ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD PARA EL FORTALECIMIENTO DE VULNERABILIDADES E INTEGRIDAD DE APLICACIONES WEB ACADÉMICAS

JIMMY FERNANDO RAMÍREZ MÁRQUEZ

Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito parcial para la obtención del grado de:

MAGÍSTER EN SEGURIDAD TELEMÁTICA

RIOBAMBA - ECUADOR

Marzo 2022

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DE TRABAJO DE TITULACIÓN CERTIFICA QUE:

El **Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo**, titulado ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD PARA EL FORTALECIMIENTO DE VULNERABILIDADES E INTEGRIDAD DE APLICACIONES WEB ACADÉMICAS, de responsabilidad del señor Jimmy Fernando Ramírez Márquez ha sido prolijamente revisado y se autoriza su presentación.

Ing. Luis Eduardo Hidalgo Almeida; Ph. D.
PRESIDENTE

Firmado electrónicamente por:
LUIS EDUARDO HIDALGO ALMEIDA

Ing. Joffre Stalin Monar Monar; Mag.
DIRECTOR

JOFFRE STALIN MONAR MONAR

Firmado digitalmente porJOFFRE STALIN MONAR MONAR
DN: cn=JOFFRE STALIN MONAR MONAR c=EC o=SECURITY DATA S.A. 2 ou=ENTIDAD DE CERTIFICACION DE INFORMACION
Motivo:Soy el autor de este documento
Ubicaci- n:
Fecha:2022-03-11 16:45-05:00

Ing. Oswaldo Geovanny Martínez Guashima; M.Sc.
MIEMBRO

OSWALDO GEOVANNY MARTINEZ GUASHIMA

Firmado digitalmente por OSWALDO GEOVANNY MARTINEZ GUASHIMA

Ing. Renny Geovanny Montalvo Armijos; Mag.
MIEMBRO

Firmado electrónicamente por:
RENNY GEOVANNY MONTALVO ARMIJOS

Riobamba, marzo de 2022

DERECHOS INTELECTUALES

Yo, Jimmy Fernando Ramírez Márquez, declaro que soy responsable de las ideas, doctrinas y resultados expuestos en el **Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo**, y que el patrimonio intelectual generado por la misma pertenece exclusivamente a la Escuela Superior Politécnica de Chimborazo.

Firmado electrónicamente por:
JIMMY FERNANDO RAMIREZ MARQUEZ

_____

Jimmy Fernando Ramírez Márquez

CI: 0801504705

DECLARACIÓN DE AUTENTICIDAD

Yo: Jimmy Fernando Ramírez Márquez., declaro que el presente Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otra fuente están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este proyecto de investigación de maestría.

Firmado electrónicamente por:
JIMMY FERNANDO RAMIREZ MARQUEZ

Jimmy Fernando Ramírez Márquez
No. Cédula: 0801504705

DEDICATORIA

Esta tesis está dedicada a: Dios quien ha sido mi guía, fortaleza y su mano de fidelidad y amor han estado conmigo hasta el día de hoy. A mi familia quienes con su amor, paciencia y comprensión me han permitido llegar a cumplir hoy un sueño más.

Jimmy

AGRADECIMIENTO

Al culminar este trabajo quiero agradecer a Dios por todas sus bendiciones, a mi familia gracias por apoyarme en cada elección y emprendimiento, a mi compadre Jhonny Quiñónez por guiarme y darme fortaleza en todo momento.

Jimmy

CONTENIDO

Páginas

CAPÍTULO V

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

El objetivo fue diseñar un sistema de fortalecimiento de la seguridad e integridad de aplicaciones web académicas para la Universidad Técnica Luis Vargas Torres (UTELVT). No existe una aplicación web cien por ciento segura se puede afirmar que todas tienen vulnerabilidades y están en riesgo de sufrir ataques que atenten contra la seguridad e integridad de su información, motivo por el cual se realizó este estudio cuasi - experimental, transversal, descriptivo en donde se busca fortalecer la seguridad e integridad de aplicaciones web académicas de la UTELVT. Se desarrollaron dos (2) escenarios diferentes; en el primero se analizaron las aplicaciones Web académicas en producción en su situación actual, y en el segundo se utilizó un escenario de pruebas controlado. Para el escaneo de vulnerabilidades en ambos escenarios se escogió la metodología OWASP, debido a que está orientada a entornos de aplicaciones web de cualquier tipo de organización. Además, el software que se utilizó para el test de penetración a aplicaciones web académicas de la UTELVT fue OWASP ZAP. Las vulnerabilidades encontradas en el primer escenario que presentan mayor riesgo a la seguridad e integridad de aplicaciones web académicas de la UTELVT durante el escaneo utilizando la aplicación OWASP ZAP fueron de tres categorías: fallos criptográficos, diseño inseguro, componentes vulnerables y obsoletos según OWASP 2021. Por lo que en el segundo escenario se diseñó e implementó un sistema de fortalecimiento de la seguridad e integridad de aplicaciones web académicas de la UTELVT, basado en salvaguardas diseñadas a partir de las soluciones ya establecidas en la metodología OWASP, además de las políticas de seguridad que las salvaguardas sugieren sean implementadas. Se logró reducir probabilidad de ocurrencia de riesgos a las aplicaciones web de la UTELVT.

Palabras Clave: <SEGURIDAD>, <FORTALECIMIENTO>, <VULNERABILIDADES>, <INTEGRIDAD>, <APLICACIONES WEB ACADÉMICAS>.

LUIS ALBERTO CAMINOS VARGAS

Firmado digitalmente por LUIS ALBERTO CAMINOS VARGAS
Nombre de reconocimiento (DN): c=EC, l=RIOBAMBA, serialNumber=0602766974, cn=LUIS ALBERTO CAMINOS VARGAS
Fecha: 2022.02.24 14:48:35 -05'00'

UNIDAD DE PROCESOS TÉCNICOS DBRA ESPOCH

0016-DBRA-UPT-IPEC-2022

SUMARY

There is no web application which is one hundred percent secure, it can be said that they all have vulnerabilities and are at risk of attacks that threaten the security and integrity of their information. For this reason, this quasi-experimental, cross-sectional, descriptive study was carried out in order to strengthen the security and integrity of academic web applications of the UTELVT. Two (2) different scenarios were developed; in the first, academic web applications in production were analyzed in their current situation, and in the second, a controlled test scenario was utilized. In order to get the vulnerability scanning in both scenarios, the OWASP methodology was selected, since it is oriented to web application environments of any type of organization. Moreover, the software used for the penetration test for academic web applications of the UTELVT was OWASP ZAP. The vulnerabilities found in the first scenario which present a greater risk to the security and integrity of UTELVT academic web applications during scanning using the OWASP ZAP application fell into three categories: cryptographic flaws, insecure design, vulnerable and obsolete components according to OWASP 2021. Hence, in the second scenario, a system to strengthen the security and integrity of academic web applications of UTELVT was designed and implemented, based on safeguards designed from the solutions already established in the OWASP methodology, in addition to the security policies which the safeguards suggested to be implemented. It was possible to reduce the probability of occurrence of risks to the web applications of the UTELVT.

Keywords: <SECURITY>, <STRENGTHENING>, <VULNERABILITIES>, <INTEGRITY>, <ACADEMIC WEB APPLICATIONS>.

CAPÍTULO I

1. INTRODUCCIÓN

Las tecnologías de la información y comunicación evolucionan día a día de forma vertiginosa, según estadísticas digitales (2021) el 60% de la población mundial está conectada a internet, mientras que en Ecuador el 57% de la población usa este servicio, es decir en este país se evidencia un incremento en el uso del internet comparado con el año anterior (Branch, 2021). Con ello también se ha incrementado el uso de dispositivos que permiten a los usuarios acceder a la red como por ejemplo (computadoras portátiles, teléfonos inteligentes, tabletas, etc.). Cada vez más actividades habituales se han trasladado a las aplicaciones web.

Mayoritariamente las aplicaciones web se desarrollan con el objetivo de proporcionar cierta funcionalidad, dejando en muchos casos la seguridad en un segundo plano. Lo que también genera riesgo de sufrir ataques informáticos, los cuales aprovechan las vulnerabilidades de las aplicaciones web en algún momento de su ciclo de vida. Motivo por el cual las aplicaciones web deben ser aseguradas, para que personas inescrupulosas no puedan acceder a éstas sin autorización.

Las instituciones de educación superior con la finalidad de asegurar la calidad académica emplean aplicaciones web para gestionar la información requerida por los procesos propios de la academia. En la provincia de Esmeraldas - Ecuador la Universidad Técnica Luis Vargas Torres (UTELVT) es la única institución de educación superior pública, en las diferentes carreras que oferta se educan aproximadamente doce mil jóvenes, los cuales acceden a las aplicaciones web académicas para realizar trámites como por ejemplo matrículas, revisión de notas, generando gran cantidad de información académica sumamente importante. Debido a esto las aplicaciones web académicas de la UTELVT deben ser resguardados ante posibles incidentes de seguridad que exploten alguna vulnerabilidad y atenten contra su disponibilidad, confidencialidad e integridad.

Este estudio tiene la siguiente estructura: la sección II se describe el marco teórico en el que se fundamenta la investigación, la sección III describe la metodología empleada en la investigación, la sección IV presenta los resultados obtenidos y la discusión sobre los mismos por último se exponen las conclusiones.

1.1. Planteamiento del problema

1.1.1. Situación problemática

La UTELVT para el desarrollo de sus actividades académicas tiene en producción varias aplicaciones web, el URL de la aplicación web principal de la universidad es http://utelvt.edu.ec/, y el URL de la aplicación web de servicios en línea es: http://sistemas.utelvt.edu.ec/.

La aplicación web de servicios en línea de la UTELVT se agrupa los siguientes aplicativos: Academia (académico SIAD, evaluación docente y eficiencia académica); Bienestar estudiantil (ficha socioeconómica, bienestar universitario-registro médico y odontología); Vinculación (prácticas pre profesionales y vinculación y, seguimiento a graduados); Administrativo (talento humano y secretaria general) y Repositorios (revistas digitales de la UTELVT y biblioteca general), cada servicio tiene acceso a su propia página Web.

En esta investigación se analizó las vulnerabilidades del servicio en línea Academia – Académico SIAD, el cual contiene: matrícula de los estudiantes, gestión académica - administrativa, docentes - registro de calificaciones y, el sistema de información académico docente. La administración general de este servicio en línea es dirigida por el departamento de TIC y supervisada por el vicerrectorado académico de la universidad.

Es obligatorio para UTELVT proteger la información académica de posibles incidentes seguridad, esto conlleva a salvaguardar prioritariamente el servicio en línea académico SIAD, tratando en lo posible que, en caso de ocurrir un ataque, este no pueda afectar la integridad de la información académica. Pero resguardar la seguridad de la información es cada día una tarea más difícil de cumplir, puesto que los delincuentes cibernéticos siempre están un paso adelante para infringir fallos de seguridad en sistemas informáticos.

1.1.2 Formulación del problema

¿Cómo proteger las aplicaciones web académicas de la Universidad Técnica Luis Vargas Torres de Esmeraldas de ataques informáticos?

1.1.3 Preguntas directrices o específicas de la investigación

- ¿Cuáles son los tipos de ataques más comunes que afectan a las aplicaciones web?
- ¿Qué herramientas se deben emplear para realizar test de seguridad a las aplicaciones web académicas de la UTELVT en un entorno controlado?

- ¿Cómo diseñar una metodología para mitigar las vulnerabilidades de las aplicaciones web académicas de la UTELVT?

1.2. Justificación de la investigación

1.2.1. Justificación Teórica

Actualmente la Universidad Técnica Luis Vargas Torres de Esmeraldas utiliza el servicio en línea académico / matricula – académico SIAD para la matrícula, gestión académica, registro de calificaciones, y el sistema integrado académico docente. Mantener la seguridad de estas aplicaciones web es una tarea muy difícil.

Esta investigación se justifica gracias a que se analizó las vulnerabilidades de las aplicaciones web académicas de la UTELVT, en base al proyecto abierto de seguridad en aplicaciones web (OWASP) el mismo que fue la base para la detección y explotación de vulnerabilidades para establecer procesos de corrección de las mismas.

En el proceso de la investigación se involucró a los encargados del desarrollo y administración de las aplicaciones web académicas de la universidad y director del departamento de TI. Con ello se hizo posible la ejecución de pruebas de penetración en un entorno controlado y establecer procesos correctivos que reduzcan las vulnerabilidades encontradas, además de diseñar el plan de fortalecimiento de la seguridad de las aplicaciones web en análisis.

El principal beneficiario es la UTELVT, gracias a que mejoró la seguridad de la información académica generada por la aplicación Web SIAD. Además, los administradores contaran con planes de contingencia, con los cuales en caso de materializarse ataques contra la seguridad los daños sean mínimos, y la recuperación sea en el menor tiempo posible.

Este proyecto es factible porque fortalece la seguridad de la información académica, a la vez que se mitigan las vulnerabilidades en las aplicaciones web académicas de la UTELVT, además se cuenta con los permisos respectivos de los administradores de TI de la Universidad.

1.2.2. Justificación Práctica

Para validar el método propuesto se plantearon dos escenarios:

En el primer caso se ejecutó un test de seguridades a las aplicaciones web académicas de la UTELVT con lo cual se establecieron las vulnerabilidades existentes actualmente.

El segundo caso se planteó las salvaguardas y se implementaron los correctivos a las aplicaciones web académicas de la UTELVT, luego de eso se ejecutó un test de seguridad a las aplicaciones web en estudio.

Se debería observar que el escenario dos existen menos riesgos de explotación de vulnerabilidades del top 10 de OWASP.

1.2.3. Justificación Metodológica

El método aplicado fue en base a la Guía OWASP para análisis de vulnerabilidades y seguridades en aplicaciones web y software en general, así como técnicas de programación segura implementaron en el escenario dos.

1.3. Objetivos de la investigación

1.3.1. Objetivo general

Diseñar un sistema de fortalecimiento de la seguridad e integridad de aplicaciones web académicas para la UTELVT.

1.3.2. Objetivos específicos

- Determinar la metodología para el escaneo de vulnerabilidades de las aplicaciones web académicas de la UTELVT.
- Detectar las vulnerabilidades de las aplicaciones web académicas de la UTELVT mediante la ejecución de test de penetración en un entorno controlado.
- Diseñar salvaguardas para el fortalecimiento de la seguridad de las aplicaciones web académicas de la UTELVT.
- Verificar la mejora de la seguridad con las salvaguardas implementadas

1.4. Hipótesis

Con la implementación de un sistema de fortalecimiento de la seguridad se reducirán las vulnerabilidades de las aplicaciones web académicas de la UTELVT.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Antecedentes del problema

2.1.1. Análisis de metodologías propuestas

A continuación, se presentan varias investigaciones previas relacionadas con el tema de este estudio:

Rincón y Albarracín en (2018) con su investigación de tipo aplicada analizaron y evaluaron la seguridad informática de la página web publicada en hosting gratuito de la institución técnica de Firavitoba, para la detección y remediación de vulnerabilidades y riesgos en la información. Con los resultados que obtuvieron concluyen que, para minimizar los riesgos en el flujo de la información, es necesario implementar controles y acciones de seguridad, lo que ayudará a fortalecer tres aspectos importantes en la seguridad de la información: la confidencialidad, integridad y disponibilidad de la información. Valiéndose del Top 10 de OWASP y su herramienta OWASP ZAP que permitió la detección de vulnerabilidades en la página web de la Institución educativa. Se encontraron 5 vulnerabilidades de nivel bajo y medio que fueron remediadas mediante la inserción de código en los archivos fuente de las páginas vulneradas. Al final, generaron una ficha con las vulnerabilidades encontradas, nivel de riesgo, confianza, descripción y remedición recomienda.

Por su parte Marulanda en (2018) ejecutó un estudio descriptivo de corte transversal, cuantitativo, con el objetivo analizar los riesgos de seguridad de la información en el sistema e-commerce siembraviva.com, haciendo uso de la guía de pruebas de la metodología OWASP versión 3.0. luego de las pruebas realizadas que con el resultado del análisis de riesgos de seguridad de la información del sistema e-commerce SiembraViva.com, aplicó pruebas de caja negra, logrando establecer los niveles de seguridad implementados en la actualidad, lo cual permite tomar acciones correctivas evitando posibles fallos y perdida de información en base a las vulnerabilidades encontradas en el sistema analizado. Para la recopilación de información utilizó herramientas de reconocimiento pasivo de acceso público como motores de búsqueda, enviando peticiones HTTP simples, forzar a la aplicación para que envíe al exterior mensajes de error, versiones o tecnología utilizada. La ejecución de la metodología la dividió en dos fases, basadas en OWASP, pasiva y activa; en la primera explora las funcionalidades de la aplicación web identificando los formularios y páginas a las que se puede acceder; en la segunda, la aplicación

es sometida a pruebas mediante herramientas de rastreo entrada y salida de resultados, Sniffers, mapeo de software y debilidades de parches, plugin, themes y validaciones de datos. Todo esto permitió identificar las vulnerabilidades de la aplicación web, para luego aplicar una metodología de mitigación de vulnerabilidades encontradas.

En otro estudio Sánchez en (2017) analizó las vulnerabilidades existentes y diseñó procesos correctivos para la página web de la dirección de educación a distancia y virtual de la universidad técnica de Ambato. Emplearon la metodología OWASP, consiguiendo focalizar el estudio y realizar los distintos análisis del sitio web. Para lo cual se dividió la investigación en dos partes, primero el análisis de la aplicación web para recolectar los datos y detectar las posibles entradas a la aplicación; en la segunda, aplica la metodología OWASP mediante diferentes Test como: de manejo de configuración y desarrollo, de manejo de identidad, de autenticación, autorización, de manejo de sesiones, de validación de entradas, de lógica de negocio, de lado cliente y Criptografía que permitieron encontrar las diferentes vulnerabilidades en la aplicación. Al final, elaboró un catálogo de vulnerabilidades mencionando en ella el nombre de la vulnerabilidad, apartado, descripción, riesgo, proceso correctivo y su respectiva solución.

Montalvo en (2017) con el objetivo de generar políticas para la gestión de riesgos de seguridad en el desarrollo de software en el Consejo de la Judicatura, desarrolló un estudio experimental en el que planteó dos escenarios, antes y después de aplicar las políticas. Durante la etapa de desarrollo de software, al aplicar las políticas, logró reducir del 39,2% al 12,5% en las pruebas ejecutadas. Para lo cual utilizó la metodología MAGERIT que sirven para el desarrollo de nuevas Aplicaciones de Software, basadas la ISO 27002 del 2013, como: trabajar en equipo, iniciar en la fase de requerimentación, trabajar con un servidor de aplicaciones que permita configurar seguridades, proteger de ataques internos, hay que prestar atención a los caracteres que se permitan ingresar, presentar una página de error personalizada, no presentar información delicada en la barra de dirección, cifrar la información, mantener el software actualizado, no permitir la opción de copiar y pegar, no permitir guardar contraseñas, mantener respaldos actualizados periódicamente, concientizar al personal interno. El tener éxito, se deben socializar todas estas políticas con el personal involucrado que forman parte de la institución.

2.1.2. Metodología de pruebas de penetración

Existen varias metodologías para pruebas de penetración que se utilizan en el ámbito de auditoría de sistemas entre ellas se puede citar: Proyecto de seguridad de aplicaciones web abiertas (OWASP); Manual de metodología de pruebas de seguridad de código abierto (OSSTMM); Estándar de ejecución de pruebas de penetración (PTES). En la tabla 1-2, se muestra las principales características de las metodologías antes mencionadas. Entre los aspectos

considerados están: 1) *ámbito y enfoque*; refiriéndose al tipo de organización que la utilizará, de los hackers que realizarán la prueba de penetración y de las diferentes áreas en las cuales se pueda emplear la metodología; 2) *Alcance*; este aspecto evidencia todas las tareas que puede abarcar la metodología incluyendo la valoración de riesgos; 3) *Profundidad*; se refiere al detalle con que trabaja la metodología; 4) *Usabilidad*; explica la facilidad con la que se puede utilizar la metodología en entorno de pruebas de penetración y riesgos. 5) *Métricas*; que son una forma objetiva de medir y clasificar las vulnerabilidades encontradas; 6) *Evaluación del riesgo*; enuncia en qué nivel de gravedad se encuentra el riesgo y como se puede mitigar el impacto de su materialización; Otro punto considerar en el análisis, son las fases que operan cada una de las metodologías para el desarrollo de pruebas (Alvarez, 2018).

A continuación, en la tabla 1-2 se presenta una comparación entre varias metodologías para análisis de vulnerabilidades en aplicaciones web:

Tabla 1-2: Tabla comparativa de metodologías de pruebas de penetración

Características	OWASP	OSSTMM	PTES
Ámbito y enfoque	Se enfoca en entornos de aplicaciones web de cualquier organización	Enfoque operativo, para cualquier empresa que quiera evaluar la seguridad de la información	Se adapta a cualquier ámbito de aplicación
Alcance	Auditoría en aplicaciones web, en todo el ciclo de la implementación	Abarca equipos y sistemas asociados a la red	Está orientado a niveles técnicos específicamente
Profundidad	Analiza en detalle lo concerniente a la seguridad de la aplicación	Análisis en detalle	Esta metodología se enmarca en la metodología OSSTMM y se combina con OWASP
Usabilidad	Usabilidad alta en aplicaciones web	Requiere de capacitación y es categorizada como nivel medio en usabilidad	Usabilidad media
Métricas	Posee métricas para categorizar y evaluar los riesgos	Definidas para medir el grado de seguridad de los activos, utiliza mediciones objetivas llamadas RAV	Puede medir o estimar la amenaza contra un activo. No hay establecidas métricas.
Evaluación del riesgo	Aplica métricas para evaluar y valorar los riesgos	Aplica los RAV para cuantificar con precisión, los niveles de riesgos; los mismos que están integrados en cada módulo de operación.	No maneja evaluación del riesgo propio, se enmarca en lo establecido en la metodología OSSTMM

Fuente: (Lopez, 2015)

Realizado por: Ramírez, J. 2021

Nota: El RAV es una medida en escala de la superficie de ataque expuesta que es un cálculo cuantitativo que busca el balance entre operaciones, limitaciones y controles

Desarrollar una guía representa un esfuerzo enorme, que implica décadas de trabajo realizado por cientos de personas en todo el mundo. Exciten diversas formas de probar fallos de seguridad, la guía OWASP recopila el consenso de los principales expertos sobre cómo realizar esta comprobación rápida, exacta y eficientemente (OWASP, 2008).. Es por ello que se decide aplicar esta metodología el desarrollo de este estudio.

2.2. Bases teóricas

2.2.1 Seguridad informática

De acuerdo con Voutssas (2010) seguridad informática es el proceso de establecer y observar un conjunto de estrategias, políticas, técnicas, reglas, guías, prácticas y procedimientos tendientes a prevenir, proteger y resguardar de daño, alteración o sustracción a los recursos informáticos de una organización.

Por su parte García & Vidal (2016) establecen que la seguridad informática se orienta a la protección de la infraestructura de tecnología de la información, a través de la minimización de los posibles riesgos que puedan afectar dicha infraestructura y la información. El objetivo primordial de la seguridad informática es mantener al mínimo los riesgos sobre los recursos informáticos, garantizando de esta manera la continuidad del negocio, al tiempo que se administra ese riesgo informático en un costo aceptable. Para ello se debe emplear estructuras organizacionales técnicas, administrativas, gerenciales o legales.

Los tres principios fundamentales en que se basa la seguridad informática son: confidencialidad, la integridad y la disponibilidad.

La confidencialidad, es la condición, que asegura que la información no pueda estar disponible o ser descubierta por personas, entidades o procesos no autorizados.

Integridad es la condición que garantiza que la información solo puede ser modificada, incluyendo su creación y borrado, por el personal autorizado. Garantiza que la información sea exacta y completa y que el sistema no modifique o corrompa la información o permita que alguien no autorizado lo haga.

Mientras que la disponibilidad es la propiedad que garantiza el acceso a los activos de información y el empleo de los recursos informáticos en el momento que sea requerido por las personas autorizadas. Un sistema seguro debe mantener la información disponible para los usuarios.

Disponibilidad significa que el sistema, tanto hardware como software, funciona de forma eficiente y que es capaz de recuperarse rápidamente en caso de fallo (García, 2003).

2.2.2 Aplicaciones web

Luján (2002) define a las aplicaciones web como herramientas que permiten a los usuarios acceder a un servidor web a través de la red mediante un navegador determinado. Por lo tanto, se define como una aplicación que accede mediante la web por una red ya sea intranet o Internet. Por lo general se menciona aplicación web a aquellos programas informáticos que son ejecutados a través del navegador.

2.2.3 Vulnerabilidades en aplicaciones web

Una vulnerabilidad es una debilidad en un activo informático la cual puede ser explotada por una amenaza y producir daños en contra de la disponibilidad, integridad y confidencialidad de dicho activo (Quiroz & Macías, 2017).

El proyecto abierto de seguridad de aplicaciones web (OWASP), es una organización que brinda información para la gestión de riesgo en seguridad de aplicaciones informáticas, publicó el top 10 de las vulnerabilidades más comunes en el año 2021, a continuación, se presenta: A01: Control de acceso roto, A02: Fallos criptográficos, A03: La inyección, A04: Diseño inseguro, A05: Configuración incorrecta de seguridad, A06: Componentes vulnerables y obsoletos, A07: Fallos de identificación y autenticación, A08: Fallos de integridad de datos y software, A09: Fallas de registro y monitoreo de seguridad, A10: Falsificación de solicitudes del lado del servidor (OWASP, 2021).

2.2.4. Análisis de vulnerabilidades

Para Castro (2016) el análisis de vulnerabilidades es el proceso por medio del cual se comprueban a través de herramientas de software y servicios de consultoría la debilidad o fortaleza ante el conjunto de amenazas conocidas al día de la evaluación tanto para elementos externos (Servicios SAAS o Software as a Service, Servicios de Cloud Computing, Servicios BYOD o Bring Your Own Device, Usuarios no autorizados, Sniffers, robots) como para elementos internos (Usuarios, sistemas implementados, estaciones de trabajo, dispositivos móviles, sistemas operativos). El correcto análisis de vulnerabilidades no solo detecta las áreas de mejora, sino que también propone la correcta arquitectura necesaria para proteger la infraestructura de una organización y los diferentes cambios de políticas de seguridad que se requiere implementar para asegurar una

continuidad de operación, la asistencia que se debe proveer cuando se ve comprometida la seguridad informática y la recuperación ante desastres, ante amenazas e intrusiones.

Los pasos para realizar el análisis de vulnerabilidades:

- Diagnóstico de Seguridad:
- Escaneo de vulnerabilidades internas y externas.
- Revisión de políticas de seguridad
- Revisión de procesos, pólizas de soporte y configuraciones que comprometan la seguridad informática.
- Reforzamiento de la topología de red.
- Generación de documento de recomendaciones de buenas prácticas de seguridad informática, arquitectura ideal para la organización,
- Planeación ante eventos que comprometan la seguridad.
- Revisión de políticas de respaldos, sistemas de redundancia, planes de recuperación de desastres.
- Generación de documento recomendaciones ante eventos de seguridad. (Castro, 2016)

2.2.5. Test de penetración o intrusión

Según Ramos (2013) un test de penetración o pent test es un procedimiento que ejecuta un conjunto de técnicas y métodos que simulan el ataque a un sistema y permite evaluar la seguridad de los sistemas informáticos, redes y aplicaciones. No importa que tan protegido este el sistema, es recomendable realizar un pent test para descubrir las vulnerabilidades de modo que se pueda plantear una defensa ante posibles ataques. El Pent Test incluye herramientas que van desde scanners de puertos, algoritmos para descifrar claves, sistemas de intrusión por fuerza bruta, herramientas de sniffing de redes y penetración de firewalls, así como también herramientas de escaneo de vulnerabilidades de aplicaciones web y mucho más. Las herramientas suelen estar agrupadas en lo que se conoce como "Toolkits", existen algunos Toolkits que son famosos por su eficiencia y por haber sido utilizados en penetraciones de alto nivel.

Se pueden clasificar las pruebas de penetración según los siguientes aspectos:

- Pruebas de penetración con objetivo: se buscan las vulnerabilidades en partes específicas de los sistemas informáticos críticos de la organización.
- Pruebas de penetración sin objetivo: consisten en examinar la totalidad de los componentes de los sistemas informáticos pertenecientes a la organización.

- Pruebas de penetración a ciegas: en estas pruebas sólo se emplea la información pública disponible sobre la organización.
- Pruebas de penetración informadas: aquí se utiliza la información privada, otorgada por la organización acerca de sus sistemas informáticos. En este tipo de pruebas se trata de simular ataques realizados por individuos internos de la organización que tienen determinado acceso a información privilegiada.
- Pruebas de penetración externas: son realizas desde lugares externos a las instalaciones de la organización. Su objetivo es evaluar los mecanismos perimetrales de seguridad informática de la organización.
- Pruebas de penetración internas: son realizadas dentro de las instalaciones de la organización con el objetivo de evaluar las políticas y mecanismos internos de seguridad de la organización.

2.2.6. Herramientas para test de penetración

Debido a la necesidad de que las pruebas de penetración sean más eficientes, se han desarrollado una variedad de herramientas de pago y open Source, que automatizan varias de las tareas que se ejecutan en las pruebas de penetración. A continuación, en la tabla 2-2 se detallan varias.

Tabla 2-2: Herramientas para test de penetración

Herramienta pent test	Ventajas	Desventajas
FOCA	• Realiza búsqueda intensiva de archivos y metadatos de una página web. • Permite descargar todos los archivos encontrados. • Trabaja con distintos formatos de documentos (.pdf, .doc, .xml). • Enlista las todas las redes conectadas a un servidor. • Realiza un análisis de archivos en donde reporta su fecha de creación, servidor y sistema operativo.	• Su interfaz es complicada.
Nmap	• Analiza todas las redes vinculadas a una IP. • Sobrepasa esquemas de Firewall. • Los paquetes que envía a las diferentes IP están muy bien construidos. • Realiza sus consultas de manera rápida. • Puede escanear un rango de IP enlistando todos los servicios. • Configurable	• Soporte para Windows deficiente. • No sobrepasa esquemas de proxys. • Puede resultar muy invasivo creando tráfico inusual en la red.
OWASP ZAP	• Herramienta de código abierto. • Multi plataforma. • Facilidad de instalación. • Soporte continuo. • Permite realizar análisis de cabeceras de una manera más simple. • Análisis pasivos	• A pesar de poseer interfaz gráfica su uso puede ser complicado. • No puede sobrepasar esquemas de firewalls. • Es invasivo.

Fuente: (Sánchez, 2017)

Realizado por: Ramírez, J. 2021

2.2.6. Política de seguridad informática

Dussan (2015) define una política de seguridad como el conjunto de reglas, normas, procedimientos que definen los criterios de seguridad para que sean adoptados a nivel local o institucional, con el objetivo de establecer, estandarizar y normalizar la seguridad tanto en el ámbito humano (usuarios, proveedores, clientes, empleados) como tecnológico (plataforma de hardware, software y telecomunicaciones. Servidores, estaciones de trabajo, sistemas operativos, bases de datos, acceso a Internet).

2.2.7. Salvaguardas

Magerit (2012) define las salvaguardas o contramedidas como aquellos procedimientos o mecanismos tecnológicos que reducen el riesgo que una amenaza se materialice. Es decir, fortalecen la seguridad. Las contramedidas persiguen conocer, prevenir, impedir, reducir y controlar el daño que podría tener un sistema de información.

2.2.8 Riesgo

Es la estimación del grado de exposición a que una amenaza se materialice sobre uno o más activos causando daños o perjuicios a la organización. El riesgo indica lo que le podría pasar a los activos si no se protegieran adecuadamente. Es importante saber qué características son de interés en cada activo, así como saber en qué medida estas características están en peligro, es decir, analizar el sistema, y se denomina análisis de riesgos que es el proceso sistemático para estimar la magnitud de los riesgos a que está expuesta una organización (Dirección General de Modernización Administrativa, Procedimientos e Impulso de la Administración Electrónica, 2012).

CAPÍTULO III

3. DISEÑO DE INVESTIGACIÓN

3.1. Tipo y diseño de investigación

Esta investigación es de tipo cuasi - experimental, transversal, debido a que se evaluaron las vulnerabilidades a las que están expuestas aplicaciones web académicas de la UTELVT en el segundo semestre académico del 2020. El alcance descriptivo facilitó particularizar las vulnerabilidades de las aplicaciones web académicas de la UTELVT

3.2. Diseño de la investigación

La presente investigación se ejecutó en la Universidad Técnica Luis Vargas Torres de Esmeraldas, su campo de acción fue el fortalecimiento de la seguridad e integridad de aplicaciones web académicas. Este estudio tiene un enfoque cuantitativo puesto que se aplicaron instrumentos en escala o rangos numéricos los cuales permitieron compilar información adecuada para el análisis de las variables en estudio.

3.3. Métodos y técnicas de la investigación

3.3.1. Métodos

La investigación se realizó a la luz del método científico, la aplicación del método deductivo en este estudio permitió establecer cuáles de las vulnerabilidades son las que deben ser atendidas con mayor prioridad, para diseñar un plan de fortalecimiento de las aplicaciones web académicas (Rodríguez, 2019).

3.3.2. Técnicas

Las técnicas utilizadas en esta investigación son:

- Revisión de documentación. - esta técnica se empeló para la recopilación de la información necesaria para el sustento científico de la investigación.
- Pruebas de laboratorio. - se realizaron pruebas en un escenario controlado con la finalidad de analizar las vulnerabilidades en las aplicaciones web académicas de la UTELVT antes y después de implementar las salvaguardas y los correctivos.

- Observación. – permitió observar el fenómeno estudiado para este estudio las vulnerabilidades de las aplicaciones web académicas de la UTELVT.
- Análisis. – se empleó para contrastar los resultados de los escaneos de vulnerabilidades realizados en los dos escenarios y determinar si existen mejoras a la seguridad de las aplicaciones web en estudio.
- Para la comprobación de la hipótesis se utilizó la estadística descriptiva.

3.4. Fuentes de información

Son todas aquellas fuentes de interés que permiten encontrar la información necesaria para el sustento de este estudio, a continuación, en la tabla 3 se las describe.

Tabla 1-3: Fuentes de información

Primarias	Secundarias
• Pruebas • Observación de Resultados	• Artículos científicos referentes al tema de estudio. • Investigaciones nacionales e internacionales. • Conferencias académicas, congresos y seminarios. • Sitios web, revistas electrónicas y blogs oficiales relacionadas al tema de investigación

Realizado por: Ramírez, J. 2021

3.5. Planteamiento de Hipótesis

3.5.1. Hipótesis General

Con la implementación de un sistema de fortalecimiento de la seguridad se reducirán las vulnerabilidades de las aplicaciones web académicas de la UTELVT.

3.5.2. Identificación de variables

Variable independiente: Vulnerabilidades de las aplicaciones web académicas de la UTELVT.

Variable dependiente: Sistema de fortalecimiento de seguridad de las aplicaciones web académicas de la UTELVT.

3.5.3 Operacionalización de variables

Tabla 2-3: Operacionalización conceptual de variables

VARIABLE	TIPO	CONCEPTO
Vulnerabilidades de las aplicaciones web académicas de la UTELVT.	Independiente	Debilidad o fallo en una aplicación web que pone en riesgo la seguridad de la información pudiendo permitir que un atacante pueda comprometer la integridad, disponibilidad o confidencialidad de la misma.
Sistema de fortalecimiento de seguridad de las aplicaciones web académicas de la UTELVT.	Dependiente	Proceso de desarrollar, añadir y probar características de seguridad dentro de las aplicaciones web reducir las vulnerabilidades de seguridad.

Realizado por: Ramírez, J. 2021

Tabla 3-3: Operacionalización Metodológica de variables

VARIABLES	INDICADORES	TÉCNICAS	INSTRUMENTOS
V. I: Vulnerabilidades de las aplicaciones web académicas de la UTELVT.	• Numero de vulnerabilidades detectadas por aplicación web • Probabilidad de ocurrencia por vulnerabilidad • Nivel de riesgo por vulnerabilidad	Pruebas Observación de resultados	Escenario actual Escenario de pruebas
V. D: Sistema de fortalecimiento de seguridad de las aplicaciones web académicas de la UTELVT	• Porcentaje de riesgo por vulnerabilidad	Pruebas Observación de resultados	Escenario actual Escenario de pruebas

Realizado por: Ramírez, J. 2021

3.6. Población y muestra

3.6.1. Población de estudio

Como la población no es extensa se empleó la totalidad de la misma como muestra la tabla 4-3.

Tabla 4-3: Población

ITEM	INFORMANTES	CANTIDAD
1	Sistemas web académicos de la UTELVT	4
TOTAL		4

Fuente: Departamento de TIC – UTELVT

Realizado por: Ramírez, J. 2021

3.7. Instrumentos de recolección de datos

Para la recolección de los datos se implementó dos escenarios, uno el actual de la UTELVT y el otro después de las medidas de fortalecimiento, las herramientas que se utilizaron son de código abierto debido a su menor presupuesto para su implementación y además permiten utilizar sus características principales sin ningún tipo de limitación, al contrario de lo que sucede con herramientas privativas que es necesario algún tipo de licenciamiento vigente. En la tabla 7-3 se muestra la escala de valoración con la que se evaluaron cada una de las características de las herramientas para escaneo de vulnerabilidades en aplicaciones web.

Tabla 5-3: Escala de valoración

	Alta	**Media**	**Baja**
Valoración	3	2	1

Realizado por: Ramírez, J. 2021

En la tabla 2-2 se muestra una comparativa de las herramientas de pruebas de penetración, para la valoración se empleó la escala contenida en la tabla 5-3. Se observa que la herramienta con mejores prestaciones para realizar el escaneo de vulnerabilidades de las aplicaciones web en estudio es OWASP ZAP debido a que está orientada a entornos de aplicaciones web de cualquier tipo.

Tabla 6-3: Comparación de herramientas para realizar pruebas de penetración

	Wireshark	**Kali Linux**	**Nmap**	**Maltego**	**Metasploit**	**Nexpose**	**Openvas**	**Owasp zap**	**Acunetix**	**Nessus**
Facilidad de uso	3	3	3	3	3	3	3	3	3	1
Requerimientos del sistema	2	2	2	2	3	2	2	2	2	2
Facilidad de instalación	2	2	1	2	1	1	2	3	2	2
Interfaz amigable	3	3	3	3	3	3	3	3	3	3
Legibilidad del reporte final	3	3	2	2	3	2	3	3	3	3
Contenido de detección	3	3	2	2	3	3	2	3	3	2
Velocidad de detección	2	2	2	1	3	1	2	3	3	2
Total	18	18	15	15	19	15	17	**20**	19	15

Realizado por: Ramírez, J. 2021

3.8. Escenario de pruebas

3.8.1 Descripción del escenario

Para la presente investigación se desarrollaron dos escenarios los cuales se describen a continuación:

Escenario 1.- Es la infraestructura real actual de los módulos web académicos de la UTELVT. En la figura 1-3 se muestra la estructura del centro de datos de la UTELVT ubicado en el departamento de TI.

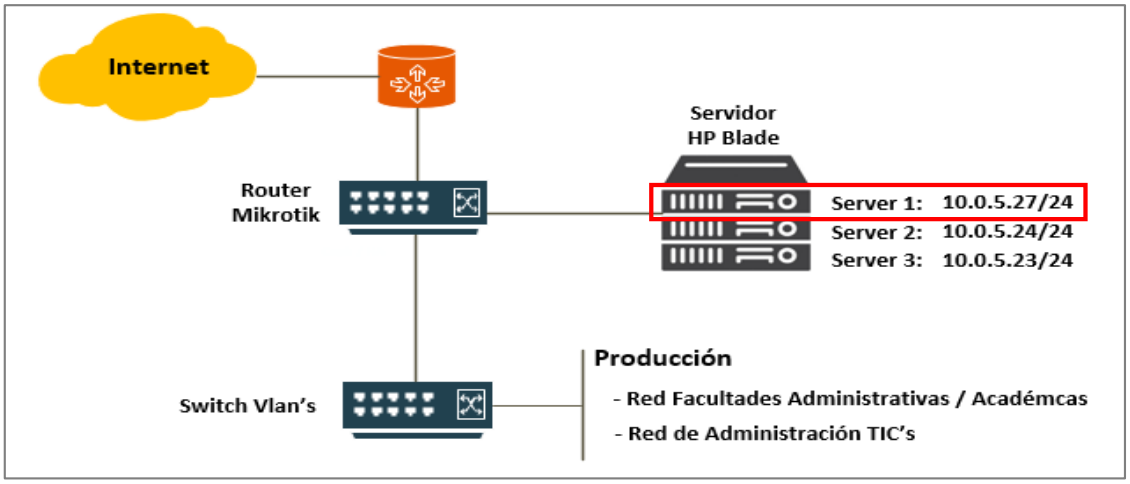


Figura 1-3: Diagrama de la estructura del centro de datos de la UTELVT
Fuente: (UTELVT, Escenario 1, 2021)

Realizado por: Ramírez, J. 2021

Las aplicaciones web académicas se encuentran alojadas en el servidor 1, los usuarios tienen acceso a éstas a través del internet/intranet. El sistema operativo de los servidores es CentOS 7. Estas aplicaciones web están desarrolladas con el framework YII 2, y tienen implementada una base de datos en Mariadb, para almacenar la información. (fuente: Programador Web UTELVT)

Escenario 2.- En este caso, se implementó un ambiente virtualizado de pruebas, donde se colocaron las salvaguardias del sistema de fortalecimiento a las vulnerabilidades de las aplicaciones web académicas de la UTELVT. En la figura 2-3 se muestra la estructura del centro de datos de la UTELVT ubicado en el departamento de TI. con el servidor de pruebas temporal.

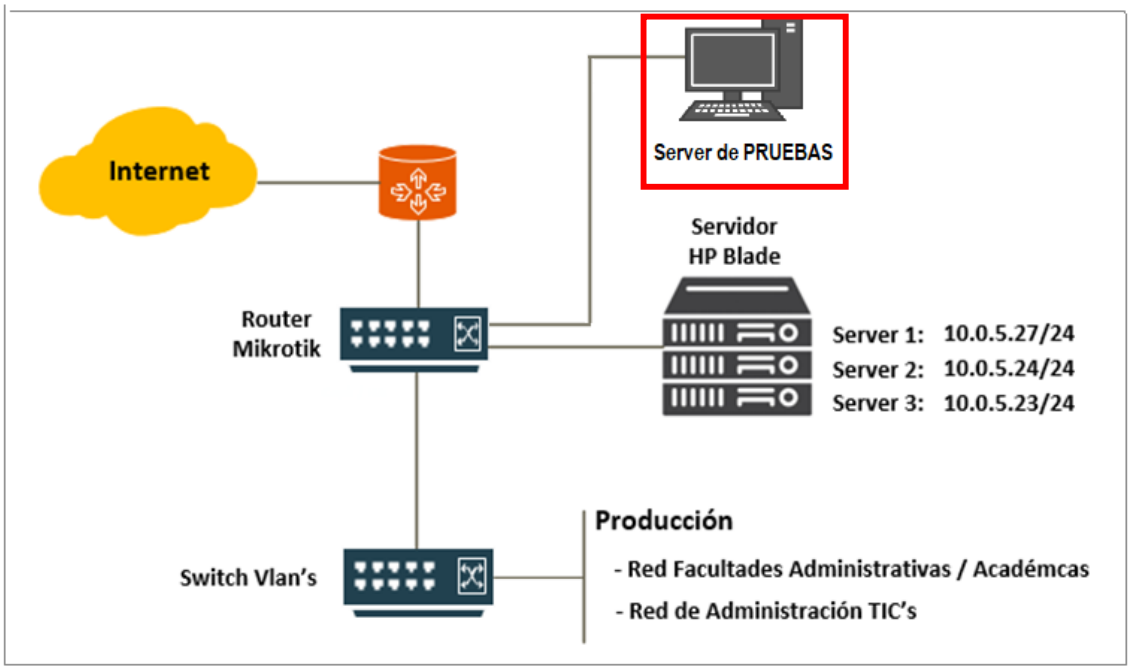


Figura 2-3: Diagrama de la estructura del centro de datos de la UTELVT con el servidor de pruebas temporal
Fuente: (UTELVT, Escenario 1, 2021)

Realizado por: Ramírez, J. 2021

3.8.2. Hardware y software del Escenario 2

Tabla 7-3: Software utilizado en el escenario de pruebas

Nombre	Versión	Descripción	Funcionalidad
Windows	10	Sistema Operativo	Simular el ataque
VMware Workstation 16 Pro	16.1.0	Máquina virtual	Virtualizar Sistema Operativo
Linux Centos	7	Sistema Operativo	Sistema operativo equipo virtual
OWASP ZAP	2.11.0	Aplicativo	Escaneo de vulnerabilidades
PHP	7.3	Lenguaje de programación	Programar líneas de código para corregir errores de seguridad
Servidor Web Apache	2	Servidor Web	Servidor web HTTP
Mariadb	10.5	SGBD	Edición de BD

Realizado por: Ramírez, J. 2021

Tabla 8-3: Hardware utilizado en el escenario de pruebas

Nombre	Descripción	Funcionalidad
Laptop	Computadora Personal	Ejecutar la máquina virtual que funcionará como atacante
Infraestructura TI UTELVT	Servidores, Pc, Router.	Infraestructura a ser atacada

Realizado por: Ramírez, J. 2021

3.8.3. Categorías de amenazas según OWASP 2021

Tabla 9-3: TOP 10 de Categorías de amenaza según OWASP 2021

Ítem	Categorías de Vulnerabilidades
1	A01: Control de acceso roto
2	A02: Fallos criptográficos
3	A03: La inyección
4	A04: Diseño inseguro
5	A05: Configuración incorrecta de seguridad
6	A06: Componentes vulnerables y obsoletos
7	A07: Fallos de identificación y autenticación
8	A08: Fallos de integridad de datos y software
9	A09: Fallas de registro y monitoreo de seguridad
10	A10: Falsificación de solicitudes del lado del servidor

Fuente: OWASP, 2021

Realizado por: Ramírez, J. 2021

3.8.4. Niveles de riesgo según OWASP 2021

Tabla 10-3: Niveles de riesgo según OWASP 2021

Nivel de Riesgo	Impacto	Descripción
Alta	Entre: 6 y <=10 Promedio: 8	Vulnerabilidad que si es explotada comprometería la seguridad de la información ocasionando un impacto negativo sobre la UTELVT. Debe solucionarse inmediatamente.
Media	Entre: 4 y <6 Promedio: 5	Vulnerabilidad que si es explotada tendría un impacto leve sobre la operativa de la UTELVT.
Baja	Entre: 1 y <4 Promedio: 2,5	Vulnerabilidad que si es explotada no ocasionaría mayores inconvenientes.
Informativo	Entre: 0 y <1 Promedio: 0,5	Advertencias y/o recomendaciones que alertan sobre posibles configuraciones que pueden ser mejoradas. Tienen muy bajo impacto en el funcionamiento de las aplicaciones.

Realizado por: Ramírez, J. 2021

Para el cálculo del Porcentaje de Riesgo de cada vulnerabilidad se utilizó la siguiente fórmula:

Riesgo *= Probabilidad de ocurrencia x Impacto*

CAPÍTULO IV

4. RESULTADOS Y DISCUSIÓN

4.1. Análisis e interpretación de resultados

Luego de realizar las pruebas respectivas y la ponderación de las vulnerabilidades a través del escaneo utilizando la herramienta OWASP ZAP 2.11.0, con el objetivo de analizarlas y priorizar las salvaguardas a ser aplicadas para minimizar los riesgos a la seguridad de las aplicaciones web académicas de la UTELVT.

Variable independiente:

- Vulnerabilidades de las aplicaciones web académicas de la UTELVT.

Indicadores:

- Numero de vulnerabilidades detectadas por aplicación web
- Probabilidad de ocurrencia por vulnerabilidad
- Nivel de riesgo por vulnerabilidad

Variable dependiente:

- Sistema de fortalecimiento de seguridad de las aplicaciones web académicas de la UTELVT Indicadores:

Indicador:

- Porcentaje de riesgo por vulnerabilidad

A continuación, se detallan las vulnerabilidades detectadas a través del escaneo de las aplicaciones web académicas, antes y después de aplicar la metodología, en cada uno de los indicadores antes mencionados. Al final, se presentan cuadros comparativos del antes y el después de esta investigación con sus respectivos gráficos.

Se empieza con el indicador Número de vulnerabilidades detectadas por aplicación web:

Tabla 1-4: Número de vulnerabilidades detectadas por aplicaciones web antes de aplicar la metodología

APLICACIÓN WEB	RIESGO ANTES					
	Informativo	**Bajo**	**Medio**	**Alto**	**Total**	**Porcentaje**
estudiante.utelvt.edu.ec/	35	166	4	0	**205**	**59,94%**
administrativo.utelvt.edu.ec/	24	45	3	0	**72**	**21,05%**
docente.utelvt.edu.ec/	21	36	3	0	**60**	**17,54%**
siad.utelvt.edu.ec/	2	3	0	0	**5**	**1,46%**
Total	**82**	**250**	**10**	**0**	**342**	**100,00%**

Realizado por: Ramírez, J. 2021

En la tabla 1-4 se encuentran las ocurrencias de las vulnerabilidades encontradas en cada una de las aplicaciones web académica de la UTELVT en el escenario 1 (sistema académico en producción), antes de aplicar la metodología, las cuales fueron **342**, que representan el 100%. También podemos apreciar que la aplicación con mayor número de ocurrencias de vulnerabilidades es *estudiante.utelvt.edu.ec/* con un 59,94%.

Una vez realizado el test de seguridad y ejecutada la intervención a las aplicaciones web en el escenario 2 (ambienten de pruebas) se pudieron obtener los siguientes resultados:

Tabla 2-4: Número de vulnerabilidades detectadas por aplicaciones web después de aplicar la metodología

APLICACIÓN WEB	RIESGO DESPUÉS					
	Informativo	**Bajo**	**Medio**	**Alto**	**Total**	**Porcentaje**
estudiante.utelvt.edu.ec/	1	0	0	0	**1**	**25,00%**
administrativo.utelvt.edu.ec/	1	0	0	0	**1**	**25,00%**
docente.utelvt.edu.ec/	1	0	0	0	**1**	**25,00%**
siad.utelvt.edu.ec/	1	0	0	0	**1**	**25,00%**
Total	**4**	**0**	**0**	**0**	**4**	**100,00%**

Realizado por: Ramírez, J. 2021

En la tabla 2-4 se demuestra que la intervención permitió reducir significativamente el número de ocurrencias de vulnerabilidades en cada una de las aplicaciones web académica de la UTELVT quedando solo 4 de tipo informativo, que representan el 100%. Se pasó de 342 amenazas a 4.

Se han encontrado **13** tipos de amenazas a través de la herramienta de análisis de vulnerabilidades de aplicaciones web OWASP ZAP, tomando en cuenta los indicadores Nivel de riesgo, Probabilidad de ocurrencia y Porcentaje de Riesgo por vulnerabilidad, detalladas a continuación:

Tabla 3-4: Nivel de riesgo, Probabilidad de ocurrencia, Porcentaje de Riesgo por vulnerabilidad antes de aplicar la metodología

No.	VULNERABILIDAD	NIVEL DE RIESGO ANTES						Porcentaje de riesgo
		Informativo	Bajo	Medio	Alto	Total	Probabilidad	
1	Exploración de directorios	0	0	2	0	**2**	**0,58%**	2,92%
2	Biblioteca JS vulnerable	0	0	8	0	**8**	**2,34%**	11,70%
3	Cookie sin bandera segura	0	44	0	0	**44**	**12,87%**	32,16%
4	Cookie sin atributo SameSite	0	49	0	0	**49**	**14,33%**	35,82%
5	Inclusión de archivos de origen JavaScript entre dominios	0	4	0	0	**4**	**1,17%**	2,92%
6	Divulgación de la marca de hora – Unix	0	25	0	0	**25**	**7,31%**	18,27%
7	El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP "X-Powered-By"	0	57	0	0	**57**	**16,67%**	41,67%
8	Conjunto de encabezados incompleto o sin control de caché	0	5	0	0	**5**	**1,46%**	3,65%
9	Las páginas seguras incluyen contenido mixto	0	3	0	0	**3**	**0,88%**	2,19%
10	Falta el encabezado X-Content-Type-Options	0	63	0	0	**63**	**18,42%**	46,05%
11	Falta el encabezado Content-Type	1	0	0	0	**1**	**0,29%**	0,15%
12	Divulgación de información - Comentarios sospechosos	79	0	0	0	**79**	**23,10%**	11,55%
13	Incompatibilidad de caracteres (Encabezado contra Conjunto de Caracteres de Tipo de Contenido Meta).-	2	0	0	0	**2**	**0,58%**	0,29%
	Pocentaje por Riesgo de Vulneabilidad	**23,98%**	**73,10 %**	**2,92 %**	**0,00 %**			
	Total	**82**	**250**	**10**	**0**	**342**	**100,00%**	

Realizado por: Ramírez, J. 2021

En la tabla 3-4 se describen los resultados obtenidos en el escenario 1 (sistema académico en producción), antes de aplicar la metodología, las cuales fueron: *Informativas* 82 con un riesgo del 23.98%, *Baja* 250 con un riesgo de 73% y *Media* 10 con un 2,92%, que representan el 100%. Cabe destacar que no se encontraron vulnerabilidades con un nivel de riesgo *Alto*. Además, desde el punto de vista del porcentaje de riesgo tenemos que a las vulnerabilidades que hay que ponerle mayor atención son: Falta el encabezado X-Content-Type-Options con 46,05%, El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP "X-Powered-By" con 41,67% y Cookie sin atributo SameSite con 35,82%.

Del mismo modo, tomando en cuenta los mismos indicadores, luego de la intervención se obtuvo:

Tabla 4-4: Nivel de riesgo, Probabilidad de ocurrencia, Porcentaje de Riesgo por vulnerabilidad después de aplicar la metodología

No.	VULNERABILIDAD	NIVEL DE RIESGO DESPUÉS						Porcentaje de riesgo
		Informativo	Bajo	Medio	Alto	Total	Probabilidad	
1	Exploración de directorios	0	0	0	0	**0**	**0,00%**	0,00%
2	Biblioteca JS vulnerable	0	0	0	0	**0**	**0,00%**	0,00%
3	Cookie sin bandera segura	0	0	0	0	**0**	**0,00%**	0,00%
4	Cookie sin atributo SameSite	0	0	0	0	**0**	**0,00%**	0,00%
5	Inclusión de archivos de origen JavaScript entre dominios	0	0	0	0	**0**	**0,00%**	0,00%
6	Divulgación de la marca de hora – Unix	0	0	0	0	**0**	**0,00%**	0,00%
7	El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP "X-Powered-By"	0	0	0	0	**0**	**0,00%**	0,00%
8	Conjunto de encabezados incompleto o sin control de caché	0	0	0	0	**0**	**0,00%**	0,00%
9	Las páginas seguras incluyen contenido mixto	0	0	0	0	**0**	**0,00%**	0,00%
10	Falta el encabezado X-Content-Type-Options	0	0	0	0	**0**	**0,00%**	0,00%
11	Falta el encabezado Content-Type	0	0	0	0	**0**	**0,00%**	0,00%
12	Divulgación de información - Comentarios sospechosos	4	0	0	0	**4**	**1,17%**	0,58%
13	Incompatibilidad de caracteres (Encabezado contra Conjunto de Caracteres de Tipo de Contenido Meta).-	0	0	0	0	**0**	**0,00%**	0,00%
	Pocentaje por Riesgo de Vulneabilidad	**1,17%**	**0,00%**	**0,00%**	**0,00%**			
	Total	**4**	**0**	**0**	**0**	**4**	**1,17%**	

Realizado por: Ramírez, J. 2021

En la tabla 4-4 se evidencia la mejora lograda al implementar las contramedidas a cada una de las ocurrencias de las vulnerabilidades; teniendo como resultado la eliminación de las amenazas con niveles de riesgo *Alto*, *Medio* y *Bajo*, por ende, se eliminó el porcentaje de riesgo; quedando solo un 1,17% que representan amenazas de tipo *Informativo*.

A continuación, se muestran los resultados anteriores, pero esta vez resumidos por categoría de amenazas según OWASP 2021 antes de aplicar la metodología, tabla 9-3:

Tabla 5-4: Nivel de Riesgo y Probabilidad de ocurrencia encontradas por categorías de amenazas según OWASP 2021 antes de aplicar la metodología

CATEGORÍAS DE AMENZAS ENCONTRADAS	RIESGO ANTES					
	Informativo	**Bajo**	**Medio**	**Alto**	**Total**	**Porcentaje**
A02: Fallos criptográficos	0	3	0	0	**3**	**0,88%**
A04: Diseño inseguro	82	93	2	0	**177**	**51,75%**
A06: Componentes vulnerables y obsoletos	0	154	8	0	**162**	**47,37%**
Pocentaje por Riesgo de Vulneabilidad	**23,98%**	**73,10%**	**2,92%**	**0,00%**		
Total	**82**	**250**	**10**	**0**	**342**	**100,00%**

Realizado por: Ramírez, J. 2021

Las categorías de vulnerabilidades halladas se describen en la tabla 5-4, donde se observan que las amenazas correspondientes a códigos que originan Diseño Inseguro son un 51,75%, seguido del uso de Componentes Vulnerables y Obsoletos con un 47,37%. Es importante señalar que las amenazas de tipo de Fallos Criptográfico son mínimas y de impacto bajo.

Así mismo, luego de la intervención los resultados son los siguientes:

Tabla 6-4: Nivel de Riesgo y Probabilidad de ocurrencia encontradas por categorías de amenazas según OWASP 2021 antes de aplicar la metodología

CATEGORÍAS DE VULNERABILIDADES ENCONTRADAS	RIESGO DESPUÉS					
	Informativo	**Bajo**	**Medio**	**Alto**	**Total**	**Porcentaje**
A02: Fallos criptográficos	0	0	0	0	**0**	**0,00%**
A04: Diseño inseguro	4	0	0	0	**4**	**100,00%**
A06: Componentes vulnerables y obsoletos	0	0	0	0	**0**	**0,00%**
Pocentaje por Riesgo de Vulneabilidad	**1,17%**	**0,00%**	**0,00%**	**0,00%**		
Total	**4**	**0**	**0**	**0**	**4**	**100,00%**

Realizado por: Ramírez, J. 2021

Como se mencionó anteriormente y como se evidencia en la tabla 6-4, se logró eliminar las vulnerabilidades de las categorías Fallos Criptográficos y, Componentes Vulnerables y Obsoletos, quedando pocas amenazas de Diseño Inseguro (4), todas con un nivel de riesgo *Informativo*.

Por último, se presentan los cuadros comparativos del antes y el después de esta investigación, donde se demuestra que se alcanzó el objetivo de mejorar la seguridad de las aplicaciones web académicas:

Tabla 7-4: Comparativo de ocurrencia de vulnerabilidades a aplicaciones web académicas antes y después de aplicar la metodología.

APLICACIÓN WEB	Vulnerabilidades			
	Antes	Después	Corregidas	% Corregidas
estudiante.utelvt.edu.ec/	205	1	204	99,51%
administrativo.utelvt.edu.ec/	72	1	71	98,61%
docente.utelvt.edu.ec/	60	1	59	98,33%
siad.utelvt.edu.ec/	5	1	4	80,00%
Total	**342**	**4**	**338**	**98,83%**

Realizado por: Ramírez, J. 2021

En la tabla 7-4 se observa que se logró corregir el 98,83% de las ocurrencias de las vulnerabilidades, demostrando la efectividad del trabajo realizado.

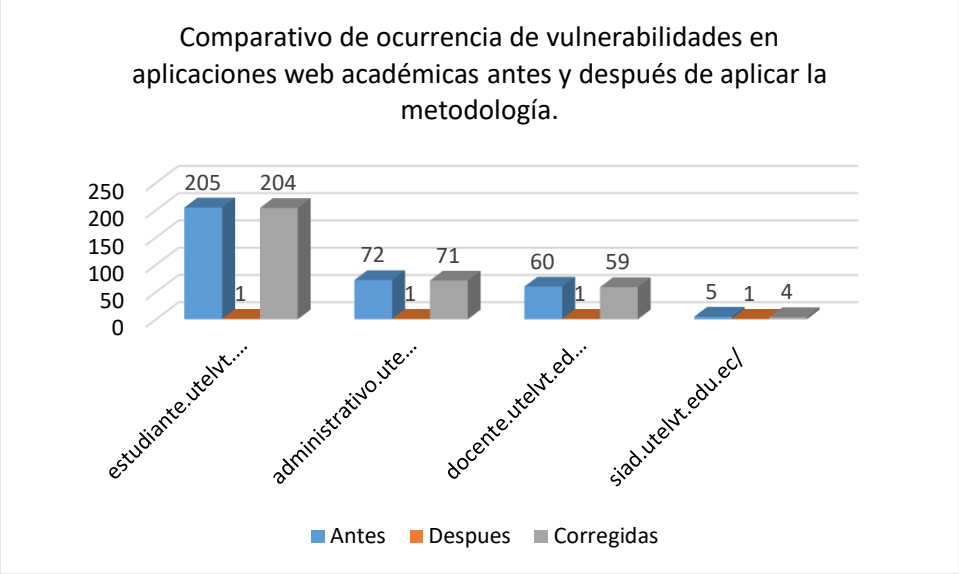


Figura 1-4: Comparativo de ocurrencia de vulnerabilidades en aplicaciones web académicas de la UTELVT antes y después de aplicar la metodología

Realizado por: Ramírez, J. 2021

En la Figura 1-4 se puede observar que antes de aplicar la metodología de solución en la aplicación web académica *estudiante.utelvt.edu.ec/* se encontraron 205 vulnerabilidades y después de aplicar la metodología se corrigieron 204, de igual manera en la aplicación *administrativo.utelvt.edu.ec/* se detectaron 72 vulnerabilidades y se corrigieron 71; además, en la aplicación *docente.utelvt.edu.ec/* inicialmente se encontraron 60 vulnerabilidades y corregidas 59; finalmente, en la aplicación *siad.utelvt.edu.ec/* se encontraron 5 vulnerabilidades y corregidas 4, se puede apreciar que la metodología para mitigar las vulnerabilidades de las aplicaciones web académicas de la UTELVT fue exitosa.

Tabla 8-4: Comparativa de Nivel de Riesgo y Probabilidad de ocurrencia encontradas por categorías de amenazas según OWASP 2021 antes de aplicar la metodología

CATEGORÍAS DE AMENAZAS ENCONTRADAS	Incidencias			
	Antes	Después	Corregidas	% Corregidas
A02: Fallos criptográficos	3	0	3	100,00%
A04: Diseño inseguro	177	4	173	97,74%
A06: Componentes vulnerables y obsoletos	162	0	162	100,00%
Total	**342**	**4**	**338**	**98,83%**

Realizado por: Ramírez, J. 2021

La información de la tabla 8-4, nos indica el número de intervenciones que se realizaron para corregir el código en cada archivo donde se generaba la vulnerabilidad y estas fueron inicialmente 342, reduciendo significativamente a 338 después de la intervención.

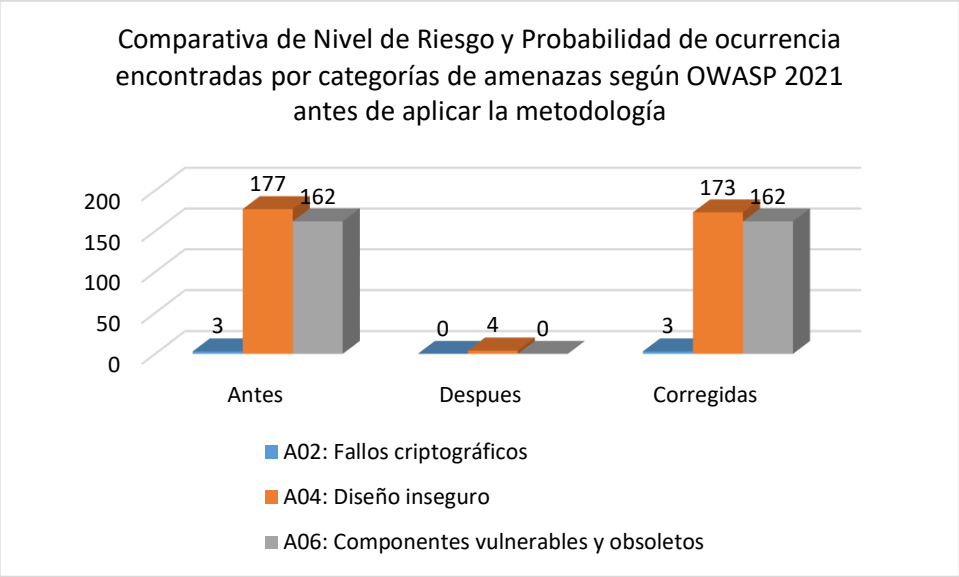


Figura 2-4: Comparativa de Nivel de Riesgo y Probabilidad de ocurrencia encontradas por categorías de amenazas según OWASP 2021 antes de aplicar la metodología

Realizado por: Ramírez, J. 2021

Se puede apreciar en la Figura 2-4 una reducción significativa en las vulnerabilidades corregidas, categorizadas en *Diseño inseguro* pasando de 177 a 173; además, se redujeron a cero (0) las vulnerabilidades encontradas en *Componentes vulnerables y obsoletos* y, *Fallos criptográficos*.

Tabla 9-4: Comparativo de ocurrencia de vulnerabilidades por nivel de riesgo en aplicaciones web académicas de la UTELVT antes y después de aplicar la metodología

NIVEL DE RIESGO VULNERABILIDADES ENCONTRADAS	INCIDENCIAS			
	Antes	Después	Corregidas	% Corregidas
Alto	0	0	0	0,00%
Medio	10	0	10	100,00%

Medio	250	0	250	100,00%
Informativo	82	4	78	95,12%
Total	**342**	**4**	**338**	**98,83%**

Realizado por: Ramírez, J. 2021

Una vez realizada la intervención se logró corregir las vulnerabilidades con nivel de riesgo Alto, Medio y Bajo. Quedando solamente vulnerabilidades de tipo informativo, las cuales tienen un impacto mínimo de riesgo en las aplicaciones intervenidas.

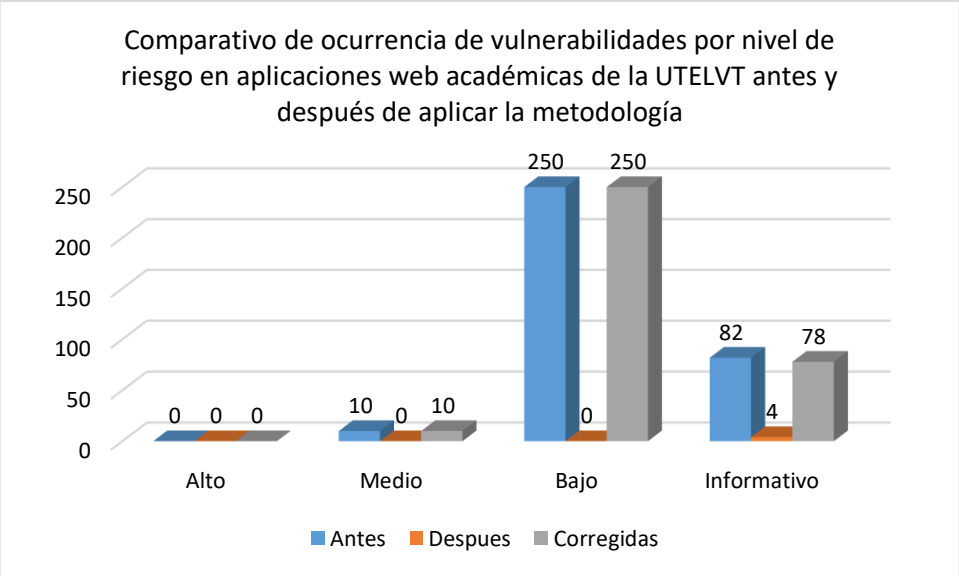


Figura 3-4: Comparativo de ocurrencia de vulnerabilidades por nivel de riesgo en aplicaciones web académicas de la UTELVT antes y después de aplicar la metodología

Realizado por: Ramírez, J. 2021

En la Figura 3-4 se aprecia la disminución significativa de ocurrencia de vulnerabilidades por nivel de riesgo *Alto, Medio y Bajo a cero (0)*, quedando unas pocas en el nivel de riesgo *Informativo* de 4 vulnerabilidades.

4.2. Verificación de hipótesis

Hipótesis

H_o= Hipótesis nula

H_1= Hipótesis alterna

H_o= Con la implementación de un sistema de fortalecimiento de la seguridad no se reducirá las vulnerabilidades de las aplicaciones web académicas de la UTELVT.

H_1= Con la implementación de un sistema de fortalecimiento de la seguridad se reducirá las vulnerabilidades de las aplicaciones web académicas de la UTELVT.

El nivel de significación escogido para la presente investigación es del 5% (0.05).

Prueba estadística Chi cuadrado

Para la verificación de la hipótesis se escogió la prueba Chi cuadrado; para este efecto, se tabuló los resultados obtenidos antes y después de la intervención,

Tabla 10-4: Frecuencia de valores observados

Valores Observados			
Vulnerabilidades	**Escenario 1**	**Escenario 2**	**Total**
Total de Vulnerabilidades detectadas x aplicación web	342	4	346
Total de Vulnerabilidades eliminadas x aplicación web	0	338	338
Total	**342**	**342**	**684**

Fuente: La investigación

Realizado por: Ramírez, J. 2021

Frecuencia esperada

$$Eij = \frac{(Total\ Columna) * (Total\ Fila)}{Suma\ Total}$$

Tabla 11-4: Frecuencia de valores esperados

Valores Esperados			
Vulnerabilidades	**Escenario 1**	**Escenario 2**	**Total**
Total de Vulnerabilidades detectadas x aplicación web	173	173	346
Total de Vulnerabilidades eliminadas x aplicación web	169	169	338
Total	**342**	**342**	**684**

Realizado por: Ramírez, J. 2021

Se calcula la prueba estadística del **Chi-cuadrado (X^2)**, de acuerdo a la siguiente fórmula:

$$x^2 calc = \sum \frac{(Oij - Eij)^2}{Eij}$$

Dónde:

Oij: son las frecuencias observadas, es el número de casos observados clasificados en la fila i columna j

Eij: son las frecuencias esperadas, es el número de casos esperados correspondiente a cada fila y columna

Tabla 12-4: Cálculo de Chi cuadrado

$x^2calc = \sum \frac{(Oij - Eij)^2}{Eij}$			***Oij***	***Eij***	***O-Eij***	***(Oij - Eij)²***	***(Oij - Eij)²*** / ***Eij***
Vulnerabilidades Escenario 1	detectadas	–	342	173	169	28.561	165,09
Vulnerabilidades Escenario 1	eliminadas	–	0	169	169	28.561	169,00
Vulnerabilidades Escenario 2	detectadas	–	4	173	169	28.561	165,09
Vulnerabilidades Escenario 2	eliminadas	–	388	219	169	28.561	130,41
Total			**23**	**23**		***X² calc*** **=**	**629,59**

Fuente: La investigación
Realizado por: Ramírez, J. 2021

Luego, Se determina los **Grados de Libertad**, para lo cual se utiliza la siguiente fórmula:

gl= (f-1)*(c-1)

Dónde:

f = Número de filas

c = Número de columnas

gl = (2-1)*(2-1)

gl= 1

Nivel de confianza de 95 % = 0.95 y Nivel de significancia del 5% = 0.05

El nivel de significancia es el error que se puede cometer al rechazar la Hipótesis Nula Ho siendo verdadera, generalmente se trabaja con un 5%, es decir 0,05, que indica que hay una probabilidad (p) del 0,95 de que la Hipótesis Nula Ho sea verdadera.

Por lo tanto, el valor crítico según la tabla de distribución (**ANEXO C**) con los valores de 1 grado de libertad y 0,95 de nivel de confianza se tiene:

X² Critico = 3,84

Criterio de decisión

Se acepta la H0: si X² calc ≤ X² Critico, caso contrario se rechaza la H0 y se acepta la H1.

El valor de X^2Critico = 3,84 < X^2 Calc= 629,59

De conformidad a lo establecido en la regla de decisión se rechaza la hipótesis nula y se acepta la hipótesis alterna, es decir, Con la implementación de un sistema de fortalecimiento de la seguridad se reducirá las vulnerabilidades de las aplicaciones web académicas de la UTELVT.

4.3. Discusión

La presente investigación logró identificar todas las vulnerabilidades de las aplicaciones web académicas de la UTELVT a través de la metodología OWASP, tal como Rincón y Albarracín en (2018) con su investigación de tipo aplicada analizaron y evaluaron la seguridad informática de la página web publicada en hosting gratuito de la institución técnica de Firavitoba, para la detección y remediación de vulnerabilidades y riesgos en la información. Dando como resultado la tabulación de las posibles soluciones

Del mismo modo se logró tomar acciones correctivas evitando posibles fallos y perdida de información en base a las vulnerabilidades encontradas en el sistema analizado; igual que en el estudio realizado por Marulanda en (2018) ejecutó un estudio descriptivo de corte transversal, cuantitativo, en el sistema e-commerce siembraviva.com y Sánchez en (2017) para la página web de la dirección de educación a distancia y virtual de la universidad técnica de Ambato. Ambos utilizaron metodología OWASP.

De la misma manera que en el estudio de Montalvo en (2017) se plantearon dos escenarios, el primero la situación actual de las aplicaciones web académicas de la UTELVT y el segundo luego de implementar el sistema de fortalecimiento de la seguridad y se realizó la evaluación del riesgo mediante la aplicación de una metodología basada en OWASP 2021.

CAPÍTULO V

5. METODOLOGÍA

En este capítulo se describe la metodología propuesta, desarrollada para el escaneo de vulnerabilidades de las aplicaciones web académicas de la UTELVT, además el sistema de la seguridad e integridad de aplicaciones web académicas en estudio.

La elección de la metodología se describe en la sección 2.1.2. (Metodología de pruebas de penetración) se utiliza como base la metodología OWASP, que contiene una serie de guías, prácticas de codificación, controles de seguridad, entre otros; que sirven como base para crear el modelo que utilizará nuevas técnicas de programación segura en PHP.

El siguiente esquema general muestra el diseño de fortalecimiento de la seguridad e integridad de las aplicaciones web académicas de la UTELVT:

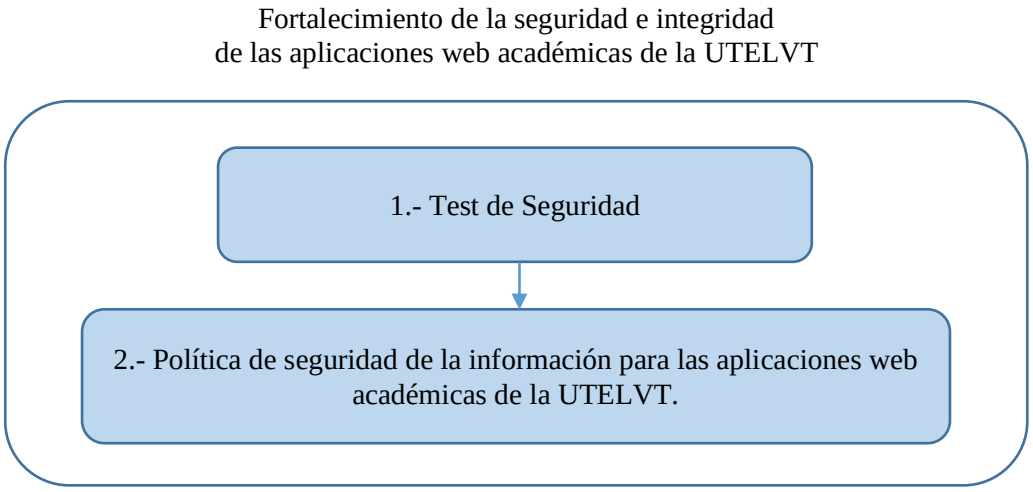


Figura 1-5: Fortalecimiento de la seguridad e integridad de las aplicaciones web académicas de la UTELVT.

Realizado por: Ramírez, J. 2021

5.1. Diseño del sistema de fortalecimiento de la seguridad e integridad de aplicaciones web académicas de la UTELVT

5.1.1. Test de Seguridad

Las fases implementadas para la reducción de vulnerabilidades en las aplicaciones web académicas de la UTELVT son las siguientes:

1. Identificación de Aplicaciones
2. Recopilación de información
3. Escaneo de vulnerabilidades de las páginas web académicas de la UTELVT.
4. Análisis de Vulnerabilidades
5. Definir Contramedidas
6. Implementación de Contramedidas
7. Test de seguridad Post Intervención
8. Pruebas de Funcionalidad
9. Puesta en Producción del sistema fortalecido

A continuación, se las detallan:

Fase 1: Identificación de Aplicaciones

Se identifica el grupo de páginas web académicas de la UTELVT en producción, que serán analizadas mediante test de penetración, con la finalidad de encontrar vulnerabilidades y de esta manera poder protegerlas. En la tabla 23 se describen las consideradas.

Tabla 1-5: Aplicaciones web académicas de la UTELVT

Grupo Academia	Dirección web	Estado
Matrícula estudiantil	https://estudiante.utelvt.edu.ec/	En servicio
Gestión académica	https://administrativo.utelvt.edu.ec/	En servicio
Docentes	https://docente.utelvt.edu.ec/	En servicio
SIAD	https://siad.utelvt.edu.ec/	En servicio

Realizado por: Ramírez, J. 2021

Fase 2: Recopilación de información.

Se centra en recoger toda la información como sea posible sobre la aplicación web objetivo.

Mediante el comando **ping** de Windows. A través de esta instrucción desde la consola de Windows se hace ping a las direcciones URL de cada página web académica de la UTELVT.

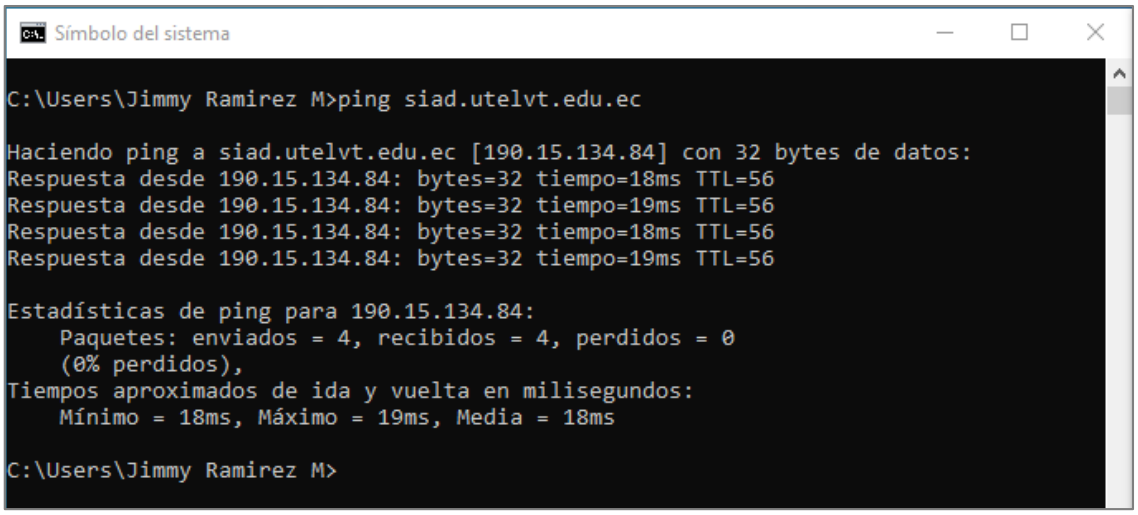


Figura 2-5*:* Ejecución del comando ping desde la consola de Windows

Realizado por: Ramírez, J. 2021

Se obtuvo las direcciones IP de cada uno de las páginas webs académicas de la UTELVT, y se pudo verificar que las páginas web de la academia están contenidas en el Servidor 1 de los dos servidores detectados: Servidor 1 (IP: 190.15.134.84), Servidor 2 (IP: 190.15.134.87) como se muestra en la Tabla 2-5.

Tabla 2-5: Direcciones IP de las páginas web académicas de la UTELVT

Grupo academia	Dirección web	Servidor	Dirección IP
Matrícula estudiantil	https://estudiante.utelvt.edu.ec/	Servidor 1	190.15.134.84
Gestión académica	https://administrativo.utelvt.edu.ec/	Servidor 1	190.15.134.84
Docentes	https://docente.utelvt.edu.ec/	Servidor 1	190.15.134.84
SIAD	https://siad.utelvt.edu.ec/	Servidor 1	190.15.134.84

Realizado por: Ramírez, J. 2021

Fase 3: Escaneo de vulnerabilidades de las páginas web académicas de la UTELVT.

Para el escaneo de vulnerabilidades se utilizó la herramienta OWASP ZAP 2.11.0 para Windows.

En la Figura 3-5, se muestra el escaneo de vulnerabilidades con la aplicación OWASP ZAP a través de la dirección URL: **http://sistemas.utelvt.edu.ec/**. en este sitio se encuentran todas las aplicaciones web que la universidad ofrece, como la academia, bienestar estudiantil, vinculación administrativo y repositorio. En la figura 3-5 se muestra el resultado del escaneo. Se observa que una de las vulnerabilidades encontradas es: ¡DOCTYPE HTML, la cual representa la declaración del **DOCTYPE**. Es decir, es un documento HTML 4.01 STRICT, de tipo público.

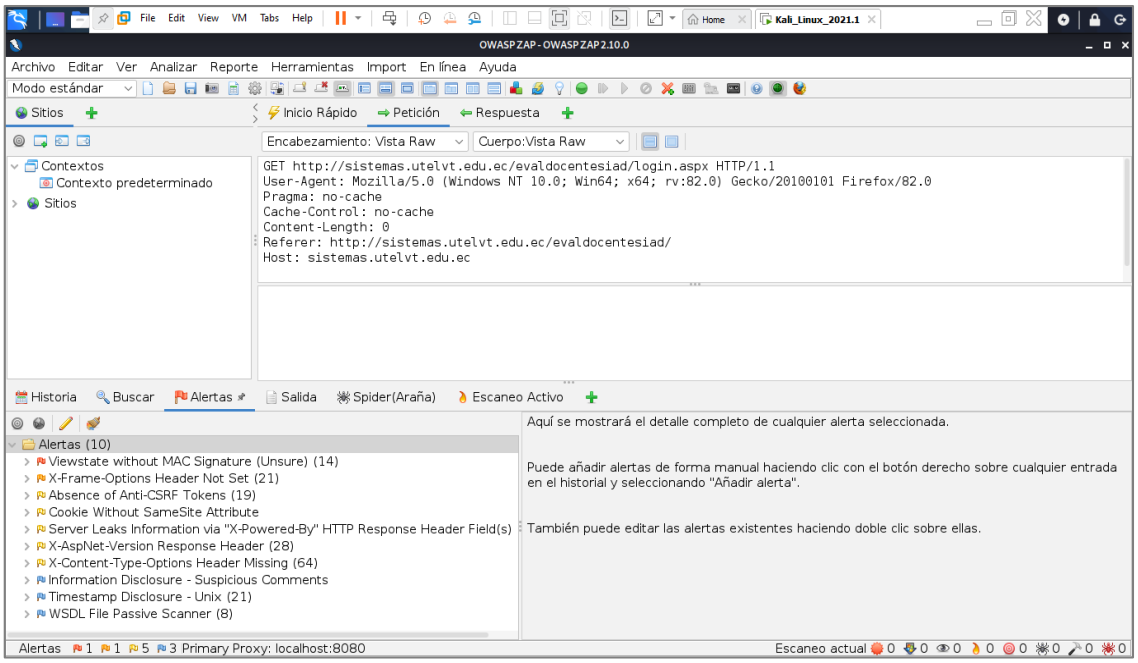


Figura 3-5: Escaneo de la página principal servicios en línea de la UTELVT

Realizado por: Ramírez, J. 2021

La figura 4-5, presenta el escaneo de vulnerabilidades de la aplicación web sistema integrado académico docente (**SIAD**) a través de la dirección URL: **https://siad.utelvt.edu.ec/**. Empleando la herramienta OWASP ZAP. En esta figura se muestra el resultado del escaneo. Se observa que las vulnerabilidades encontradas son bajas (Alertas bajas).

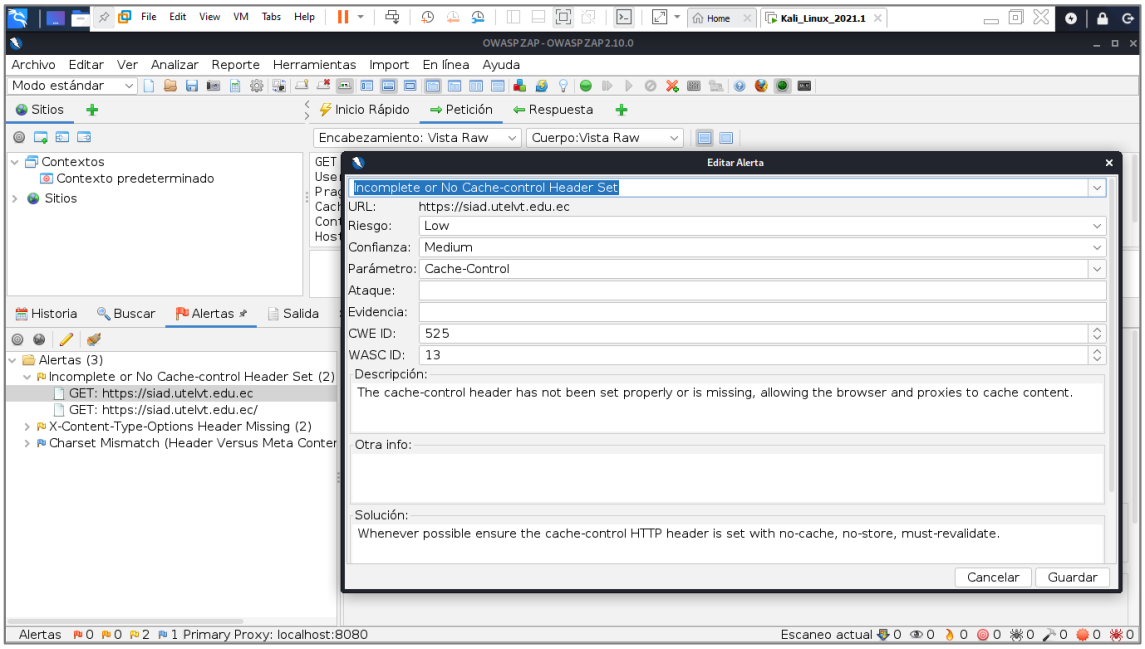


Figura 4-5: Escaneo de vulnerabilidades de SIAD mediante OWASP ZAP

Realizado por: Ramírez, J. 2021

Del mismo modo, la figura 5-5, nos grafica los detalles de otras vulnerabilidades encontradas es de tipo encabezado HTTP de respuesta X-Content-Type-Options. Este marcador MIME (Multipurpose Internet Mail Extensions), es utilizado en servidores para que indiquen los tipos MIME (Multipurpose Internet Mail Extensions) anunciados en los encabezados.

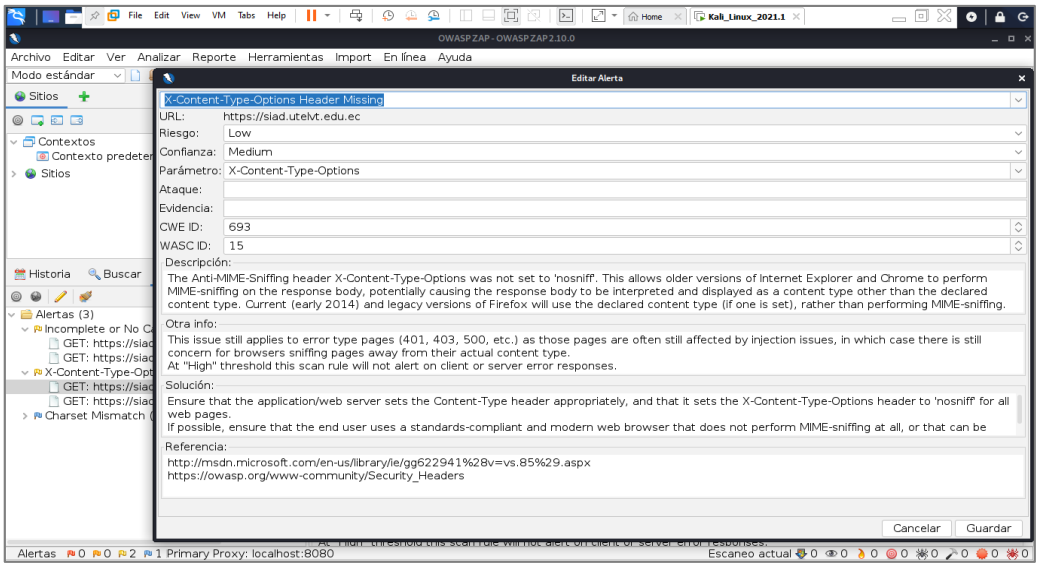


Figura 5-5: Vulnerabilidades de tipo encabezado HTTP de respuesta X-Content-Type-Options
Realizado por: Ramírez, J. 2021

Continuando, en la figura 6-5, podemos observar el escaneo de vulnerabilidades de la aplicación web matrícula estudiantil a través de la URL: **https://estudiante.utelvt.edu.ec/**, en esta figura se muestra el resultado del escaneo. Se observa 7 vulnerabilidades críticas (Alertas Rojas). Estas son de SQL Injection y 4 en vulnerabilidades de Librería.

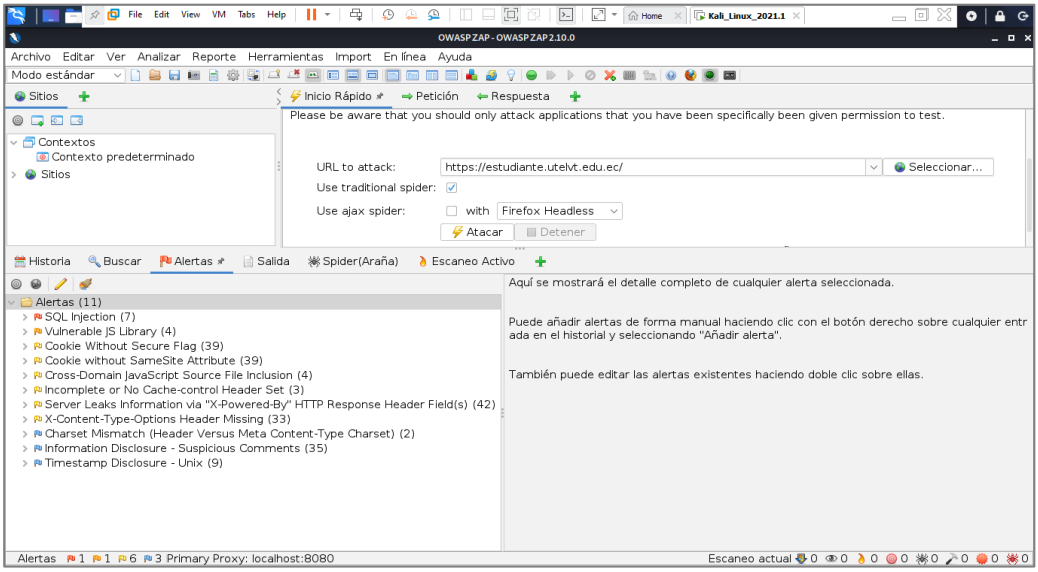


Figura 6-5: Escaneo de vulnerabilidades de la aplicación web matrícula estudiantil.
Realizado por: Ramírez, J. 2021

Fase 4: Análisis de Vulnerabilidades

Se realiza el resumen y análisis de las vulnerabilidades encontradas en el escaneo de las aplicaciones web académicas de la UTELVT en producción.

Después del escaneo de las aplicaciones web académicas de la UTELVT en producción, se encontraron trece (13) tipos vulnerabilidades, las cuales se muestran en la tabla 27-5

Tabla 27-5: Vulnerabilidades encontradas en las aplicaciones web de la UTELVT en producción

Código	Alerta CWE / WASC ID (Enumeración de debilidades comunes)	Clasificación (Top 10 OWASP)	Riesgo / confianza	Solución	N. Incidencias
AM001	**Exploración de directorios. -** Es posible ver la lista de directorios. La lista de directorios puede revelar scripts ocultos, incluir archivos, archivos fuente de copia de seguridad, etc. a los que se puede acceder para leer información confidencial.	Diseño inseguro	Medio / Medio	Deshabilite la exploración de directorios. Si esto es necesario, asegúrese de que los archivos enumerados no induzcan riesgos.	2
AM002	**Biblioteca JS vulnerable. -** El bootstrap de biblioteca identificado, versión 3.3.7 es vulnerable.	Componentes vulnerables y obsoletos	Medio / Medio	Actualice a la última versión de bootstrap.	8
AM003	**Cookie sin bandera segura. -** Se ha establecido una cookie sin la bandera segura, lo que significa que se puede acceder a la cookie a través de conexiones no cifradas.	Configuración incorrecta de seguridad	Bajo / Medio	Siempre que una cookie contenga información confidencial o sea un token de sesión, siempre debe pasarse utilizando un canal cifrado. Asegúrese de que el indicador seguro esté configurado para las cookies que contienen dicha información confidencial.	44
AM004	**Cookie sin atributo SameSite.-** Una cookie ha sido enviada sin el atributo SameSite, lo que significa que la cookie puede ser enviada como un resultado de una solicitud 'cross-site'.	Configuración incorrecta de seguridad	Bajo / Medio	Asegúrese que el atributo SameSite está establecido como 'lax' o idealmente 'strict' para todas las cookies.	49
AM005	**Inclusión de archivos de origen JavaScript entre dominios. -** La página incluye uno o más archivos de script de un dominio de terceros.	Configuración incorrecta de seguridad	Bajo / Medio	Asegúrese de que los archivos de origen de JavaScript se carguen solo desde fuentes de confianza y que los usuarios finales de la aplicación no puedan controlar las fuentes.	4
AM006	**Divulgación de la marca de hora – Unix** Una marca de tiempo ha sido divulgada por el servidor de la aplicación/el navegador – Unix.	Diseño inseguro	Bajo / Bajo	Confirmar manualmente que los datos de marca de hora no son sensibles, y que los datos no pueden ser agregados a patrones explotables de divulgación.	25
AM007	**El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP ''X-Powered-By''** El servidor de la web/aplicación está divulgando información mediante uno o más encabezados de respuesta HTTP "X-Powered-By". El acceso a tal información podría facilitarles a los atacantes la identificación de otros marcos / componentes de los que su aplicación web depende y las vulnerabilidades a las que pueden estar sujetos tales componentes.	Configuración incorrecta de seguridad	Bajo / Medio	Asegúrese que su servidor web, servidor de aplicación, equilibrador de carga, etc. está configurado para suprimir encabezados "X-Powered-By".	57

AM008	**Conjunto de encabezados incompleto o sin control de caché** Encabezado de control de caché no se ha configurado correctamente o falta, lo que permite que el navegador y los proxies almacenen en caché el contenido.	Diseño inseguro	Bajo / Medio	Siempre que sea posible, asegúrese de que el encabezado HTTP de control de caché esté configurado con no-cache, no-store, must-revalidate.	5
AM009	**Las páginas seguras incluyen contenido mixto. -** La página incluye contenido mixto, es decir, contenido al que se accede a través de HTTP en lugar de HTTPS.	Fallos criptográficos	Baja / Medios	Una página que está disponible a través de SSL / TLS debe estar compuesta completamente de contenido que se transmite a través de SSL / TLS. La página no debe contener ningún contenido que se transmita a través de HTTP sin cifrar.	3
AM010	**Falta el encabezado X-Content-Type-Options.-** El encabezado Anti-MIME-Sniffing X-Content-Type-Options no se estableció en 'nosniff'. Esto permite que las versiones anteriores de Internet Explorer y Chrome realicen el rastreo MIME en el cuerpo de la respuesta, lo que podría hacer que el cuerpo de la respuesta se interprete y se muestre como un tipo de contenido distinto del tipo de contenido declarado.	Diseño inseguro	Bajo / Medio	Asegúrese de que la aplicación/servidor web establece el encabezado Content-Type correctamente y que establece el encabezado X-Content-Type-Options en 'nosniff' para todas las páginas web.	63
AM011	**Falta el encabezado Content-Type.-** El encabezado Content-Type faltaba o estaba vacío.	Diseño inseguro	Informativa / Medio	Asegúrese de que cada página establece el valor de tipo de contenido específico y apropiado para el contenido que se está entregando.	1
AM012	**Divulgación de información - Comentarios sospechosos. -** La respuesta parece contener comentarios sospechosos que pueden ayudar a un atacante. Nota: Las coincidencias realizadas dentro de bloques de script o archivos son contra todo el contenido, no solo contra los comentarios.	Diseño inseguro	Informativa / Bajo	Eliminar todos los comentarios que devuelvan información que podría ayudar a un atacante y arreglar cualquier problema subyacente al que se refieran.	79
AM013	**Incompatibilidad de caracteres (Encabezado contra Conjunto de Caracteres de Tipo de Contenido Meta**).- Esta comprobación identifica las respuestas en las que el encabezado HTTP Content-Type declara un conjunto de caracteres diferente del conjunto de caracteres definido por el cuerpo del HTML o XML Cuando hay una discrepancia entre el encabezado HTTP y el cuerpo del contenido, los navegadores web pueden ser forzados a un modo de detección de contenido no deseado para determinar el conjunto de caracteres correcto del contenido.	Diseño inseguro	Informativa / Bajo	Forzar UTF-8 para todo el contenido de texto tanto en el encabezado HTPP y etiquetas meta en HTML o declaraciones de codificación en XML	2

Realizado por: Ramírez, J. 2021

Fase 5: Definir Contramedidas

Se realizó una investigación documental acerca de cada una de las posibles soluciones para la mitigación de las vulnerabilidades descritas en la tabla 29-5. En dicha investigación se tomaron en cuenta características relevantes de las aplicaciones académicas de la UTELVT y los servidores web donde residen; esto con la finalidad de seleccionar la contramedida más adecuada.

Fase 6: Implementación de Contramedidas

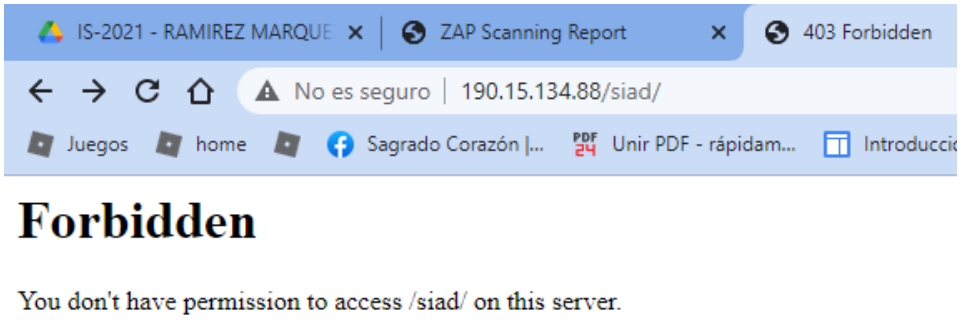
Para esta fase, se preparó un ambiente de pruebas con una réplica del servidor web en producción con las siguientes características: En un computador de escritorio Core I7 con Sistema Operativo anfitrión Windows 10 Pro; y un servidor virtualizado en VMware con sistema operativo Centos 7, lenguaje de programación PHP 7.3, Servidor Web Apache 2, sistema gestor de base de datos (SGBD) Mariadb y las aplicaciones web académicas analizadas, con una copia de los directorios de las aplicaciones web académicas de la UTELVT.

Los detalles de la puesta en marcha del servidor de pruebas de las aplicaciones web de la UTELVT se encuentran en el **ANEXO A.**

La Guía de fortalecimiento de las aplicaciones web académicas de la UTELVT se detalla a continuación:

Tabla 28-5: Guía de fortalecimiento de las aplicaciones web académicas de la UTELVT

CÓDIGO	SOLUCIÓN DETALLADA A VULNERABILIDADES ENCONTRADAS

AM001	Para deshabilitar la exploración de directorios para todas las aplicaciones residentes en el servidor web, se realizará el siguiente procedimiento:
	1. Abrir el archivo de configuración de APACHE2, cuya ruta es: **/etc/httpd/conf/httpd.conf**
	2. Añadir la sentencia "***Options -Indexes***"; en la sección de configuración del directorio donde están almacenados los archivos de las aplicaciones web; en este caso "/**var/www/html**"; tal como muestra a continuación:
	`<Directory /var/www/html>`
	`  Options -Indexes`
	`</Directory>`
	3. Reiniciar el servidor web, utilizando el comando ***service httpd restart***
	4. Al intentar acceder a los directorios desde el navegador, mostrará esta respuesta.
	 
	NOTA: Existen otros métodos para restringir el listado de directorios a través del archivo ***.htaccess***

AM002

Para actualizar el Bootstrap se lo realiza de la siguiente manera:

Actualizar Bootstrap en YII 2:

1. Instalar Bootstrap5

```
php composer.phar require --prefer-dist yiisoft/yii2-bootstrap5
```

En caso de error realizar lo siguiente:

a. Editar el archivo ***composer.json*** ubicado en el directorio raíz del proyecto y aumentar el repositorio:

```
"repositories": [
    {
        "type": "composer",
        "url": "https://asset-packagist.org"
    }
]
```

n

b. Editar el archivo **config/web.php**, y añadir los respectivos aliases, como se muestra continuación:

```
$config = [
    ...
    'aliases' => [
        '@bower' => '@vendor/bower-asset',
        '@npm'   => '@vendor/npm-asset',
    ],
    ...
];
```

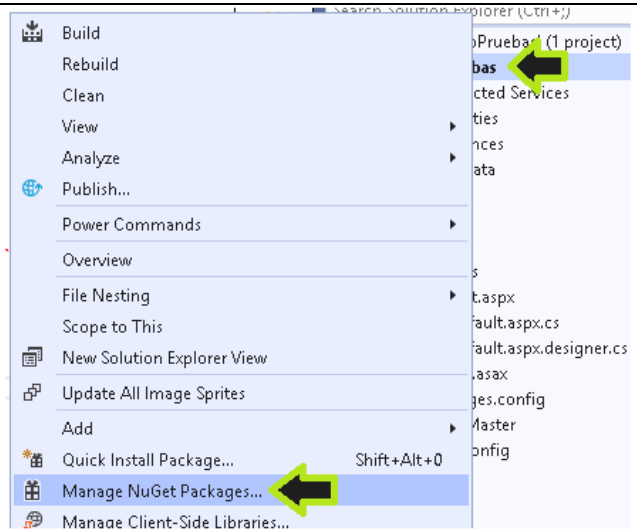
c. Ejecute el comando ***composer update***

d. Instalar Bootstrap5

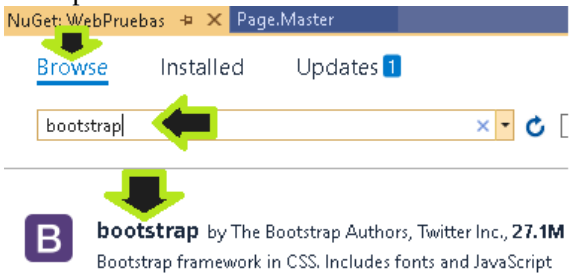
Actualizar Bootstrap en .NET

1. Actualizar Bootstrap, abriendo la herramienta ***Manage Nuget Packages:***

Sobre el nombre del proyecto se hace clic derecho y seleccionae ***Manage Nuget Packages***, como se muestra a continuación:

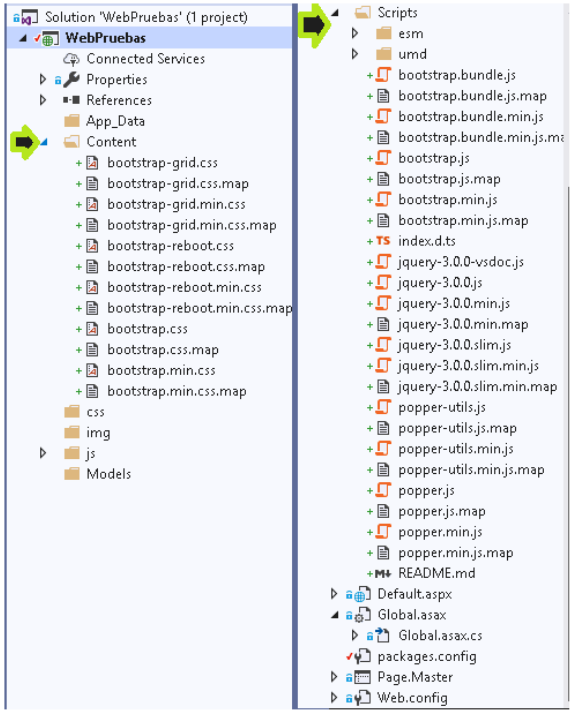


2. En la pestaña Browse del buscador se escribe Bootstrap, y seleccionar registro Bootstrap



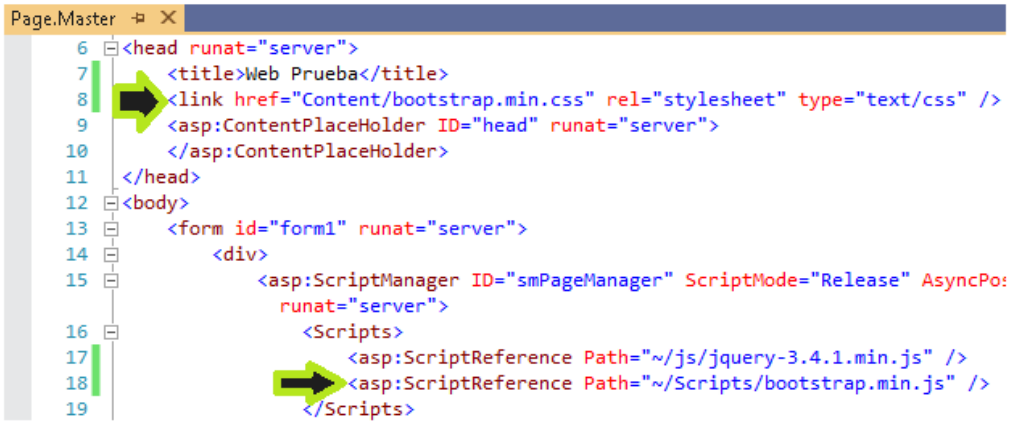
3. Selecciona el paquete de la versión a instalar:

4. La instalación genera dos carpetas en el proyecto:
   Content, hojas de estilo o css.
   Scripts, archivos javascript.



5. Referenciar Bootstrap
   Para utilizar el framework en todo el proyecto se agregan las referencias en archivo Master:

   En el **head** agrega el archivo ***css*** de nombre ***bootstrap.min.css***
   En la sección de **scripts** dentro del **body** agrega el **javascript** de nombre ***bootstrap.min.js***

<table>
<tr>
<td></td>
<td>


Actualizar código

Una vez instalada la nueva versión de Bootstrap se debe proceder a actualizar todos los elementos de presentación de los formularios que utilicen Bootstrap y CSS; como son Barras de menú, Botones, InputBox, DropBox, etc.
</td>
</tr>
<tr>
<td>AM003</td>
<td>
Para impedir el acceso a las Cookies a través de conexiones no cifradas, se debe realizar el procedimiento de acuerdo al tipo de servidor:

Procedimiento 1: Servidor web

1. Abrir el archivo de configuración de APACHE2, cuya ruta es: /etc/httpd/conf/httpd.conf

2. Añadir la sentencia al final del archivo según sea el caso:

Caso 1: Si el servidor web soporta solamente trafico en HTTP:


```
Header edit Set-Cookie ^(.*)$ $1;HttpOnly
```


Caso 2: Si el servidor web soporta solamente trafico en HTTPS:


```
Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure
```


3. Reiniciar el servidor web, utilizando el comando service httpd restart
</td>
</tr>
</table>

<table>
<tr><td></td><td>Procedimiento 2: Servidor PHP
1. En el archivo de configuración de PHP, php.ini; añadir la siguiente línea
session.cookie_httponly = True
2. Reiniciar el servidor web, utilizando el comando service httpd restart</td></tr>
<tr><td>AM004</td><td>Para evitar que las cookies se envíen en peticiones de sitios diferentes donde se originaron; se debe realizar la siguiente configuración:
Procedimiento 1: Servidor web
1. Abrir el archivo de configuración de APACHE2, cuya ruta es: /etc/httpd/conf/httpd.conf
2. Añadir la sentencia al final del archivo según sea el caso:
<pre>Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure;SameSite=Strict</pre>3.- Reiniciar el servidor web, utilizando el comando service httpd restart

Procedimiento 2: Servidor PHP
1. En el archivo de configuración de PHP, php.ini; añadir la siguiente línea
session.cookie_samesite="Strict"
2.- Reiniciar el servidor web, utilizando el comando service httpd restart</td></tr>
<tr><td>AM005</td><td>Para evitar los ataques que inyecten scripts de lado cliente, a través del sitio web, hasta los otros usuarios, se debe implementar los siguientes procesos:
Procedimiento 1: Servidor web
1. Abrir el archivo de configuración de APACHE2, cuya ruta es: /etc/httpd/conf/httpd.conf
2. Añadir la sentencia como se indica a continuación:
<pre>X-XSS-Protection: 1; mode=block</pre>3. Reiniciar el servidor web, utilizando el comando service httpd restart

Procedimiento 2: Archivo .htaccess
1. Añadir el siguiente segmento de codigo en el archivo .htaccess, ubicado en el directorio raíz de cada aplicación web; como muestra a continuación:
<pre>Header set X-XSS-Protection "1; mode=block"</pre>2. Reiniciar el servidor web, utilizando el comando service httpd restart</td></tr>
</table>

<table>
<tr><td></td><td>

Procedimiento 3: Recomendaciones en PHP

1. Para evitar los ataques XSS es necesario validar todas las entradas del codigo, se recomienda seguir las directrices de los ejemplos recomendados a continuación:

```php
# example 1
$name = htmlspecialchars($_GET['name']); // same for $_POST, $_REQUEST etc.

# example 2
$name = filter_input(INPUT_GET, 'name', FILTER_SANITIZE_SPECIAL_CHARS);

# example 3: URL Encoding
$input = urlencode($_GET['input']);
// or
$input = filter_input(INPUT_GET, 'input', FILTER_SANITIZE_URL);
echo '<a href="http://dev.com/page?input="' . $input . '">Link</a>';
```

2. Reiniciar el servidor web, utilizando el comando ***service httpd restart***

</td></tr>
<tr><td>AM006</td><td>Revisar todos los archivos que componen la aplicación web y eliminar cualquier línea de comentario que haga referencia a fechas específicas.

Las herramientas de escaneo de vulnerabilidades podrían señalar erróneamente valores numéricos de configuración (diseños como ancho de tablas, altos de filas, códigos numéricos de colores) como evidencia de este tipo de vulnerabilidad. En este caso ignorar la advertencia de vulnerabilidad.</td></tr>
<tr><td>AM007</td><td>Para desabilitar el envio de información a traves de los encabezados se debe realizar los siguientes procedimientos:

Procedimiento 1: Servidor web
1. Abrir el archivo de configuración de APACHE2, cuya ruta es: /etc/httpd/conf/httpd.conf

Alternativa 1: Añadir la sentencia “xPoweredByHeader = off”; como muestra a continuación:</td></tr>
</table>

	`xPoweredByHeader = off`
	
	Alternativa 2: Se puede configurar a traves de la directiva ***Header***, para esto añadir el siguiente el segmento de codigo; como muestra a continuación:
	`Header unset X-Powered-By`
	
	Alternativa 3: Para incrementar la seguridad, además de aplicar las alternativas 1 y 2, se recomienda desactivar todos los mensajes de respuest del servidor web, para esto añadir el siguiente el segmento de codigo; como muestra a continuación:
	`ServerSignature Off`
	`ServerTokens Prod`
	
	2. Reiniciar el servidor web, utilizando el comando ***service httpd restart***
	
	Procedimiento 2: Configurar PHP
	1. Localizar y abrir el archivo ***php.ini***
	2. Añadir "***expose_php = off***" ; tal como muestra la figura.
	`expose_php = off`
	
	3. Reiniciar el servidor web, utilizando el comando ***service httpd restart***

AM008	Para la configuración correcta del acceso a cache, se debe realizar los siguientes procedimientos, según sea el caso: ***Caso 1: Cabeceras en archivos .html*** 1. Acceder al archivo **html** a configurar 2. Identificar la sección de cabeceras, que está delimitada por las etiquetas **<head> ….. </head>** 3. En esta sección añadir una de las cabeceras alternativas ***meta***, como se describe a continuación: ***Alternativa 1:*** `<meta http-equiv="Cache-control" content="no-cache">` ***Alternativa 2:*** `<meta http-equiv="Pragma" content="no-cache" />`

	Caso 2: Cabeceras en archivos .php 1. Acceder al archivo ***.php*** a configurar 2. Identificar la sección de cabeceras, que está delimitada por las etiquetas ***header***, luego de la apertura de sección del código php *"**<?php header …… ?>**"* 3. En esta sección añadir una de las cabeceras alternativas como se describe a continuación: ***Alternativa 1:*** `header("Cache-Control: no-cache");` ***Alternativa 2:*** `header("Pragma: no-cache");` ***Caso 3: Archivos vacíos*** En algunas ocasiones las herramientas de diagnóstico presentan esta alerta ante la carencia de las cabeceras en archivos vacíos que se encuentran dentro de los directorios de las aplicaciones web. En este caso se recomienda eliminar estos archivos ya que no contribuyen al desempeño de la aplicación.

AM009	Para que las aplicaciones no se muestren como contenido mixto y las conexiones que están direccionadas en modo HTTP se reconfiguren al ser ejecutada por el navegador, se deben realizar las siguientes fases de solución:
	
	Fase 1: Configurar la actualización automática de conexiones inseguras
	Alternativa 1: Servidor Web
	1. Abrir el archivo de configuración de **APACHE2**, cuya ruta es: **/etc/httpd/conf/httpd.conf**
	2. En la sección de configuración del Host Añadir la sentencia como muestra a continuación:
	`Content-Security-Policy: upgrade-insecure-requests`
	
	Alternativa 2: Archivos html
	1. Acceder al archivo html a configurar
	2. Identificar la sección de cabeceras, que está delimitada por las etiquetas <**head> ….. </head**>
	3. En esta sección añadir una de las cabeceras alternativas **meta**, como se describe a continuación:
	`<meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests">`
	Fase 2: Configurar todas de conexiones inseguras.
	Este proceso se debe editar todas las líneas de código que configuren conexiones que estén en formato http://, como muestra a continuación:

<table>
<tr><td></td><td>

```
<img src="http://example.com/image.png">
<img src="http://not-example.com/image.png">
```

Luego, rescribir las **url** en formato https://, como muestra a continuación:

```
<img src="https://example.com/image.png">
<img src="https://not-example.com/image.png">
```

Realizar esta edición en todos los archivos que configuran la aplicación web.
</td></tr>
<tr><td>AM010</td><td>
Para proteger las vulnerabilidades de tipo MIME sniffing a las aplicaciones residentes en el servidor web, se realizará a traves de uno de los siguientes procedimientos, según sea la necesidad de protección:

Procedimiento 1: Servidor web
1. Abrir el archivo de configuración de APACHE2, cuya ruta es: /etc/httpd/conf/httpd.conf
Alternativa 1: Añadir la sentencia “X-Content-Type-Options: nosniff”; como muestra a continuación:

```
X-Content-Type-Options: nosniff
```

Alternativa 2: Añadir el siguiente el segmento de codigo; como muestra a continuación:

```
<IfModule mod_headers.c>
    Header set X-Content-Type-Options "nosniff"
</IfModule>
```

2. Reiniciar el servidor web, utilizando el comando service httpd restart

Procedimiento 2: Archivo .htaccess
1. Añadir el siguiente el segmento de codigo en el archivo .htaccess, ubicado en el directorio raíz de cada aplicación web; como muestra a continuación:

```
Header set X-Content-Type-Options nosniff
```

2. Reiniciar el servidor web, utilizando el comando service httpd restart
</td></tr>
</table>

<table>
<tr><td></td><td>

Procedimiento 3: Archivo index.php

1. Abrir el archivo ***index.php*** ubicado en el directorio raíz de cada aplicación web.
2. Añadir el siguiente el segmento de codigo en la sección cabecera, al inicio de la sección de código *php*, tal como muestra la figura.

```
<?php header('X-Content-Type-Options: nosniff'); ?>
```

3. Reiniciar el servidor web, utilizando el comando ***service httpd restart***

</td></tr>
<tr><td>AM011</td><td>

Para configurar la directiva *X-Content-Type-Options* se deben realizar los siguientes cambios:

Procedimiento 1: Archivos .html

1. Localizar y abrir los archivos html
2. En la sección cabecera **(<head>…..</head>**), añadir la siguiente meta:

```
<meta content="text/html; charset=UTF-8; X-Content-Type-Options=nosniff" http-equiv="Content-Type" />
```

3. Reiniciar el servidor web, utilizando el comando ***service httpd restart***

Procedimiento 2: Archivos .php

1. Abrir los archivos. **php**
2. Añadir el siguiente el segmento de código en la sección cabecera, al inicio de la sección de código php, como muestra a continuación:

```
<?php header('X-Content-Type-Options: nosniff'); ?>
```

3. Reiniciar el servidor web, utilizando el comando ***service httpd restart***

Procedimiento 3: Servidor web

1. Abrir el archivo de configuración de **APACHE2**, cuya ruta es: **/etc/httpd/conf/httpd.conf**

Alternativa 1: Añadir la sentencia "***X-Content-Type-Options: nosniff***"; tal como muestra a continuación:

```
X-Content-Type-Options: nosniff
```

Alternativa 2: Añadir el siguiente el segmento de codigo; tal como muestra la figura.

```
<IfModule mod_headers.c>
    Header set X-Content-Type-Options "nosniff"
</IfModule>
```

2. Reiniciar el servidor web, utilizando el comando ***service httpd restart***

</td></tr>
</table>

<table>
<tr><td>AM012</td><td>Revisar todos los archivos que componen la aplicación web y eliminar cualquier línea de comentario que haga referencia a fechas específicas.

Las herramientas de escaneo de vulnerabilidades podrían señalar erróneamente valores numéricos de configuración (diseños como ancho de tablas, altos de filas, códigos numéricos de colores) como evidencia de este tipo de vulnerabilidad. En este caso ignorar la advertencia de vulnerabilidad.</td></tr>
<tr><td>AM013</td><td>Procedimiento 1:
Para cambiar la configuración de la codificación de manera global para todas las aplicaciones residentes en el servidor web, se realizará el siguiente procedimiento:
1. Abrir el archivo de configuración de APACHE2, cuya ruta es: /etc/httpd/conf/httpd.conf
2. Añadir la sentencia “AddDefaultCharset utf-8”; en este archivo de configuración; como muestra a continuación:
<pre>AddDefaultCharset utf-8</pre>3. Reiniciar el servidor web, utilizando el comando service httpd restart

Procedimiento 2:
Para cambiar la configuración de la codificación en cada una de las aplicaciones de manera unitaria, se realizará el siguiente procedimiento:
Caso 1: Archivos *.html
1. Abrir el archivo a configurar (generalmente index.html).
2. En la sección cabecera, en la configuración de las meta, añadir la sentencia content="text/html;charset=UTF-8”; como muestra a continuación:
<pre><meta http-equiv="Content-Type" content="text/html;charset=UTF-8"></pre>3. Reiniciar el servidor web, utilizando el comando service httpd restart

Caso 2: Archivos *.php
1. Abrir el archivo a configurar (generalmente index.php).
2. En la sección cabecera, al inicio de la sección de código php, añadir la sentencia "content-type: text/html; charset: utf-8"; como muestra a continuación:
<pre><?php header("content-type: text/html; charset: utf-8"); ?></pre>3. Reiniciar el servidor web, utilizando el comando service httpd restart</td></tr>
</table>

Realizado por: Ramírez, J. 2021

Fase 7: Test de seguridad Post Intervención

Se realizaron las pruebas de penetración después de implementar las salvaguardas diseñadas en base a las vulnerabilidades encontradas en las aplicaciones web académicas de la UTELVT con la finalidad de probar el fortalecimiento de la seguridad de las aplicaciones web en estudio, todo esto en un ambiente de pruebas como se muestra en la figura 11.

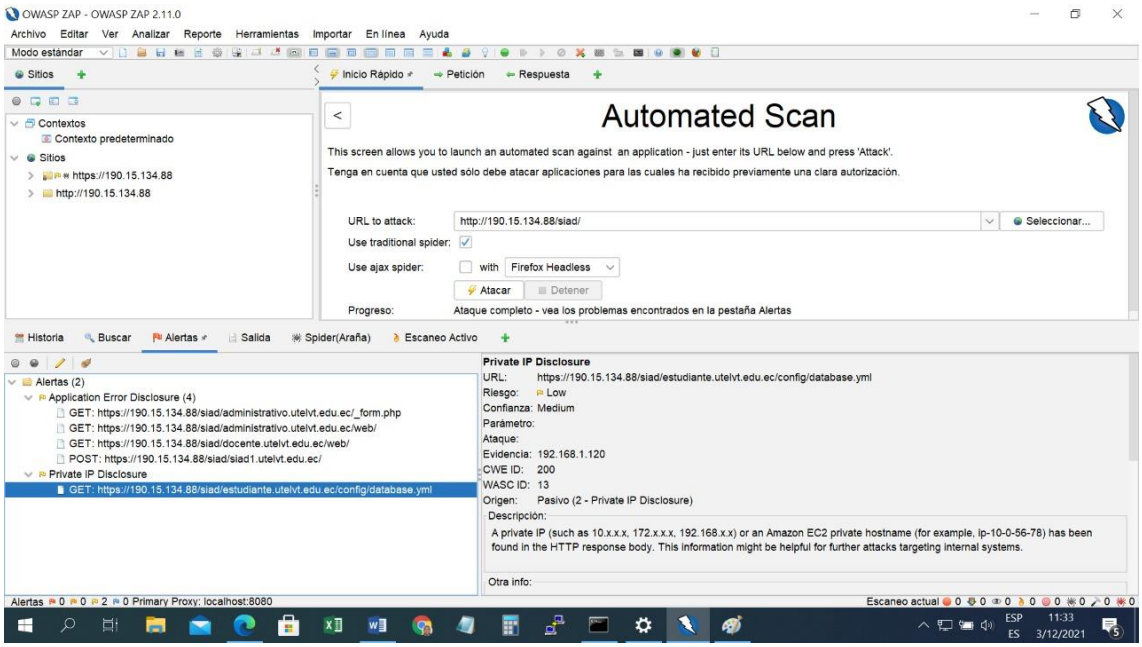


Figura 7-5: Escaneo de vulnerabilidades Escenario 2

Realizado por: Ramírez, J. 2021

Del escaneo se observa que existen amenazas que aún prevalecen, pero son de tipo informativo y están motivadas por los cambios que se realizaron a las aplicaciones web en el ambiente de pruebas, estas vulnerabilidades desaparecerán al ser implementadas en el ambiente de producción.

Fase 8: Pruebas de Funcionalidad

Una vez que se comprobó la corrección de las vulnerabilidades, se procedió a realizar una verificación de las aplicaciones, mediante la navegación en el browser examinando la funcionalidad y operatividad de todos sus componentes.

Fase 9: Puesta en Producción del sistema fortalecido

Cumplidas todas las fases anteriores, primero reconfiguramos el servidor web real, tomando en cuenta las contramedidas propuestas y luego copiamos las aplicaciones intervenidas.

5.1.2. Política de seguridad de la información para las aplicaciones web académicas de la UTELVT.

Alcance

La presente política de seguridad de la información se diseña con el objetivo de fortalecer la seguridad e integridad de las aplicaciones web académicas de la UTELVT. Su aplicación es obligatoria para docentes y estudiantes de la universidad.

Política de seguridad de la información

Control de Accesos.

a. Requerimientos para el Control de Acceso.

Política de Control de Accesos.

En lo referente a controles de acceso, se contemplarán los siguientes aspectos:

- Identificar los requerimientos de seguridad de cada una de las aplicaciones.
- Identificar la información relacionada con las aplicaciones.
- Clasificación de Información de los diferentes sistemas y redes
- Definir los perfiles de acceso de usuarios docente, estudiante.
- Administrar los derechos de acceso en un ambiente distribuido y de red, que reconozcan todos los tipos de conexiones disponibles.

b. Administración de Accesos de Usuarios.

Con el objetivo de impedir el acceso no autorizado a la información el responsable de las aplicaciones web académicas implementará procedimientos formales para controlar la asignación de derechos de acceso a los sistemas, datos y servicios de información.

Registro de Usuarios.

El responsable de seguridad de las aplicaciones web académicas de la UTELVT definirá el procedimiento de registro de usuarios a fin de otorgar y revocar acceso a todos los servicios académicos de la universidad.

- Utilizar identificadores de usuario únicos, de manera que se pueda identificar a los usuarios por sus acciones.

- Verificar que el usuario tiene autorización del propietario de la información para el uso de las aplicaciones web académicas.
- Verificar que el nivel de acceso otorgado es adecuado para el propósito de la función del usuario.
- Entregar a los usuarios un detalle escrito de sus derechos de acceso.
- Requerir que los usuarios firmen declaraciones señalando que comprenden y aceptan las condiciones para el acceso.
- Mantener un registro formal de todas las personas registradas para utilizar el servicio.
- Cancelar inmediatamente los derechos de acceso de los usuarios que se desvincularon de la UTELVT.
- Realizar revisiones periódicas con el objeto de usuarios que se desvincularon de la UTELVT
- Garantizar que los identificadores de usuario redundantes no se asignen a otros usuarios.
- Incluir cláusulas en los contratos de personal y de servicios que especifiquen sanciones si el personal o los agentes que prestan un servicio intentan accesos no autorizados.

Administración de Privilegios.

El propietario de la información restringirá y vigilará la asignación y uso de privilegios, debido a que inadecuado el uso de los privilegios de la página web resulta a menudo en el factor preponderante que contribuye a la falla de los sistemas a los que se ha accedido ilegalmente.

Se debe tener en cuenta los siguientes aspectos:

- Identificar los privilegios asociados a cada proceso de la aplicación web,
- Establecer los privilegios a los usuarios sobre la base de la necesidad de uso y evento por evento.
- Mantener un proceso de autorización y un registro de todos los privilegios asignados. Una vez que se haya completado el proceso formal de autorización.
- Establecer un período de vigencia para el mantenimiento de los privilegios (basado al uso que se le dará a los mismos) luego del cual los mismos serán revocados.

Los propietarios de información serán los encargados de aprobar la asignación de privilegios a usuarios y solicitar su implementación, lo cual será supervisado por el responsable de seguridad informática.

Administración de contraseñas de usuario.

La asignación de contraseñas se controlará a través de un proceso de administración formal, a continuación, se lo describe:

- Requerir que los usuarios acepten la declaración por la cual se comprometen a mantener sus contraseñas personales en secreto.
- Certificar que los usuarios cambien las contraseñas provisionales que les han sido asignadas la primera vez que ingresan al sistema.
- Generar contraseñas provisionales seguras para otorgar a los usuarios.
- El responsable de seguridad informática establecerá los procedimientos de manejo de contraseñas apropiados para cada sistema.

c. Responsabilidades del Usuario.

Uso de Contraseñas.

Los usuarios deben seguir buenas prácticas de seguridad en la selección y uso de contraseñas. Las contraseñas constituyen un medio de validación y autenticación de la identidad de un usuario, y consecuentemente un medio para establecer derechos de acceso a los servicios de las aplicaciones web académicas de la UTELVT. Los usuarios deben cumplir las siguientes directivas:

- Mantener las en secreto contraseñas.
- Seleccionar contraseñas de calidad, de acuerdo a las prescripciones informadas por el responsable del Activo de Información de que se trate, que:
  - Sean fáciles de recordar.
  - No estén basadas en algún dato que otra persona pueda adivinar u obtener fácilmente mediante información relacionada con la persona, por ejemplo, nombres, números de teléfono, fecha de nacimiento, etc.
  - No consistan en caracteres idénticos consecutivos.
- Cambiar las contraseñas cada vez que el sistema se lo solicite.
- Cambiar las contraseñas provisionales en el primer inicio de sesión.

CONCLUSIONES

- La metodología OWASP presentó características tecnológicas más favorables y fiables, de acuerdo a la tabla comparativa 1-2, para el escaneo de vulnerabilidades de las aplicaciones web académicas de la UTELVT, gracias a que se enfoca en entornos de aplicaciones web de cualquier organización y permite la auditoría de aplicaciones web durante todo el ciclo de implementación.
- La herramienta de pruebas de seguridad OWASP ZAP, permitió determinar de manera efectiva la categoría, el tipo, el nivel de riesgo de las 342 vulnerabilidades encontradas en las aplicaciones web académicas de la UTELVT.
- Se implementaron dos escenarios de pruebas: el primero sin aplicar la metodología, ambiente en producción; y el segundo con su aplicación, ambiente de pruebas; sobre cada escenario se ejecutaron las pruebas de seguridad. Obteniendo como resultado lo siguiente: en el primer escenario se detectaron 342 vulnerabilidades de diferentes categorías, tipos, y niveles de riesgo; mientras que en el segundo escenario se detectaron 4 vulnerabilidades con nivel de riesgo informativo. De esta manera se contrastó que la metodología elaborado redujo las vulnerabilidades en un 98,88%, dando lugar al uso de aplicaciones académicas web más seguras.
- Luego del análisis de riesgo de la situación actual de las aplicaciones web académicas de la UTELVT, se diseñó un sistema de fortalecimiento de la seguridad e integridad, basado en salvaguardas diseñadas a partir de las soluciones ya establecidas en la metodología OWASP, además de las políticas de seguridad que las salvaguardas sugieren sean implementadas.
- Se verifico la mejora de la seguridad con las salvaguardas implementadas mediante la comparativa del análisis de riesgo en cada uno de los escenarios planteados en la investigación, es decir, antes y después del sistema de fortalecimiento de la seguridad e integridad de aplicaciones web académicas de la UTELVT.

RECOMENDACIONES

- Se debe ejecutar periódicamente el análisis de las vulnerabilidades de las aplicaciones web académicas de la UTELVT, o cuando se realicen actualizaciones, tratando en lo posible de utilizar las últimas versiones tanto de la metodología OWASP como del aplicativo OWASP ZAP.
- Mantener en lo posible registro de los fallos en la seguridad de las aplicaciones web académicas.
- El responsable de la seguridad de la información de la UTELVT debe en lo posible implementar el sistema de fortalecimiento propuesto, y mantenerlo actualizado.
- Mantener actualizados todos los sistemas operativos, servidores web y demás componentes que configuran el diseño y la funcionalidad de las aplicaciones web de la UTELVT.
- Como continuación de este trabajo de tesis, existen diversas líneas de investigación que quedan abiertas en las que es factible continuar investigando. Durante el proceso de esta tesis surgieron varias líneas que necesitan ser investigadas en el futuro, entre ellas se puede citar: el sistema de gestión de la seguridad informática de la UTELVT y diseño de software seguro.

BIBLIOGRAFÍA

- **Alvarez, V**. (2018). *Propuesta de una metodología de pruebas de penetración orientada a riesgos.* Guayaquil.
- **Arboleda, M**. (2021). *Estadísticas digitales 2020.* Obtenido de https://www.hablemosdemarcas.com/estadisticas-digital-2020/
- **Ascencio, M. y Moreno, P**. (2011). *Desarrollo de una Propuesta Metodológica para determinar la seguridad en una aplicación web.* Colombia: Universidad Tecnológica de Pereira.
- **Branch.** (2021). *Estadísticas de la situación digital de Ecuador en el 2020-2021.* Obtenido de https://branch.com.co/marketing-digital/estadisticas-de-la-situacion-digital-de-ecuador-en-el-2020-2021/
- **Castro, I.** (Julio de 2016). *¿Qué es un Análisis de Vulnerabilidades Informáticas?* Recuperado el Abril de 2018, de http://blog.cerounosoftware.com.mx/que-es-un-analisis-de-vulnerabilidades-inform%C3%A1ticas
- **Dirección General de Modernización Administrativa, P. e**. (2012). *MAGERIT – versión 3.0.. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Libro I - Método.* Madrid: Ministerio de Hacienda y Administraciones Públicas.
- **Dirección General de Modernización Administrativa, Procedimientos e Impulso de la Administración Electrónica.** (2012). *Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información - libro 1.* Madrid: Ministerio de Hacienda y Administraciones Públicas.
- D**ussan, C.** (2015). Políticas de seguridad informatica. *Universidad Automnma del estado de Mexico Redaly*.
- **García, G.** (2003). Seguridad Informática. *Servigraf.*
- **García, G., & Vidal, M.** (2016). La informática y la seguridad. Un tema de importancia para el directivo. *Infodir*(22), 47-58.
- **Hernández, R.** (2010). *Metodología de la inevstigación.* México D.F.: McGraw-Hill / Interamericana editores, S.A. De C.V.
- **Lopez, F.** (2015). *Metodologías para el Desarrollo de Software Seguro.* España: Universidad de Cataluña.

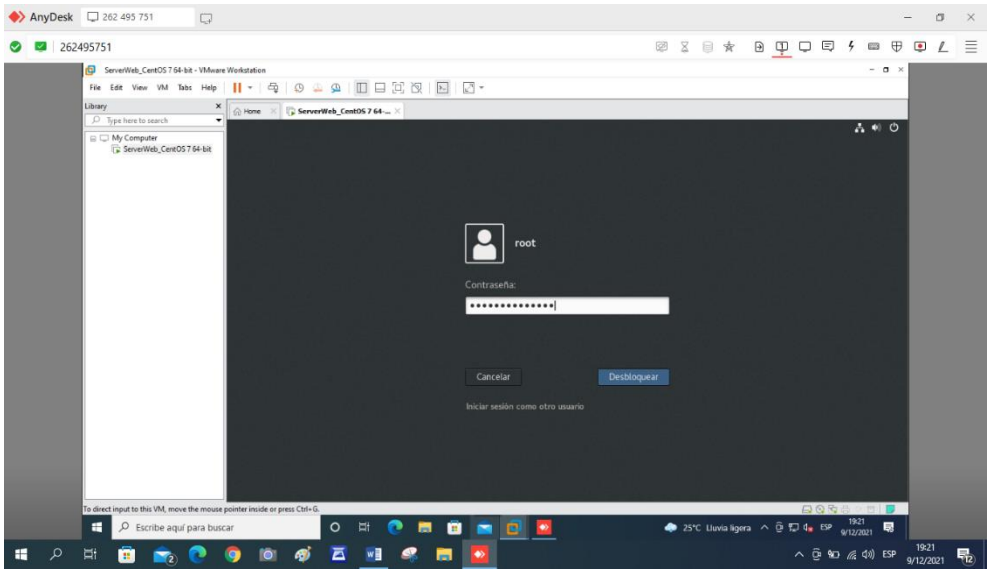
- **Luján, S.** (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web.* Alicante: Club universitario.
- **Marulanda, M**. (2018). *Aplicación de la metodología de pruebas OWASP para mejoramiento de la seguridad en el sistema e-commerce siembraviva.com.* Maizales.
- **Montalvo, R.** (2017). *Generación de políticas para la gestión de riesgos de seguridad en el desarrollo de software.* Riobamba.
- **OWASP.** (2008). *Guía de pruebas owasp versión 3*. Obtenido de https://www.owasp.org/images/8/80/Gu%C3%ADa_de_pruebas_de_OWASP_ver_3.0.pdf
- **OWASP.** (2021). *Introduction to OWASP Top 10 2021*. Obtenido de https://owasp.org/Top10/
- **Quiroz, S., & Macías, D.** (2017). Seguridad en informática: consideraciones. *Dominio de la ciencia*.
- **Ramos, J.** (2013). Pruebas de penetración o pent test. *Revista de Información, Tecnología y Sociedad*.
- **Rincón , M., & Albarracin, F.** (2018). *Análisis y evaluación de la seguridad informática para la página web publicada en hosting gratuito de la institución técnica de Firavitoba, para la detección y remediación de vulnerabilidades y riesgos en la información.* Sogamoso - Boyacá.
- **Rodríguez, D.** (2019). *Método deductivo: características y ejemplos.* Obtenido de https://www.lifeder.com/metodo-deductivo/
- Sánchez, J. (2017). *Análisis de vulnerabilidades y diseño de procesos correctivos de la página web de la Dirección de Educación a Distancia y Virtual de la Universidad Técnica de Ambato.* Ambato.
- **Voutssas, J.** (2010). Preservación documental digital y seguridad informática. *Investigación bibliotecológica, 24*(50).
- **Yánez, E.** (2014). *Guía de buenas prácticas de desarrollo de aplicaciones web seguras aplicado al sistema control de nuevos aspirantes Empresa Grupo LAAR.* Riobamba: ESPOCH.

ANEXOS

Anexo A. Implementación del Escenario 2 (Ambiente de pruebas)

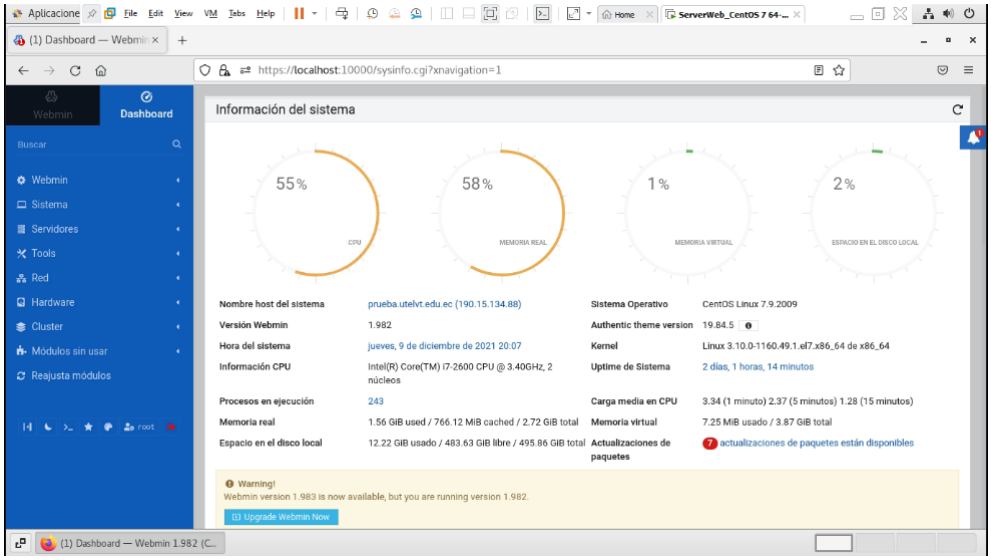
Servidor web virtual con CentOS 7, en el ambiente de pruebas

Se ha implementado el Escenario 2 - Ambiente de pruebas utilizando la plataforma de virtualización VMware Workstation Pro 16.2.1, en el cual se virtualizó un servidor Web con el Sistema Operativo Linux CentOS 7, el sistema se ejecutó con normalidad. En la figura 12 se muestra el proceso de logon al servidor de pruebas. Para proceder a copiar y configurar las aplicaciones web académicas de la UTELVT en el ambiente de pruebas, se ingresa al sistema con el usuario **root** para tener todos los privilegios de administración.



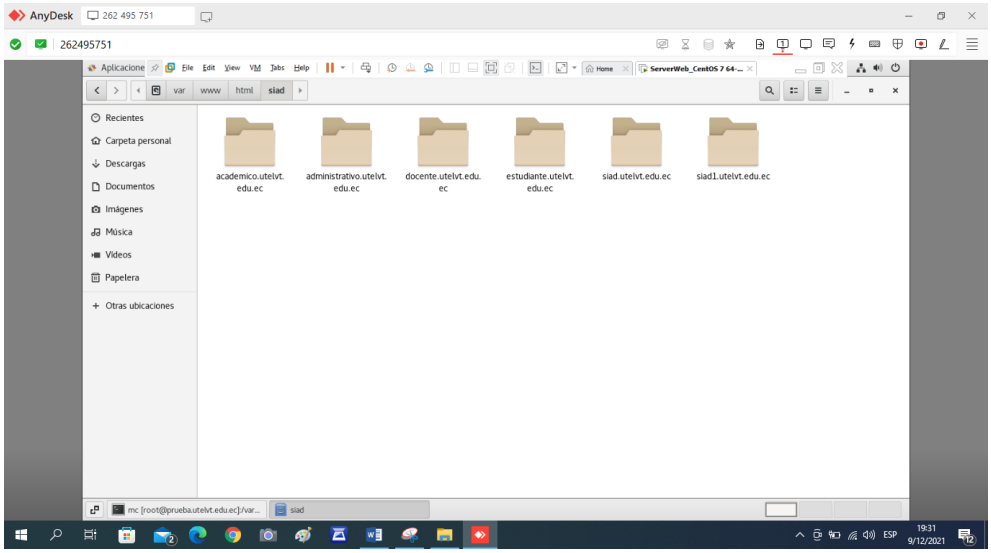
Logon al Servidor Web Virtual de Pruebas CentOS 7

A continuación, se presenta el Dashboard del Servidor a través del Servicio Webmin donde se observa de manera gráfica e intuitiva todos los parámetros de funcionamiento y características del equipo virtual.



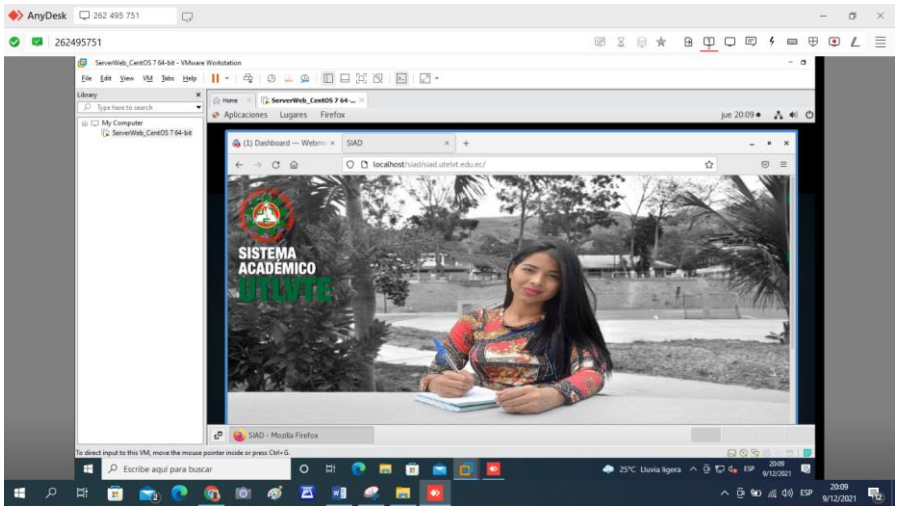
Información del Servidor Web Virtual de Pruebas

Se comprobó, como se observa en la siguiente figura, que todos los archivos correspondientes a las aplicaciones Web, estén copiadas de manera completa y correcta en el respectivo directorio del Servidor Web Virtual de Prueba, el cual es /var/www/html/siad.



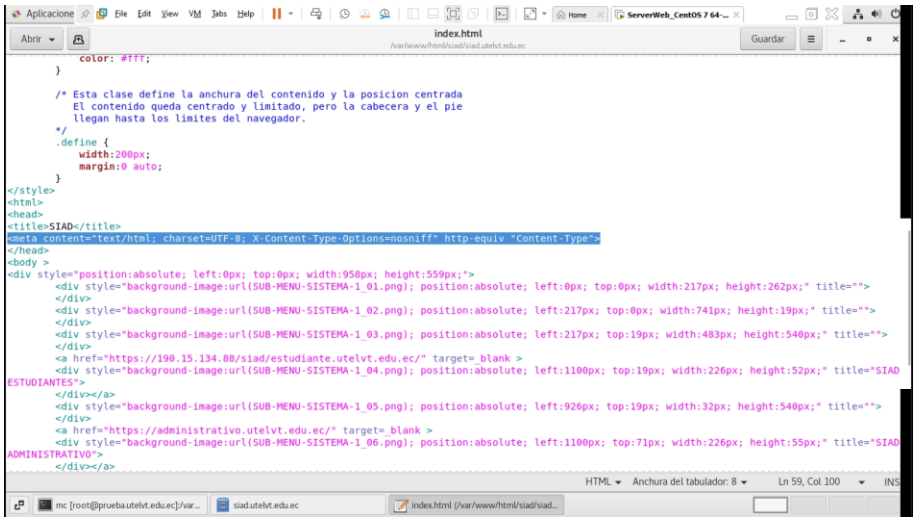
Directorios de Aplicaciones Web del Servidor Virtual de Pruebas

En la figura siguiente se verifica el correcto funcionamiento del servidor web de pruebas y las aplicaciones web académicas de la UTELVT cargadas en el servidor web virtual de pruebas.



Verificación del funcionamiento del Servidor Web Virtual de Pruebas

A continuación, se muestra el entorno de edición de códigos de las aplicaciones web académicas, es allí donde se hicieron las correcciones sugeridas por la metodología OWASP.



Edición del código de Aplicaciones Web

Anexo B. Autorización para realizar la investigación

UNIVERSIDAD TÉCNICA
"LUIS VARGAS TORRES"
DE ESMERALDAS

Dirección de Tecnologías de la Información y Comunicación

Oficio Nro. UTLVTE-TICS-2020-0079-O

Esmeraldas, 17 de marzo de 2020

Señor Ingeniero
Ramírez Márquez Jimmy Fernando
Docente FACI UTLVTE

Ciudad. -

ASUNTO: AUTORIZACIÓN DE REALIZACIÓN DE ESTUDIO.

En respuesta al oficio S/N, de fecha 11 de marzo de 2020, en el cual usted expresa "Yo, Jimmy Fernando Ramírez Márquez, con CI. 0801504705, docente de la carrera de Ingenierías de Tecnologías de la Información de la UTE-LVT, solicito muy respetuosamente, acepte y autorice la realización en el Departamento de TIC's el estudio ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD PARA EL FORTALECIMIENTO DE VULNERABILIDADES E INTEGRIDAD DE APLICACIONES WEB ACADEMICAS, proyecto de investigación (…)"; al respecto otorgo a usted, **VISTO BUENO** para la ejecución del estudio propuesto como parte de su proyecto de investigación, bajo el compromiso de entregar a ésta Dirección los resultados obtenidos a manera de informe, incluyendo sugerencias o recomendaciones de mejoras bajo un criterio de carácter técnico, de así ameritarse.

Las fechas de realización de las pruebas técnicas a efectuarse, serán acordadas con anticipación mediante comunicado por vía electrónica a la dirección leonardo.reyes@utelvt.edu.ec, o llamada telefónica al número de contacto +593 993160691.

Atentamente,

Ing. Leonardo Gabriel Reyes Vélez, Msc.
DIRECTOR DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN UTLVTE

Cc..

Señor Magister
Plata Cabrera Carlos Simón
Analista Programador de la UTLVTE

Cc. Archivo...

Anexo C. Tabla de distribución de CHI cuadrado utilizado para la demostración de la hipótesis

n	*0,995*	*0,99*	*0,975*	*0,95*	*0,9*	*0,75*	*0,5*	*0,25*	*0,05*	*0,025*	*0,01*	*0,005*
1	7,879	6,635	5,024	3,841	2,706	1,323	0,455	0,102	0,004	0,001	0,000	0,000
2	10,597	9,210	7,378	5,991	4,605	2,773	1,386	0,575	0,103	0,051	0,020	0,010
3	12,838	11,345	9,348	7,815	6,251	4,108	2,366	1,213	0,352	0,216	0,115	0,072
4	14,860	13,277	11,143	9,488	7,779	5,385	3,357	1,923	0,711	0,484	0,297	0,207
5	16,750	15,086	12,833	11,070	9,236	6,626	4,351	2,675	1,145	0,831	0,554	0,412
6	18,548	16,812	14,449	12,592	10,645	7,841	5,348	3,455	1,635	1,237	0,872	0,676
7	20,278	18,475	16,013	14,067	12,017	9,037	6,346	4,255	2,167	1,690	1,239	0,989
8	21,955	20,090	17,535	15,507	13,362	10,219	7,344	5,071	2,733	2,180	1,646	1,344
9	23,589	21,666	19,023	16,919	14,684	11,389	8,343	5,899	3,325	2,700	2,088	1,735
10	25,188	23,209	20,483	18,307	15,987	12,549	9,342	6,737	3,940	3,247	2,558	2,156
11	26,757	24,725	21,920	19,675	17,275	13,701	10,341	7,584	4,575	3,816	3,053	2,603
12	28,300	26,217	23,337	21,026	18,549	14,845	11,340	8,438	5,226	4,404	3,571	3,074

espoch | Dirección de Bibliotecas y Recursos del Aprendizaje

UNIDAD DE PROCESOS TÉCNICOS Y ANÁLISIS BIBLIOGRÁFICO Y DOCUMENTAL

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 14 / 03 / 2022

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: *Jimmy Fernando Ramírez Márquez*
INFORMACIÓN INSTITUCIONAL
Instituto de Posgrado y Educación Continua
Título a optar: Magíster en Seguridad Telemática
f. Analista de Biblioteca responsable: Lic. Luis Caminos Vargas Mgs.

LUIS ALBERTO CAMINOS VARGAS

Firmado digitalmente por LUIS ALBERTO CAMINOS VARGAS
Nombre de reconocimiento (DN): c=EC, l=RIOBAMBA, serialNumber=0602766974, cn=LUIS ALBERTO CAMINOS VARGAS
Fecha: 2022.03.14 12:26:17 -05'00'

UNIDAD DE PROCESOS TÉCNICOS DBRA DIRECCION DE BIBLIOTECAS Y RECURSOS DEL APRENDIZAJE ESPOCH