REPÚBLICA DEL ECUADOR

CASO DE ESTUDIO

"EJECUTAR PRUEBAS DE SEGURIDAD PARA EL SITIO WEB ENTORNO VIRTUAL DE APRENDIZAJE (EVA) DEL INSTITUTO SUPERIOR TECNOLÓGICO TENA"

MODALIDAD COMPLEXIVO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN DESARROLLO DE SOFTWARE

AUTOR: JHORDAN JESÚS HUATATOCA AVILEZ

TUTOR: ING. DARWIN NÚÑEZ

Tena - Ecuador

2025 - IS

ÍNDICE DE CONTENIDO

INDICE DE FIGURAS

INDICE DE TABLAS

APROBACIÓN DEL TUTOR

ING. DARWIN NÚÑEZ

PROFESOR DEL INSTITUTO SUPERIOR TECNOLÓGICO TENA.

CERTIFICA:

En calidad de Tutor Examen de carácter complexivo práctico denominado: **Ejecutar pruebas de seguridad para el sitio web Entorno Virtual de Aprendizaje (EVA) del Instituto Superior Tecnológico Tena**, de autoría del señor **Jhordan Jesús Huatatoca Avilez**, con CC. 1500983182 estudiante de la Carrera de Tecnología Superior en Desarrollo de Software del Instituto Superior Tecnológico Tena, CERTIFICO que se ha realizado la revisión prolija del Examen de carácter complexivo práctico antes citado, cumple con los requisitos de fondo y de forma que exigen el respectivo reglamento e institución.

Tena, 16 de julio de 2025

Ing. Darwin Núñez

TUTOR DEL EXAMEN DE CARACTER COMPLEXIVO PRÁCTICO

RESUMEN

Este estudio evalúa la seguridad del sitio web del Entorno Virtual de Aprendizaje (EVA) del Instituto Superior Tecnológico Tena, mediante la ejecución de pruebas especializadas orientadas a identificar y mitigar vulnerabilidades críticas. La investigación se centró en el análisis de amenazas comunes en aplicaciones web, como la inyección SQL (SQLi), el cross-site scripting (XSS) y los ataques de falsificación de petición en sitios cruzados (CSRF), alineándose con las recomendaciones de seguridad establecidas por el proyecto OWASP.

Para alcanzar los objetivos propuestos, se utilizaron herramientas de análisis dinámico como **Pentest-Tools** y **OWASP ZAP**, que permitieron realizar escaneos automáticos del sitio en ejecución para detectar vulnerabilidades sin necesidad de acceso al código fuente. OWASP ZAP se empleó para simular ataques y evaluar el comportamiento del sistema ante posibles explotaciones, mientras que Pentest-Tools ofreció una exploración más amplia y detallada del entorno web, incluyendo configuraciones inseguras, cabeceras HTTP ausentes y tecnologías desactualizadas.

Los hallazgos fueron documentados minuciosamente y se establecieron medidas correctivas específicas para cada vulnerabilidad identificada. Como resultado, se propuso un protocolo de pruebas de seguridad que puede ser replicado en futuras versiones del sistema, fortaleciendo la postura de seguridad institucional y fomentando una cultura de desarrollo responsable y protección digital en entornos educativos.

Palabras clave: Seguridad web, EVA, Pentest-Tools, OWASP ZAP, XSS, CSRF, SQL Injection, pruebas de penetración, análisis dinámico, plataformas educativas seguras.

ABSTRACT

This study evaluates the security of the Virtual Learning Environment (EVA) website of the Instituto Superior Tecnológico Tena through the execution of specialized tests aimed at identifying and mitigating critical vulnerabilities. The research focused on analyzing common web application threats such as SQL Injection (SQLi), Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF), in alignment with the security guidelines established by the OWASP project.

To achieve the proposed objectives, dynamic analysis tools such as Pentest-Tools and OWASP ZAP were used. These tools enabled automatic scanning of the live site to detect vulnerabilities without requiring access to the source code. OWASP ZAP was utilized to simulate attacks and assess the system's behavior under potential exploitation, while Pentest-Tools provided a broader and more detailed scan of the web environment, including insecure configurations, missing HTTP security headers, and outdated technologies.

The findings were thoroughly documented, and specific corrective actions were proposed for each identified vulnerability. As a result, a security testing protocol was developed that can be replicated in future versions of the system, strengthening the institution's security posture and promoting a culture of secure development and digital protection in educational environments.

Keywords: Web security, EVA, Pentest-Tools, OWASP ZAP, XSS, CSRF, SQL Injection, penetration testing, dynamic analysis, secure educational platforms.

Revieward by

Carolina Romero A.

BA. Carolina Romero

Language Center Professor

1 INTRODUCCIÓN

En la actualidad, el uso de plataformas web en entornos educativos ha adquirido una gran relevancia, permitiendo a instituciones y estudiantes gestionar contenidos, recursos y actividades de forma remota y eficiente. Sin embargo, este crecimiento también ha traído consigo nuevos desafíos, particularmente en lo que respecta a la seguridad de la información. Las aplicaciones web, como el Entorno Virtual de Aprendizaje (EVA) del Instituto Superior Tecnológico Tena, manejan datos sensibles de estudiantes, docentes y administrativos, lo que las convierte en objetivos potenciales de ciberataques si no se implementan mecanismos de protección adecuados.

La presencia de vulnerabilidades como la inyección SQL (SQL Injection), el Cross-Site Scripting (XSS) o la falsificación de petición en sitios cruzados (CSRF) puede comprometer seriamente la confidencialidad, integridad y disponibilidad de la información. A pesar de la existencia de estándares y herramientas que permiten mitigar estos riesgos, muchas instituciones aún no aplican procesos sistemáticos de pruebas de seguridad, dejando expuestas sus plataformas a ataques que podrían prevenirse con prácticas adecuadas.

Ante esta problemática, esta investigación busca responder a la siguiente pregunta: ¿Cómo se pueden identificar y mitigar las vulnerabilidades más comunes en el sitio web EVA mediante la implementación de pruebas de seguridad y herramientas de análisis especializadas?

1.1 OBJETIVOS ESPECÍFICOS

- Identificar las vulnerabilidades frecuentes en aplicaciones web, enfocándose en XSS, SQL Injection y CSRF.

- Aplicar herramientas de análisis estático como OWASP ZAP para realizar pruebas de seguridad sobre el sitio EVA.
- Documentar los hallazgos y proponer soluciones correctivas que fortalezcan la seguridad del sistema.

En este estudio se utilizarán herramientas reconocidas en el ámbito de la ciberseguridad. OWASP ZAP facilitará pruebas de penetración automatizadas para evaluar la resistencia del sistema ante ataques externos. La combinación de estas herramientas permitirá obtener una visión completa del estado actual de la seguridad del sistema.

El objetivo general de esta investigación es fortalecer la seguridad del Entorno Virtual de Aprendizaje (EVA) mediante la ejecución de pruebas técnicas que permitan identificar vulnerabilidades, aplicar correcciones y establecer un protocolo que sirva como referencia para futuras auditorías.

La importancia de este estudio radica en su aporte a la protección de los datos y al fortalecimiento de la confianza en el uso de plataformas educativas digitales. Al aplicar un enfoque basado en estándares y herramientas especializadas, se busca no solo resolver vulnerabilidades actuales, sino también fomentar una cultura institucional de desarrollo seguro y prevención de riesgos cibernéticos.

2 ANÁLISIS

El presente análisis se fundamenta en la descomposición del problema de seguridad web en sus componentes esenciales, permitiendo identificar vulnerabilidades críticas en el sitio web del Entorno Virtual de Aprendizaje (EVA) del Instituto Superior Tecnológico Tena. Para garantizar la fiabilidad, se aplicaron metodologías y herramientas alineadas con los principios del modelo ISO/IEC 27001, especialmente en lo relacionado con la seguridad y la mantenibilidad del software.

El enfoque metodológico fue descriptivo y aplicado, combinando técnicas de análisis estático y dinámico para la detección de vulnerabilidades como inyección SQL (SQL Injection), secuencias de comandos en sitios cruzados (XSS) y falsificación de solicitudes en sitios cruzados (CSRF). Estas pruebas fueron ejecutadas con el objetivo de diagnosticar fallas de seguridad frecuentes en aplicaciones desarrolladas en PHP.

2.1 Herramientas Utilizadas

Para lograr un análisis integral, se utilizaron herramientas automáticas y manuales que permiten evaluar tanto el código fuente como el comportamiento del sistema en ejecución. Además, para obtener una evaluación integral, se emplearon tanto herramientas técnicas como encuestas a usuarios:

2.1.1 Recolección de datos cualitativos

Se aplicaron encuestas estructuradas a usuarios de la plataforma EVA (Entorno Virtual de Aprendizaje) a través de la plataforma digital como WhatsApp, lo cuales son los estudiantes del Instituto Superior Tecnológico Tena (ISTT). Se obtuvo un total de 53 respuestas, lo que permitió recopilar datos sobre la percepción de seguridad en páginas web.

- **Resultados de encuesta sobre la Usabilidad en Aplicaciones Móviles**

Figura 1

Pregunta N°1 – Conocimiento sobre seguridad en páginas web

Nota: La mayoría de los encuestados mencionan conocer sobre la seguridad de páginas web.

Figura 2

Pregunta N°2 – Consecuencia de una página web insegura

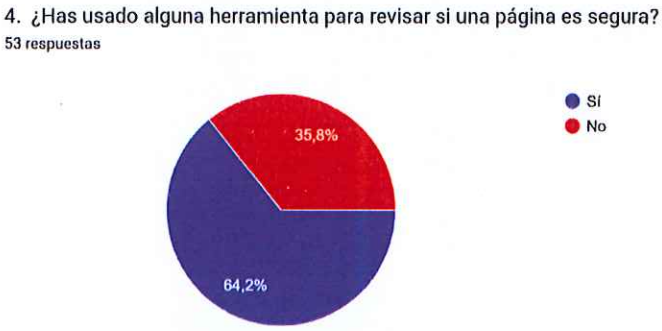
Figura 3

Pregunta N°3 – Robo de información

Nota: El 94,3% de los encuestado tiene conocimiento sobre qué sucedería si alguna página que utiliza frecuentemente se viera afectado por su seguridad deficiente.

Figura 4

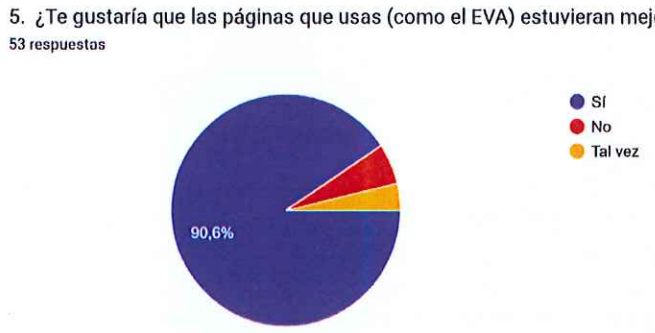
Pregunta N°4 – Herramientas de seguridad



Nota: El 64,2% utiliza alguna herramienta para verificar su seguridad antes de ingresar sus datos personales y así evitar el riesgo de ser afectados por robo de información.

Figura 5

Pregunta N°5 – Seguridad en el EVA



Nota: El 90,6% de los encuestados mostraron una respuesta positiva a mejoras para el Entorno Virtual de Aprendizaje (EVA). Sin embargo, el 5,7% se muestran inconformes de implementar nuevas formas de seguridad y el 3,8% están indecisos ante tal pregunta.

Figura 6

Pregunta N°6 – Precaución ante páginas que manejan información personal

Nota: El 94,3% de respuestas mencionan que revisan la página antes de ingresar información delicada en la misma. Mientras tanto, el 3,8% y 1,9% menciona que no les parece relevante revisar si una página es segura para la protección de sus datos

Figura 7

Pregunta N°7 – Páginas no seguras

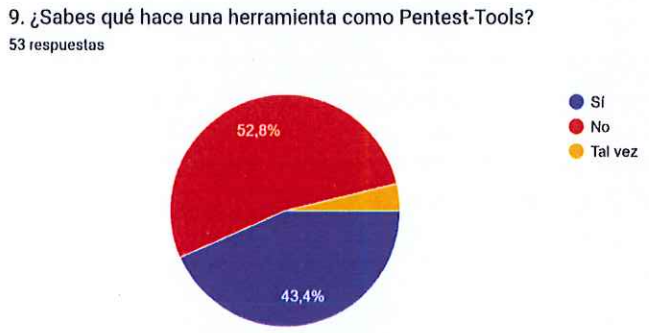
Figura 8

Pregunta N°8 – Protección de páginas web

Nota: El 56,6% de respuestas están interesados en aprender diferentes formas de proteger una página web. Por otro lado, el 39,6% aún no están decididos si aprender o no sobre la protección de páginas web, el resto 3,8% están completamente decididos en no aprender.

Figura 9

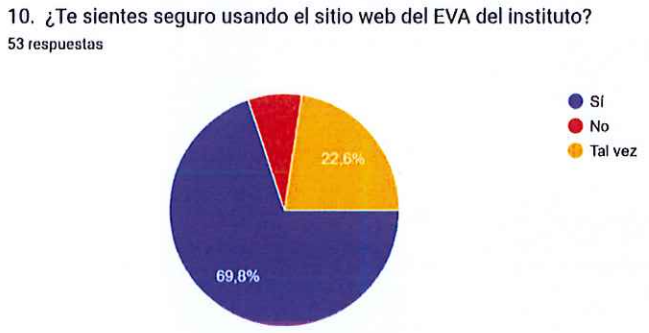
Pregunta N°9 – Herramienta Pentest-Tools



Nota: La mayor parte de los encuestados, es decir el 52,8% mencionan desconocer de dicha herramienta. Sin embargo, el 43,4% están seguros de conocer la herramienta Pentest-Tools.

Figura 10

Pregunta N°10 – Seguridad en el EVA del ISTTena



Nota: El 69,8% de respuesta mencionan que sí se sienten seguros al usar la plataforma EVA. Mientras tanto, el 22,6% se siente inseguros de usar la plataforma. Por otro lado, el 7,5% de los encuestados no se sienten seguros al usar la plataforma antes mencionada.

2.1.2 Identificación de vulnerabilidades de forma manual

Se realizaron pruebas manuales sobre los formularios públicos del sistema EVA del Instituto Superior Tecnológico Tena. Estas pruebas se centraron en vulnerabilidades comunes como Cross-Site Scripting (XSS), Inyección SQL y CSRF, sin el uso de herramientas automáticas.

a) XSS

Se inyectó el siguiente código en el campo de búsqueda del EVA:

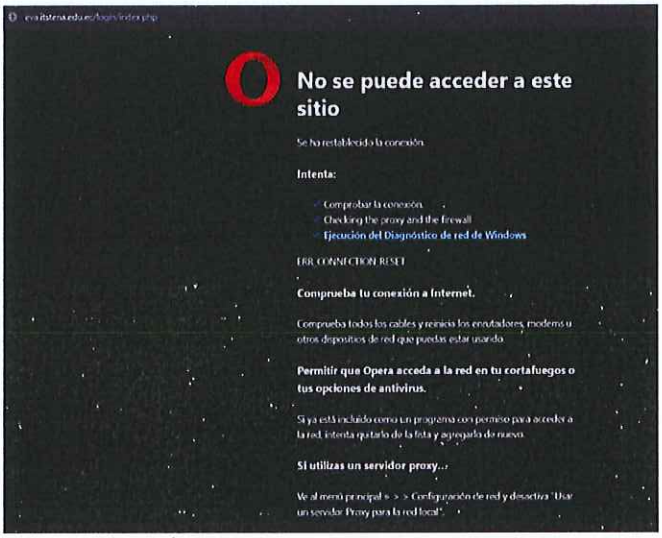
<script>alert("XSS")</script>

Figura 11

Inyección del código XSS

Figura 12

Rechazo de la página al script



Nota: La inserción del script genera una respuesta de rechazo por parte del sistema.

Por lo que evidencia que la aplicación no es vulnerable a ataques de XSS reflejado. Esto indica que existen mecanismos de validación que detectan y neutralizan entradas potencialmente maliciosas.

b) SQL Injection

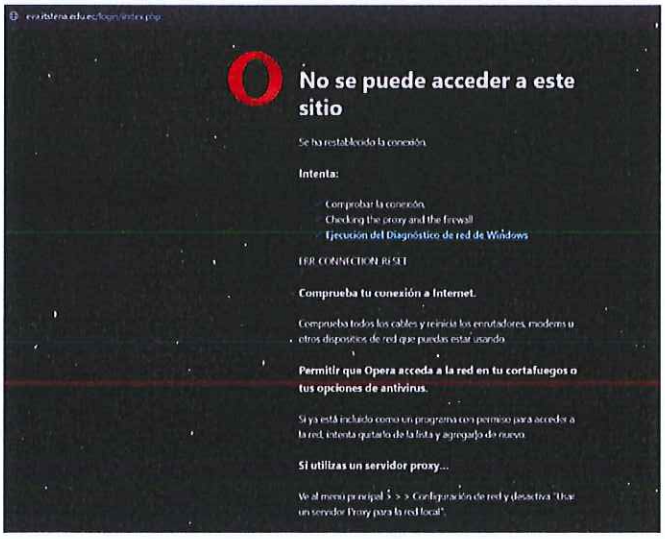
Se probó el siguiente código “payload ' OR '1'='1” en el formulario de login del sistema EVA.

Figura 13

Inyección SQL

Figura 14.

Rechazo a la inyección SQL



Nota: El sitio web EVA no permite la ejecución de inyecciones SQL.

El sitio bloquea el acceso cuando se intenta introducir código malicioso. Esto indica que la protección no reside únicamente en el código de la aplicación, sino que también está implementada a nivel externo, como un WAF (Firewall de Aplicaciones Web) o un firewall del servidor.

c) CSRF

Se analizó el formulario de inicio de sesión de los usuarios. Se comprobó que, si incluía tokens de seguridad, lo que no lo hace susceptible a ataques CSRF.

Figura 15

Código fuente del formulario de inicio de sesión con token CSRF visible

Nota: El código fuente de la página exhibe claramente el token anti-CSRF.

Esto indica que la aplicación implementa mecanismos efectivos para protegerse contra ataques CSRF, contribuyendo así a la seguridad del entorno de trabajo EVA.

2.1.3 Identificación de vulnerabilidades de forma automática

2.1.3.1 Herramientas utilizadas

Tabla 1

Herramientas que se utilizaron para los dos diferentes análisis

HERRAMIENTA	*TIPO DE ANÁLISIS*	*OBJETIVO PRINCIPAL*	*VERSIÓN UTILIZADA*
OWASP ZAP	*Análisis dinámico (DAST)*	*Encontrar vulnerabilidades como XSS, CSRF y SQL Injection mediante la ejecución dinámica sobre la web.*	*2.14.0*
PENTEST-TOOLS	*Análisis dinámico (DAST)*	*Detectar de forma automática las vulnerabilidades más frecuentes presentes en el sitio web analizado.*	*Plataforma online actualizada (versión web, julio 2025)*

a) Análisis general con la herramienta OWASP ZAP

Se utilizó la herramienta OWASP ZAP, una solución automatizada que permite realizar pruebas de penetración mediante análisis activo de sitios web.

Durante la simulación de ataque al entorno http://eva.itstena.edu.ec, se ejecutaron múltiples pruebas de seguridad, los más relevantes son:

- Inyección SQL
- Exploración de directorios (Directory Browsing)
- Inyección de código del lado del servidor (Server Side Code Injection)
- Inyección XPath y XML
- Manipulación de parámetros (Parameter Tampering)
- Redirección externa y ejecución remota de comandos (RCE)
- Inyeccion CRLF

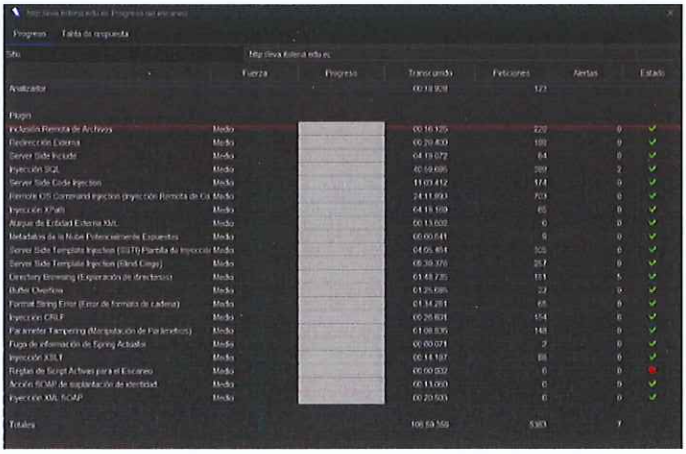
Tabla 2

Términos y Siglas

Nombre Técnico en Español	*Sigla (si aplica)*	*Nombre Técnico en Inglés*	*Descripción*
Inyección SQL	**SQLi**	SQL Injection	Permite ejecutar consultas maliciosas sobre bases de datos.
Exploración de directorios	—	Directory Browsing	Expone archivos o carpetas no protegidas del servidor.
Inyección de código del lado del servidor	—	Server-Side Code Injection	Inserta código en el servidor que puede ser ejecutado.
Inyección XPath y XML	**XPathi / XMLi**	XPath Injection / XML Injection	Manipula consultas XPath/XML para acceder a datos.
Manipulación de parámetros	—	Parameter Tampering	Cambia valores de parámetros en URLs o formularios para alterar el comportamiento.
Redirección externa y ejecución remota de comandos	**RCE**	Remote Code Execution	Permite ejecutar comandos maliciosos de forma remota.
Inyección CRLF (Retorno de Carro y Salto de Línea)	**CRLFi**	CRLF Injection	Es una vulnerabilidad web que permite a un atacante inyectar caracteres especiales de salto de línea (`\r\n`) en las respuestas HTTP del servidor

Figura 16

Progreso del escaneo con detección de múltiples pruebas realizadas



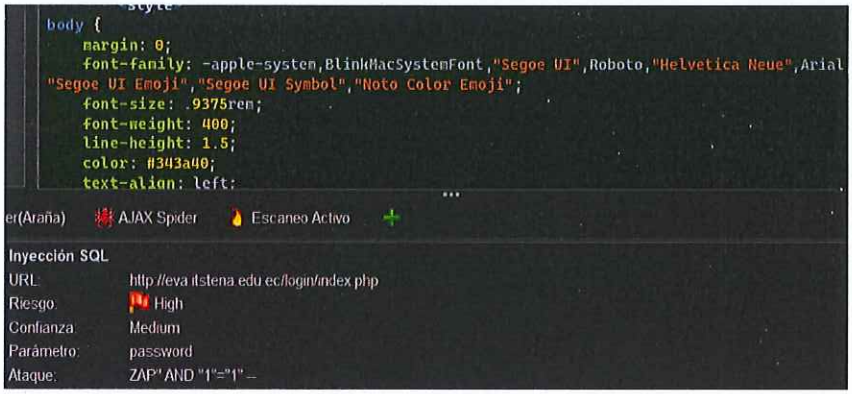
Nota: Pruebas generales automáticas que contiene la herramienta OWASP ZAP

b) Análisis de seguridad de inyección SQL

Durante el escaneo activo realizado con OWASP ZAP, se identificó una vulnerabilidad de Inyección SQL en el parámetro "password" del formulario de autenticación "(/login/index.php)" de la plataforma evaluada.

Figura 17

Vulnerabilidad de inyección SQL



Nota: Alerta de SQLi generada por OWASP ZAP.

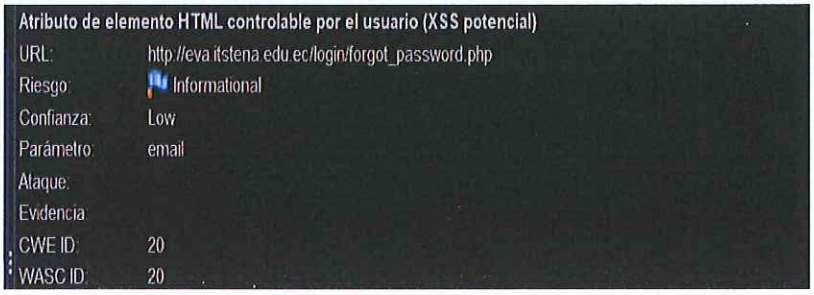
Se evidencia que el sistema es vulnerable al ataque de inyección SQL con una cadena simple que manipula la lógica del inicio de sesión.

c) Análisis de seguridad de XSS

Durante el escaneo activo con OWASP ZAP, se detectó una vulnerabilidad de Cross-Site Scripting (XSS) en el parámetro "email" de la URL "(eva.itstena.edu.ec/login/forgot_password.php)" con una confianza baja.

Figura 18

Vulnerabilidad de XSS en el parámetro



Nota: Comprobación examina la entrada proporcionada por el usuario en parámetros de cadenas.

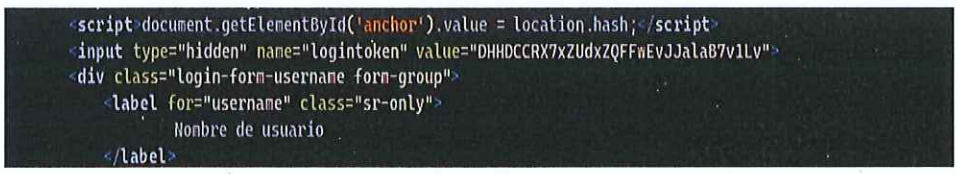
Las cadenas están formadas de consulta y datos POST para identificar dónde podrían estar controlados ciertos valores de atributos HTML. Esto proporciona detección de puntos calientes para XSS (cross-site scripting).

d) Análisis de seguridad de CSRF

Durante el escaneo activo con OWASP ZAP, se localizó el token anti-CSRF lo que significa que tiene seguridad, por otro lado, en otros formularios, no se encuentra el token anti-CSRF lo cual puede tener riesgo de vulnerabilidad hacia el sistema.

Figura 19

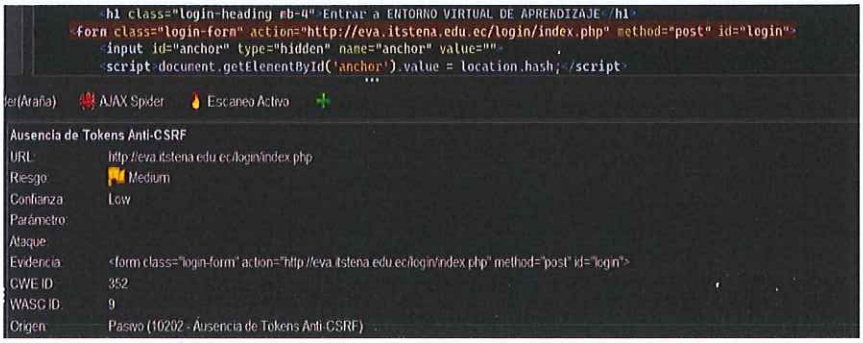
Token anti-CSRF



Nota: *El token anti-CSRF se encuentra nombrada "logintoken" en la línea de código.*

Figura 20

Ausencia de Tokens Anti-CSRF



Nota: No existe ningún tipo de Token Anti-CSRF.

2.1.4 OWASP ZAP

OWASP ZAP (Zed Attack Proxy) es una herramienta de código abierto desarrollada por el proyecto OWASP para realizar pruebas de seguridad en aplicaciones web. Está diseñada tanto para principiantes como para expertos en seguridad, y permite detectar automáticamente vulnerabilidades comunes como XSS, inyección SQL, CSRF, entre otras. Su interfaz gráfica, funciones automatizadas y herramientas manuales lo convierten en una de las soluciones más populares para realizar auditorías de seguridad web.

- **Características**

Tabla 3

Características principales de OWASP ZAP

Característica	*Descripción*
CÓDIGO ABIERTO Y GRATUITO	*Disponible para todos sin costo, bajo licencia libre.*
ANÁLISIS AUTOMATIZADO	*Detecta vulnerabilidades comunes en aplicaciones web sin intervención manual.*
PROXY DE INTERCEPCIÓN	*Permite interceptar y modificar tráfico HTTP/HTTPS entre el navegador y el servidor.*
ESCANEO ACTIVO Y PASIVO	*Escanea tanto de forma pasiva (sin alterar el sistema) como activa (con pruebas).*
HERRAMIENTAS MANUALES	*Incluye utilidades como fuzzing, fuerza bruta, escaneo de puertos, etc.*
INFORMES DETALLADOS	*Genera reportes con las vulnerabilidades encontradas, su riesgo y recomendaciones.*

- **Relación**

Tabla 4

Relación de OWASP ZAP con ISO/IEC 27001

Control ISO 27001	*Descripción*	*Relación con OWASP ZAP*
A.12.6.1	*Gestión de vulnerabilidades técnicas*	*OWASP ZAP identifica y reporta vulnerabilidades web comunes.*
A.14.2.1	*Principios de desarrollo seguro*	*OWASP ZAP valida que el desarrollo cumple buenas prácticas.*
A.18.2.3	*Evaluación técnica del sistema*	*OWASP ZAP realiza pruebas automatizadas para evaluar seguridad.*

- **Aplicación con este estudio**

OWASP ZAP se utilizó para identificar vulnerabilidades de seguridad en el sitio web del Entorno Virtual de Aprendizaje (EVA), mediante el escaneo automático y manual de componentes críticos. La herramienta permitió detectar fallos como inyecciones SQL, XSS y CSRF, así como analizar encabezados HTTP y configuraciones inseguras, proporcionando información clave para proponer mejoras en la protección de la aplicación.

2.1.5 Pentest-Tools

Pentest-Tools es una plataforma en línea que permite analizar la seguridad de páginas web y servidores mediante herramientas automáticas. No necesita instalación y es ideal tanto para profesionales de ciberseguridad como para usuarios que quieren detectar vulnerabilidades en su sitio web.

- **Características**

Tabla 5

Características de Pentest - Tools

CARACTERISTICAS	*DESCRIPCION*
WEBSITE VULNERABILITY SCANNER	*Analiza automáticamente una página web para encontrar fallos como XSS, SQL Injection, y otros riesgos comunes. Ideal para un análisis completo.*
CMS DETECTION	*Detecta si el sitio utiliza sistemas como WordPress, Joomla o Drupal, y verifica si están actualizados o expuestos a vulnerabilidades.*
SUBDOMAIN FINDER	*Encuentra subdominios ocultos que podrían representar puertas traseras o sistemas olvidados. Útil para reconocimiento web.*
SSL/TLS SCANNER	*Verifica el estado de seguridad del certificado SSL de la página, comprobando cifrado, fechas de validez y configuración segura del HTTPS.*

- **Relación**

Tabla 6

Relación de Pentest-Tools con ISO/IEC 27001

CONTROL ISO 27001	*DESCRIPCIÓN*	*RELACIÓN CON PENTEST-TOOLS*
A.12.6.1	*Gestión de vulnerabilidades técnicas*	*Permite escanear sitios web para detectar vulnerabilidades.*
A.18.2.3	*Evaluación técnica del sistema*	*Realiza evaluaciones externas simulando ataques reales.*
A.13.1.1	*Protección de redes*	*Identifica fallos en protocolos, puertos y configuración web.*

- **Aplicación en este estudio**

Con el fin de complementar el análisis de vulnerabilidades, se utilizó la plataforma Pentest-Tools, una solución basada en la web que permite ejecutar pruebas de seguridad automatizadas en aplicaciones web. Esta herramienta permite realizar escaneos sin necesidad de instalación local, facilitando la evaluación rápida de sistemas accesibles públicamente.

2.2 Análisis de vulnerabilidades del sistema EVA utilizando una herramienta automatizada como Pentest-Tools.

Esta sección describe el análisis de seguridad realizado al entorno web del Entorno Virtual de Aprendizaje (EVA) del Instituto Superior Tecnológico Tena, utilizando la herramienta automatizada Pentest-Tools. El objetivo fue identificar vulnerabilidades frecuentes, tales como inyecciones SQL, ataques XSS, ausencia de tokens CSRF y exposición de cabeceras HTTP inseguras. Estos hallazgos permiten evaluar el nivel de riesgo al que está expuesto el sistema, así como detectar malas configuraciones o prácticas que podrían comprometer la seguridad, disponibilidad e integridad del sitio web.

Pentest-Tools permitió realizar un escaneo externo del sistema EVA simulando ataques reales sin necesidad de acceso al código fuente, generando un reporte técnico con recomendaciones específicas para mitigar los problemas detectados.

2.2.1 Herramienta utilizada: Pentest-Tools

Pentest-Tools es una plataforma en línea para la realización de pruebas de penetración automatizadas, diseñada para identificar vulnerabilidades de seguridad en aplicaciones web sin necesidad de acceder al código fuente. Esta herramienta facilita la detección de fallos mediante escaneos externos, emulando ataques reales que permiten evaluar el nivel de exposición de un sitio frente a amenazas comunes.

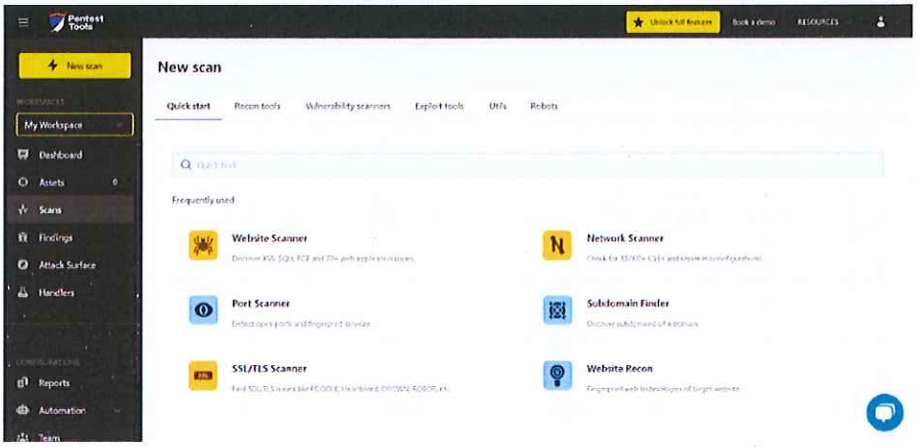
Entre las vulnerabilidades que puede detectar se encuentran:

- Inyección SQL (SQLi).
- Cross-Site Scripting (XSS).
- Falta de protección contra ataques CSRF.
- Cabeceras HTTP inseguras o mal configuradas.
- Exposición de directorios o archivos sensibles.
- Fallas relacionadas con la seguridad del servidor o configuración del sitio.

Su facilidad de uso y generación automática de informes técnicos la convierten en una herramienta ideal para evaluaciones preliminares de seguridad en plataformas web como el sistema EVA.

Figura 21

Dashboard de la herramienta Pentest-tools

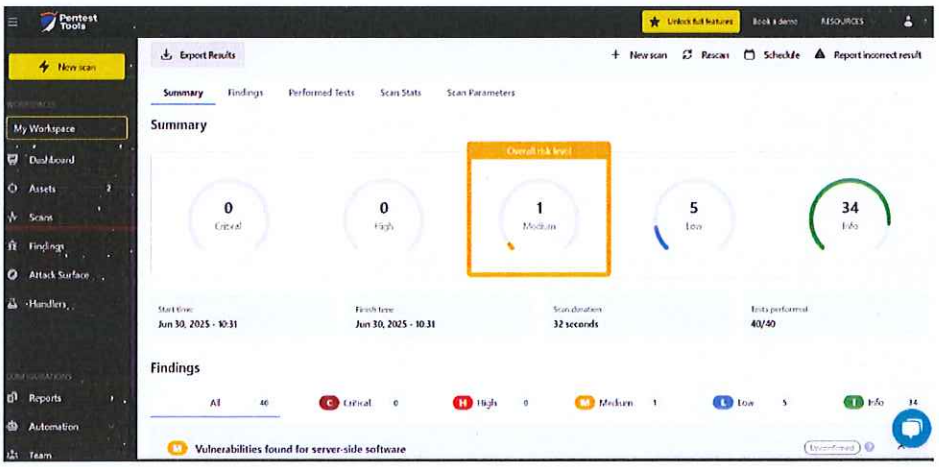


2.2.2. Hallazgos de vulnerabilidades con la herramienta Pentest-tools

Durante el análisis del entorno EVA utilizando Pentest-Tools, se detectaron dos vulnerabilidades de tipo Cross-Site Scripting (XSS) en la librería Bootstrap v3.3.7.

Figura 22

Vulnerabilidades detectadas con la herramienta Pentest-tools



I. Vulnerabilidades nivel de riesgo medio

Figura 23

Primera vulnerabilidad media detallada por la herramienta Pentest-tools

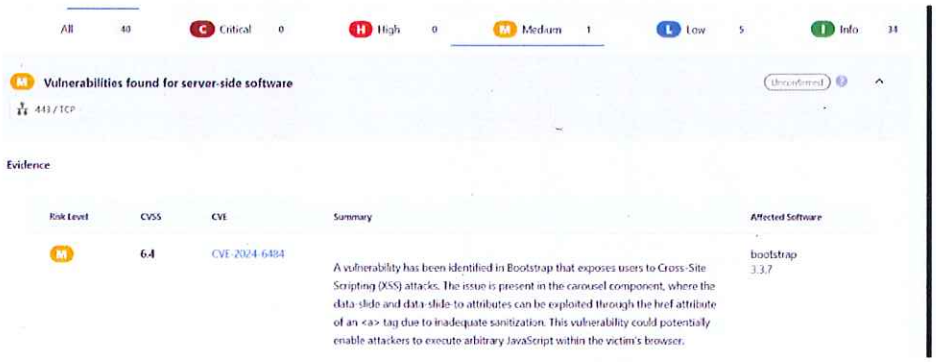


Figura 24

Segunda vulnerabilidad media detallada por la herramienta Pentest-tools

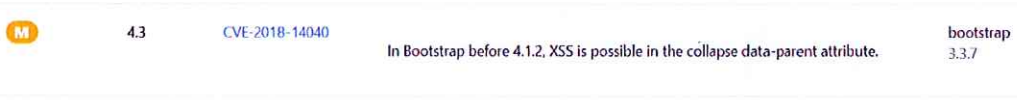


Tabla 7

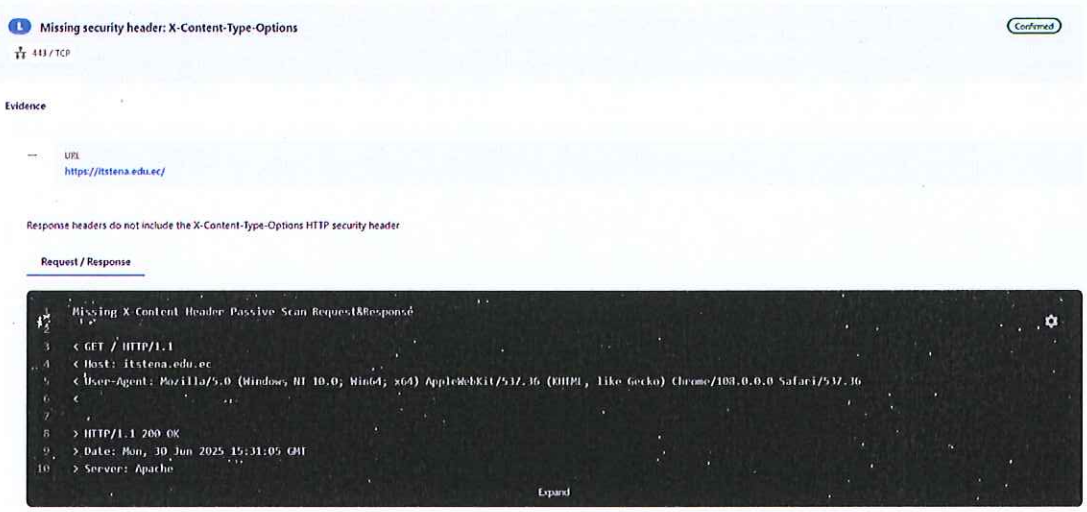
Vulnerabilidades nivel de riesgo medio

Vulnerabilidad	*¿Qué es?*	*Consecuencias*
Cross-Site Scripting (XSS)	*Es una vulnerabilidad que permite al atacante inyectar scripts maliciosos en páginas web.*	*Robo de información del usuario, redirección a sitios maliciosos o ejecución de acciones sin autorización.*
Uso de Bootstrap 3.3.7	*Se trata de una versión antigua de la librería que contiene errores de seguridad conocidos, como XSS.*	*Facilita la explotación de vulnerabilidades si no se actualiza; permite la ejecución de código malicioso.*

II. Vulnerabilidades nivel de riesgo bajo

Figura 25

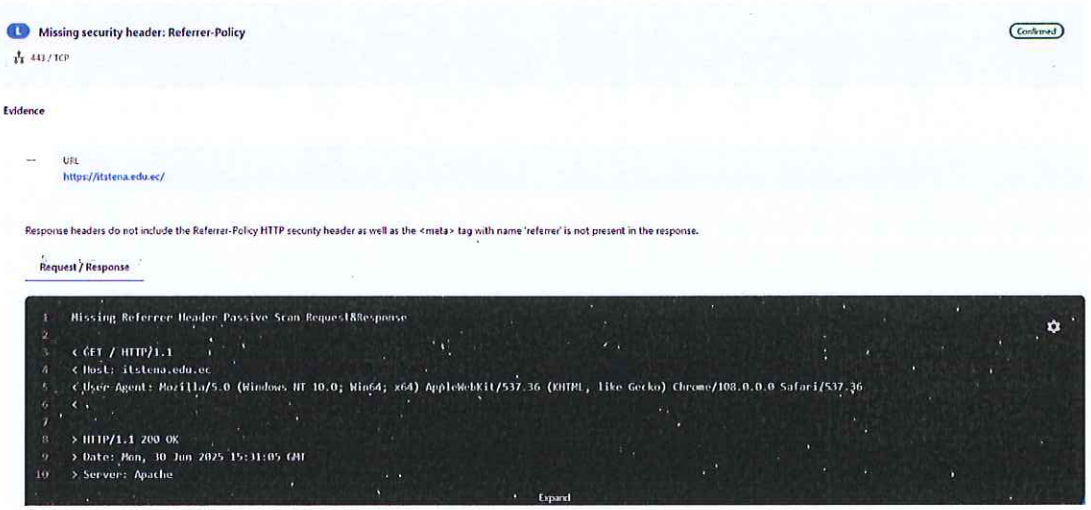
Primera vulnerabilidad nivel de riesgo bajo



Nota: No incluyen el encabezado de seguridad HTTP X-Content-Type-Options

Figura 26

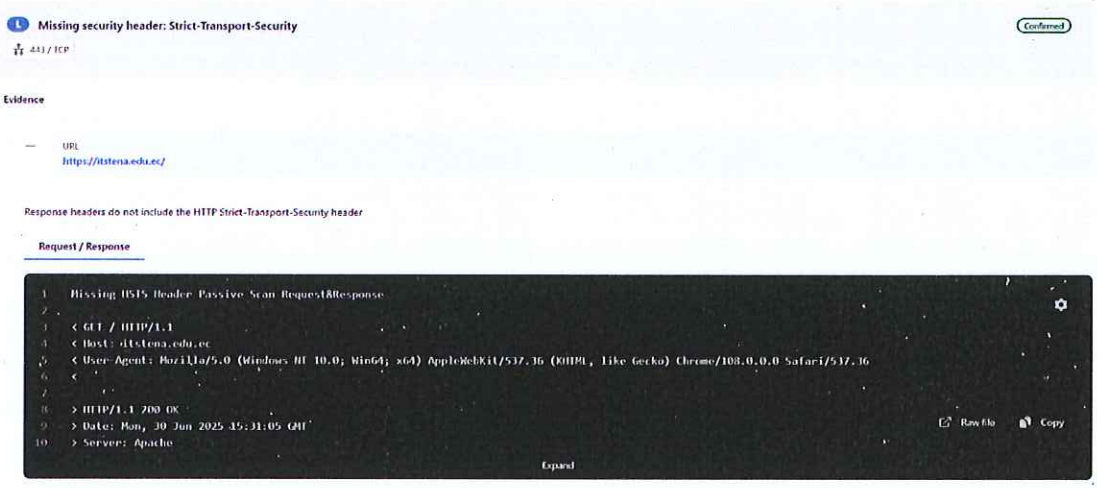
Segunda vulnerabilidad nivel de riesgo bajo



Nota: No incluyen el encabezado de seguridad HTTP Referrer-Policy y la etiqueta <meta> con el nombre 'referrer' no está presente en la respuesta.

Figura 27

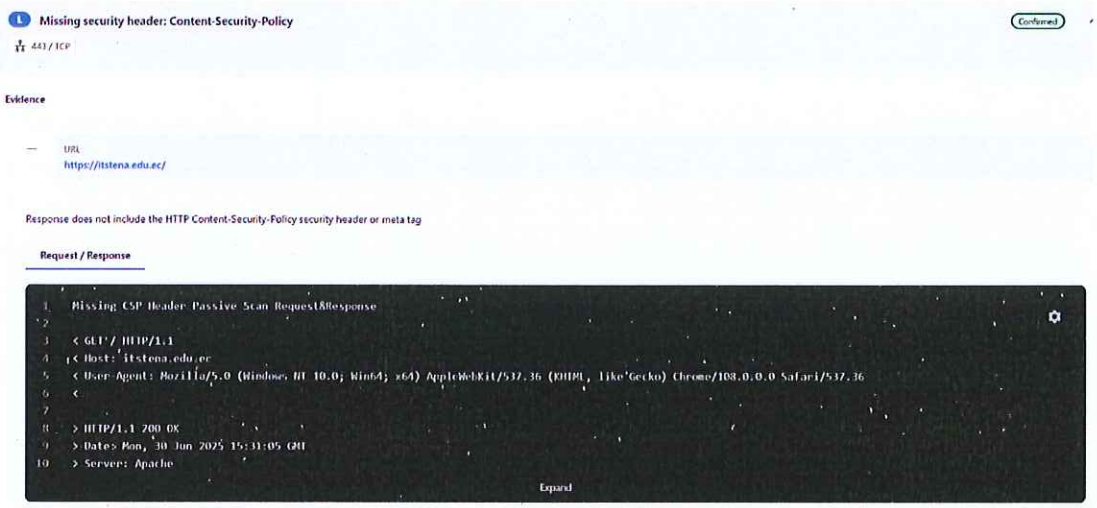
Tercera vulnerabilidad nivel de riesgo bajo



Nota: No incluyen el encabezado HTTP Strict-Transport-Security

Figura 28

Cuarta vulnerabilidad nivel de riesgo bajo



Nota: No incluye el encabezado de seguridad HTTP Content-Security-Policy ni la metaetiqueta

Tabla 8

Vulnerabilidades encontradas con la herramienta Pentest-tools

Nº	*VULNERABILIDAD*	*¿QUÉ ES?*	*CONSECUENCIA POTENCIAL*
1	***Falta de X-Content-Type-Options***	*El servidor no evita que el navegador intente adivinar el tipo de archivo (MIME).*	*Puede ejecutar archivos como scripts cuando no debería, permitiendo ataques como XSS.*
2	***Falta de Referrer-Policy***	*No se indica cómo manejar la información del sitio anterior al seguir un enlace.*	*Puede filtrar URLs sensibles o tokens al sitio de destino, afectando la privacidad.*
3	***Falta de Strict-Transport-Security (HSTS)***	*No se obliga al navegador a usar HTTPS.*	*Podría permitir ataques tipo "man-in-the-middle" si el usuario accede por HTTP.*
4	***Falta de Content-Security-Policy (CSP)***	*No se define qué scripts o recursos son seguros de cargar.*	*Aumenta el riesgo de XSS, ya que no se limita qué contenido externo se puede ejecutar.*

3 PROPUESTA

Como resultado del análisis de seguridad realizado al Entorno Virtual de Aprendizaje (EVA) del Instituto Superior Tecnológico Tena, se ha identificado la presencia de vulnerabilidades de nivel bajo y medio, especialmente relacionadas con configuraciones inadecuadas en las cabeceras HTTP de seguridad, uso de librerías desactualizadas (como Bootstrap 3.3.7) y la ausencia de mecanismos de defensa contra ataques comunes como XSS, CSRF e Inyección SQL.

Estas fallas, aunque no críticas en su mayoría, representan una puerta de entrada potencial para atacantes que busquen comprometer la confidencialidad, integridad y disponibilidad del sistema y los datos de sus usuarios. En este contexto, se propone un conjunto de acciones estructuradas orientadas a mejorar significativamente la seguridad del sitio web EVA y reducir su superficie de ataque.

3.1 Objetivo general de la propuesta

Implementar un conjunto de medidas técnicas, organizacionales y formativas que fortalezcan la seguridad del sitio EVA, enfocándose en prevenir ataques web comunes, proteger la información de los usuarios y mejorar la calidad del sistema conforme a buenas prácticas de ciberseguridad.

3.2 Acciones Propuestas

I. Implementación de cabeceras HTTP de seguridad

Se detectó la ausencia de varias cabeceras de seguridad esenciales. Se propone configurar en el servidor web las siguientes:

- X-Content-Type-Options: nosniff :

Evita que el navegador adivine el tipo de contenido y reduzca el riesgo de XSS.

- Strict-Transport-Security (HSTS):

Obliga a que los navegadores solo se conecten mediante HTTPS, evitando ataques tipo Man-in-the-Middle.

- Content-Security-Policy (CSP):

Define qué contenido puede cargarse en la página, limitando la ejecución de scripts no autorizados.

- Referrer-Policy:

Controla qué información se envía al seguir enlaces, protegiendo URLs internas.

Estas cabeceras se deben añadir al archivo de configuración del servidor para que acompañen cada respuesta HTTP emitida por la aplicación.

II. Sustitución de librerías vulnerables

Se encontró que EVA utiliza Bootstrap v3.3.7, una versión antigua con vulnerabilidades conocidas de tipo XSS. Se recomienda:

- Actualizar a Bootstrap 5 o superior, versiones que han corregido las vulnerabilidades identificadas.
- Realizar pruebas funcionales tras la migración para asegurar que la interfaz mantenga su comportamiento esperado.
- Validar que las nuevas versiones sean compatibles con navegadores y dispositivos utilizados por los estudiantes y docentes.

III. Integración de pruebas automatizadas con Pentest-Tools

Para mantener una vigilancia constante, se propone realizar escaneos periódicos utilizando la herramienta Pentest-Tools, aprovechando su capacidad de detectar vulnerabilidades como:

- Inyección SQL.
- Cross-Site Scripting (XSS).
- Falta de configuraciones seguras en el servidor.
- Exposición de archivos sensibles.

Estos escaneos deben programarse al menos una vez cada tres meses y ser ejecutados por personal autorizado. Sus reportes deben ser documentados y revisados con el equipo de desarrollo.

IV. Implementación de validaciones en formularios

Aunque no se analizó directamente el código fuente, se recomienda asegurar que todos los formularios:

- Utilicen tokens CSRF (si la tecnología lo permite).
- Tengan validación tanto del lado del cliente como del servidor.
- Restringan correctamente los caracteres ingresados, especialmente en campos como búsqueda, nombre, comentarios, etc.
- Estas medidas evitarán que usuarios maliciosos puedan inyectar comandos o scripts en el sistema.

V. Respaldo periódico del sistema y control de versiones

Se sugiere:

- Implementar un sistema de copias de seguridad (backups) semanales del sistema y su base de datos.
- Usar un repositorio con control de versiones (como Git) para tener trazabilidad de los cambios realizados en el sistema EVA.

Esto permitirá restaurar rápidamente el sistema ante fallos o ataques, y mantener un registro claro de su evolución.

3.3 Beneficios de la propuesta

- Reducción significativa del riesgo de ataques XSS, CSRF e inyección SQL.
- Mayor protección de la información académica y personal de estudiantes y docentes.
- Cumplimiento de buenas prácticas de desarrollo seguro.
- Mejor percepción de seguridad por parte de los usuarios.

3.4 Consideraciones finales

La propuesta aquí presentada busca no solo resolver las vulnerabilidades actuales, sino establecer una cultura de mejora continua en materia de seguridad. Su implementación no requiere cambios drásticos en la estructura del sistema, pero sí compromiso institucional y técnico para aplicarla correctamente y mantenerla en el tiempo.

4 CONCLUSIONES

El presente estudio ha demostrado la relevancia de aplicar pruebas de seguridad sistemáticas en plataformas web educativas como el Entorno Virtual de Aprendizaje (EVA) del Instituto Superior Tecnológico Tena. A medida que el uso de estos sistemas se intensifica en contextos académicos, también aumenta la exposición a riesgos cibernéticos que podrían comprometer la confidencialidad, integridad y disponibilidad de la información institucional y personal.

Mediante la implementación de herramientas especializadas como OWASP ZAP y Pentest-Tools, fue posible identificar vulnerabilidades comunes como inyección SQL (SQLi), Cross-Site Scripting (XSS) y falsificación de peticiones en sitios cruzados (CSRF). Estas amenazas, si bien en algunos casos fueron catalogadas como de bajo riesgo, tienen el potencial de ser explotadas si no se toman medidas correctivas oportunas.

Se concluye que la aplicación de pruebas de seguridad automatizadas y manuales, combinadas con un proceso de documentación de hallazgos y recomendaciones, permite no solo detectar fallas técnicas, sino también establecer un protocolo de mejora continua en el desarrollo web. Esto contribuye significativamente a aumentar la protección de los datos sensibles gestionados por la plataforma, mejorar la calidad del servicio ofrecido a los usuarios, y elevar los estándares de seguridad institucional.

5 RECOMENDACIONES

Adoptar una postura proactiva en ciberseguridad, incorporando la seguridad informática como un componente esencial en cada fase del ciclo de vida del desarrollo de software.

Implementar medidas de seguridad básicas como la correcta configuración del servidor web, el uso de cabeceras HTTP seguras, validaciones en formularios y la actualización regular de componentes y librerías.

Establecer un cronograma de revisiones periódicas que incluya pruebas de penetración internas y externas, así como escaneos automatizados con herramientas reconocidas.

Promover la capacitación continua del personal técnico en prácticas seguras de desarrollo y gestión de sistemas web

Fomentar una cultura institucional de prevención, concienciando a usuarios y administradores sobre los riesgos existentes y la importancia del uso responsable del sistema EVA.

Documentar e implementar protocolos de respuesta a incidentes, para actuar de manera oportuna frente a cualquier posible vulnerabilidad o intento de ataque.

6 REFERENCIAS BIBLIOGRÁFICAS

Rojas Villanueva, H. J. (2024). "Implementación y evaluación de la eficiencia de desempeño, bajo las normas ISO 25010 de un sistema web para el registro de colegiados y control de pago en el Colegio de Economistas de Huánuco.

Arni, S. A., Mongkau, D. C., & Berelaku, A. (2023). Analisis Performa Website Menggunakan GTMetrix: . Jurnal Minfo Polgan, 12(1), 857-861.

Urrutia Simó, P. (2022). ANÁLISIS DE SEGURIDAD EN WEBS PÚBLICAS A TRAVÉS DE TEST DE INTRUSIÓN (Doctoral dissertation, Universitat Politècnica de València).

Amaya Melo, A. P. (2024). Análisis de seguridad para despliegues de plataformas educativas tecnológicas moodle en docker swarm desarrolladas por la empresa TRASCEND-IT (Doctoral dissertation, PUCE Ibarra).

Weichbroth, P. (2020). Usability of mobile applications: a systematic literature study. Ieee Access, 8, 55563-55577.

Lucas, G. I. C., Tejena, L. E. D., Solorzano, B. R. P., & Merino, M. J. M. (2022). Riesgos de seguridad de los datos en la web. Journal TechInnovation, 1(2), 43-49.

Fernández Gutiérrez, L. A., Álvarez Valencia, L., Alvarez Matos, N., González Brito, H. R., & Trujillo Casañola, Y. (2024). Tendencias actuales de las vulnerabilidades y ataques XSS. Serie Científica de la Universidad de las Ciencias Informáticas, 17(7), 120-132.

Rivera García, A. (2023). Sistema para determinar si un código fuente es vulnerable a XSS (cross-site scripting).

Segundo, C. J. N. (2020). Hacking web (Análisis de ataques SQL Inyección, XSS).

Bonilla Mosquera, D., Gonzalez Lopez, J. F., González Brito, H. R., & Trujillo Casañola, Y. (2024). Tendencia en las vulnerabilidades CSRF entre 2018 y 2024. Serie Científica de la Universidad de las Ciencias Informáticas, 17(7), 133-143.

Carballo, F. N., Olivera, V. P., Diaz, M. S. R., & Turello, M. G. (2024). Vulnerabilidad CSRF. Revista Digital del Departamento de Ingeniería e Investigaciones Tecnológicas, 9(2).

Ramos Mena, Á. E. (2022). Análisis de seguridad de XSS, SQL Injection y CSRF en Laravel, Django, Express y Spring.

Corredor, A. L. P., Lasso, M. A. B., & Estrada, E. G. (2025). Identificación de vulnerabilidades basados en programación: buenas prácticas de programación para sistemas informáticos más seguros. Revista Ingeniería, Matemáticas y Ciencias de la Información, 12(23), 121-137.

Castaño Gómez, J. (2022). Ataques SQL Injection: Caso de estudio con MySQL y PHP.

Fernández Gayol, P. (2022). Estudio de los principales tipos de ataques por inyección de código a aplicaciones web y sistema para determinar si un código fuente es vulnerable a SQL Injection.

Ontiveros, J. M. B., Briones, F. Z., Morales, N. R. R., Estrada, M. B., & Reyes, M. P. (2022). Proceso de seguridad para evitar la infiltración de inyección SQL (SQL injection). REVISTA IPSUMTEC, 5(1), 52-62.

Fernández, S. A. C. (2024). Estudio de Tipos de Vulnerabilidades SQL Injection y Protección contra los Mismos.

7 ANEXOS

ANEXO *I*

Preguntas de la encuesta aplicada en redes sociales y a estudiantes

1. ¿Has escuchado alguna vez sobre la seguridad en páginas web?
   - o Sí
   - o No
   - o Tal vez
2. ¿Sabes qué puede pasar si una página web no protege la información?
   - o Sí
   - o No
   - o Tal vez
3. ¿Crees que una página web puede ser hackeada si no está bien protegida?
   - o Sí
   - o No
   - o Lo dudo
4. ¿Has usado alguna herramienta para revisar si una página es segura?
   - o Sí
   - o No
5. ¿Te gustaría que las páginas que usas (como el EVA) estuvieran mejor protegidas?
   - o Sí
   - o No
   - o Tal vez
6. ¿Te parece importante revisar si una página es segura antes de ingresar datos personales?

- Sí
- No
- Tal vez

7. ¿Has visto alguna vez un mensaje como "Esta página no es segura" en el navegador?

- Sí
- No
- Tal vez

8. ¿Te gustaría aprender cómo proteger una página web?

- Sí
- No
- Tal vez

9. ¿Sabes qué hace una herramienta como Pentest-Tools?

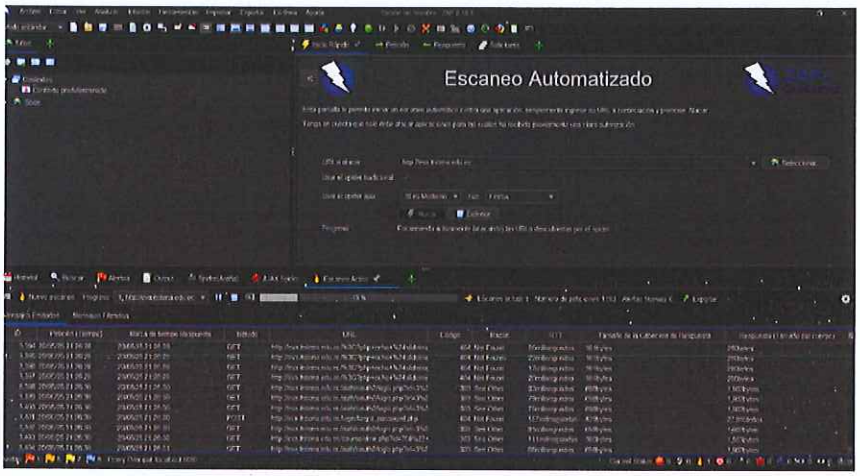
- Sí
- No
- Tal vez

10. ¿Te sientes seguro usando el sitio web del EVA del instituto?

- Sí
- No
- Tal vez

ANEXO II

Escaneo del EVA



ANEXO III

Informe técnico de pruebas de seguridad

Ver Informe Técnico de Pruebas de Seguridad

ANEXO IV

Oficio de autorización para la ejecución de pruebas de seguridad

REPÚBLICA DEL ECUADOR

INSTITUTO SUPERIOR TECNOLÓGICO TENA
Tecnología, Innovación y Desarrollo

Secretaría de Educación Superior, Ciencia, Tecnología e Innovación

Oficio Nº ISTT-R-2025-526-OF
Tena, 04 de agosto de 2025

Señor
Huatatoca Avilez Jhordan Jesús
ESTUDIANTE DEL IST TENA
Presente

De mi consideración:

Con un cordial saludo y en atención al documento s/n de fecha 31 de julio del 2025, que en su parte pertinente manifiesta: *(...) me encuentro en el proceso de titulación con el tema: **"Ejecutar Pruebas de Seguridad para el sitio web Entorno Virtual De Aprendizaje (EVA) del Instituto Superior Tecnológico Tena"**, por ende, me dirijo respetuosamente para solicitar el permiso de realizar pruebas de seguridad al sitio web antes mencionado (...)*

Al respecto, informo a usted que se autoriza la petición realizada; para lo cual, deberá coordinar con el Ing. Fernando Núñez y con el Lic. Héctor Lozada para la ejecución de dicha actividad.

Con sentimiento de distinguida consideración.

Atentamente,

LORENA PILAR YÁNEZ PALACIOS

Ing. Lorena Pilar Yánez Palacios, MEd.
RECTORA DEL INSTITUTO SUPERIOR TECNOLÓGICO TENA

c.c.: *Ing. Fernando Núñez, DOCENTE DEL IST TENA*
Lic. Héctor Lozada, DOCENTE DEL IST TENA

Punto de Atención al Usuario: km 1 ½ vía Tena – Archidona
Teléfono: 062311709
secretaria.general@itstena.edu.ec
https://www.itstena.edu.ec

EL NUEVO ECUADOR