementación del SGSI:**

* + Se estableció un SGSI conforme a **ISO 27001**, que incluyó:
  + Análisis de riesgos.
  + Definición de políticas de seguridad.
  + Configuración de respaldos periódicos y cifrado de datos sensibles.
  + Controles de acceso y auditorías regulares.

**Implementación de Políticas DLP:**

* + Se adoptaron medidas y controles para prevenir la fuga de datos sensibles.

## 4. Detalle de Fallos Detectados y Acciones Correctivas

### 4.1 MySQL

* **Fallo:**
  + Existencia de un usuario con contraseña débil que permitía accesos no autorizados.
* **Acciones Correctivas:**
  + Actualización de la contraseña utilizando algoritmos de hash seguros.
  + Restricción de privilegios y configuración de la base de datos para aceptar conexiones únicamente de IP autorizadas.

### 4.2 FTP

* **Fallo:**
  + Configuración errónea que permitía conexiones inseguras y acceso a archivos sensibles.
* **Acciones Correctivas:**
  + Reconfiguración para deshabilitar accesos anónimos.
  + Refuerzo de reglas en el firewall para limitar el acceso desde rangos IP confiables.
  + Implementación de cifrado en la transferencia de datos.

### 4.3 SSH

* **Fallo:**
  + Autenticación mediante contraseñas débiles, aumentando el riesgo de ataques de fuerza bruta.
* **Acciones Correctivas:**
  + Implementación de autenticación mediante claves públicas y desactivación de la autenticación por contraseña.
  + Ajustes en el archivo sshd\_config para limitar intentos fallidos y mejorar el registro de accesos.

### 4.4 Puertos Abiertos Innecesarios

* **Fallo:**
  + Puertos abiertos no esenciales que amplían la superficie de ataque.
* **Acciones Correctivas:**
  + Mapeo completo del sistema con **Nmap** y cierre de puertos innecesarios mediante ajustes en el firewall.

### 4.5 Permisos en wp-config.php

* **Fallo:**
  + Permisos excesivamente permisivos en el archivo wp-config.php, exponiendo credenciales críticas.
* **Acciones Correctivas:**
  + Restricción de permisos mediante chmod 600 wp-config.php.
  + Ajuste de la propiedad con chown para que solo el usuario autorizado pudiera acceder.

### 4.6 Directorio Web Listable

* **Fallo:**
  + Configuración que permitía la indexación y listado de directorios, facilitando el acceso a archivos sensibles.
* **Acciones Correctivas:**
  + Modificación en la configuración del servidor web (Apache/Nginx) para deshabilitar el listado de directorios.
  + Implementación de archivos index predeterminados en directorios críticos.

## 5. Herramientas Utilizadas

Se utilizó un conjunto amplio de herramientas para abordar tanto el análisis forense como la detección y corrección de vulnerabilidades:

* **grep:** Análisis de logs para detectar patrones y accesos sospechosos.
* **ps, ls, find:** Identificación de procesos y archivos inusuales.
* **chkrootkit:** Escaneo en busca de malware y rootkits.
* **Nmap:** Escaneo de puertos y detección de servicios expuestos.
* **systemctl:** Gestión y detención de servicios comprometidos.
* **Gestores de paquetes (apt/yum):** Actualización del sistema y aplicación de parches.
* **chmod, chown:** Ajuste de permisos y propiedades de archivos críticos.
* **Scripts de explotación controlada:** Evaluación de vulnerabilidades en servicios como FTP y web.
* **Netstat:** Monitoreo de conexiones activas para detectar accesos no autorizados.
* **Lynis:** Auditoría de seguridad del sistema Linux, detectando vulnerabilidades y configuraciones inseguras.
* **Nikto:** Escaneo del servidor web para identificar vulnerabilidades y configuraciones erróneas.
* **Wireshark:** Análisis del tráfico de red para identificar actividades sospechosas.
* **Fail2ban:** Monitorización de logs y bloqueo automático de IPs sospechosas ante intentos de fuerza bruta.
* **OpenVAS:** Escáner de vulnerabilidades para una auditoría completa del sistema.

## 6. Recomendaciones y Próximos Pasos

Para garantizar la seguridad del servidor y prevenir futuros incidentes, se recomienda lo siguiente:

**Fortalecimiento de Credenciales:**

* + Actualizar todas las contraseñas, aplicando políticas de complejidad y caducidad.

**Revisión y Reconfiguración de Servicios:**

* + Asegurar que servicios como FTP y SSH se configuren siguiendo las mejores prácticas, empleando cifrado y autenticación robusta.

**Cierre y Monitoreo de Puertos:**

* + Realizar auditorías periódicas para identificar y cerrar puertos no esenciales.

**Control de Acceso a Archivos Sensibles:**

* + Revisar y ajustar permisos en archivos críticos (por ejemplo, wp-config.php) para limitar accesos no autorizados.

**Implementación de Monitoreo Continuo:**

* + Utilizar herramientas (como Fail2ban, Wireshark y soluciones de auditoría) para detectar actividades anómalas y responder de forma inmediata.

**Actualización de Planes y Capacitación:**

* + Revisar y actualizar el plan de respuesta a incidentes (basado en NIST SP 800-61) y continuar la implementación y auditoría del SGSI conforme a ISO 27001.

**Simulacros y Evaluaciones Periódicas:**

* + Realizar pruebas de penetración regulares y simulacros de respuesta para validar la eficacia de las medidas implementadas.

## 7. Conclusiones

El análisis detallado permitió identificar múltiples fallos de configuración y vulnerabilidades que facilitaron el compromiso inicial del sistema. La aplicación inmediata de medidas correctivas, junto con la utilización de un amplio conjunto de herramientas (incluyendo Lynis, Nikto, Wireshark, Fail2ban y OpenVAS), ha permitido mitigar eficazmente el riesgo de explotación y fortalecer la seguridad del servidor. Además, la implementación de un plan de respuesta a incidentes y el establecimiento de un SGSI proporcionan una base sólida para prevenir y responder ante futuros ataques.