UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERIA
CARRERA INGENIERIA EN TELECOMUNICACIONES

Análisis, diseño e implementación de un proceso hardening para la protección del servidor de la página web institucional de la Dirección de Tecnologías de la Información y Comunicación de la UNACH.

Trabajo de Titulación para optar al título de Ingeniero en Telecomunicaciones

Autor:
Baloa Rodriguez, Nazareth Andreina

Tutor:
Mag. Eduardo Daniel Haro Mendoza

Riobamba, Ecuador. 2023

DECLARATORIA DE AUTORÍA

Yo, Nazareth Andreina Baloa Rodriguez, con cédula de ciudadanía 175885257-6, autora del trabajo de investigación titulado: **Análisis, diseño e implementación de un proceso hardening para la protección del servidor de la página web institucional de la Dirección de Tecnologías de la Información y Comunicación de la UNACH**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autora de la obra referida será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 31 de octubre de 2023.

.

Nazareth Andreina Baloa Rodriguez
C.I:175885257-6

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

Quien suscribe, **Eduardo Daniel Haro Mendoza** catedrático adscrito a la Facultad de Ingeniería por medio del presente documento certifico haber asesorado y revisado el desarrollo del trabajo de investigación titulado "**Análisis, diseño e implementación de un proceso hardening para la protección del servidor de la página web institucional de la Dirección de Tecnologías de la Información y Comunicación de la UNACH"**, bajo la autoría de **Nazareth Andreina Baloa Rodriguez;** por lo que se autoriza ejecutar los trámites legales para su sustentación.

Es todo cuanto informar en honor a la verdad; en Riobamba, a los 11 días del mes de octubre del 2023.

Firmado electrónicamente por:
EDUARDO DANIEL HARO MENDOZA

Mgs. Eduardo Daniel Haro Mendoza

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **Análisis, diseño e implementación de un proceso hardening para la protección del servidor de la página web institucional de la Dirección de Tecnologías de la Información y Comunicación de la UNACH**, presentado por **Nazareth Andreina Baloa Rodriguez**, con cédula de identidad número 175885257-6, bajo la tutoría del Mg. Eduardo Daniel Haro Mendoza; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba el 31 de octubre de 2023.

Mgs. Deysi Inca.
PRESIDENTE DEL TRIBUNAL DE GRADO

Mgs Alejandra Pozo.
MIEMBRO DEL TRIBUNAL DE GRADO

PhD. Antonio Meneses
MIEMBRO DEL TRIBUNAL DE GRADO

CERTIFICADO ANTIPLAGIO

Dirección
Académica
VICERRECTORADO ACADÉMICO

en movimiento
SGC
SISTEMA DE GESTIÓN DE LA CALIDAD

UNACH-RGF-01-04-08.15
VERSIÓN 01: 06-09-2021

C E R T I F I C A C I Ó N

Que, **BALOA RODRIGUEZ NAZARETH ANDREINA** con CC: **175885257-6**, estudiante de la Carrera **TELECOMUNICACIONES**, Facultad de **INGENIERÍA**; ha trabajado bajo mi tutoría el trabajo de investigación titulado "**Análisis, diseño e implementación de un proceso hardening para la protección del servidor de la página web institucional de la Dirección de Tecnologías de la Información y Comunicación de la UNACH.**", cumple con el 5%, de acuerdo al reporte del sistema Anti plagio **URKUND**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 23 de octubre de 2023.

Firmado electrónicamente por:
EDUARDO DANIEL HARO MENDOZA

Mgs. Eduardo Daniel Haro Mendoza
TUTOR

DEDICATORIA

Este trabajo va dedicado primero a Dios, porque sin el nada sería posible. A mi abuela Antonia Silva, por apoyarme desde la distancia y alentarme a ser mejor cada día; a mi papá, mamá y hermana, por ser mi apoyo incondicional en todo este proceso. Y al resto de familiares y amigos, que estuvieron presente en todo este tiempo.

AGRADECIMIENTO

Agradezco primero a Dios, a mi madre Marlet Rodriguez y a mi padre Cesar Baloa, gracias por la educación y todos los valores que me dan. A mi hermana Nathaly Baloa por llenar de alegría mis días. A toda mi familia, que a pesar de la distancia siempre me alentaron y apoyaron para que lograra todos mis objetivos.

A mis primos Lorenzo y Leonel Montilla, por apoyarme desde que era muy pequeña y enseñarme a luchar por mis sueños.

A mis amigos Vanessa, Jessica, Jefferson y Roberto, porque son personas muy importantes para mí en este proceso, simplemente gracias por su apoyo y amistad.

A mi tutor y amigo Ing. Daniel Haro, por ayudarme en mi formación como profesional. Al Ing. Javier Montalvo por ser un guía en esta etapa tan importante.

A Ecuador, la Universidad Nacional de Chimborazo y a todos mis profesores que me guiaron en este camino para lograr ser una excelente profesional.

Y, por último, a Danilo Orna por ser parte de mi vida y ser mi apoyo incondicional siempre.

Índice General

ÍNDICE DE TABLAS.

ÍNDICE DE FIGURAS

RESUMEN

En la actualidad, la seguridad informática se ha vuelto un aspecto fundamental para las empresas o instituciones, ya que la mayoría de las actividades educativas y laborables dependen de sistemas informáticos conectados a redes o internet, es por ello, que estas están expuestas considerablemente a ciberataques.

Debido a esto, en el presente proyecto se elabora un proceso hardening para la mitigación de algunas de las vulnerabilidades más comunes en servidores web. Aplicando y comprobando este proceso en el servidor de la página web institucional de la UNACH.

Se realiza un análisis con la herramienta Pentest Tools preproceso hardening y se compara con una análisis post proceso hardening, en donde se detallan las vulnerabilidades encontradas y su proceso para mitigarlas. Finalmente, se demuestra la mitigación del 55% de las vulnerabilidades encontradas aplicando el proceso hardening.

Palabras claves: hardening, ciberseguridad, servidores, vulnerabilidades, ciberataques.

ABSTRACT

At the present time, computer security has become a fundamental aspect for companies or institutions since most educational and work activities depend on computer systems connected to networks or the Internet, which is why they are considerably exposed to cyber-attacks.

Due to this, a hardening process is developed in this project to mitigate some of the most common vulnerabilities in web servers. It has been applied, and this process has been checked on the UNACH institutional website server.

An analysis is carried out with the Pentest Tools pre-hardening process and compared with a post-hardening process analysis, which details the vulnerabilities found and the process to mitigate them. Finally, the mitigation of 55% of the vulnerabilities found by applying the hardening process is demonstrated.

Keywords: hardening, cybersecurity, servers, vulnerabilities, cyber-attacks.

Firmado electrónicamente por:
DARIO JAVIER
CUTIOPALA LEON

Reviewed by:
Mg. Dario Javier Cutiopala Leon
ENGLISH PROFESSOR
c.c. 0604581066

1. CAPÍTULO I

1.1 INTRODUCCIÓN

Las telecomunicaciones han evolucionado con el pasar del tiempo logrando un gran impacto en el ámbito de las redes. Debido a ello se realizan muchos estudios e investigaciones para lograr una correcta administración y gestión de red, en donde, el administrador de la red pueda manejar, controlar y monitorear toda la red de manera sencilla y rápida.

Ahora bien, en las grandes empresas o instituciones se suelen manejar gran cantidad de datos e información y, por ende, estas buscan tener una infraestructura de red sólida y con equipos diseñados para facilitar la realización de algunas tareas. Teniendo en cuenta que las instituciones están expuestas continuamente a ataques cibernéticos [1], se debe implementar un proceso para proteger un sistema o conjunto de sistemas informáticos. Este proceso es conocido como hardening [2]. El hardening es el proceso de protección de una infraestructura mediante la aplicación de configuraciones de seguridad específicas para prevenir ataques informáticos.

El proceso de hardening consiste en modificar las características específicas de un sistema de tal manera que se aumente su nivel de seguridad. Algunos de los ajustes que se proponen en el proceso son deshabilitar servicios o funciones que no se utilicen y sustituir algunas aplicaciones por versiones más seguras o actuales [2]. Además, se debe considerar que al implementar la metodología de hardening [2] esta permitirá robustecer los activos computacionales a nivel de seguridad, siempre y cuando la implementación se realice de tal forma que no afecte el rendimiento ni acceso a los servicios de la infraestructura computacional existente [3].

Por otro lado, los ataques cibernéticos [1] se han vuelto más frecuentes en la actualidad, siendo las instituciones públicas las más afectadas. En 2021 ESET [3] saco un reporte donde menciona que los códigos maliciosos más usados por los hackers [4] son los virus troyanos [5], gusanos [6], spyware [7] y ransomware [5]; y hace mención que solo en el 2020, en Ecuador ocurrieron alrededor de 140 mil detecciones de exploits [5], 6 mil detecciones de ransomware [5] y casi 8 mil detecciones de spyware [8]. Es por ello, que al desarrollar este proyecto se busca contribuir con la Dirección de tecnologías de información y comunicación (DTIC) de la UNACH para mejorar las carencias que puedan existir en su servidor de la página web institucional, logrando sobre guardar de manera más optima toda su información.

1.2 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

En la actualidad ninguna institución esta absuelta de sufrir un ciberataque, es por ello, que se realizan constantes estudios sobre cómo mejorar de manera continua la seguridad en una red. Ahora bien, la Universidad Nacional de Chimborazo cuenta con procesos de seguridad para sobre guardar sus datos, pero que pasaría si estos fallan y se logra entrar al sistema teniendo acceso a todos ellos, es ahí donde surge el problema para desarrollar este proyecto.

Al analizar la red de los servidores, se busca encontrar en donde existen debilidades del sistema y así plantear futuras mejoras en ellas. Se realizará un trabajo detallado y específico sobre el servidor web institucional. Se pretenden realizar algunos escaneos en este servidor para así determinar los puntos vulnerables a ataques cibernéticos para mitigarlos. Asimismo, se debe tener en consideración, que uno de los problemas que más impacta en la institución es que los usuarios pueden descargar diferentes softwares [9] o programas en los laboratorios que existen y esto sin necesidad de tener autorización de ningún supervisor o administrador de la red.

El servidor de la página web institucional de la UNACH, es de suma importancia, este contiene información relevante y es por ello, que para el DTIC surge la necesidad de protegerlo. Es ahí donde entra el problema de este proyecto. Pues con la realización del proceso Hardening en el servidor se garantiza la mitigación de las vulnerabilidades neutralizando casi por completo que este servidor pueda sufrir algún ataque.

1.3 OBJETIVOS

1.3.1 Objetivo General

- Analizar, diseñar e implementar un proceso de hardening para la protección del servidor de la página web institucional de la Dirección de Tecnologías de la Información y comunicación (DTIC) de la Universidad Nacional de Chimborazo.

1.3.2 Objetivos Específicos

- Estudiar el proceso Hardening que existe para minimizar los ciberataques en servidores.
- Analizar la infraestructura de red del servidor web institucional encargado de la DTIC.
- Identificar y evaluar las vulnerabilidades o riesgos en el servidor web institucional.
- Desarrollar una metodología para aplicar el proceso hardening para la protección de los servidores web institucional de la DTIC de la UNACH.
- Comprobar que el proceso hardening aplicado sea efectivo para la protección del servidor.

2. CAPÍTULO II

2.1 MARCO TEÓRICO

2.1.1 Infraestructura de red

Una infraestructura de red es el conjunto de hardware y software que forma parte de una red como servidores, routers, switch, computadoras, impresoras, aplicaciones, servicios de seguridad como antivirus, entre otros. Además, al tener una infraestructura de red bien establecida se optimiza el tráfico, se mejora la comunicación y se logra implantar un orden dentro de esa red.

2.1.2 Servidores

Los servidores se definen como computadoras pertenecientes a una red informática, las cuales, proveen determinados servicios a las demás computadoras conectada. El servidor debe tener una aplicación determinada que logre atender las peticiones de los usuarios conectados y brindarles una respuesta adecuada [10]. Además, estos suelen trabajar en un modelo de comunicación de cliente- servidor, distribuyendo las tareas entre los proveedores de recursos libres, para brindar así a sus usuarios la oportunidad de compartir datos, información determinada y dar accesos a ciertos recursos de hardware como impresoras, fax, entre otros.

2.1.3 Tipos de Servidores

Existen una variedad amplia de tipos de servidores que se distinguen por las diferentes tareas que pueden realizar y si son de hardware o software. A continuación, detallaremos algunos de ellos:

Servidor web: un servidor web o conocido también como servidor HTTP es un computador que procesa, almacena y entrega archivos de sitios web a los navegadores [11], es decir, permite que las páginas web, los videos e imágenes puedan ser difundidas en internet logrando establecer un enlace que comunica el servidor donde se encuentran los datos con el usuario que los solicita.

Servidor DNS: un servidor DNS es aquel que permite desde un dominio de internet encontrar el servidor que aloja ese sitio web o el sistema de correo. [10]

Servidor VPN: un servidor VPN, es el encargado de establecer comunicaciones seguras por medio de un túnel cifrado en redes inseguras como el Internet.[12]

Servidor de Correo: como su nombre lo dice, este es el encargado de los servicios de correo electrónico delegado de dar acceso al correo, recibir o almacenar información.

Servidor Proxy: Es un servidor que permite acceder a otras redes a través de él, es utilizado mayormente para permitir que cierto número de computadoras puedan navegar en internet de manera inspeccionada [13].

Servidor de Archivos: Son servidores que permiten almacenar o guardar ficheros informáticos[12].

Servidor Backup: Este servidor puede ser físico o lógico y su función es sustituir al servidor principal en caso de que exista un error o fallo que no deje que este siga dando su servicio[13].

Servidor de Seguridad: Este tipo de servidores básicamente se encarga de ofrecer servicios de protección a toda una red, como pueden ser antivirus, sistemas de intrusión, cortafuegos, etc[13].

2.1.4 Ciberseguridad

La ciberseguridad se refiere a la acción de proteger sistemas, redes y programas de posibles ataques digitales. Esta es la encargada de descubrir posibles fallos que logren poner en riesgo los dispositivos o sistemas de una red [14]. Además, esta establece medios que garanticen la defensa e impenetrabilidad al interior de una red.

2.1.5 Seguridad

Según [15] la seguridad es la falta de riesgos por la confianza que se tiene a alguien o algo. Es decir, la seguridad busca gestionar los riesgos tratando de evitarlos o prevenirlos de alguna manera evitando así, situaciones de comprometan la integridad del objeto o persona. Además, en la seguridad siempre están inmersas cuatro acciones, que son; prevención del riesgo, transferir el riesgo, mitigar el riesgo y aceptar el riesgo.

2.1.6 Seguridad informática

La seguridad informática se define como la ciencia encargada de los procesos, técnicas y métodos que buscan procesar, almacenar y transmitir información [15], teniendo como principal objetivo minimizar los riesgos que pueden provenir de cualquier parte, ya sea a la entrada o salida de datos, del medio donde se transportan los dato o incluso del hardware existente en una red. Ahora bien, la seguridad de la información va de la mano de la seguridad informática, pero está ya no abarca solo el medio informático, sino que se preocupa por todo aquello que pueda contener información y esta se basa en cuatro pilares que observamos en la figura 1.

Figura 1: Pilares de la seguridad de la información. Fuente: De Autor.

2.1.7 Vulnerabilidad informática

Es un componente del código o del software que identifica los defectos de la seguridad de las aplicaciones, sistemas y redes para que los cibercriminales puedan beneficiarse de ellos [13]. Es decir, una vulnerabilidad informática hace referencia a un error en un sistema que pueda ser provechado por un atacante generando situación de riesgo hacia la información existente en una red o servidor.

2.1.8 Hardening

Hardening (en español endurecimiento), se define como el proceso de fortalecer un sistema o equipo mediante la mitigación de vulnerabilidades en el mismo. El cual, consiste en modificar determinadas características de un sistema, de modo tal que aumente su nivel de seguridad. Según [16] El hardening del sistema se refiere a las acciones realizadas para reducir la superficie de ataque, asegurando las configuraciones de los componentes del sistema (servidores, aplicaciones, etc.).

2.1.9 Hardening en servidores

El hardening en servidores (en ingles Server hardening), es un proceso que se basa en fortalecer o robustecer un sistema, teniendo como objetivo proteger y asegurar un servidor de los ciberataques, reduciendo todos los posibles puntos vulnerables del sistema donde un atacante no autorizado pueda tener acceso. Para el hardening en servidores existen cuatro fases que se recomiendan aplicar, y estas son descritas en la figura 2.

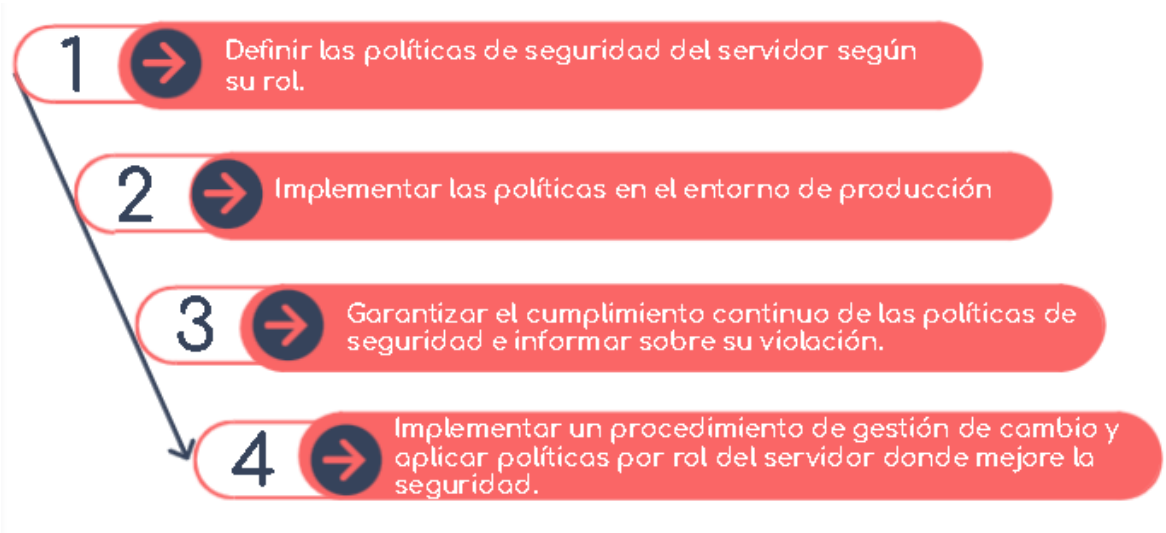


Figura 2: Fases para hardening en Servidores. Fuente: De Autor.

La ventaja que nos da realizar hardening en servidores es reducir los riesgos relacionados con fraude y error humano, además facilita un despliegue de configuración más limpio y seguro, y certifica que los recursos críticos se encuentren con parches actualizados siendo capaces de defenderse contra vulnerabilidades conocidas [13].

3. CAPÍTULO III

3.1 METODOLOGIA

3.1.1 Enfoque, Alcance y tipo

Este proyecto de investigación se considera que es de tipo mixto, puesto que, primero se realizara un escaneo del servidor, en donde se identifican los puntos vulnerables y luego, se diseñara e implementara un proceso hardening para así mejorar la seguridad del servidor reduciendo la posibilidad de sufrir un ciberataque.

Además, tiene un tipo de investigación descriptiva, que según [17]se define como "una investigación que se orienta a describir el fenómeno e identificar las características de su estado actual. Lleva a las caracterizaciones y diagnostico descriptivo". Al desarrollar esta investigación se busca centrarse primero en evaluar las vulnerabilidades del servidor, para después poder generar una solución a los problemas que se presenten.

Asimismo, se puede mencionar que es de índole experimental debido a que se buscan identificar las vulnerabilidades del servidor web institucional para mitigarlas y, por último, se efectuará una comparación del escáner inicial, con el escáner realizado luego de implementar hardening en el servidor de la página web institucional.

3.1.2 Proceso de la metodología.

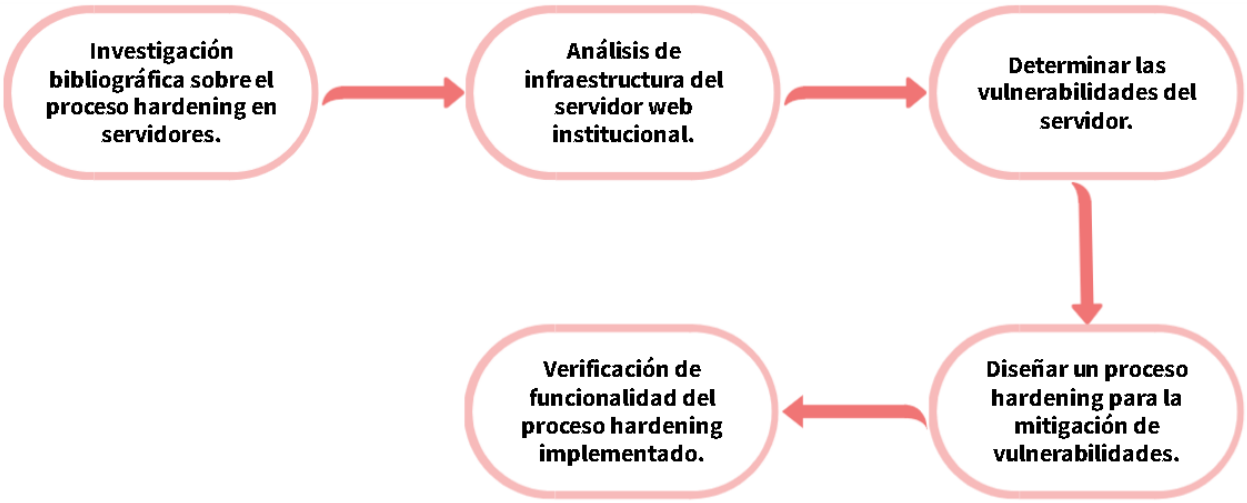


Figura 3: Diagrama del proceso de la metodología.

3.1.2.1 Fase 1 - Investigación bibliográfica sobre el proceso hardening en servidores.

En esta primera etapa se realizará una investigación exhaustiva en libros, artículos científicos, notas de revistas científicas y trabajos de grados que tengan relación al proceso

hardening y a la aplicación de este en servidores. En donde, profundizaremos en los aspectos claves para lograr la mitigación de vulnerabilidades y las diferentes metodologías que existen para aplicar dicho proceso.

3.1.2.2 Fase 2 - Análisis de infraestructura del servidor web institucional.

En esta etapa se busca detallar la topología o infraestructura de la red en la cual se encuentra operando el servido, puesto que, al tratarse de una institución educativa grande, se tiene una infraestructura de red un tanto compleja. Es por ello, que al analizarla lograremos tener un ambiente claro de trabajo. Y a su vez, será más fácil determinar las vulnerabilidades de este para obtener una correcta mitigación.

3.1.2.3 Fase 3 - Determinar las vulnerabilidades del servidor.

En la etapa 3, es cuando empezamos a realizar la búsqueda de vulnerabilidades en el servidor. Esto, se obtiene mediante escaneos que se realizan directamente al servidor web. Ahora bien, para obtener las vulnerabilidades se utiliza la herramienta de pentest tools, que esta básicamente realiza una prueba de penetración que valora los posibles fallos de seguridad informática que puede tener un sistema y qué alcance tienen dichos fallos[18].

Para realizar el escáner con la herramienta se realizaron los siguientes pasos:

1. Ingresamos a https://pentest-tools.com/website-vulnerability-scanning/website-scanner .

Figura 4: Pantalla de Pentest tools

2. Ingresamos en la parte derecha el URL de la página web de la Universidad Nacional de Chimborazo.

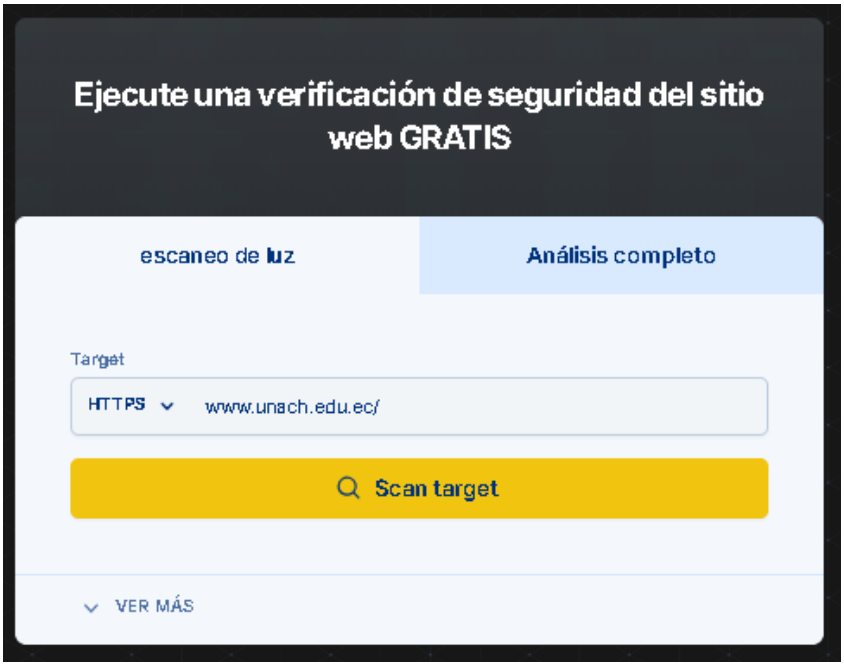


Figura 5: Ingreso del url de la página web a escanear

3. Iniciamos el escáner y esperamos que la herramienta nos muestre el informe con los resultados.

Figura 6: Inicio del escáner

4. Al obtener el informe que nos proporciona la herramienta podemos analizar y clasificar las vulnerabilidades.

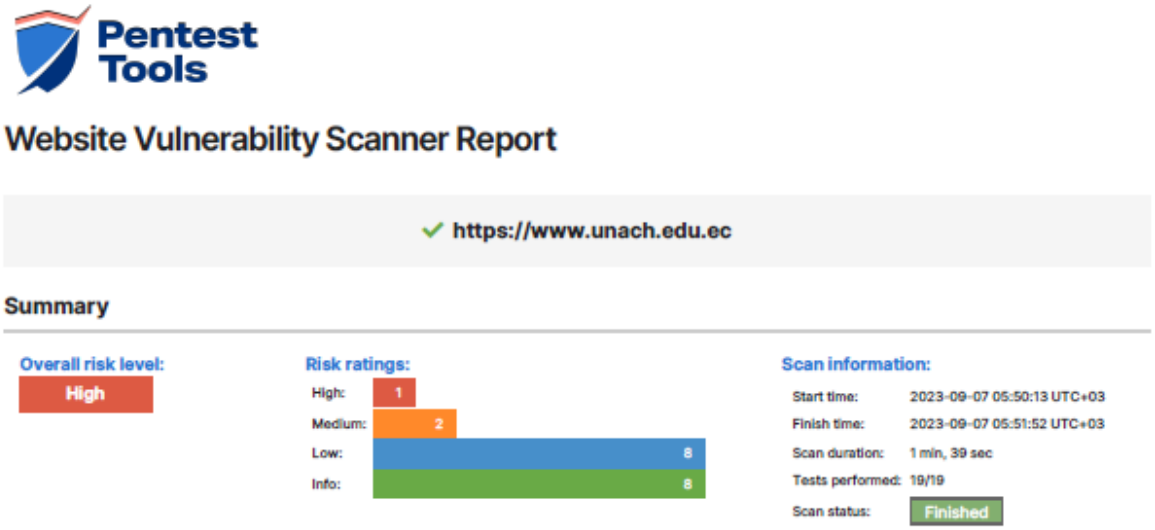


Figura 7: Resumen del informe arrojado por Pentest Tools.

3.1.2.4 Fase 4 - Diseñar un proceso hardening para la mitigación de vulnerabilidades.

En esta etapa, se plantea realizar un proceso hardening para la mitigación de seis vulnerabilidades de las encontradas. Ahora bien, según el tipo de vulnerabilidad se plantea una solución especifica en donde, se abordan todos los parámetros de esta para poder corregirla sin afectar el funcionamiento del servidor.

3.1.2.5 Fase 5 - Verificación de funcionalidad del proceso hardening implementado.

En la etapa final, lo que se busca es comprobar que el proceso hardening diseñado e implementado funciona de la mejor manera. Esto se verifica realizando un escaneo final en donde, compararemos el primero antes de realizar hardening con el escaneo realizado luego de aplicar el proceso hardening en las vulnerabilidades seleccionadas.

Es importante mencionar, que el proceso hardening se aplicara en una copia del servidor web institucional proporcionada por la Dirección de tecnologías de la información y comunicación de la UNACH, para evitar caídas en el servidor principal.

3.1.3 Población y muestra

Debido al tipo de investigación planteada, se establece que la población son las diferentes vulnerabilidades que existen en la actualidad en los sistemas y de esta forma, podemos definir que tenemos un muestreo no probabilístico, ya que, de las 11 vulnerabilidades encontradas en el servidor solo se realizó la mitigación de 6 de ellas. Teniendo en cuenta, que estas vulnerabilidades no son producto de un proceso aleatorio, sino que fueron seleccionadas de acuerdo al nivel de riesgo en el servidor. Es importante mencionar que para la investigación no se tenía un acceso pleno al servidor y sus configuraciones y por ello, no se mitigaron todas la vulnerabilidades.

3.1.4 Variables

Tabla 1: Variables Independientes y dependientes.

	Variable	Definición	Dimensión	Indicadores
Independiente	Hardening	Se define como el proceso de fortalecer un sistema o equipo mediante la mitigación de vulnerabilidades en el mismo.	• Configuraciones de seguridad.	• Configuración actual del servidor. • Actividades de protección realizadas.
Dependiente	Seguridad del servidor web institucional	Se define como la acción destinada a salvaguardar la información, el software y los sistemas vinculados al servidor.	• Escaneo de sistema para ver vulnerabilidades. • Consumo de recursos.	• Número de vulnerabilidades. • Nivel de confidencialidad de la información

4. CAPÍTULO IV

4.1 RESULTADOS Y DISCUSIÓN

4.1.1 Procesos de hardening en servidores web.

El hardening en servidores se refiere a tomar medidas para fortalecer la seguridad de un servidor, reduciendo así su superficie de ataque y minimizando posibles vulnerabilidades. Esto incluye la implementación de varias medidas y configuraciones de seguridad para proteger los servidores de amenazas y ataques cibernéticos. Ahora bien, existen aspectos claves con relación al proceso hardening en servidores que son:

- Actualización del sistema operativo
- Configuración de firewall
- Eliminación de servicios innecesarios
- Configuración de políticas de acceso
- Auditoria y registro
- Protección contra Malware y antivirus
- Control de privilegios
- Monitoreo de seguridad
- Copias de seguridad

No solo se deben tener en cuenta los aspectos antes señalados, sino que también se debe tener presente que el proceso hardening puede ser diferente según el sistema operativo del servidor y las funciones de seguridad de este. Ahora bien, para profundizar más en el tema, se realizó una investigación bibliográfica con relación a trabajos sobre el proceso de hardening en servidores web, obteniendo como resultados relevantes y beneficiosos para este trabajo los que se describen en la tabla 2.

Tabla 2: Investigación bibliográfica sobres trabajos relacionados al proceso hardening en servidores web

Título del trabajo	Autor(es)/año de publicación	Resumen de proceso realizado/Herramientas utilizadas.
Análisis y Proceso de Hardening de Servidor Virtual Web, Facultad de Ingeniería (IngeTic).	Toto Ronald Sánchez Cari (2021)	Se creo un ambiente virtual con un sistema operativo Linux Ubuntu para analizar las vulnerabilidades que puedan existir, realizando un escáner con la herramienta Nikto. Se realizaron acciones para mejorar la seguridad del servidor como; actualizaciones, denegaciones de servicios, ocultar versiones del sistema operativo y seguridad de encabezados. Al tratarse de un ambiente virtualizado, realizaron configuraciones al servidor con comandos para actualizar el SO, autenticación de usuarios, permisos de usuarios, etc.
Diseño de un esquema de hardening para los servidores web de la oficina TIC de la Unidad Administrativa Especial de Servicios Públicos de la ciudad de Bogotá.	Oscar Ricardo Rodriguez Martínez (2023)	Se trabajo sobre una infraestructura de red grande en donde, el investigador divide el trabajo en verificación de puertos abierto, análisis de vulnerabilidades y pruebas de vulnerabilidades sobre aplicaciones web. además, detalla cada servidor con sus funciones y aplicaciones. Utiliza herramientas como NMAP, GreenBone y OWASP ZAP.
Uso Práctico De Encabezados Http Para Mejorar La Seguridad En El Protocolo	Andrés Iván Anturi Figueroa (2020)	Este trabajo amplía el proyecto OWASP y su elemento de escaneo para ofrecer una herramienta que permite a los desarrolladores aprender sobre el uso seguro de los encabezados de respuesta en sus aplicaciones web. Además, la propuesta incluye tres categorías para ayudar aún más a los desarrolladores a seleccionar el encabezado que mejor satisfaga sus necesidades.
Automatización de pruebas de seguridad a servidores web	Leobel Rodríguez Chang1, Henry Raúl Gonzáles Brito, Dayana Pérez Fernández (2021)	En esta investigación se plantea el desarrollo de una aplicación con el propósito de automatizar la realización de pruebas de seguridad en servidores web, enfocándose en tres áreas principales: la seguridad de los archivos de configuración, la evaluación de la encriptación TLS/SSL, y la realización de pruebas desde la perspectiva del cliente.

Proceso De Hardening de Servidor Web	Erick Giovanni Varela Guzmán (2020)	En este trabajo se basan en los servidores Nginx y Apache debido a que son los más populares a nivel mundial. Plantean un proceso práctico de seguridad recomendado e implementado sobre un servidor web en un escenario de trabajo utilizando máquinas virtuales. Esta investigación se enfoca en el punto 6 de la lista de vulnerabilidades comunes de OWASP, "mala gestión de configuración se seguridad", puesto que, está relacionada con muchas más vulnerabilidades comúnmente encontradas.
Técnicas de programación segura para mitigar vulnerabilidades en aplicaciones web.	Monar Joffre, Pástor Danilo, Arcos Gloria, Oñate Alejandra (2018)	En este estudio, se presenta un conjunto de estrategias de programación segura diseñadas para mitigar las vulnerabilidades en aplicaciones web desarrolladas en el entorno PHP. En este contexto, se identificaron inicialmente diez vulnerabilidades mediante el cumplimiento de las recomendaciones de OWASP TOP-10. Luego, se exponen siete técnicas específicas junto con sus respectivas pautas de implementación. Y la efectividad de estas técnicas se confirma a través de una evaluación que compara la seguridad de una aplicación web en dos escenarios distintos: uno con la aplicación de las técnicas propuestas y otro sin ellas.

4.1.2 Infraestructura del servidor web institucional.

La figura 8, se muestra la infraestructura en la que se encuentra funcionando el servidor que contiene la página web institucional. Recalcando, que en esta investigación se trabajó sobre una copia proporcionada por el departamento de tecnologías de la información y comunicación. Teniendo una Ip pública "192.168.150.152" y un dominio https://hweb.unach.edu.ec/.

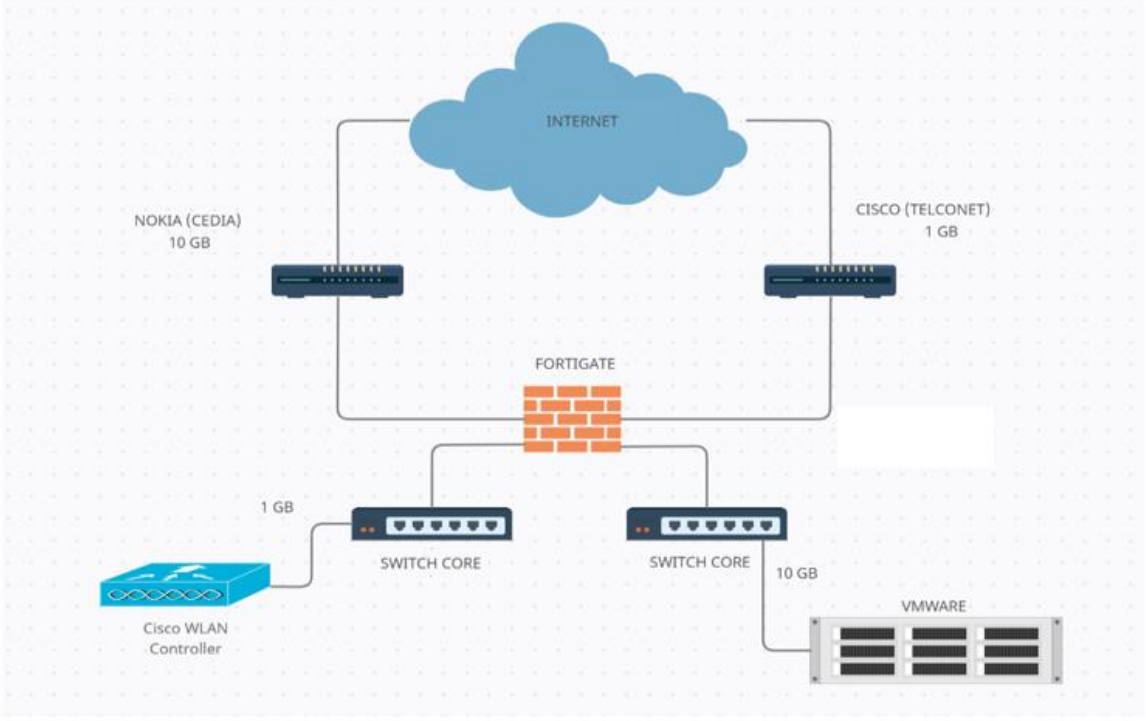


Figura 8: Topología de la red del servidor. Fuente: DTIC Unach

4.1.3 Identificación de vulnerabilidades en el servidor web institucional.

Para lograr identificar las vulnerabilidades del servidor web institucional, primero se realizó una entrevista con el Ingeniero Javier Montalvo, donde él nos indicó cual es la estructura del servidor y cómo funciona. Posteriormente, se realizó un escaneo de la red actual con la aplicación Pentest Tools[19].

Ahora bien, Pentest Tools es una herramienta web creada en el 2013 por profesionales dedicados a la seguridad informática, esta permite realizar escaneos para evaluar la seguridad de una red o de aplicaciones web[20]. Además, la plataforma combina más de 20 herramientas que optimizan el trabajo de las pruebas de seguridad, teniendo así una excelente garantía. Pentest Tools realiza diferentes tareas como:

Mapeo de la superficie de ataque: Descubriendo los principales objetivos de ataque en una red, puestos abiertos, archivos ocultos, etc.

Escaneo de vulnerabilidades: Tanto de redes como de aplicaciones o sitios web, mediante herramientas para detectar vulnerabilidades comunes o específicas.

Explotación: Mostrando a los clientes cuán importante es la seguridad realizándoles hacking ético para explorar, accesos, vulnerabilidades, etc.

Monitoreo continuo de la seguridad: Programando análisis periódicos de vulnerabilidades, teniendo constante monitoreo en la red o aplicación web y así prevenir un ataque reducir riesgos considerables.

Agregando a lo anterior, el escaneo realizado con la herramienta menciona nos arrojó un informe final, en donde se encontraron 11 vulnerabilidades clasificadas en riesgo alto, medio y bajo como se describen en la tabla 3.

Tabla 3: Vulnerabilidades encontradas en el servidor.

Ítem	Vulnerabilidad	Riesgo	Descripción
1	CVE-2022-37454 CVE-2017-8923 CVE-2022-31629 CVE-2021-3618 CVE-2022-31628	Alto	Desbordamiento de enteros y un desbordamiento de búfer resultante que permite a los atacantes ejecutar código arbitrario o eliminar propiedades criptográficas esperadas. Esto ocurre en la interfaz de función de esponja[21].
			No hay prohibición de cambios en los objetos de cadena que resultan en una longitud negativa, lo que permite a atacantes remotos provocar una denegación de servicio o posiblemente tener otro impacto no especificado al aprovechando el uso de en un script con una cadena larga[22].
			En las versiones de PHP anteriores a 7.4.31, 8.0.24 y 8.1.11, la vulnerabilidad permite a atacantes de red y del mismo sitio establecer una cookie insegura estándar en el navegador de la víctima que se trata como una cookie `__Host-` o `__Secure-`. mediante aplicaciones PHP[23].
			Un atacante MiTM que tenga acceso al tráfico de la víctima en la capa TCP/IP puede redirigir el tráfico de un subdominio a otro, lo que resultará en una sesión TLS válida. Esto rompe la autenticación de TLS y pueden ser posibles ataques entre protocolos donde el comportamiento de un servicio de protocolo puede comprometer al otro en la capa de aplicación[24].
			En las versiones de PHP anteriores a 7.4.31, 8.0.24 y 8.1.11, el código del descompresor phar descomprimiría recursivamente archivos gzip "quines", lo que resultaba en un bucle infinito[25].
2	Configuración de cookies inseguras: falta el indicador de seguridad	Medio	Cuando una cookie no tiene configurada la bandera segura, se enviará en cada solicitud a través de HTTP y HTTPS. Incluso si la aplicación web en sí se envía a través de HTTPS, un atacante aún podría robar la sesión en uso al obligar al usuario a realizar una solicitud HTTP y luego robar la cookie de sesión allí[26].
3	Configuración de cookies insegura:	Medio	Cuando una cookie no tiene un indicador HttpOnly, se puede acceder a ella a través de JavaScript, lo que significa que un XSS podría provocar el robo de cookies. Estas cookies

	falta el indicador HttpOnly		incluyen, entre otras, tokens CSRF y sesiones de clientes que pueden facilitar la toma de control de cuenta/sesión[27].
4	Archivo robots.txt encontrado	Bajo	No se presenta un riesgo de seguridad específico al utilizar un archivo robots.txt. No obstante, es común que los administradores de sitios web lo utilicen de manera inapropiada en un intento de ocultar ciertas páginas a los usuarios. Es importante destacar que esto no debe considerarse una medida efectiva de seguridad, ya que estas URL pueden leerse con facilidad directamente desde el archivo robots.txt
5	Software y tecnología de servidor encontrados	Bajo	Un atacante podría emplear estos datos para llevar a cabo ataques dirigidos hacia el software en cuestión, aprovechando la información sobre su tipo y versión.
6	Falta encabezado de seguridad: Contenido-Seguridad-Política	Bajo	El encabezado Content-Security-Policy habilita una función de seguridad integrada en los navegadores web que previene la explotación de vulnerabilidades de Cross-Site Scripting. Cuando la aplicación de destino es propensa a XSS y no se utiliza este encabezado, los atacantes tienen una vía sencilla para aprovecharla.
7	Falta encabezado de seguridad: X-Frame-Options	Bajo	Dado que el servidor no proporciona el encabezado X-Frame-Options, existe la posibilidad de que un atacante pueda insertar este sitio web dentro de un iframe en un sitio web de terceros.
8	Falta encabezado de seguridad: Política de referencia	Bajo	El encabezado Referrer-Policy en el protocolo HTTP controla la cantidad de información de referencia que el navegador enviará junto con cada solicitud que se origine desde la aplicación web actual. Es decir, si un usuario visita la página web y hace clic en un enlace que lleva a otro sitio, el navegador enviará la URL de origen completar en el encabezado "Referer" si el encabezado Referrer-Policy no está configurado. Dicha URL de origen puede contener información sensible y se podría utilizar para rastrear a los usuarios.
9	Falta encabezado de seguridad: X-	Bajo	El encabezado HTTP `X-Content-Type-Options` está dirigido al navegador Internet Explorer y le impide reinterpretar el contenido de una página web. La falta de este encabezado podría provocar ataques como Cross-Site Scripting o phishing.

	Content-Type-Options		
10	Falta encabezado de seguridad: X-XSS-Protection	Bajo	El encabezado HTTP X-XSS-Protection notifica al navegador que detenga la carga de páginas web cuando detecta intentos de ataques de Cross-Site Scripting. La ausencia de este encabezado deja a los usuarios de la aplicación vulnerables a posibles ataques XSS si la aplicación web tiene esta debilidad de seguridad.
11	Falta encabezada de seguridad: Estricta-Transporte-Seguridad	Bajo	El encabezado HTTP Strict-Transport-Security (HSTS) instruye al navegador a establecer únicamente conexiones seguras con el servidor web, rechazando cualquier intento de conexión HTTP no cifrada. Si este encabezado está ausente, un atacante podría forzar a un usuario a establecer una conexión HTTP no cifrada con el servidor, lo que podría facilitar la interceptación del tráfico de la red y la extracción de información confidencial.

4.1.4 Proceso hardening para mitigar las vulnerabilidades encontradas.

Al tener identificada las vulnerabilidades con las que se trabajara, se plantea un proceso de hardening para cada una de ella:

1. ***Vulnerabilidad 1 - Falta encabezado de seguridad: Contenido-Seguridad-Política.***

El encabezado Content Security Policy (*Contenido-Seguridad-Política)*, permite especificar desde donde se pueden cargar recursos en la página. Cuando se incluyen fuentes de contenido aptas, evita que se cargue código malicioso en el sitio web y con ellos, se reducen los riesgos de XSS[28].

Para mitigar esta vulnerabilidad debemos agregar en el archivos de configuración del servidor la siguiente línea de código:

add_header Content-Security-Policy “default-src ‘self’” always;

Esta línea agrega el encabezado Content Security Policy a todas las respuestas del servidor y especifica desde donde se pueden cargar recursos como scripts, estilos y fuentes.

2. ***Vulnerabilidad 2 - Falta encabezado de seguridad: X-Frame-Options.***

El encabezado X-Frame-Options, permite mejorar la protección de las aplicaciones web en contra del Click-jacking, declarando una política comunicada desde un host al navegador del cliente acerca de si debe mostrar o no el contenido transmitido en marcos de otras páginas web.

Para mitigar esta vulnerabilidad debemos agregar en el archivos de configuración del servidor la siguiente línea de código:

add_header X-Frame-Options “deny” always;

Esta línea agrega el encabezado X-Frame-Options a todas las respuestas del servidor y especifica que se prohíbe que el sitio web pueda ser embebido en marcos.

3. ***Vulnerabilidad 3 - Falta encabezado de seguridad: Política de referencia.***

El encabezado Referrer-Policy, es el que rige qué información de remisión debe ser incluida en las peticiones realizadas. Enviando solo el origen del documento como referencia a un destino a-priori priori hacia el más seguro (HTTPS->HTTPS), pero no se envía a un destino menos seguro (HTTPS->HTTP).

Para mitigar esta vulnerabilidad debemos agregar en el archivos de configuración del servidor la siguiente línea de código:

add_header Referrer-Policy “strict-origin-when-cross-origin” always;

Esta línea agrega el encabezado Referrer-Policy a todas las respuestas del servidor, indicando que se enviara información referer solo cuando la solicitud se realice a un dominio diferente.

4. ***Vulnerabilidad 4 - Falta encabezado de seguridad: X-XSS-Protection.***

El encabezado X-XSS-Protection, habilita el filtro de Cross-site scripting (XSS) en el navegador. Es decir, al detectar un ataque XSS, el navegador en vez de desinfectar bloquea la reproducción de la página.

Para mitigar esta vulnerabilidad debemos agregar en el archivos de configuración del servidor la siguiente línea de código:

add_header X-XSS-Protection “1; mode=block” always;

Esta línea agrega el encabezado X-XSS-Protection a todas las respuestas del servidor, activando el filtro XSS y bloqueando la ejecución de scripts si se detecta un ataque XSS.

5. ***Vulnerabilidad 5 - Falta encabezada de seguridad: Estricta-Transporte-Seguridad.***

El encabezado Strict Transport Security, permite a los servidores web determinar que los navegadores sólo pueden interactuar utilizando conexiones HTTPS seguras y no sobre HTTP inseguro. Esta política de seguridad web que protege a los sitios web contra los ataques de degradación de protocolo y el secuestro de cookies.

Para mitigar esta vulnerabilidad debemos agregar en el archivos de configuración del servidor la siguiente línea de código:

add_header Strict-Transport-Security “max-age=63072000; includeSubdomains;” always;

Esta línea agrega el encabezado Strict Transport Security a todas las respuestas del servidor, Indicando que cada 63072000 segundos el navegador recuerda que el sitio solo puede ser visitado utilizando HTTPS y aplicando a su vez, a todos los subdominios.

6. ***Vulnerabilidad 6 - Configuración de cookies inseguras: falta el indicador de seguridad***

La configuración de cookies inseguras hace referencia a que en la configuración del servidor web no existe el atributo “Secure”. Ahora bien, este atributo es una particularidad de las cookies que especifica que la está solo puede enviarse mediante conexiones cifradas (HTTPS).

Para mitigar esta vulnerabilidad debemos agregar en el archivos de configuración del servidor la siguiente línea de código:

add_header Set-Cookie “Path=/; HttpOnly; Secure”;

Esta línea agrega para indicar al navegador que debe configurar una cookie, estableciendo ruta y atributos.

Figura 9: Captura de la configuración del servidor.

4.1.5 Comprobación del proceso hardening aplicado.

Luego de aplicar el proceso hardening establecido para cada una de las vulnerabilidades seleccionadas, procedemos a realizar otro escáner, en donde, verificaremos si fueron mitigadas o no las vulnerabilidades. En las figuras 9-10, se resumen los resultados obtenidos de los escáner. Además, se realiza una hipótesis justificando su eficiencia mediante un chi-cuadrado.

Prueba de hipótesis:

> ***Ho:*** La aplicación de un proceso hardening se relaciona significativamente con las vulnerabilidades que se presenten en un servidor.
>
> ***Ha:*** La aplicación de un proceso hardening no se relación significativamente con las vulnerabilidades que se presentan en un servidor.

Tabla 4: Resultados del Chi-Cuadrado.

	X-squared	p-value	Df
Chi-Cuadrado	2,25	0.1336	1

Debido a que p-value>0,05. Se acepta la hipótesis nula, es decir, se comprueba que la aplicación de un proceso hardening se relaciona significativamente con las vulnerabilidades que se presentan en un servidor.

```
> cuadro
               Vulnerabilidades
Sin Hardening                11
Con Hardening                 5

> chisq.test(cuadro)

        Chi-squared test for given probabilities

data:  cuadro
X-squared = 2.25, df = 1, p-value = 0.1336
```

Figura 10: Captura del software R mostrando los resultados de chi-cuadrado

- Escáner Pre-Proceso Hardening

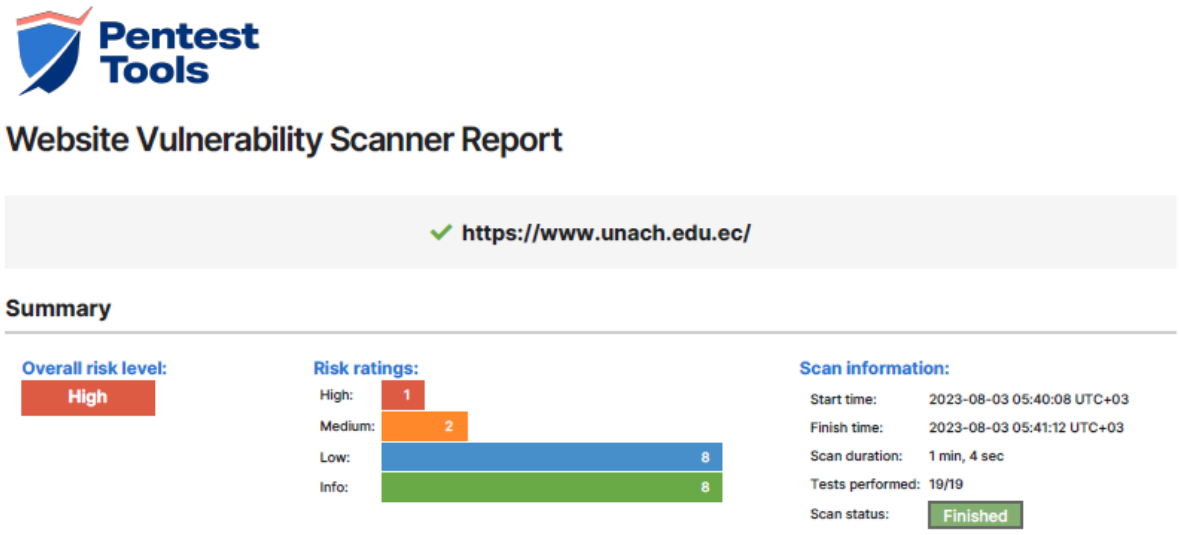


Figura 11: Resumen del informe Pentest Tools, Pre-Hardening

- Escáner Post-Proceso hardening

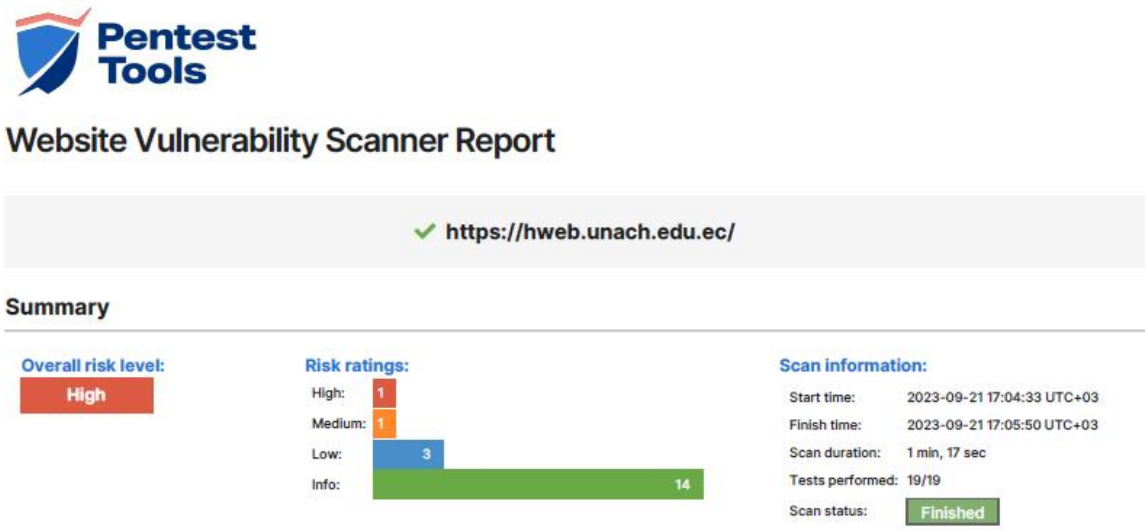


Figura 12: Resumen del informe Pentest Tools, Post-Hardening

En la figura 13, se presenta una gráfica de comparación entre las vulnerabilidades del servidor antes de ser implementado el proceso hardening y luego de aplicar el proceso hardening, obteniendo resultados satisfactorios, puesto que, se logra la mitigación de 6 vulnerabilidades, 1 de riesgo medio y 5 de riesgo alto.

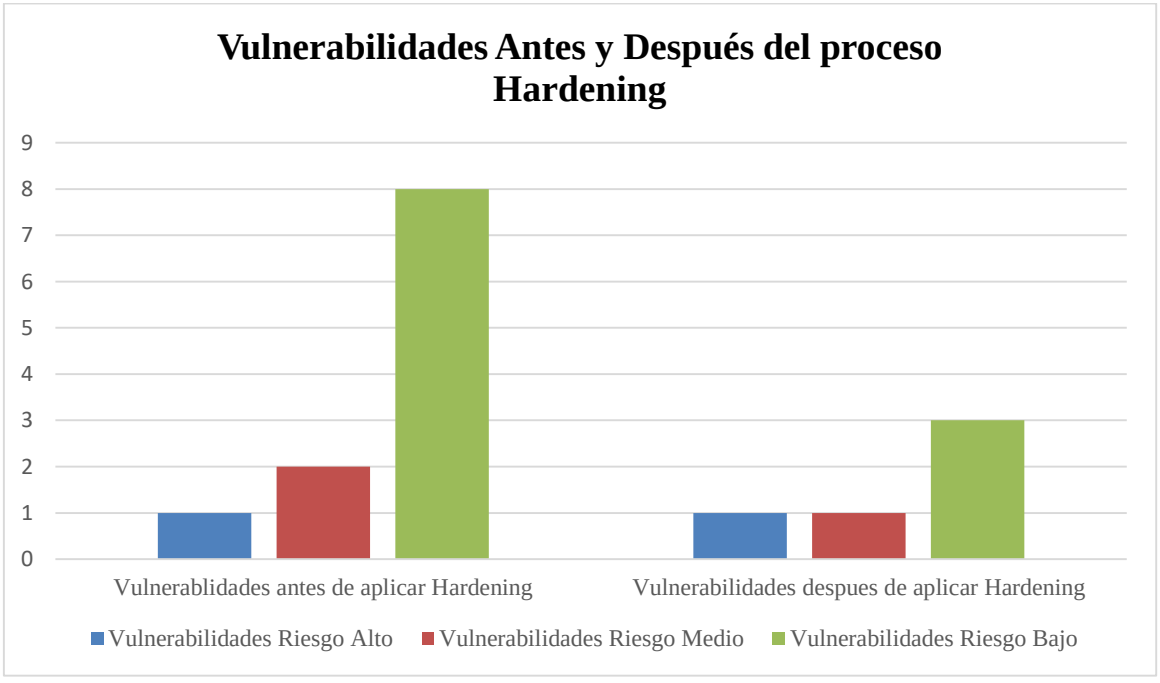


Figura 13: Grafica de vulnerabilidades antes y después del proceso hardening.

5. CONCLUSIONES

Se alcanzaron con satisfacción los objetivos de la investigación, que incluyen el análisis del proceso Hardening en servidores y la implementación efectiva en el servidor de la página web institucional.

Al estudiar las vulnerabilidades del servidor de la página web institucional de la UNACH se identificaron que las vulnerabilidades más comunes son de actualización del sistema operativo y de encabezados.

El proceso hardening implementado demostró ser eficaz en la mejora de la seguridad del servidor de la página web institucional, logrando identificar y mitigar riesgos significativos.

Al trabajar con una copia del servidor se deja un ambiente claro planteado para el Departamento de tecnologías de la información y comunicación de la UNACH, que sirve como modelo para realizar mejoras en la seguridad del servidor de la página web institucional en producción.

6. RECOMENDACIONES

Se recomienda mantener el sistema operativo actualizado, puesto que, muchas vulnerabilidades se relacionan directamente con esto y hace que el servidor sea más vulnerable.

También, al desarrollar este proyectos nos dimos cuenta de que las carpetas de configuración del servidor se encuentran desordenadas, entonces se recomienda tener un poco más de orden a la hora de la programación, ya que con ello se lograría tener un ambiente más claro y así poder implementar políticas de seguridad más eficientes.

7. BIBLIOGRÁFIA

[1] Carisio Emanuele, “MEDIACLOUD.”

[2] Benchimol Daniel, *Hacking Cero desde*, 1era ed. Buenos Aires, 2011.

[3] ESET, “Seguridad digital en latinoamerica.”

[4] Vélez Cuauhtémoc, “Instituto de ingenierìa UNAM,” Hackers.

[5] Belcic Ivan, “Academy,” ¿Qué es un malware troyano?

[6] Belcic Ivan, “Avast Academy,” ¿Qué es un gusano informático?

[7] Seguin PAtrick, “Avast Academy-Spyware,” Spyware: detección, prevención y eliminación.

[8] Lizette Abril, “El comercio,” *Ecuador está entre los países con más ciberataques en América Latina*, 2021.

[9] “Significados-Software.”

[10] Equipo Editorial Etecé, “Servidor,” Servidor.

[11] Equipo Editorial Etecé, “Servidor Web,” Servidor Web.

[12] J. L. Martinez, “PRORED,” ¿Qué tipos de servidores hay?

[13] K. Paola Pinduisaca Guashpa, “UNIVERSIDAD NACIONAL DE CHIMBORAZO FACULTAD DE INGENIERÍA.”

[14] Cisco, “Ciberseguridad,” Cisco.

[15] M. I. Romero *et al.*, “INTRODUCCIÓN A LA SEGURIDAD INFORMÁTICA Y EL ANÁLISIS DE VULNERABILIDADES.”

[16] Redacción KeepCoding, “¿Qué es el Hardening en Ciberseguridad?,” KeepCoding.

[17] H. Hugo Sánchez, C. Carlos, R. Romero, and K. Mejía Sáenz, “Vicerrectorado de Investigación.”

[18] Shirly Nowak, “Nuclio,” CIBERSEGURIDAD-¿Qué es el Pentesting?

[19] Pentest Tools, “Pentest-Tools.com.”

[20] C. DE Ingeniería En Sistemas Y Computación, K. Paola Pinduisaca Guashpa, and I. Lorena Molina Valdiviezo, “UNIVERSIDAD NACIONAL DE CHIMBORAZO FACULTAD DE INGENIERÍA.”

[21] National Vulnerability Database, “NVD - CVE-2022-37454.”

[22] National Vulnerability Database, “NVD - CVE-2017-8923.”

[23] National Vulnerability Database, “NVD - CVE-2022-31629.”

[24] National Vulnerability Database, “NVD - CVE-2021-3618.”

[25] National Vulnerbility Database, “NVD - CVE-2022-31628.”

[26] Detectify, “Cookie lack Secure flag.”

[27] Detectify, “Missing HttpOnly flag on cookies.”

[28] DreamsHost, “Encabezados de seguridad.”

8. ANEXOS

Anexo 1: Informe de escáner realizado por pentest Tools antes de aplicar el proceso Hardening.

Pentest Tools

Website Vulnerability Scanner Report

✔ https://www.unach.edu.ec/

Summary

Overall risk level: High

Risk ratings:

High:	1
Medium:	2
Low:	8
Info:	8

Scan information:

Start time:	2023-08-03 05:40:08 UTC+03
Finish time:	2023-08-03 05:41:12 UTC+03
Scan duration:	1 min, 4 sec
Tests performed:	19/19
Scan status:	Finished

Findings

Vulnerabilities found for server-side software

UNCONFIRMED

Risk Level	CVSS	CVE	Summary	Exploit	Affected software
	9.8	CVE-2022-37454	The Keccak XKCP SHA-3 reference implementation before fdc6fef has an integer overflow and resultant buffer overflow that allows attackers to execute arbitrary code or eliminate expected cryptographic properties. This occurs in the sponge function interface.	N/A	php 7.3.33
	7.5	CVE-2017-8923	The zend_string_extend function in Zend/zend_string.h in PHP through 7.1.5 does not prevent changes to string objects that result in a negative length, which allows remote attackers to cause a denial of service (application crash) or possibly have unspecified other impact by leveraging a script's use of .= with a long string.	N/A	php 7.3.33
	6.5	CVE-2022-31629	In PHP versions before 7.4.31, 8.0.24 and 8.1.11, the vulnerability enables network and same-site attackers to set a standard insecure cookie in the victim's browser which is treated as a `__Host-` or `__Secure-` cookie by PHP applications.	N/A	php 7.3.33
	5.8	CVE-2021-3618	ALPACA is an application layer protocol content confusion attack, exploiting TLS servers implementing different protocols but using compatible certificates, such as multi-domain or wildcard certificates. A MiTM attacker having access to victim's traffic at the TCP/IP layer can redirect traffic from one subdomain to another, resulting in a valid TLS session. This breaks the authentication of TLS and cross-protocol attacks may be possible where the behavior of one protocol service may compromise the other at the application layer.	N/A	nginx 1.20.1
	5.5	CVE-2022-31628	In PHP versions before 7.4.31, 8.0.24 and 8.1.11, the phar uncompressor code would recursively uncompress "quines" gzip files, resulting in an infinite loop.	N/A	php 7.3.33
	4.3	CVE-2019-11358	jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.	N/A	jquery 1.12.4
	4.3	CVE-2020-11022	In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.	N/A	jquery 1.12.4
	4.3	CVE-2020-11023	In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.	N/A	jquery 1.12.4
	4.3	CVE-2015-9251	jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.	N/A	jquery 1.12.4

1 / 8

Details

Risk description:
These vulnerabilities expose the affected applications to the risk of unauthorized access to confidential data and possibly to denial of service attacks. An attacker could search for an appropriate exploit (or create one himself) for any of these vulnerabilities and use it to attack the system.

Recommendation:
We recommend you to upgrade the affected software to the latest version in order to eliminate the risk of these vulnerabilities.

Classification:
CWE : CWE-1026
OWASP Top 10 - 2013 : A9 - Using Components with Known Vulnerabilities
OWASP Top 10 - 2017 : A9 - Using Components with Known Vulnerabilities

Insecure cookie setting: missing HttpOnly flag

CONFIRMED

URL	Cookie Name	Evidence
https://www.unach.edu.ec/	PHPSESSID	The server responded with Set-Cookie header(s) that does not specify the HttpOnly flag: Set-Cookie: PHPSESSID=culb5cu6eemaec7hkikp67u0jd

Details

Risk description:
A cookie has been set without the `HttpOnly` flag, which means that it can be accessed by the JavaScript code running inside the web page. If an attacker manages to inject malicious JavaScript code on the page (e.g. by using an XSS attack) then the cookie will be accessible and it can be transmitted to another site. In case of a session cookie, this could lead to session hijacking.

Recommendation:
Ensure that the HttpOnly flag is set for all cookies.

References:
https://owasp.org/www-community/HttpOnly

Classification:
CWE : CWE-1004
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Insecure cookie setting: missing Secure flag

CONFIRMED

URL	Cookie Name	Evidence
https://www.unach.edu.ec/	PHPSESSID	Set-Cookie: PHPSESSID=culb5cu6eemaec7hkikp67u0jd; path=/, cookiesession1=678B28BD944EC5AF89872CC961B1770E;Expires=Fri, 02 Aug 2024 02:40:14 GMT;Path=/;HttpOnly

Details

Risk description:
Since the `Secure` flag is not set on the cookie, the browser will send it over an unencrypted channel (plain HTTP) if such a request is made. Thus, the risk exists that an attacker will intercept the clear-text communication between the browser and the server and he will steal the cookie of the user. If this is a session cookie, the attacker could gain unauthorized access to the victim's web session.

Recommendation:
Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.

References:
https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html

Classification:
CWE : CWE-614

OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: X-Frame-Options

CONFIRMED

URL	Evidence
https://www.unach.edu.ec/	Response headers do not include the HTTP X-Frame-Options security header

Details

Risk description:
Because the `X-Frame-Options` header is not sent by the server, an attacker could embed this website into an iframe of a third party website. By manipulating the display attributes of the iframe, the attacker could trick the user into performing mouse clicks in the application, thus performing activities without user consent (ex: delete user, subscribe to newsletter, etc). This is called a Clickjacking attack and it is described in detail here:
https://owasp.org/www-community/attacks/Clickjacking

Recommendation:
We recommend you to add the `X-Frame-Options` HTTP header with the values `DENY` or `SAMEORIGIN` to every page that you want to be protected against Clickjacking attacks.

References:
https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: Content-Security-Policy

CONFIRMED

URL	Evidence
https://www.unach.edu.ec/	Response headers do not include the HTTP Content-Security-Policy security header

Details

Risk description:
The Content-Security-Policy (CSP) header activates a protection mechanism implemented in web browsers which prevents exploitation of Cross-Site Scripting vulnerabilities (XSS). If the target application is vulnerable to XSS, lack of this header makes it easily exploitable by attackers.

Recommendation:
Configure the Content-Security-Header to be sent with each HTTP response in order to apply the specific policies needed by the application.

References:
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: X-Content-Type-Options

CONFIRMED

URL	Evidence
https://www.unach.edu.ec/	Response headers do not include the X-Content-Type-Options HTTP security header

Details

Risk description:
The HTTP header `X-Content-Type-Options` is addressed to the Internet Explorer browser and prevents it from reinterpreting the content of a web page (MIME-sniffing) and thus overriding the value of the Content-Type header). Lack of this header could lead to attacks such as Cross-Site Scripting or phishing.

Recommendation:
We recommend setting the X-Content-Type-Options header such as `X-Content-Type-Options: nosniff` .

References:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: Strict-Transport-Security

CONFIRMED

URL	Evidence
https://www.unach.edu.ec/	Response headers do not include the HTTP Strict-Transport-Security header

Details

Risk description:
The HTTP Strict-Transport-Security header instructs the browser to initiate only secure (HTTPS) connections to the web server and deny any unencrypted HTTP connection attempts. Lack of this header permits an attacker to force a victim user to initiate a clear-text HTTP connection to the server, thus opening the possibility to eavesdrop on the network traffic and extract sensitive information (e.g. session cookies).

Recommendation:
The Strict-Transport-Security HTTP header should be sent with each HTTPS response. The syntax is as follows:

```
Strict-Transport-Security: max-age=<seconds>[; includeSubDomains]
```

The parameter `max-age` gives the time frame for requirement of HTTPS in seconds and should be chosen quite high, e.g. several months. A value below 7776000 is considered as too low by this scanner check.
The flag `includeSubDomains` defines that the policy applies also for sub domains of the sender of the response.

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: Referrer-Policy

CONFIRMED

URL	Evidence
https://www.unach.edu.ec/	Response headers do not include the Referrer-Policy HTTP security header as well as the <meta> tag with name 'referrer' is not present in the response.

Details

Risk description:
The Referrer-Policy HTTP header controls how much referrer information the browser will send with each request originated from the current web application.
For instance, if a user visits the web page "http://example.com/pricing/" and it clicks on a link from that page going to e.g. "https://www.google.com", the browser will send to Google the full originating URL in the `Referer` header, assuming the Referrer-Policy header is not set. The originating URL could be considered sensitive information and it could be used for user tracking.

Recommendation:
The Referrer-Policy header should be configured on the server side to avoid user tracking and inadvertent information leakage. The value `no-referrer` of this header instructs the browser to omit the Referer header entirely.

References:
https://developer.mozilla.org/en-US/docs/Web/Security/Referer_header:_privacy_and_security_concerns

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: X-XSS-Protection

CONFIRMED

URL	Evidence
https://www.unach.edu.ec/	Response headers do not include the HTTP X-XSS-Protection security header

Details

Risk description:
The `X-XSS-Protection` HTTP header instructs the browser to stop loading web pages when they detect reflected Cross-Site Scripting (XSS) attacks. Lack of this header exposes application users to XSS attacks in case the web application contains such vulnerability.

Recommendation:
We recommend setting the X-XSS-Protection header to `X-XSS-Protection: 1; mode=block`.

References:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Robots.txt file found

CONFIRMED

URL
https://www.unach.edu.ec/robots.txt

Details

Risk description:
There is no particular security risk in having a robots.txt file. However, this file is often misused by website administrators to try to hide some web pages from the users. This should not be considered a security measure because these URLs can be easily read directly from the robots.txt file.

Recommendation:
We recommend you to manually review the entries from robots.txt and remove the ones which lead to sensitive locations in the website (ex. administration panels, configuration files, etc).

References:
https://www.theregister.co.uk/2015/05/19/robotstxt/

Classification:
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Server software and technology found

UNCONFIRMED

Software / Version	Category
PHP 7.3.33	Programming languages
WordPress	CMS, Blogs
MySQL	Databases
Webpack	Miscellaneous

Open Graph	Miscellaneous
Module Federation	Miscellaneous
WPMU DEV Smush 3.12.6	WordPress plugins
Elementor 3.1.4	Page builders, WordPress plugins
Yoast SEO Premium 15.3	SEO
GSAP	JavaScript frameworks
Yoast SEO	SEO, WordPress plugins
Nginx 1.20.1	Web servers, Reverse proxies
Slider Revolution 6.6.14	Widgets, Photo galleries
Lodash 1.8.3	JavaScript libraries
LazySizes	JavaScript libraries, Performance
jQuery 1.12.4	JavaScript libraries
Google Analytics UA	Analytics
core-js 2.6.11	JavaScript libraries
Swiper	JavaScript libraries
Google Font API	Font scripts
Font Awesome 4.7.0	Font scripts

Details

Risk description:
An attacker could use this information to mount specific attacks against the identified software type and version.

Recommendation:
We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

References:
https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server.html

Classification:
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Security.txt file is missing

CONFIRMED

URL
Missing: https://www.unach.edu.ec/.well-known/security.txt

Details

Risk description:
We have detected that the server is missing the security.txt file. There is no particular risk in not creating a valid Security.txt file for your server. However, this file is important because it offers a designated channel for reporting vulnerabilities and security issues.

Recommendation:
We recommend you to implement the security.txt file according to the standard, in order to allow researchers or users report any security issues they find, improving the defensive mechanisms of your server.

References:

https://securitytxt.org/

Classification:
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Website is accessible.

Nothing was found for client access policies.

Nothing was found for use of untrusted certificates.

Nothing was found for enabled HTTP debug methods.

Nothing was found for secure communication.

Nothing was found for directory listing.

Nothing was found for domain too loose set for cookies.

Scan coverage information

List of tests performed (19/19)

- Checking for website accessibility...
- Checking for HttpOnly flag of cookie...
- Checking for Secure flag of cookie...
- Checking for missing HTTP header - X-Frame-Options...
- Checking for missing HTTP header - Content Security Policy...
- Checking for missing HTTP header - X-Content-Type-Options...
- Checking for missing HTTP header - Strict-Transport-Security...
- Checking for missing HTTP header - Referrer...
- Checking for missing HTTP header - X-XSS-Protection...
- Checking for website technologies...
- Checking for vulnerabilities of server-side software...
- Checking for client access policies...
- Checking for robots.txt file...
- Checking for absence of the security.txt file...
- Checking for use of untrusted certificates...
- Checking for enabled HTTP debug methods...
- Checking for secure communication...
- Checking for directory listing...
- Checking for domain too loose set for cookies...

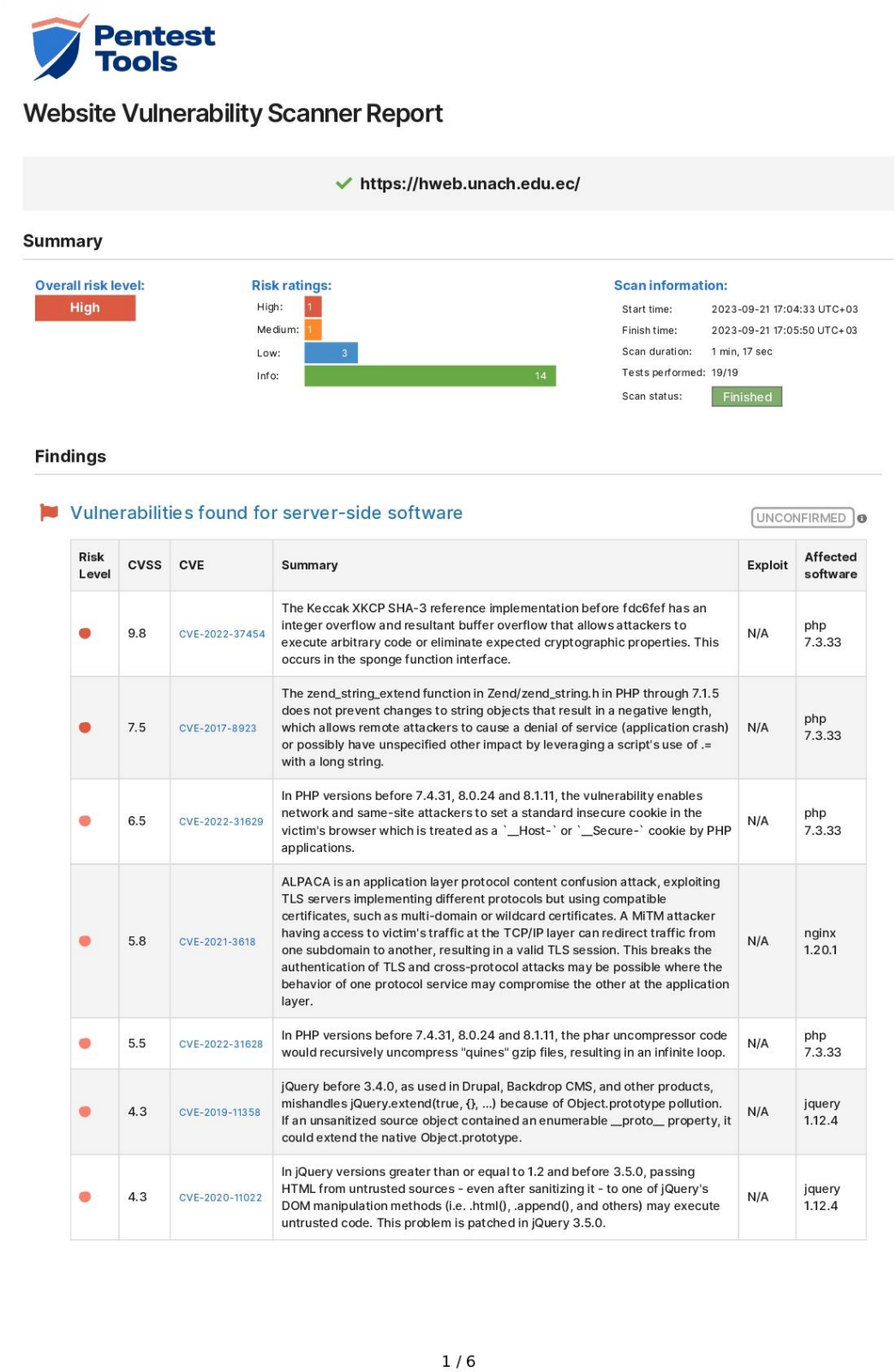
Scan parameters

Website URL:	https://www.unach.edu.ec/
Scan type:	Light
Authentication:	False

Scan stats

Unique Injection Points Detected:	1
URLs spidered:	1
Total number of HTTP requests:	9

Anexo 2: Informe de escáner realizado por pentest Tools después de aplicar el proceso Hardening.



Pentest Tools

Website Vulnerability Scanner Report

✔ https://hweb.unach.edu.ec/

Summary

Overall risk level: High

Risk ratings:

High:	1
Medium:	1
Low:	3
Info:	14

Scan information:

Start time:	2023-09-21 17:04:33 UTC+03
Finish time:	2023-09-21 17:05:50 UTC+03
Scan duration:	1 min, 17 sec
Tests performed:	19/19
Scan status:	Finished

Findings

Vulnerabilities found for server-side software

UNCONFIRMED

Risk Level	CVSS	CVE	Summary	Exploit	Affected software
	9.8	CVE-2022-37454	The Keccak XKCP SHA-3 reference implementation before fdc6fef has an integer overflow and resultant buffer overflow that allows attackers to execute arbitrary code or eliminate expected cryptographic properties. This occurs in the sponge function interface.	N/A	php 7.3.33
	7.5	CVE-2017-8923	The zend_string_extend function in Zend/zend_string.h in PHP through 7.1.5 does not prevent changes to string objects that result in a negative length, which allows remote attackers to cause a denial of service (application crash) or possibly have unspecified other impact by leveraging a script's use of .= with a long string.	N/A	php 7.3.33
	6.5	CVE-2022-31629	In PHP versions before 7.4.31, 8.0.24 and 8.1.11, the vulnerability enables network and same-site attackers to set a standard insecure cookie in the victim's browser which is treated as a `__Host-` or `__Secure-` cookie by PHP applications.	N/A	php 7.3.33
	5.8	CVE-2021-3618	ALPACA is an application layer protocol content confusion attack, exploiting TLS servers implementing different protocols but using compatible certificates, such as multi-domain or wildcard certificates. A MiTM attacker having access to victim's traffic at the TCP/IP layer can redirect traffic from one subdomain to another, resulting in a valid TLS session. This breaks the authentication of TLS and cross-protocol attacks may be possible where the behavior of one protocol service may compromise the other at the application layer.	N/A	nginx 1.20.1
	5.5	CVE-2022-31628	In PHP versions before 7.4.31, 8.0.24 and 8.1.11, the phar uncompressor code would recursively uncompress "quines" gzip files, resulting in an infinite loop.	N/A	php 7.3.33
	4.3	CVE-2019-11358	jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.	N/A	jquery 1.12.4
	4.3	CVE-2020-11022	In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.	N/A	jquery 1.12.4

1 / 6

	4.3	CVE-2020-11023	In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.	N/A	jquery 1.12.4
	4.3	CVE-2015-9251	jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.	N/A	jquery 1.12.4

Details

Risk description:
These vulnerabilities expose the affected applications to the risk of unauthorized access to confidential data and possibly to denial of service attacks. An attacker could search for an appropriate exploit (or create one himself) for any of these vulnerabilities and use it to attack the system.

Recommendation:
We recommend you to upgrade the affected software to the latest version in order to eliminate the risk of these vulnerabilities.

Classification:
CWE : CWE-1026
OWASP Top 10 - 2013 : A9 - Using Components with Known Vulnerabilities
OWASP Top 10 - 2017 : A9 - Using Components with Known Vulnerabilities

Insecure cookie setting: missing HttpOnly flag

CONFIRMED

URL	Cookie Name	Evidence
https://hweb.unach.edu.ec/	PHPSESSID	The server responded with Set-Cookie header(s) that does not specify the HttpOnly flag: Set-Cookie: PHPSESSID=e6gaj55qcd1n773nqqs31rh184

Details

Risk description:
A cookie has been set without the `HttpOnly` flag, which means that it can be accessed by the JavaScript code running inside the web page. If an attacker manages to inject malicious JavaScript code on the page (e.g. by using an XSS attack) then the cookie will be accessible and it can be transmitted to another site. In case of a session cookie, this could lead to session hijacking.

Recommendation:
Ensure that the HttpOnly flag is set for all cookies.

References:
https://owasp.org/www-community/HttpOnly

Classification:
CWE : CWE-1004
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Missing security header: X-Content-Type-Options

CONFIRMED

URL	Evidence
https://hweb.unach.edu.ec/	Response headers do not include the X-Content-Type-Options HTTP security header

Details

Risk description:
The HTTP header `X-Content-Type-Options` is addressed to the Internet Explorer browser and prevents it from reinterpreting the content of a web page (MIME-sniffing) and thus overriding the value of the Content-Type header). Lack of this header could lead to attacks such as Cross-Site Scripting or phishing.

Recommendation:
We recommend setting the X-Content-Type-Options header such as `X-Content-Type-Options: nosniff`.

References:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options

Classification:
CWE : CWE-693
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Robots.txt file found

CONFIRMED

URL
https://hweb.unach.edu.ec/robots.txt

Details

Risk description:
There is no particular security risk in having a robots.txt file. However, this file is often misused by website administrators to try to hide some web pages from the users. This should not be considered a security measure because these URLs can be easily read directly from the robots.txt file.

Recommendation:
We recommend you to manually review the entries from robots.txt and remove the ones which lead to sensitive locations in the website (ex. administration panels, configuration files, etc).

References:
https://www.theregister.co.uk/2015/05/19/robotstxt/

Classification:
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Server software and technology found

UNCONFIRMED

Software / Version	Category
PHP 7.3.33	Programming languages
WordPress	CMS, Blogs
MySQL	Databases
Webpack	Miscellaneous
Open Graph	Miscellaneous
Module Federation	Miscellaneous
WPMU DEV Smush	WordPress plugins
Elementor 3.1.4	Page builders, WordPress plugins
Yoast SEO Premium 15.3	SEO
GSAP	JavaScript frameworks
Yoast SEO	SEO, WordPress plugins
Nginx 1.20.1	Web servers, Reverse proxies
Slider Revolution 6.6.16	Widgets, Photo galleries
Lodash 1.8.3	JavaScript libraries
LazySizes	JavaScript libraries, Performance

jQuery 1.12.4	JavaScript libraries
Google Analytics UA	Analytics
core-js 2.6.11	JavaScript libraries
Swiper	JavaScript libraries
Google Font API	Font scripts
Font Awesome 4.7.0	Font scripts
HSTS	Security

Details

Risk description:
An attacker could use this information to mount specific attacks against the identified software type and version.

Recommendation:
We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

References:
https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server.html

Classification:
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Security.txt file is missing

CONFIRMED

URL
Missing: https://hweb.unach.edu.ec/.well-known/security.txt

Details

Risk description:
We have detected that the server is missing the security.txt file. There is no particular risk in not creating a valid Security.txt file for your server. However, this file is important because it offers a designated channel for reporting vulnerabilities and security issues.

Recommendation:
We recommend you to implement the security.txt file according to the standard, in order to allow researchers or users report any security issues they find, improving the defensive mechanisms of your server.

References:
https://securitytxt.org/

Classification:
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Website is accessible.

Nothing was found for client access policies.

Nothing was found for use of untrusted certificates.

Nothing was found for enabled HTTP debug methods.

Nothing was found for secure communication.

Nothing was found for directory listing.

Nothing was found for missing HTTP header - Strict-Transport-Security.

Nothing was found for missing HTTP header - Content Security Policy.

Nothing was found for missing HTTP header - X-Frame-Options.

Nothing was found for missing HTTP header - Referrer.

Nothing was found for domain too loose set for cookies.

Nothing was found for Secure flag of cookie.

Nothing was found for unsafe HTTP header Content Security Policy.

Scan coverage information

List of tests performed (19/19)

- Checking for website accessibility...
- Checking for missing HTTP header - X-Content-Type-Options...
- Checking for HttpOnly flag of cookie...
- Checking for website technologies...
- Checking for vulnerabilities of server-side software...
- Checking for client access policies...
- Checking for robots.txt file...
- Checking for absence of the security.txt file...
- Checking for use of untrusted certificates...
- Checking for enabled HTTP debug methods...
- Checking for secure communication...
- Checking for directory listing...
- Checking for missing HTTP header - Strict-Transport-Security...
- Checking for missing HTTP header - Content Security Policy...
- Checking for missing HTTP header - X-Frame-Options...
- Checking for missing HTTP header - Referrer...
- Checking for domain too loose set for cookies...
- Checking for Secure flag of cookie...
- Checking for unsafe HTTP header Content Security Policy...

Scan parameters

Website URL:	https://hweb.unach.edu.ec/
Scan type:	Light
Authentication:	False

Scan stats

Unique Injection Points Detected:	1838
URLs spidered:	2
Total number of HTTP requests:	10
Average time until a response was received:	1924ms