# Análisis vulnerabilidades en aplicaciones web

## Contenido

Índice de Hardening ..... 2
Análisis en una aplicación web ..... 5
Descripción del software ..... 5
Desarrollo ..... 5

# Índice de Hardening

El término "índice de hardening" generalmente se refiere a una medida que evalúa el nivel de seguridad de un sistema o aplicación. El "hardening" se refiere al proceso de fortalecer la seguridad de un sistema mediante la aplicación de medidas y configuraciones de seguridad. Un índice de hardening podría ser una métrica que cuantifica la efectividad de estas medidas de seguridad.

Voy a calcular el índice de hardening e intentare subirlo hasta un 70 mínimo.

Usare Lynis, esta herramienta nos va a permitir realizar auditorías bastante completas sobre un sistema operativo Linux. Esta herramienta, como se explicará a continuación ha sido probada en dos distribuciones Linux: Arch y Fedora. Lynis escanea el sistema buscando el software y determinando el cumplimiento con los estándares y las buenas prácticas en seguridad de la información, también detecta errores de configuración. Cuando finaliza el escaneado muestra un informe con la información correspondiente.

Esta tarea ha sido realizada en dos sistemas operativos distintos: Manjaro y Fedora. En ambos, originalmente el índice de hardening se acercaba al 70, aun así realizo diferentes pruebas para intentar mejorarlo.

Hay que comentar que una de las fases más difíciles de esta tarea ha sido averiguar qué modificaciones en el sistema podrían mejorar la seguridad de este, ya que, a pesar de que la herramienta Lynis sugiere gran número de modificaciones no todas ellas aumentan el hardening. En el archivo adjunto al proyecto se muestran capturas de las modificaciones realizadas en Manjaro, pero no se va a entrar en detalles puesto que muchas de ellas no mejoraron el hardening, viendo que los resultado obtenidos en este sistema no eran favorables se decidió trabajar sobre una máquina virtual con sistema Fedora.

Los pasos son los siguientes, las imágenes de las pruebas están adjuntadas y nombradas según el apartado:

***Instalación Lynis:***

sudo dnf install lynis

***Ejecución (resultado hardening):***

Usamos el comando:

sudo lynis audit system

La ejecución viene acompañada de una lista de sugerencias detalladas en el archivo adjunto. Siguiendo estas sugerencias hacemos los siguientes pasos.

***Modificación archivo etc/login.defs***

Ejecutemos el comando:

sudo nano /etc/login.defs

y realizo las siguientes modificaciones.

***Encryption method***

***Pass***

PASS_MIN_DAYS: indica el número mínimo de días que el usuario debe mantener la contraseña antes de cambiarla.

PASS_MAX_DAYS: cuánto tarda en expirar la contraseña.

***Umask:***

Se trata de la configuración que controla los permisos predeterminados asignados a nuevos archivos y directorios creados por un usuario (user mask). Cuando un usuario crea un nuevo archivo o directorio, el sistema utiliza la configuración umask para determinar los permisos iniciales. El valor es un nº octal que se resta de los permisos máximos para calcular los permisos reales. Cambiando este valor de 22 a 027 he conseguido desactivar permisos específicos al crear nuevos archivos o directorios.

Con estos pasos, ha sido suficiente para aumentar el índice de hardening de nuestro sistema, que ya supera los 70 exigidos por la empresa.

**Browser seguro:**

Además de comprobar el nivel de seguridad de los dispositivos debería de haberse configurado el browser de forma segura para la comunicación. Los pasos llevados a cabo quedan capturados en el archivo adjunto. Aquí se explica brevemente cuáles son los pasos que se han seguido.

Lo primero que se sugiere es la actualización del sistema desde el que se va a trabajar, a partir de entonces entramos en la configuración de seguridad del navegador.

Una vez dentro de la privacidad y personalización de nuestro navegador empiezo eliminando datos de actividad de forma automática estableciendo un periodo de eliminación. Hacemos lo mismo con el historial de YouTube.

Seguido de esto se realiza una revisión de seguridad activando navegación segura, comprobando los dispositivos desde los que se ha iniciado sesión, los acceso de terceros, las contraseñas guardadas…

Con estos simples pasos se ha conseguido configurar el browser de forma segura.

# Análisis en una aplicación web

Voy a simular una serie de ataques muy recurrentes según el OWASP Top 10 en un entorno simulado como Vulnweb y Mutillidae con la herramienta OWASP ZAP.

## Descripción del software

OWASP ZAP (Zed Attack Proxy) es una herramienta de prueba de seguridad de código abierto diseñada para encontrar vulnerabilidades en aplicaciones web durante el desarrollo y las pruebas. Desarrollado por la Open Web Application Security Project (OWASP), ZAP proporciona una plataforma integral para identificar posibles riesgos de seguridad en aplicaciones web y servicios web.

Características principales de OWASP ZAP: Escaneo Activo y Pasivo, spider y Scanners Automáticos, interceptación de Proxy, informes y Resultados, automatización y soporte para Scripts

## Desarrollo

Usando Manjaro me descargue la herramienta software OWASP ZAP por docker siguiendo estos pasos:

sudo systemctl start docker

sudo systemctl enable docker

sudo usermod -aG docker $USER

docker pull citizenstig/nowasp

docker run -d -p 80:80 citizenstig/nowasp

Una vez descargada comienzo con la configuración del proxy, una vez que conocemos el puerto del proxy (8080) trataremos de “engañar” al navegador, en lugar de realizar peticiones directamente a internet lo hará a través de nuestro proxy.

El siguiente paso es arrancar el navegador ya preparado con el proxy funcionando(manual explore). La aplicación captura peticiones.

Primero buscamos una página que tuviera un formulario html probamos en esta página https://www.htmlquick.com/es/tutorials/forms.html, luego añadiendo un breakpoint en el zap con el browser de prueba de chrome que proporciona la aplicación cambiamos los datos del formulario de Julio a Luis, adjuntamos las imágenes en prueba.

Al ver que la captura ha funcionado pruebo a hacer una captura http de una contraseña:

Accedo a la página del login, OWASP sigue capturando tráfico lo que nos permite intentar cambiar la password, para ello seleccionamos la parte del código correspondiente a esta propiedad y clicamos en FUZZ, seleccionamos sql injection para cargar todas las cadenas que intentan bypasear la construcción de la consulta, he hecho captura de estos pasos.

**Prueba XSS:**

En el formulario en lugar de ingresar una IP introducimos un código Javascript:

<script> alert('Prueba1'); </script>

Este código hace que el sistema web envíe el código, la cadena la ejecuta el servidor que la inserta en el código html resultante, cuando se renderiza la ejecución del código JS, salta la alerta que he enviado.

Otro ataque es el Path/Directory, que consiste en una vulnerabilidad crítica que consta de en una url como esta (ficticia): https://docs.google.com/documnt/d/PwertwuNc129Hacu1llf9tE=JD77C3qFPl96KU/edit, que al poseer un = podemos añadir ilimitadamente ../../../../etc y acceder a los archivos internos del ordenador, esto es especialmente preocupante con archivos como ../etc/passwd, que nos permitirá ver hasta el usuario que maneja la máquina donde corre la página, aunque los archivos con solo acceso de admin seguirán bloqueados.