INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO

FACULTAD DE INGENIERÍA, DISEÑO E INNOVACIÓN

ESPECIALIZACIÓN EN SEGURIDAD DE LA INFORMACIÓN

EVALUACIÓN DEL ESTADO DE SEGURIDAD DE LA INFORMACIÓN DEL APLICATIVO WEB DE GESTIÓN Y SEGUIMIENTO DE NOVEDADES DE APRENDICES DEL PROGRAMA DE ARTICULACIÓN CON LA EDUCACIÓN MEDIA DEL SERVICIO NACIONAL DE APRENDIZAJE - SENA

PRESENTA:

ELIANA PATRICIA LOPEZ BERNAL
1712010481
CARMEN LILIANA HERRERA MARTÍNEZ
1712010016

ASESOR TEMÁTICO:

MAG. WILMAR JAIMES FERNANDEZ

Mayo de 2018

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

Hoy en día, el gran auge que se le ha dado al uso de las aplicaciones Web en el entorno laboral, las hace imprescindibles para el manejo de sus procesos de negocios, lo que obliga a la implementación de medidas de seguridad para controlar y proteger la información que se almacena dentro de ellas. Es así como atendiendo a la necesidad de agilizar los procesos administrativos propios del programa de Articulación de la Educación Media del SENA, se decidió implementar un prototipo para un sistema, que de acuerdo a los lineamientos institucionales, apoye las gestiones administrativas que exige éste programa.

Con miras a que el aplicativo Web pueda ser reconocido como parte de la plataforma tecnológica institucional, se requiere implementar controles técnicos de seguridad de la información para mitigar las amenazas a las que potencialmente se encuentra expuesta y que de esta manera estar enmarcada dentro de los lineamientos del Sistema de Gestión de Seguridad de la Información que actualmente maneja el SENA como estrategia institucional y competitiva.

Se utilizó el modelo de planeación estratégica situacional, para la identificación de la situación problema, se verificó la viabilidad del problema formulado y la posibilidad de ejecutarlo en una situación real de trabajo. Como estrategia metodológica, se realizaron pruebas de seguridad utilizando herramientas automatizadas para la detección de vulnerabilidades en el aplicativo Web y se hizo un análisis de resultados con el fin de medir el nivel de seguridad de la información gestionado en el aplicativo Web. Se pretende que a corto plazo se pueda mejorar la seguridad del sistema, implementando estrategias de remediación para las vulnerabilidades detectadas, como mecanismo de defensa para asegurar un software de calidad que garantice como principios fundamentales, la disponibilidad, la integridad y la confidencialidad de la información para todos sus usuarios.

PALABRAS CLAVES

Aplicaciones Web, Seguridad de la información, detección de vulnerabilidades, herramientas, pruebas de seguridad

ABSTRACT

Nowadays, the great boom that has been given to the use of Web applications in the work environment, makes them essential for the management of their business processes, which forces the implementation of security measures to control and protect the information that is stored inside them. Thus, in response to the need to streamline the administrative processes of the SENA Media Education articulation program, it was decided to develop a Web information system that, in accordance with institutional guidelines, supports the administrative procedures required by this program.

In order for the Web application to be recognized as part of the institutional technology platform, the implementation of technical information security controls is required to mitigate the threats to which it is potentially exposed and thus be framed within the guidelines of the Information Security Management System currently managed by SENA as an institutional guideline.

The strategic situational planning model was used to identify the problem situation, the viability of the problem formulated and the possibility of executing it in a real work situation were verified. As a methodological strategy, security tests were carried out using automated tools for the detection of vulnerabilities in the Web application and an analysis of the results was made in order to measure the current state of information security in the Web application. It is intended that in the short term the security of the system can be improved, implementing remediation strategies for the detected vulnerabilities, as a defense mechanism to ensure quality software that guarantees as fundamental principles, integrity, confidentiality and availability of information for all its users.

KEY WORDS

Web applications, Information security, vulnerability detection, tools, security tests.

INTRODUCCIÓN

Como parte del recurso tecnológico que proveen las TICs[1], las aplicaciones Web son actualmente el medio principal para la gestión de información utilizada por la mayoría de las empresas en el manejo de sus procesos administrativos y por las múltiples ventajas que ofrecen, entre ellas la facilidad de uso e instalación multiplataforma y la comunicación interactiva entre el usuario y la información.

Una organización como el SENA[2], que utiliza Internet como medio para acceder a sus aplicaciones y que gestiona la información sensible de sus procesos internos, está expuesta a peligros de acceso indebido a la información, por lo que sus aplicativos como los sistemas de información y las bases de datos, constituyen para esta entidad, uno de los activos de información más importantes que debe proteger, como factor clave en la administración de la seguridad de su información.

Planteamiento del Problema

Descripción de la Situación de Interés. El SENA, al ser una entidad pública colombiana, tiene como finalidad la capacitación técnica del recurso humano, formando personas para su vinculación al mundo laboral y realizando actividades para el desarrollo empresarial, tecnológico y comunitario. Dentro de sus diferentes programas de formación, se encuentra el programa de Articulación con la Media que busca que los jóvenes de grados décimo y once de bachillerato accedan al desarrollo de competencias técnicas que les permitan tener un desempeño laboral al terminar la educación media. Para la gestión académica y administrativa de los procesos de formación que gestiona el SENA en sus diferentes programas a nivel general, cuenta como plataforma tecnológica con el aplicativo SOFIA PLUS[3], la cual se ha constituido como una nueva infraestructura tecnológica y digital, que genera una forma diferente de relacionarse con los procesos de aprendizaje.

Actualmente el programa de Articulación con la Media que ofrece el SENA, no cuenta con un software propio que apoye las gestiones administrativas acorde con los lineamientos que exige el programa y que conlleve a la gestión de una información más completa, exacta, e inmediata y que garantice la disponibilidad, la integridad y la confidencialidad de la información para todos sus usuarios, quienes como integrantes de la comunidad educativa SENA, se convierten en actores claves de apoyo para llevar a cabo con éxito los objetivos del programa de articulación. Es por esta razón que con miras a

[1] Tecnologías de la Información y la Comunicación.
[2] Servicio Nacional de Aprendizaje. Página Web oficial: www.sena.edu.co.
[3] Sistema Optimizado para la Formación Integral y el Aprendizaje Activo

agilizar los procesos administrativos propios del programa, se decidió desarrollar el Sistema de Información Web para el Programa de Articulación con la Educación Media del SENA en su versión 1. El primer prototipo de desarrollo del aplicativo, se enfoca en la gestión y seguimiento de novedades de aprendices del programa de articulación, en el cual se han detectado fallos por la falta de implementación de controles técnicos de seguridad y por lo tanto su utilización pone en riesgo la información que los usuarios manejan.

Red de Actores Relevantes. Los actores relevantes que intervienen directamente con el Sistema de información Web para el Programa de Articulación con la Educación Media SENA como usuarios, y que se ven afectados por los inconvenientes de seguridad que ofrece el aplicativo Web son los siguientes:

- ***Instructores:*** Personas que asumen el rol de facilitador del aprendizaje y quienes orientan, apoyan, retroalimentan y evalúa al aprendiz durante su proceso formativo.
- ***Instituciones de Educación Media:*** Conjunto de personas y bienes establecidos por las autoridades públicas o privadas que ofertan educación media. Son entidades que cuentan con planta física, licencia de funcionamiento, infraestructura administrativa y medios educativos adecuados para impartir educación.
- C***oordinador Académico y Líder del Programa de Articulación con la Media:*** Personal administrativo del SENA encargado de dirigir, controlar y evaluar las acciones de formación profesional integral conforme a las políticas institucionales, la normatividad vigente en la entidad y la programación de la oferta educativa requerida para el programa de Articulación con la Educación Media en los centros de formación.
- ***Administrador Web:*** Es el responsable del contenido, publicación y mantenimiento del aplicativo Web. Se asegura de que la información esté correcta y actualizada. Se encarga de las cuestiones administrativas y técnicas del hosting del aplicativo.

Figura 1. Actores Relevantes Identificados en la Situación Problema.

Fuente. Elaboración propia.

Causas. Los diferentes actores relevantes, detectaron fallos que se traducen en vulnerabilidades[4] que comprometen la seguridad de la información gestionada por el aplicativo Web y que al ser analizadas, se convierten en cadenas causales que conllevan a la situación problémica planteada, identificándose como causas críticas las siguientes:

- Falta de definición de una política de tratamiento y protección de datos personales: En la actualidad, el aplicativo Web aún no integra el marco normativo sobre la política de tratamiento y protección de datos personales conforme a lo señalado en la Ley 1266 de 2008, la Ley 1581 de 2012 y del Decreto1377 de 2013, lo que no garantiza que la información suministrada cuente con los con los principios de seguridad, integridad y confidencialidad.
- Detección de errores en la validación de entradas de datos a través de formularios: Actualmente el aplicativo no realiza un filtro bien restrictivo de la información procedente de los formularios HTML, lo que conlleva a duplicidad en el registro de información y fallas en cuanto a la completitud y exactitud de los datos gestionados por el aplicativo.
- Falta de implementación de un mecanismo robusto que permita la autenticación del usuario: Esta situación conlleva a que la gestión de sesiones de usuarios sea criptográficamente insegura, por lo cual es conveniente la implementación de cifrado de en las variables de sesión mediante técnicas criptográficas, además se pueden producir fallas en la gestión del control de acceso de usuarios de acuerdo al perfil asignado y esto ocasiona que se visualice información no asociada con su perfil.
- Falta de implementación de logs para registro de errores y actividades del sistema: Lo cual dificulta la trazabilidad de las acciones realizadas por los usuarios y el registro de actividades realizadas para procesos de auditorías de seguridad.
- Falta de implementación de políticas para validar el uso de contraseñas seguras: La aplicación no valida la creación de contraseñas robustas, exponiéndose a que un atacante pueda deducirla fácilmente afectando la integridad y confidencialidad de la información.
- Falta de definición de una política de respaldo de la información crítica mediante archivos de copia de seguridad: Actualmente se sigue realizando el respaldo de la información con copias de seguridad generadas de forma manual y sin periodicidad alguna, lo que ha ocasionado retrasos en los procesos gestionados por el aplicativo en casos de pérdida de información por no contar con copias de seguridad recientes.
- La aplicación no posee un protocolo de seguridad para evitar ataques internos o externos: En el momento no se cuenta con un sistema de prevención de intrusos mediante la utilización de canales de transacción segura de datos para los servicios ofrecidos entre el usuario y el servidor, además de la aplicación de estrategias para el acceso y aseguramiento de la base de datos del aplicativo.

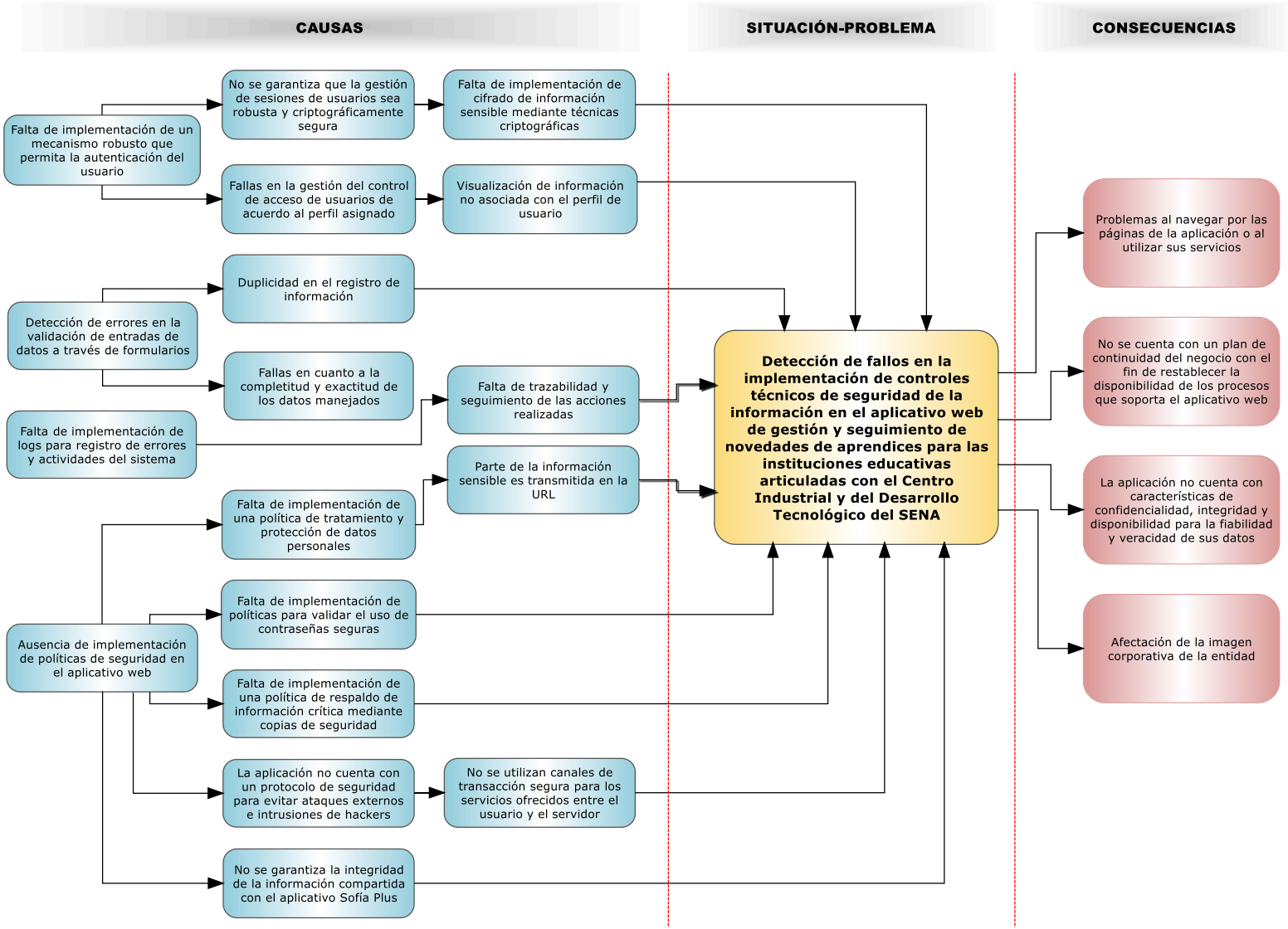
[4] Las vulnerabilidades en un software son los fallos de seguridad a través de las cuales, un atacante puede llegar a comprometer seguridad de todo un sistema sobre el que se ejecuta una aplicación.

- No se garantiza que la información compartida con el aplicativo Sofía Plus sea íntegra: No existe una política idónea que garantice condiciones de seguridad adecuadas para evitar la adulteración, pérdida o consulta en la importación de datos compartidos con el aplicativo central.

Consecuencias. La detección de fallos técnicas de seguridad de la información en el aplicativo Web de gestión y seguimiento de novedades de aprendices para las instituciones educativas articuladas con el SENA, ocasiona las siguientes inconvenientes en cuanto a su uso:

- Problemas al navegar por las páginas de la aplicación o al utilizar sus servicios: El grado de satisfacción de uso del aplicativo por parte de los usuarios no es deseado, por lo que la aplicación Web no ofrece el nivel de confianza requerido para su uso.
- No se cuenta con un plan de contingencia del negocio con el fin de restablecer la disponibilidad de los procesos que soporta el aplicativo Web: El SENA se encuentra vulnerable ante incidentes o eventos inesperados e indeseados que afecten la seguridad de la información, presentados por el uso del aplicativo en estas condiciones.
- La aplicación no cuenta con características de confidencialidad, integridad y disponibilidad para la fiabilidad y veracidad de sus datos: Aún no se tienen en cuenta la utilización de técnicas de desarrollo seguro del software, como tampoco los planes de concientización y sensibilización del personal para el manejo de la seguridad de la información que gestiona, en la implantación de buenas prácticas de seguridad en cuanto al uso de la infraestructura tecnológica de servidores y aplicativos Web.
- Afectación de la imagen corporativa de la entidad: El SENA, es reconocido por ser la entidad más querida por todos los colombianos, posee una sistema de gestión de calidad que garantiza que los servicios que ofrece den cumplimiento a los requisitos de la comunidad y en el logro de la satisfacción del mismo, por lo tanto los sistemas de información y las aplicaciones Web que conforman las plataformas tecnológicas institucionales, deben cumplir con los mismos estándares de calidad.

Figura 2. Flujograma Explicativo de la Situación Problema.



Fuente. Elaboración propia.

Justificación

Actualmente el SENA posee una política de Gestión de Seguridad de la Información[5] como parte del Sistema Integrado de Gestión y Autocontrol, que contempla como marco normativo entre otros, la privacidad y seguridad de la información y de los sistemas de información que se manejan como plataformas tecnológicas institucionales, pero el aplicativo Web de gestión y seguimiento de novedades de aprendices para las instituciones educativas articuladas con el SENA que se está implementando en el programa de Articulación con la Media, es una propuesta para el apoyo administrativo de los actores que intervienen de manera directa en este programa, pero aún no está reconocido como parte de la plataforma tecnológica institucional, por lo tanto se desconoce lo planteado en el dominio de seguridad que tiene que ver con la adquisición, desarrollo y mantenimiento de sistemas de información que plantea el nuevo enfoque de la norma ISO/IEC 27001:2013[6], como modelo para estructurar todos los procesos del SGSI del SENA. Cuando se especificaron los requerimientos para el desarrollo de éste aplicativo Web, no se tuvieron en cuenta los requerimientos de seguridad alineados con el Sistema de Gestión de la Seguridad de la Información del SENA.

Teniendo en cuenta algunas estadísticas en cuanto a la gestión de la inseguridad de las aplicaciones Web, se toma como referencia lo planteado en el documento de los 10 riesgos de seguridad más relevantes en aplicaciones Web presentados por la organización OWASP[7] [1]. Como se observa en la figura 4, El OWASP Top 10 muestra una comparativa de los riesgos de seguridad de aplicaciones Web más comunes y más importantes presentados desde el año 2013 hasta el año 2017 y que de acuerdo al nivel de riesgo crítico se presenta con mayor incidencia en las organizaciones.

---

[5] Política del Subsistema Gestión de Seguridad de la Información SGSI. Fue aprobada por el Consejo Directivo Nacional mediante Acuerdo 0007 de 2016, en donde la entidad asumen el compromiso de implementar el Subsistema de Gestión de Seguridad de la información para proteger los activos de información de los procesos misionales de la entidad.

[6] ISO 27001 es una norma internacional emitida por la ISO (Organización Internacional de Normalización) que describe cómo se debe gestionar la seguridad de la información en una empresa. En el año 2013 se publicó la revisión más reciente de esta norma y desde entonces su nombre completo es ISO/IEC 27001:2013.

[7] OWASP cuyas siglas en inglés son Open Web Application Security Project y en español Proyecto Abierto de Seguridad de Aplicaciones Web. Es un proyecto creado para generar conciencia acerca de la seguridad en aplicaciones mediante la identificación de los riesgos más críticos que se presentan en las organizaciones. Esta lista es publicada y actualizada cada tres años por esta organización. Su última publicación oficial fue en el año 2017.

Figura 3. OWASP Top 10 Comparativa 2013 – 2017.

OWASP Top 10 2013	±	OWASP Top 10 2017
A1 – Inyección	→	A1:2017 – Inyección
A2 – Pérdida de Autenticación y Gestión de Sesiones	→	A2:2017 – Pérdida de Autenticación y Gestión de Sesiones
A3 – Secuencia de Comandos en Sitios Cruzados (XSS)	↘	A3:2017 – Exposición de Datos Sensibles
A4 – Referencia Directa Insegura a Objetos [Unido+A7]	∪	A4:2017 – Entidad Externa de XML (XXE) [NUEVO]
A5 – Configuración de Seguridad Incorrecta	↘	A5:2017 – Pérdida de Control de Acceso [Unido]
A6 – Exposición de Datos Sensibles	↗	A6:2017 – Configuración de Seguridad Incorrecta
A7 – Ausencia de Control de Acceso a las Funciones [Unido+A4]	∪	A7:2017 – Secuencia de Comandos en Sitios Cruzados (XSS)
A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)	☒	A8:2017 – Deserialización Insegura [NUEVO, Comunidad]
A9 – Uso de Componentes con Vulnerabilidades Conocidas	→	A9:2017 – Uso de Componentes con Vulnerabilidades Conocidas
A10 – Redirecciones y reenvíos no validados	☒	A10:2017 – Registro y Monitoreo Insuficientes [NUEVO, Comunidad]

Fuente. Recuperado de: https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf.

Esta información recopila las vulnerabilidades más importantes que se presentan en la industria del uso de aplicaciones Web de cientos de organizaciones y más de cien mil aplicaciones y APIs alrededor del mundo. Las 10 principales categorías son seleccionadas y categorizadas de acuerdo con la estimación de datos de prevalencia, explotabilidad, detectabilidad e impacto.

En la última publicación se observa que nuevamente los ataques de inyección de código (A1) vuelven a ser el mayor riesgo de seguridad, al igual en las publicaciones anteriores. En orden descendente le siguen la pérdida de autenticación y gestión de sesiones (A2) y la exposición de datos sensibles (A3), como las vulnerabilidades más representativas en la actualidad, entre otras.

La empresa de seguridad Veracode[8], afirma que casi el 80% de las aplicaciones desarrolladas para la Web contienen al menos una vulnerabilidad en su evaluación inicial, cerca del 70% de las aplicaciones analizadas no pasa una auditoría de seguridad respecto de las 10 medidas del OWASP Top 10. En conclusión esta empresa determinó que en el 2017 un 27,6 % de las aplicaciones analizadas son fácilmente vulnerables mediante inyecciones de código SQL.

[8] Veracode. Empresa que ofrece soluciones de aplicaciones y servicios de seguridad en software. Participa en la elaboración del OWASP Top 10. Página Oficial: https://www.veracode.com.

En éste caso en particular, la mayoría de los fallos de seguridad detectados por los usuarios en el uso del aplicativo Web de gestión y seguimiento de novedades de aprendices implementado en el programa de Articulación con la Educación Media del SENA, se encuentran contenidos en las estadísticas de riesgos de seguridad presentados en la última publicación OWASP Top 10.

La necesidad de tener un sistema de información autónomo que facilite la gestión de información dentro del programa de Articulación con la Educación Media y la falta de implementación de buenas prácticas de desarrollo seguro de software, incrementaron el riesgo de no descubrir vulnerabilidades de forma rápida y precisa, lo que hace de éste sistema de información un software inseguro. Además se hace necesario incluir la seguridad en las etapas finales de desarrollo del software, lo cual no resulta nada rentable, pues se requiere inversión en cuanto a tiempo y recursos, en la implementación de cambios y corrección de errores detectados en la fase de mantenimiento para incluir el tema de la seguridad en la prevención de vulnerabilidades potenciales y que según las estadísticas publicadas por especialistas y expertos en el tema, ascienden más o menos a un 40% del costo total del desarrollo del software.

Alcance

Ser conscientes de la necesidad de integrar metodologías para el desarrollo seguro del software, así como el aseguramiento de aplicaciones y bases de datos, permite la profundización de conocimientos específicos en el área de desarrollo de software seguro, con el fin de proporcionar técnicas básicas para protegerse contra las vulnerabilidades detectados en la implementación de controles técnicos de seguridad informática en el aplicativo Web de gestión y seguimiento de novedades de aprendices para el programa de Articulación con la Educación Media del SENA, en una propuesta que defina las políticas o lineamientos que incorporen la seguridad en las aplicaciones.

El alcance de esta propuesta debe tener en cuenta los lineamientos planteados en el Sistema de Gestión de Seguridad de la Información del SENA, en lo que tiene que ver con la adquisición, desarrollo y mantenimiento de software, ya que el aplicativo Web se pretende utilizar como plataforma tecnológica institucional para el programa de articulación con el fin de que se encuentre diseñada y alineada con el contexto de la entidad, pues se requiere presentar una propuesta de un desarrollo de software seguro alineado con las necesidades, los procesos, los objetivos y la operación del SENA, dentro del marco normativo y regulatorio aplicable. Para ello se requiere de contar con un informe de medición del estado de seguridad del aplicativo Web para adoptar medidas estratégicas que puedan mitigar las vulnerabilidades detectadas.

De igual manera cabe especificar que las acciones que serán ejecutas en este proyecto no contempla la implementación de los controles técnicos de seguridad en el prototipo actual de desarrollo del aplicativo Web, pues se requiere tomar como referencia el informe de seguridad obtenido como resultado de la detección de vulnerabilidades, para implementar en su desarrollo final, las soluciones propuestas que permitan mitigar los fallos de seguridad detectados y así evidenciar la utilización de buenas prácticas de desarrollo seguro de software.

Objetivos del Proyecto de Grado

Evaluar el estado de la seguridad de la información de una solución tecnológica que apoye la gestión y seguimiento de novedades de aprendices del programa de Articulación con la Educación Media del SENA, mediante la utilización de herramientas automáticas para el escaneo de vulnerabilidades Web que permita:

- Desarrollar destrezas específicas en la ejecución de proyectos en el área de desarrollo de software, aplicando la metodología de planeación estratégica situacional (PES), orientada a la identificación y resolución de problemáticas aplicado en el entorno laboral.
- Identificar los principios fundamentales de la seguridad de la información como: Confidencialidad, integridad, autenticidad, disponibilidad, como falencias detectadas en una aplicación Web.
- Implementar pruebas de seguridad, como estrategia para la detección de fallos y validación de requerimientos de seguridad que garanticen el desarrollo de una aplicación Web segura y confiable.
- Interpretar resultados de vulnerabilidades detectadas para proponer mecanismos de solución que ayuden a prevenir ataques y mitigar el problema de un software no seguro.

MARCO TEÓRICO

A continuación se hace una revisión de investigaciones realizadas acerca de las posturas teóricas planteadas en la situación problema. Este sustento teórico se enmarca principalmente dentro de los conceptos de desarrollo seguro de software, las amenazas y vulnerabilidades a las que son expuestas las aplicaciones Web y las distintas técnicas utilizadas para su detección.

Aplicaciones Web

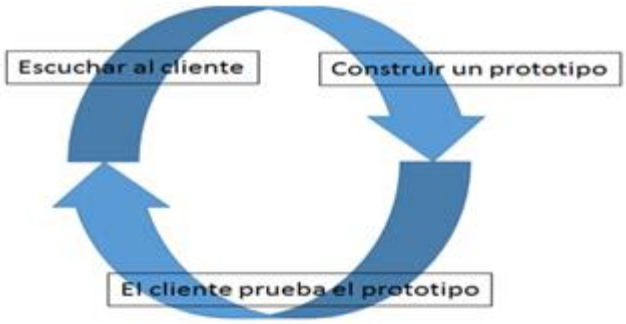
Descripción General. Según [2], las aplicaciones Web han surgido como una tendencia mundial, debido al auge económico que este sistema de comunicación ha representado para el mundo, avanzando en las tecnologías empleadas para este tipo de plataformas, en la media en que las grandes y pequeñas empresas utilizan más este medio de comunicación para realizar compras, transacciones en línea y manejar la información sensible de la empresa, esta tecnología de las aplicaciones Web, resulta muy atractiva para que muchas personas quieran sacar un provecho malicioso de estas.

Por consiguiente se debe saber implementar y no escatimar en los controles de seguridad en las aplicaciones Web que un especialista en seguridad de la información debe aplicar. Es un tema que está en constante evolución y a medida que surge un nuevo avance tecnológico, paralelo a ello surge el riesgo o vulnerabilidad para esta nueva alternativa informática.

[3] plantea en su artículo que las aplicaciones Web de gestión académica y administrativa son sistemas que manejan información sensible por lo que requieren de mecanismos de protección, con el fin de proteger los datos de cada individuo y mantenerlos íntegros así como disponibles y con los niveles de confidencialidad adecuada.

En la actualidad tomando como referencia las revisiones realizadas por [4], se debe incorporar en el desarrollo de un producto software, un plan para el diseño y modelado que en ingeniería del software se conoce como modelo de procesos. Entre los modelos de procesos que más utilizados se encuentran el modelo de construcción de prototipos, el modelo lineal secuencial, el modelo para el Desarrollo Rápido de Aplicaciones, el modelo incremental, el modelo de desarrollo basado en componentes y el modelo en espiral. De acuerdo al tipo de aplicación a desarrollar se recomienda escoger un modelo en particular. En la siguiente tabla se muestra enfoque de los modelos de procesos más representativos y una descripción de los diferentes roles del proceso.

Tabla 1. Modelos para el Desarrollo del Software.

Modelos de Procesos para Desarrollo de del Software	Descripción
Modelo Lineal Secuencial 	Este modelo sugiere un enfoque incremental paso a paso del desarrollo de software. Comienza con las actividades de análisis de requisitos y progresa con el diseño, codificación, pruebas y mantenimiento.
Modelo de Construcción de Prototipos 	Es un modelo no secuencial, basado en la construcción de simulaciones o modelos ejecutables de aplicaciones (prototipos). Su objetivo principal es la participación directa del cliente en la construcción del software requerido.
Modelo para el Desarrollo Rápido de Aplicaciones 	Este modelo de desarrollo de software conocido como RAD por sus siglas en ingles Rapid Application Development, es un modelo cuyo ciclo de desarrollo permite la creación de sistemas funcionales en un periodo de tiempo corto, comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE.

Fuente. Elaboración propia a partir de: http://www.redalyc.org/pdf/849/84921327034.pdf

Tecnologías de Desarrollo. La Web funciona bajo este principio: Los servidores Web alojan el contenido y los navegadores que deberán estar instalados en el equipo del usuario (Mozilla Firefox, Google Chrome), utilizan los protocolos para realizar el puente entre el servidor Web y contenido que requiere utilizar el usuario. Las tecnologías de desarrollo del lado del cliente son utilizadas para la integración en las páginas Web.

A continuación se presenta un paralelo entre las tecnologías de programación del lado del cliente y del lado del servidor:

Tabla 2. Programación del Lado del Cliente y del Lado del Servidor.

Programación del lado del cliente	Programación del lado del servidor
Los programas se instalan en el servidor pero se ejecutan en el cliente	Los programas se instalan y son ejecutados por el servidor
Se descarga de trabajo a los servidores	El trabajo recae sobre los servidores pudiéndose presentar sobrecarga
Para la ejecución de los scripts es necesario una transmisión de código por la red	Al cliente solo se les transfiere el resultado de la ejecución de un script
No es necesario realizar transmisiones en la red para invocar las respuestas a las acciones de los usuarios sobre el script	Una vez enviada al usuario la respuesta del programa, cualquier petición adicional del cliente requiere una nueva conexión con el servidor
Se requiere que el cliente tenga instalados programas o plugins adecuados para la correcta ejecución del script	Todo el software necesario debe estar instalado en el servidor
La página no podrá ser ejecutada correctamente si en el cliente no se encuentra instalado alguno de los programas intérpretes o plugins	Todos los clientes podrán visualizar correctamente la página
El cliente tiene acceso al código que puede transferirse, pudiendo obtener a partir de él información que pueda resultar comprometida	Los clientes no tienen acceso al código fuente, ya que permanece en el servidor conservando su privacidad

Fuente. Recuperado de: http://www.editdiazdesantos.com/wwwdat/pdf/9788479787066.pdf.

- **Tecnologías de programación del lado del cliente**

Los lenguajes de programación del lado del cliente se usan para ser integrados en las páginas Web, es decir que, un código escrito en un lenguaje de script es embebido directamente dentro de un código HTML y se ejecuta interpretado y no compilada como otros lenguajes de programación. Esta tecnología de programación es comúnmente utilizada para la validación de datos en el equipo cliente antes de enviarlos al servidor.

A continuación se presentan las principales tecnologías de desarrollo utilizadas del lado del cliente.

Figura 4. Descripción de las Tecnologías de Programación del Lado del Cliente.

Tecnología de Programación de Lado del Cliente	Descripción
HTML 	Es un lenguaje de hipertexto que se basa en etiquetas que le indican al navegador dónde colocar texto, imagen, sonido, video, formularios, tablas, etc., y la forma que tendrán estos al ser colocados en la página. Estas etiquetas se pueden utilizar para definir la forma o estilo que se quiere aplicar al documento.
Applets de Java 	Son otra manera de incluir código a ejecutar en los clientes que visualizan una Web. Son pequeñas aplicaciones que se transfieren con las webs y que el navegador ejecuta en la página. Están precompilados, por lo que su forma de responder varía de los de JavaScript, y son más difíciles de programar, pero también son más potentes e independientes del navegador o plataforma que se utilice.
VBScript 	Es un lenguaje que también programa scripts, se basa en Visual Basic pero sólo es compatible con Internet Explorer, lo cual lo limita ampliamente. Su modo de funcionamiento es muy similar al utilizado en JavaScript y los recursos a los que se puede acceder también son los mismos: el navegador.
CSS 	Hojas de estilo en cascada (o CSS, siglas en inglés de Cascading Stylesheets) es un lenguaje de diseño gráfico que permite crear estilos que generalicen el comportamiento de la página Web en general. Así, si en algún momento se quiere cambiar algún estilo, automáticamente se actualizaría en todo el sitio.
JavaScript 	Es un robusto lenguaje de programación que puede ser aplicado a un documento HTML y usado para crear interactividad dinámica en los sitios Web. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Fuente. Elaboración propia a partir de:
http://adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html

- **Tecnologías de programación del lado del servidor**

Un lenguaje de programación del lado del servidor es aquel que es ejecutado en el servidor Web antes del envío de la página al cliente a través de Internet. Las páginas que se ejecutan en el lado del servidor tienen acceso principalmente a las bases de datos instaladas en el servidor, además de otras tareas que se envían al cliente para que puedan ser interpretados directamente por el navegador.

A continuación se presentan las principales tecnologías de desarrollo utilizadas del lado del servidor.

Figura 5. Descripción de las Tecnologías de Programación del Lado del Servidor

Tecnología de Programación de Lado del Servidor	Descripción
PERL 	Perl es un lenguaje de programación interpretado, lo que viene a ser que el código de los scripts en Perl no se compila, sino que cada vez que se va a ejecutar se lee el código y arranca interpretando lo que hay escrito. Es muy dinámico, ya que desde Perl se llama a otros subprogramas escritos en otros lenguajes.
ASP.NET 	ASP es un lenguaje desarrollado por Microsoft para la creación de páginas dinámicas del servidor. Escribe en la propia Web utilizando el lenguaje Visual Basic Script o Jscript. Para el funcionamiento de las páginas se necesita tener instalado IIS (Internet Information Services) con el Framework .Net.
PHP 	PHP es el acrónimo de Hipertext Pre-Pprocesor. Es un lenguaje gratuito e independiente de plataforma, rápido, con una librería de funciones enorme y con mucha documentación. Se usa para la generación de páginas Web dinámicas, embebidas en páginas HTML. Para su funcionamiento necesita tener instalado un servidor de Apache.
JSP 	Es un lenguaje desarrollado por Sun Microsystems para la creación de sitios Web dinámicos y potentes. JSP es el acrónimo de Java Server Pages, comparte características similares a las de ASP.NET. JSP tiene un motor de páginas basado en los servlets2 de Java. Para su funcionamiento necesita tener instalado un servidor Tomcat.

Fuente. Elaboración propia a partir de:
http://adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html

- **Sistema de gestión de bases de datos**

Un sistema de gestor de bases de datos (SGBD) es un programa que permite crear, gestionar y administrar una base de datos. Trabaja con tablas que permiten realizar vinculaciones o relaciones entre sí para acceder a la información ejecutando consultas en lenguaje SQL. A continuación se presenta una descripción de los principales sistemas gestores de bases de datos utilizados actualmente.

Figura 6. Descripción de los Sistemas Gestores de Bases de Datos

Sistema Gestor de Bases de Datos	Descripción
MySQL 	MySQL es un software de sistema gestión de base de datos relacional que se puede ejecutar en los sistemas operativos GNU/Linux, Windows y Mac, se ejecuta de forma multi-thread (realiza varias tareas en paralelo de forma concurrente) y multiusuario y es distribuido por Oracle bajo la licencia GPL y comercial. Es el sistema gestor de base de datos más popular en el mundo.
Oracle 	Oracle es un sistema de gestión de base de datos desarrollado por la compañía Oracle, este sistema es de tipo modelo objeto relacional, por el cual es uno de los gestores de bases de datos más completo como: soporte de transacciones, estabilidad, escalabilidad y puede correr en los sistemas operativos GNU/LINUX, Windows, Mac y entre otros.
PostGreSQL 	PostgreSQL es un sistema gestor de base de datos relacional de código abierto de muchos otros proyectos, multiplataforma, orientado a objetos bajo la licencia PostgreSQL que es similar a la BSD de la MIT.
Microsoft SQL Server 	Microsoft SQL Server un software propietario de gestión de base de datos creado por la compañía Microsoft disponible, lamentablemente solo se puede usar en el sistema operativo Windows, aunque recientemente anunciaron que SQL Server 2016 estaría disponible para GNU/Linux para este fin de año.
MariaDB 	MariaDB es un programa sistema de manejo de bases de datos multiplataforma descendiente de MySQL creado en el año 2009 por el descontento de modelo de desarrollo. Se distribuye bajo la licencia GPL que es una software completamente libre y además se ha introducido dos nuevos motores de almacenamiento Aria y XtraDB en sustitución de MyISAM y InnoDB.
SQLite 	SQLite es un sistema de manejo de bases de datos de tipo modelo relacional multiplataforma, este gestor de base datos se diferencia entre los demás que son cliente-servidor, si no es una biblioteca en proceso que implementa un sistema autónomo, sin necesidad de hacer configuración.
MogoDB 	MongoDB es un sistema de base de datos NoSQL multiplataforma, orientado a documentos desarrollado bajo la filosofía de software libre, los datos son guardados en la base datos en estructuras de datos similar a JSON de JavaScript e incluso tiene la capacidad de realizar consultas utilizando JavaScript por el cual también existen APIS para distintos lenguajes de programación para realizar consultas e informes.

Tabla 5. Elaboración propia a partir de: https://www.mindmeister.com/es/687810410/clasificaci-n-de-los-sistemas-de-gestion-bases-de-datos.

- **Servidor Web Apache**

Apache es el Servidor Web más utilizado a nivel mundial, líder con el mayor número de instalaciones por encima de IIS (Internet Information Server) de Microsoft. [5] en su investigación sobre el aseguramiento en la configuración del Servidor Web Apache, lo define de la siguiente manera: "Apache es un extraordinario servidor Web (servidor para el protocolo HTTP). Apache tiene una participación superior al 60 % de los servidores en todo el mundo. Apache se caracteriza por ser estable, multiplataforma, modular y altamente configurable, lo cual significa que se puede adaptar para satisfacer diferentes necesidades. Apache registra los diferentes eventos que ocurren cuando está en servicio a través de archivos log. De esta manera facilita la obtención de estadísticas que son usadas para la toma de decisiones por parte del administrador. Además, dispone de componentes de seguridad, los cuales pueden ser aprovechados para fortalecer las condiciones de acceso a recursos Web disponibles para ser recuperados a través de solicitudes HTTP realizadas por un navegador, siempre y cuando sean configurados apropiadamente. Apache se caracteriza también por ser Open Source y gratuito. La configuración de Apache se realiza mediante la edición del archivo de texto *httpd.conf*, el cual tiene todas las instrucciones que debe seguir Apache para su funcionamiento".

Desarrollo Seguro de Software

De acuerdo con [6] es importante concienciar a la comunidad de desarrolladores sobre la importancia de incluir temas enfocados con la seguridad del software en las etapas de su desarrollo y la necesidad de implementar el ciclo de vida de desarrollo de software seguro SDLC, como proceso que se centra en la seguridad para minimizar las vulnerabilidades que pueden ser objeto de un ataque.

Problemática de la Seguridad en el Software. "Actualmente, la seguridad es un concepto que todo sistema debe incorporar. La Ingeniería del Software todavía no es capaz de brindar un mecanismo para implementar adecuadamente la seguridad: Los lenguajes tienen primitivas inseguras, el código relativo a la seguridad es siempre un código confuso y complejo debido a técnicas de abstracción insuficientes, etc", son algunos de los conceptos que nos menciona el autor [7] en su artículo "*Aplicación de la Programación Orientada a Aspectos como Solución a los Problemas de la Seguridad en el Software"*.

Se concuerda con este autor en que una de las aproximaciones más comúnmente utilizada para la seguridad es el "ataque-parche" (En inglés "penetrate and patch"), en donde la seguridad es tratada a medida que las fallas se van presentando. Así se va desarrollando un sistema con consideraciones mínimas relacionadas con la seguridad. Posteriormente y una vez que el sistema esté funcionando, se detectarán los ataques e inmediatamente se buscará la forma de corregirlos. Bajo esta aproximación, claramente se observa que poco factible implementar la seguridad de una manera adecuada.

Propiedades del Software Seguro. De acuerdo con lo planteado por [8] el software seguro debe contar con las siguientes propiedades básicas o atributos fundamentales de seguridad: "

- Confidencialidad. El software debe asegurar que cualquiera de sus características (incluidas sus relaciones con su ambiente de ejecución y sus usuarios), los activos que administra y/o su contenido son accesibles sólo para las entidades autorizadas e inaccesibles para el resto.
- Integridad. El software y los activos que administra son resistentes y flexibles a la subversión (modificaciones no autorizadas del código, los activos administrados, la configuración o el comportamiento del software por parte de entidades autorizadas). Esta propiedad se debe preservar durante el desarrollo del software y su ejecución.
- Disponibilidad. El software debe estar operativo y accesible a sus usuarios autorizados (humanos o procesos) siempre que se lo requiera; y desempeñarse con una performance adecuada para que los usuarios puedan realizar sus tareas en forma correcta y dar cumplimiento a los objetivos de la organización que lo utiliza" (p.3,4).

Metodologías para Desarrollar Software Seguro. Según lo comentado por [9] para hacer frente a las amenazas a las que potencialmente está expuesto un software, es necesaria la utilización de metodologías que integren en su proceso de desarrollo, estrategias para eliminar vulnerabilidades y la incorporación de la seguridad en la arquitectura de cualquier producto software como elemento esencial. [9] menciona en su artículo que: "Existen varias metodologías que establecen una serie de pasos en búsqueda de un software más seguro y capaz de resistir ataques. Entre ellas se encuentran Correctness by Construction (CbyC), Security Development Lifecycle (SDL), Cigital Touchpoints, Common Criteria, Comprehensive, Lightweight Application Security Process (CLASP), TSP-Secure" (p.4).

El autor en su artículo hace una descripción muy detallada de las dos primeras metodologías por ser ampliamente conocidas, especificando las particulares de las fases que las conforman y al final, hace una comparativa donde se concluye que no existe una metodología mejor que otra sino aquella que se pueda adaptar al tipo de proyecto de software que se requiera desarrollar.

El Modelado de Amenazas. Conocido también como "Threat Modeling" – TM. [10] menciona en su artículo la importancia de aplicar el modelo de Modelo de Amenazas para incorporar la seguridad en el modelado de sistemas de información, como técnica que apoya la definición de los requerimientos de seguridad además del desarrollo de casos de abuso o del mal uso. Estas técnicas constituyen fuentes adicionales de información para la realización de pruebas de seguridad.

Esta técnica consiste la definición de un esquema estructurado y repetible que ayude a identificar los riesgos y las amenazas en el software desde las etapas tempranas de su desarrollo. Ayuda en un mejor entendimiento de los requerimientos de seguridad del equipo involucrado en el desarrollo y a reconocer problemas en el diseño. El análisis de requerimientos de seguridad del sistema realizado con la técnica TM, debe servir de base para el desarrollo de pruebas de seguridad durante la integración.

Según [10]: "El proceso de TM consta de las siguientes etapas:

1. Conformar un grupo de análisis de riesgo.
2. Descomponer la aplicación e identificar componentes claves.
3. Determinar las amenazas a cada componente de la aplicación.
4. Asignar un valor a cada amenaza.
5. Decidir cómo responder a las amenazas.
6. Identificar las técnicas y tecnologías necesarias para mitigar los riesgos identificados" (p.5).

[10] plantea que el proceso de TM constituye un marco de trabajo para el proceso de análisis de riesgo estructurado, pues permite reconocer las amenazas a las que está expuesta de una aplicación y cuantificar sus riesgos. En su investigación, [10] aplicó esta técnica en un caso concreto y concluyo que el éxito de implementar un modelado de amenazas, constituye un proceso iterativo que pasa por múltiples revisiones y mejoras

durante la evolución del proyecto y favorece la reutilización de componentes que pueden servir para su uso en otros proyectos similares como las aplicaciones Web o posteriores mejoras.

Al igual que [10], [11] también realizó una investigación donde describe la técnica de análisis y gestión de riesgo modelado de amenazas, y la asocia con el entorno de desarrollo seguro de software y de aplicaciones. También concuerda en que los casos de mal uso y de abuso, los modelos de amenazas y los requerimientos de seguridad del software, son fundamentales para desarrollar pruebas de seguridad orientadas a la construcción de casos de prueba basados en la simulación de un ataque y el conocimiento del software y hardware típico, además de ayudar a descubrir defectos de diseño, entre otros.

Amenazas y Vulnerabilidades en Aplicaciones Web

Amenazas de Seguridad según OWASP. Basados en las categorías de riesgos de seguridad de OWASP, en este tema se abordara la descripción de principales vulnerabilidades con mayor riesgo y ataques realizados a las aplicaciones Web. Esta información es de vital importancia, debido a que si alguna de estas vulnerabilidades son detectadas durante la realización de las pruebas de seguridad en un aplicativo, se requiere de su inmediata solución, pues compromete en gran medida la seguridad de la información que gestiona el aplicativo, causando un impacto negativo al negocio.

[12], refieren en su artículo que la comunidad de seguridad de OWASP ha publicado su informe anual en 2015 que captura los principales riesgos en el desarrollo de aplicaciones Web como una relación de la probabilidad de un evento y su consecuencia. La siguiente figura enumera los principales riesgos expuestos por OWASP Top 10, clasificados en categorías y presentados en orden de prevalencia, con su respectiva descripción.

Figura 7. Categorías de Riesgos de Seguridad según OWASP Top 10.

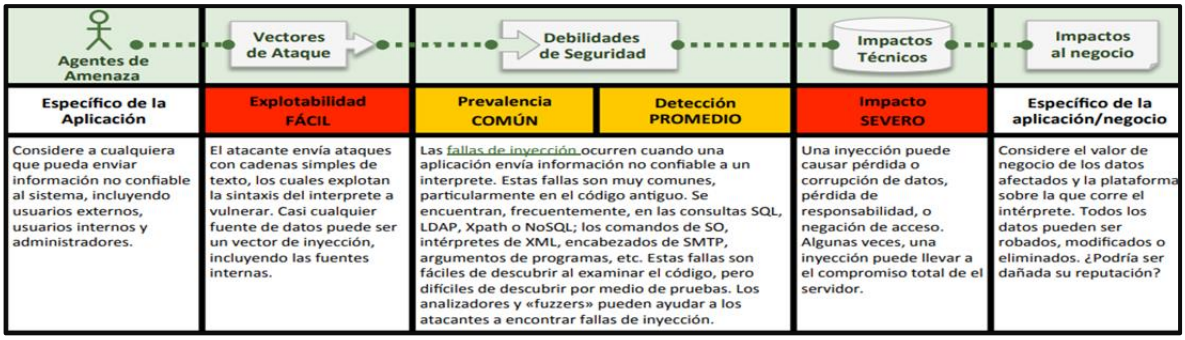
Fuente. Elaboración propia a partir de: [12]. Evaluación del Rendimiento de los Escáneres de Seguridad de Aplicaciones Web para una Defensa más Efectiva.

Como se puede observar, los dos riesgos más comunes en el entorno Web son la inyección SQL, que permite a los atacantes alterar las consultas SQL enviadas a una base de datos y scripts de sitios cruzados (XSS).

Para determinar el riesgo global que una aplicación Web insegura puede causar en una organización, es necesario valorar la probabilidad que se asocia a cada agente de amenaza, el vector de ataque y las debilidades en la seguridad, combinándolas con una estimación del impacto a nivel técnico y de negocios y con esto evaluar finalmente si el daño causado puede no tener consecuencias, o si por el contrario, puede colocar a la organización por fuera del negocio, es decir, determinar si la aplicación Web es vulnerable, junto con las medidas de seguridad que debería tener en cuenta para prevenir un posible ataque. La siguiente tabla evalúa cada riesgo de seguridad según los factores mencionados anteriormente.

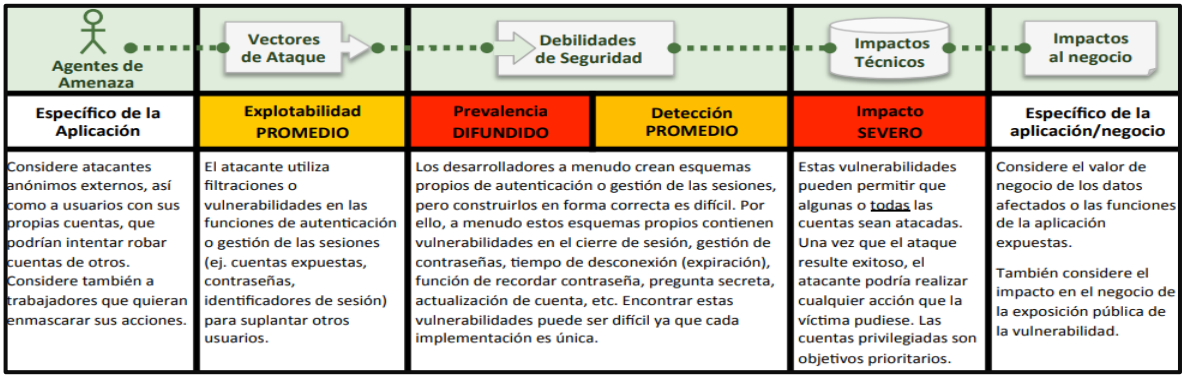
Tabla 3. Evaluación del Riesgo de Seguridad.

A1. Inyección



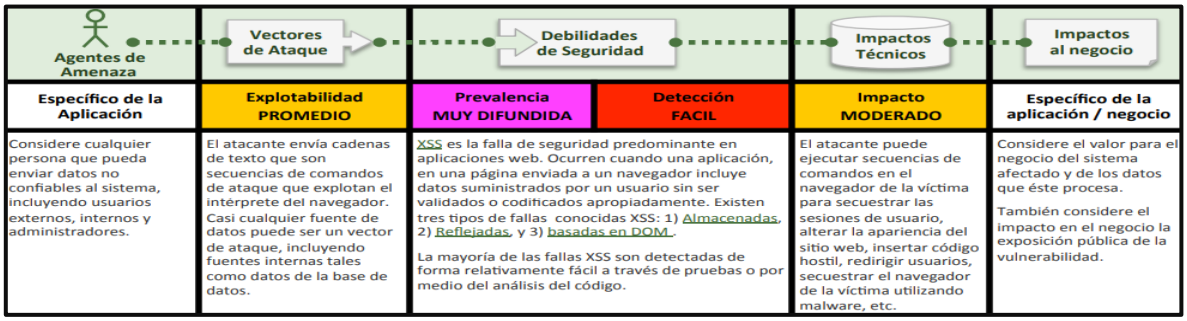
Específico de la Aplicación	Explotabilidad FÁCIL	Prevalencia COMÚN	Detección PROMEDIO	Impacto SEVERO	Específico de la aplicación/negocio
Considere a cualquiera que pueda enviar información no confiable al sistema, incluyendo usuarios externos, usuarios internos y administradores.	El atacante envía ataques con cadenas simples de texto, los cuales explotan la sintaxis del interprete a vulnerar. Casi cualquier fuente de datos puede ser un vector de inyección, incluyendo las fuentes internas.	Las fallas de inyección ocurren cuando una aplicación envía información no confiable a un interprete. Estas fallas son muy comunes, particularmente en el código antiguo. Se encuentran, frecuentemente, en las consultas SQL, LDAP, Xpath o NoSQL; los comandos de SO, intérpretes de XML, encabezados de SMTP, argumentos de programas, etc. Estas fallas son fáciles de descubrir al examinar el código, pero difíciles de descubrir por medio de pruebas. Los analizadores y «fuzzers» pueden ayudar a los atacantes a encontrar fallas de inyección.		Una inyección puede causar pérdida o corrupción de datos, pérdida de responsabilidad, o negación de acceso. Algunas veces, una inyección puede llevar a el compromiso total de el servidor.	Considere el valor de negocio de los datos afectados y la plataforma sobre la que corre el intérprete. Todos los datos pueden ser robados, modificados o eliminados. ¿Podría ser dañada su reputación?

A2. Pérdida de Autenticación y Gestión de Sesiones



Específico de la Aplicación	Explotabilidad PROMEDIO	Prevalencia DIFUNDIDO	Detección PROMEDIO	Impacto SEVERO	Específico de la aplicación/negocio
Considere atacantes anónimos externos, así como a usuarios con sus propias cuentas, que podrían intentar robar cuentas de otros. Considere también a trabajadores que quieran enmascarar sus acciones.	El atacante utiliza filtraciones o vulnerabilidades en las funciones de autenticación o gestión de las sesiones (ej. cuentas expuestas, contraseñas, identificadores de sesión) para suplantar otros usuarios.	Los desarrolladores a menudo crean esquemas propios de autenticación o gestión de las sesiones, pero construirlos en forma correcta es difícil. Por ello, a menudo estos esquemas propios contienen vulnerabilidades en el cierre de sesión, gestión de contraseñas, tiempo de desconexión (expiración), función de recordar contraseña, pregunta secreta, actualización de cuenta, etc. Encontrar estas vulnerabilidades puede ser difícil ya que cada implementación es única.		Estas vulnerabilidades pueden permitir que algunas o todas las cuentas sean atacadas. Una vez que el ataque resulte exitoso, el atacante podría realizar cualquier acción que la víctima pudiese. Las cuentas privilegiadas son objetivos prioritarios.	Considere el valor de negocio de los datos afectados o las funciones de la aplicación expuestas. También considere el impacto en el negocio de la exposición pública de la vulnerabilidad.

A3. Secuencia de Comandos en Sitios Cruzados (XSS)



Específico de la Aplicación	Explotabilidad PROMEDIO	Prevalencia MUY DIFUNDIDA	Detección FACIL	Impacto MODERADO	Específico de la aplicación / negocio
Considere cualquier persona que pueda enviar datos no confiables al sistema, incluyendo usuarios externos, internos y administradores.	El atacante envía cadenas de texto que son secuencias de comandos de ataque que explotan el intérprete del navegador. Casi cualquier fuente de datos puede ser un vector de ataque, incluyendo fuentes internas tales como datos de la base de datos.	XSS es la falla de seguridad predominante en aplicaciones web. Ocurren cuando una aplicación, en una página enviada a un navegador incluye datos suministrados por un usuario sin ser validados o codificados apropiadamente. Existen tres tipos de fallas conocidas XSS: 1) Almacenadas, 2) Reflejadas, y 3) basadas en DOM. La mayoría de las fallas XSS son detectadas de forma relativamente fácil a través de pruebas o por medio del análisis del código.		El atacante puede ejecutar secuencias de comandos en el navegador de la víctima para secuestrar las sesiones de usuario, alterar la apariencia del sitio web, insertar código hostil, redirigir usuarios, secuestrar el navegador de la víctima utilizando malware, etc.	Considere el valor para el negocio del sistema afectado y de los datos que éste procesa. También considere el impacto en el negocio la exposición pública de la vulnerabilidad.

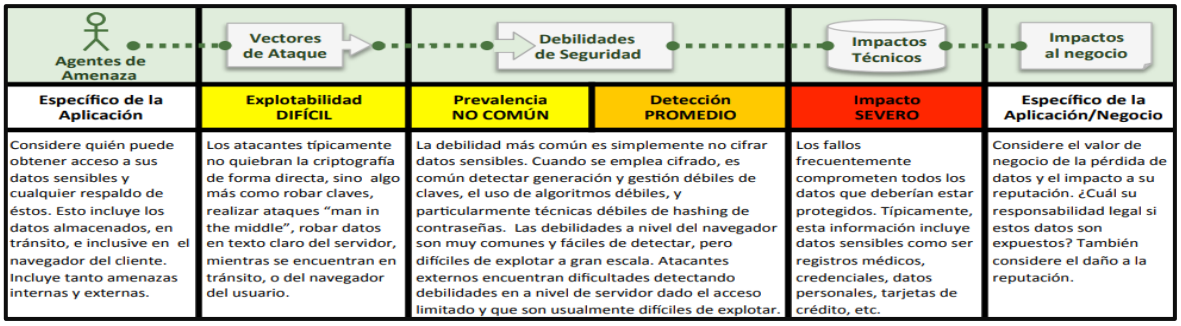
A4. Referencia Directa Insegura a Objetos

Específico de la Aplicación	Explotabilidad FÁCIL	Prevalencia COMÚN	Detección FÁCIL	Impacto MODERADO	Específico de la aplicación/negocio
Considere los tipos de usuarios en su sistema. ¿Existen usuarios que tengan únicamente acceso parcial a determinados tipos de datos del sistema?	Un atacante, como usuario autorizado en el sistema, simplemente modifica el valor de un parámetro que se refiere directamente a un objeto del sistema por otro objeto para el que el usuario no se encuentra autorizado. ¿Se concede el acceso?	Normalmente, las aplicaciones utilizan el nombre o clave actual de un objeto cuando se generan las páginas web. Las aplicaciones no siempre verifican que el usuario tiene autorización sobre el objetivo. Esto resulta en una vulnerabilidad de referencia de objetos directos inseguros. Los auditores pueden manipular fácilmente los valores del parámetro para detectar estas vulnerabilidades. Un análisis de código muestra rápidamente si la autorización se verifica correctamente.		Dichas vulnerabilidades pueden comprometer toda la información que pueda ser referida por parámetros. A menos que el espacio de nombres resulte escaso, para un atacante resulta sencillo acceder a todos los datos disponibles de ese tipo.	Considere el valor de negocio de los datos afectados o las funciones de la aplicación expuestas. También considere el impacto en el negocio de la exposición pública de la vulnerabilidad.

A5. Configuración de Seguridad Incorrecta

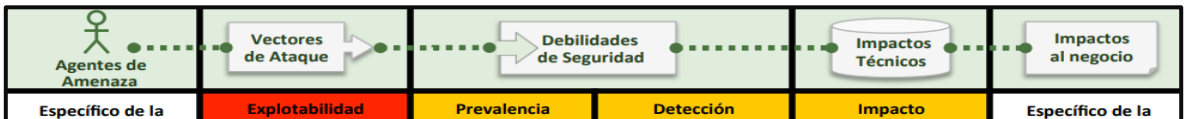
Específico de la Aplicación	Explotabilidad FÁCIL	Prevalencia COMÚN	Detección FÁCIL	Impacto MODERADO	Específico de la aplicación / negocio
Considere atacantes anónimos externos así como usuarios con sus propias cuentas que pueden intentar comprometer el sistema. También considere personal interno buscando enmascarar sus acciones.	Un atacante accede a cuentas por defecto, páginas sin uso, fallas sin parchear, archivos y directorios sin protección, etc. para obtener acceso no autorizado o conocimiento del sistema.	Las configuraciones de seguridad incorrectas pueden ocurrir a cualquier nivel de la aplicación, incluyendo la plataforma, servidor web, servidor de aplicación, base de datos, framework, y código personalizado. Los desarrolladores y administradores de sistema necesitan trabajar juntos para asegurar que las distintas capas están configuradas apropiadamente. Las herramientas de detacción automatizadas son útiles para detectar parches omitidos, fallos de configuración, uso de cuentas por defecto, servicios innecesarios, etc.		Estas vulnerabilidades frecuentemente dan a los atacantes acceso no autorizado a algunas funcionalidades o datos del sistema. Ocasionalmente provocan que el sistema se comprometa totalmente.	El sistema podría ser completamente comprometido sin su conocimiento. Todos sus datos podrían ser robados o modificados lentamente en el tiempo. Los costes de recuperación podrían ser altos.

A6. Exposición de Datos Sensibles



Específico de la Aplicación	Explotabilidad DIFÍCIL	Prevalencia NO COMÚN	Detección PROMEDIO	Impacto SEVERO	Específico de la Aplicación/Negocio
Considere quién puede obtener acceso a sus datos sensibles y cualquier respaldo de éstos. Esto incluye los datos almacenados, en tránsito, e inclusive en el navegador del cliente. Incluye tanto amenazas internas y externas.	Los atacantes típicamente no quiebran la criptografía de forma directa, sino algo más como robar claves, realizar ataques "man in the middle", robar datos en texto claro del servidor, mientras se encuentran en tránsito, o del navegador del usuario.	La debilidad más común es simplemente no cifrar datos sensibles. Cuando se emplea cifrado, es común detectar generación y gestión débiles de claves, el uso de algoritmos débiles, y particularmente técnicas débiles de hashing de contraseñas. Las debilidades a nivel del navegador son muy comunes y fáciles de detectar, pero difíciles de explotar a gran escala. Atacantes externos encuentran dificultades detectando debilidades en a nivel de servidor dado el acceso limitado y que son usualmente difíciles de explotar.		Los fallos frecuentemente comprometen todos los datos que deberían estar protegidos. Típicamente, esta información incluye datos sensibles como ser registros médicos, credenciales, datos personales, tarjetas de crédito, etc.	Considere el valor de negocio de la pérdida de datos y el impacto a su reputación. ¿Cuál su responsabilidad legal si estos datos son expuestos? También considere el daño a la reputación.

A7. Inexistente Control de Acceso a Nivel de Funcionalidades



Específico de la Aplicación	Explotabilidad FÁCIL	Prevalencia COMÚN	Detección PROMEDIO	Impacto MODERADO	Específico de la aplicación/negocio
Cualquiera con acceso a la red puede enviar una petición a su aplicación. ¿Un usuario anónimo podría acceder a una funcionalidad privada o un usuario normal acceder a una función que requiere privilegios?	El atacante, que es un usuario legítimo en el sistema, simplemente cambia la URL o un parámetro a una función con privilegios. ¿Se le concede acceso? Usuarios anónimos podrían acceder a funcionalidades privadas que no estén protegidas.	Las aplicaciones no siempre protegen las funcionalidades adecuadamente. En ocasiones la protección a nivel de funcionalidad se administra por medio de una configuración, y el sistema está mal configurado. Otras veces los programadores deben incluir un adecuado chequeo por código, y se olvidan. La detección de este tipo de vulnerailidad es sencillo. La parte más compleja es identificar qué páginas (URLs) o funcionalidades atacables existen.		Estas vulnerabilidades permiten el acceso no autorizado de los atacantes a funciones del sistema. Las funciones administrativas son un objetivo clave de este tipo de ataques.	Considere el valor para su negocio de las funciones expuestas y los datos que éstas procesan. Además, considere el impacto a su reputación si esta vulnerabilidad se hiciera pública.

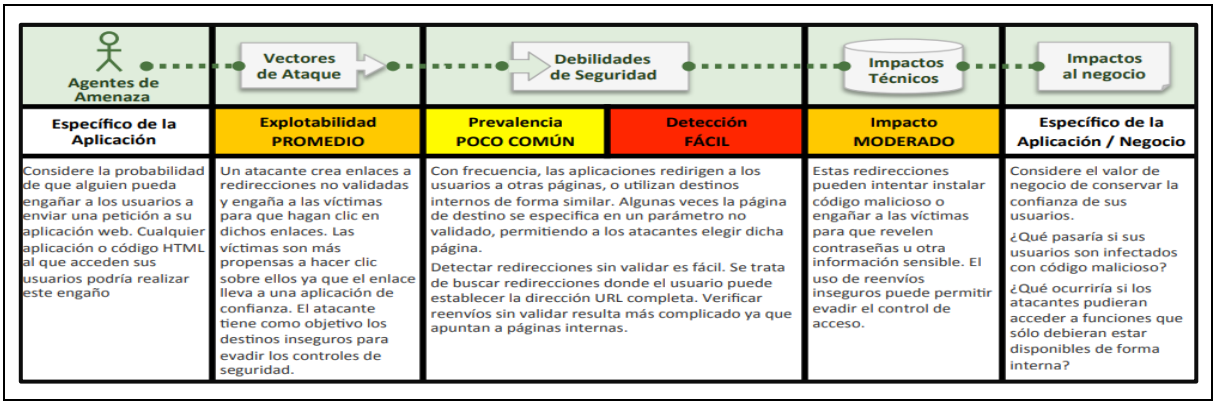
A8. Falsificación de Peticiones en Sitios Cruzados (CSRF)

Específico de la Aplicación	Explotabilidad PROMEDIO	Prevalencia COMÚN	Detección FÁCIL	Impacto MODERADO	Específico de la aplicación/negocio
Considere cualquier persona que pueda cargar contenido en los navegadores de los usuarios, y así obligarlos a presentar una solicitud para su sitio web. Cualquier sitio web o canal HTML que el usuario acceda puede realizar este tipo de ataque.	El atacante crea peticiones HTTP falsificadas y engaña a la víctima mediante el envío de etiquetas de imágenes, XSS u otras técnicas. Si el usuario está autenticado, el ataque tiene éxito.	CSRF aprovecha el hecho que la mayoría de las aplicaciones web permiten a los atacantes predecir todos los detalles de una acción en particular. Dado que los navegadores envían credenciales como cookies de sesión de forma automática, los atacantes pueden crear páginas web maliciosas que generan peticiones falsificadas que son indistinguibles de las legítimas. La detección de fallos de tipo CSRF es bastante fácil a través de pruebas de penetración o de análisis de código.		Los atacantes pueden cambiar cualquier dato que la víctima esté autorizada a cambiar, o a acceder a cualquier funcionalidad donde esté autorizada, incluyendo registro, cambios de estado o cierre de sesión.	Considerar el valor de negocio asociado a los datos o funciones afectados. Tener en cuenta lo que representa no estar seguro si los usuarios en realidad desean realizar dichas acciones. Considerar el impacto que tiene en la reputación de su negocio.

A9. Uso de Componentes con Vulnerabilidades Conocidas

Específico de la Aplicación	Explotabilidad PROMEDIO	Prevalencia DIFUNDIDO	Detectabilidad DIFÍCIL	Impacto MODERADO	Específico de la aplicación / negocio
Algunos componentes vulnerables (por ejemplo frameworks) pueden ser identificados y explotados con herramientas automatizadas, aumentando las opciones de la amenaza más allá del objetivo atacado.	El atacante identifica un componente débil a través de escaneos automáticos o análisis manuales. Ajusta el exploit como lo necesita y ejecuta el ataque. Se hace más difícil si el componente es ampliamente utilizado en la aplicación.	Virtualmente cualquier aplicación tiene este tipo de problema debido a que la mayoría de los equipos de desarrollo no se enfocan en asegurar que sus componentes / bibliotecas se encuentren actualizadas. En muchos casos, los desarrolladores no conocen todos los componentes que utilizan, y menos sus versiones. Dependencias entre componentes dificultan incluso más el problema.		El rango completo de debilidades incluye inyección, control de acceso roto, XSS, etc. El impacto puede ser desde mínimo hasta apoderamiento completo del equipo y compromiso de los datos.	Considere qué puede significar cada vulnerabilidad para el negocio controlado por la aplicación afectada. Puede ser trivial o puede significar compromiso completo.

A10. Redirecciones y Reenvíos no Válidos



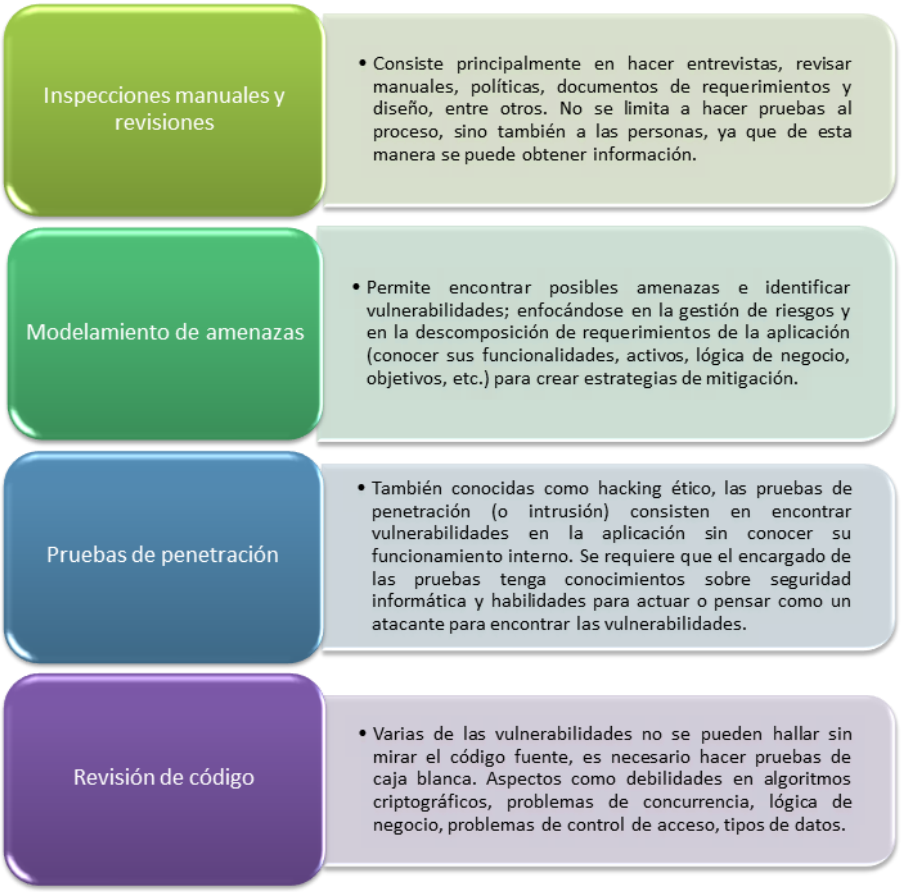
Fuente. Recuperado de: https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Espa%C3%B1ol.pdf.

Pruebas de Seguridad para Detección de Vulnerabilidades. En muy importante que cada proyecto de software definan un modelo a seguir para el correcto manejo de los riesgos y vulnerabilidades que se puedan presentar. Estos modelos de desarrollo abarcan los objetivos, el alcance y todo el proceso de evolución del software.

[13] define las pruebas de seguridad como un conjunto de actividades que ejecutan para encontrar fallas y vulnerabilidades en aplicaciones Web, buscando disminuir el impacto de ataques a ellas y pérdida de información importante. La seguridad en aplicaciones Web busca asegurar la integridad, la confidencialidad y la disponibilidad de los datos y funciones que maneja el software, teniendo en cuenta el impacto que pueden tener fallas de seguridad según el contexto empresarial.

Técnicas de Pruebas de Seguridad. [14] menciona en su artículo que la guía de pruebas OWASP, define cuatro técnicas principales para llevar a cabo pruebas de seguridad en aplicaciones Web, las cuales se describen a continuación:

Figura 8. Técnicas para Realizar Pruebas de Seguridad en Aplicaciones Web.



Fuente. Elaboración propia a partir de [13]. Pruebas de Seguridad en Aplicaciones Web como Imperativo en la Calidad de Desarrollo del Software.

La metodología de pruebas de OWASP se basa en las distintas categorías de las vulnerabilidades de las aplicaciones Web. El enfoque de OWASP, además de ser orientada a las pruebas de seguridad en todas las fases de desarrollo del software, también se centra en las pruebas de intrusión o caja negra; que es la manera como la mayoría de atacantes puede tener acceso a la aplicación (teniendo en cuenta que obtener el código fuente no es una tarea fácil, pudiendo explotar vulnerabilidades del software). [13].

El artículo de [15] presenta, al igual que las guías de pruebas de OWASP, otras técnicas para detección de vulnerabilidades:

Figura 9. Técnicas para Detectar Vulnerabilidades en Aplicaciones Web

Fuente. Elaboración propia a partir de [15]. Guía de Ataques, Vulnerabilidades, Técnicas y Herramientas para Aplicaciones Web.

Herramientas para la Detección de Vulnerabilidades. En la revisión bibliográfica realizada, se encontraron varios autores que han investigado no solo las vulnerabilidades de las aplicaciones Web, sino en las herramientas que se utilizan para realizar los procesos automáticos de detección de vulnerabilidades, los cuales son de bastante ayuda para las personas encargadas de realizar este tipo de pruebas.

La siguiente tabla es un compendio de herramientas utilizadas para el análisis estático y dinámico de código, las describen los autores [16], [15] y [12].

Tabla 4. Herramientas Utilizadas para la Detectar Vulnerabilidades Web

Herramientas para Detección de Vulnerabilidades Web	Descripción
Acunetix 	Es una herramienta para la detección de vulnerabilidades en los aplicativos web. Esta herramienta consiste en detectar fallos de seguridad y advertir a los desarrolladores y administradores web de posibles ataques de usuarios que pudieran tener acceso a los datos y en general al sistema de información.
Netsparker 	Es una herramienta fácil de usar, tiene una interfaz de usuario muy buena y realiza un buen trabajo al detectar las vulnerabilidades más importantes. Genera buenos informes que son fáciles de leer y de diseño intuitivo, además tiene la capacidad de confirmar las vulnerabilidades detectadas.
Burp Suite 	Es una herramienta fácil de usar e intuitivo. También es altamente configurable y contiene numerosas funciones poderosas para ayudar a los probadores más experimentados con su trabajo.
VEGA 	Es una herramienta utilizada para realizar pruebas de seguridad a los aplicativos web. Utiliza código abierto. Esta herramienta ofrece el escaneo de vulnerabilidades a los aplicativos web basada en técnicas comunes de ataque a los Ambientes web.
Appscan 	Es una herramienta de pruebas que permite el rápido desarrollo de la seguridad. Detecta los efectos de la seguridad automáticamente, con un componente integrado al desarrollo.
Webinspect 	Es una solución de prevención de riesgos y ataques a nivel del código de aplicación. Facilita a los usuarios la identificación de todas las vulnerabilidades que afecten a las aplicaciones.
OWASP ZAP 	Es una herramienta con gran reconocimiento dentro del programa OWASP. El diseño de esta herramienta está orientado a monitorear los aplicativos en un enfoque orientado a la gestión de auditorías.
QualysGuard Web Application Scanning WAS 	Es una herramienta con enfoque en la detección de vulnerabilidades de penetración, refiriendo también las técnicas encontradas en OWASP.

WebSite Security Audit- WSSA	Es una herramienta utilizada para realizar pruebas en cuanto a la seguridad de aplicativos, servidores y páginas web. Su enfoque esta para pruebas como SQL injection y Cross Site.
Retina Web Security Scanner	Esta herramienta contiene componentes adicionales para la detección de vulnerabilidades ya que es capaz de reportarlas por su jerarquía en los diferentes niveles de riesgo, lo que aprueba que se consideren unas mejores prácticas de seguridad en los aplicativos que puedan acceder a dicho software.
WEBAPP 360	Es una herramienta que trabaja con las técnicas de riesgos de OWASP, además de realzar escaneo a los aplicativos web, revisa también la subestructura del aplicativo, generando una mayor confianza de utilización.
Frame-C	Es una herramienta específica para analizar aplicativos realizados en lenguaje C, tiene la ventaja de incorporar reportes estadísticos.
Parasoft C/C++ Test	Es una herramienta delimitada para aplicativos desarrollados en lenguaje C, incorpora las técnicas de OWASP
Fortify Static Code Analyzer	Es una herramienta que proporciona a los desarrolladores análisis del código para tener presente los controles de seguridad en el software, generando buenas prácticas en el desarrollo seguro.
MCAfee Vulnerability Manager	Es una herramienta que tiene un enfoque paralelo de monitoreo. Está diseñada para las pruebas de penetración. Trabaja con las técnicas del OWASP.
Nessus Vulnerability Scanner	Esta herramienta está diseñada para evidenciar los faltantes de parches actualizados en el software. Los reportes los exporta en formatos pdf y archivos planos.
Nexpose Vulnerability Manager	Es una herramienta diseñada para dar trámite a la detección de vulnerabilidades ajustándole paralelamente los controles y la respectiva planificación del riesgo.

	
Whatweb	Es una herramienta enfocada en la administración del software del aplicativo y en las versiones que se cuenta para el manejo de los correos electrónicos.

Fuente. Elaboración propia a partir de [16]. Capacidades de Detección de las Herramientas de Análisis de Vulnerabilidades en Aplicaciones Web, [15]. Guía de Ataques, Vulnerabilidades, Técnicas y Herramientas para Aplicaciones Web y [12]. Evaluación del Rendimiento de los Escáneres de Seguridad de Aplicaciones Web para una Defensa más Efectiva.

ESTRATEGIA METODOLÓGICA

Existen diferentes técnicas utilizadas para comprobar el estado de seguridad de un aplicativo Web. A continuación se hará un análisis del procedimiento para la detección de vulnerabilidades del sistema basado en los componentes de un modelo de pruebas de seguridad y también la aplicación del enfoque del modelado de amenazas que ayuda a explorar vulnerabilidades potenciales y crear estrategias de mitigación, para finalmente seleccionar el enfoque metodológico más conveniente a seguir de acuerdo al estado de desarrollo del aplicativo.

Enfoque Metodológico Basado en Pruebas de Seguridad

Componentes Fundamentales del Modelo de Pruebas de Seguridad. Como estrategia para evidenciar las vulnerabilidades en el sistema de información Web, se tendrán en cuenta algunos lineamientos propuestos en la guía de pruebas de seguridad de aplicaciones Web según OWASP[9], como marco de referencia debido a la problemática planteada, pues no se puede garantizar la seguridad de una aplicación Web sin hacer las respectivas pruebas de seguridad en ella. Estas pruebas de seguridad son utilizadas para validar los requerimientos de seguridad del aplicativo Web, dependiendo de su naturaleza y de los lineamientos de la organización para la cual está siendo desarrollada. Su aplicación durante la fase de desarrollo garantiza que los componentes del software desarrollados, contengan un nivel de seguridad comprobado antes de ser integrados con otros componentes del aplicativo y avanzar de forma más segura a la siguiente fase.

Teniendo en cuenta que la metodología de pruebas que plantea la guía OWASP, se basa en una técnica de comprobación de caja negra, es decir que la persona que realiza las pruebas no posee ninguna información sobre la aplicación que va a ser comprobada. El modelo de pruebas de ésta metodología, tiene tres compones fundamentales:

1. El auditor que es la persona encargada de realizar las actividades de comprobación de vulnerabilidades. Estos auditores de seguridad pueden tener los siguientes perfiles:

    - Desarrolladores de Software: Personas que necesitan asegurarse de que el código implementado no es vulnerable a ataques, ya que son los únicos que entienden como se ha desarrollado el software desde la fase de implementación y por lo tanto son las personas idóneas para diseñar pruebas efectivas que comprueben el nivel de seguridad de la misma.

[9] La Guía de Pruebas de OWASP (OWASP Testing Guide). Es un guía del Proyecto Abierto de Seguridad de Aplicaciones Web (OWASP) que recopila y estructura todas las posibles pruebas de seguridad orientadas hacia las aplicaciones Web.
https://www.owasp.org/images/8/80/Gu%C3%ADa_de_pruebas_de_OWASP_ver_3.0.pdf

- Testers de Software: Personas encargadas de detectar la mayor cantidad de fallas posibles en el aplicativo con el mínimo esfuerzo con el objetivo de asegurar la máxima calidad del producto.
- Especialistas de Seguridad: Personas que tiene la responsabilidad de asegurarse de que las aplicaciones no salgan a producción con vulnerabilidades, verificando su seguridad de forma completa.

2. Las herramientas y metodología empleada para la realización de pruebas relacionadas con los riesgos de seguridad más críticos en aplicaciones Web según el TOP 10 de OWASP[10]. En este caso se utilizaran herramientas automatizadas para realización de pruebas unitarias de seguridad, las cuales están enfocadas en el análisis de código fuente para verificar que en tiempo de ejecución, los componentes funcionan como se esperaba.

3. La aplicación vulnerable que es la caja sobre la que se van a realizar las pruebas aplicando las herramientas de acuerdo a la metodología empleada. En este caso se trata del aplicativo Web de Gestión y Seguimiento de Novedades de Aprendices del Programa de Articulación con la Educación Media del Servicio Nacional del Aprendizaje - SENA, el cual se encuentra en su fase de implementación, por lo tanto se considera muy pertinente la detección de potenciales vulnerabilidades que afecten la calidad del software antes de probar el prototipo en el entorno de trabajo, ya que el objetivo es que el aplicativo sea integrado como plataforma institucional de gestión de procesos administrativos en el programa de articulación con la educación media que lidera el SENA y se ajuste a los lineamientos del Sistema de Gestión de Seguridad de la Información que esta institución tiene implementado como ventaja competitiva.

A continuación se hace una descripción aplicada de cada uno de los componentes fundamentales del modelo de pruebas que plantea la guía OWASP, tomándolos como referencia para establecer una estrategia que se pueda utilizar de manera práctica para la identificación rápida de vulnerabilidades como mecanismo de seguridad en el aplicativo Web.

Auditor de Seguridad. Actualmente el aplicativo Web se encuentra en su fase de implementación, en donde es fundamental la utilización de buenas prácticas en cuanto a la codificación segura de software. En éste caso el perfil del auditor de seguridad debe ser

[10] Es un documento que especifica los diez riesgos de seguridad más importantes en aplicaciones Web según del Proyecto Abierto de Seguridad de Aplicaciones Web (OWASP).Esta lista se publica y actualiza cada 3 años.

el desarrollador senior que lidera el grupo de programadores del software y de la base de datos y quien debe ser también el especialista en materia de software seguro para tomar decisiones sobre los cambios que se integrarán en la construcción de la aplicación, pues sobre éstos está la difícil tarea de desarrollar una aplicación Web segura. Particularmente para el Sistema de Información Web de Gestión y Seguimiento de Novedades de Aprendices del Programa de Articulación con la Educación Media del Servicio Nacional del Aprendizaje - SENA, se requiere la identificación de potenciales vulnerabilidades Web que no han sido detectadas en ésta fase de desarrollo y poderlas mitigar mediante planes de remediación y corrección, reduciendo así los riesgos en materia de seguridad. Posteriormente cuando el aplicativo se encuentre totalmente desarrollado, se requiere que el perfil de un especialista en seguridad de la Información pueda certificar la calidad del aplicativo mediante la realización de auditorías, revisión de código y la verificación de requerimientos de seguridad cumplidos en aplicativo Web.

Herramientas Seleccionadas para la Detección de Vulnerabilidades Web. Los escáneres de vulnerabilidades Web son herramientas de software que realiza pruebas de caja negra sobre la aplicación Web detectando vulnerabilidades o fallos de seguridad, no tienen acceso al código fuente y sólo realizan pruebas funcionales. En el mercado existen una gran variedad de éstas herramientas, algunas de éstas son comerciales y garantizan facilidad en la instalación y utilización; otras herramientas son gratuitas pero actúan con menor eficacia de detección y dificultad en su instalación y utilización; también existen herramientas online que ofrecen soluciones gratuitas, funcionan desde el navegador y dan resultados muy completos en poco tiempo. En general, estas herramientas ofrecen una serie de ventajas para el análisis automático de vulnerabilidades, principalmente el ahorro en tiempo empleado para la detección de las vulnerabilidades que afectan a los sistemas escaneados, el planteamiento de soluciones propuestas para mitigarlos problemas encontrados y su facilidad de uso, brindando la posibilidad de producir un código más seguro antes de poner la aplicación en un entorno de producción. Aunque estas herramientas son ampliamente empleadas para la detección de vulnerabilidades en aplicaciones Web, no son la única técnica utilizada para la comprobación de la seguridad de las aplicaciones.

En la investigación realizada por los autores [16] sobre la efectividad de detección de las herramientas de escaneo de vulnerabilidades en aplicaciones Web, se concluyó que OWASP ZAP y ACUNETIX WVS son herramientas que detectaron la mayor cantidad de vulnerabilidades clasificadas en ULWeVA (clasificación unificada de 63 vulnerabilidades), por encima de otras. A continuación se describirán las características de estas herramientas seleccionadas para realizar un escaneo de vulnerabilidades en el aplicativo Web en el ámbito de la realización de pruebas de seguridad.

- **OWASP Zed Attack Proxy Project**

El Zed Attack Proxy (ZAP)[11] de OWASP es una herramienta gratuita de seguridad considerada una de las más populares en el mundo. Es una de las aplicaciones del proyecto OWASP más activas utilizadas en la realización de auditorías o pruebas de seguridad. Es utilizada para encontrar una amplia variedad de vulnerabilidades de seguridad en las aplicaciones Web mediante la realización de pentest[12], ya sea de forma manual o mediante análisis automáticos.

Esta herramienta está disponible para Unix/Linux, Windows y Mac OS. Se puede ejecutar como un escáner introduciendo la URL para realizar el escaneo automático, o se puede utilizar como proxy de interceptación para realizar manualmente pruebas en páginas específicas.

Para controlar la seguridad de las aplicaciones Web, la herramienta dispone en general de las siguientes funciones y análisis específicos:

- ✓ Capacidad de comprobar todas las peticiones y respuestas entre el equipo cliente y el equipo servidor.
- ✓ Capacidad de localizar recursos en un servidor.
- ✓ Configuración de un sistema de políticas para elegir las reglas que formarán parte del análisis.
- ✓ Diálogos de escaneo con opciones de configuración avanzadas.
- ✓ Análisis automáticos.
- ✓ Escaneos pasivos y activos.
- ✓ Capacidad de lanzar varios ataques en forma simultánea.
- ✓ Capacidad para utilizar certificados SSL dinámicos.
- ✓ Soporte para utilizar tarjetas inteligentes y certificados personales.
- ✓ Análisis de sistemas de autenticación.
- ✓ Disponibilidad de una serie de plugins o extensiones para añadirle más funcional a la herramienta, por ejemplo para generación de reportes de alertas personalizadas.

La interfaz gráfica de usuario de OWASP ZAP contiene los siguientes elementos:

1. Barra de menús: Brinda acceso a muchas de las herramientas automáticas y manuales.
2. Barra de herramientas: Incluye botones que brindan fácil acceso a las características comúnmente utilizadas.

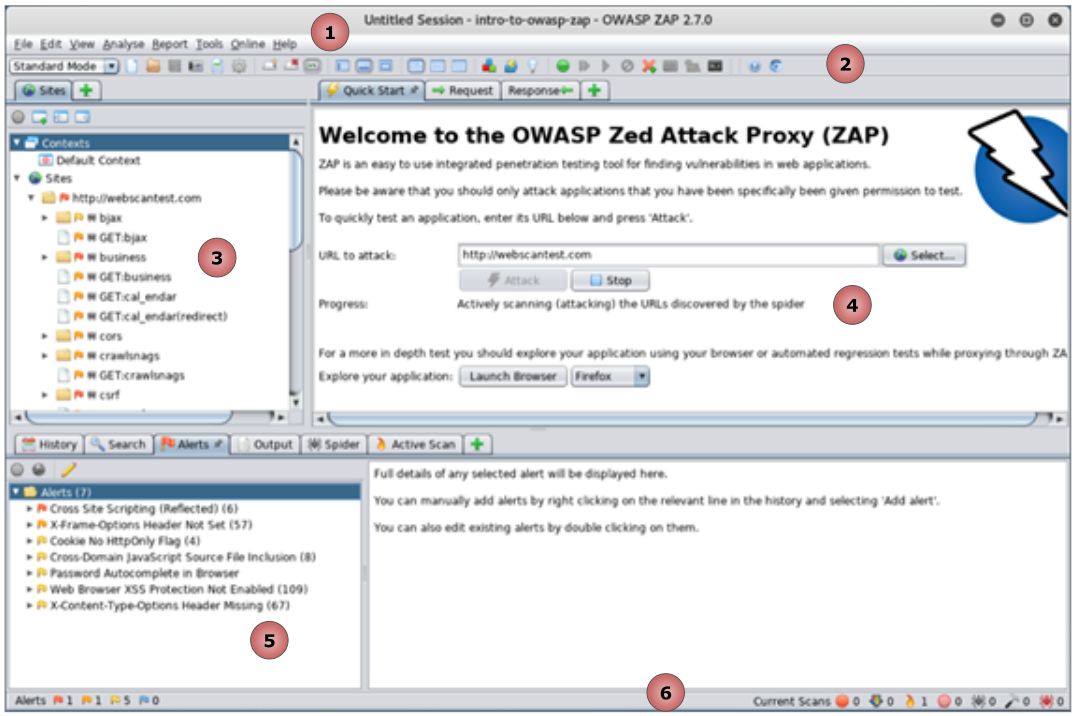
---

[11] Es un proyecto desarrollado por la comunidad de OWASP. Su página oficial es https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project.

[12] También son conocidas como pruebas de penetración, su objetivo es encontrar las debilidades de seguridad mediante el ataque a un sistema informático. Ayudan a determinar si un sistema es vulnerable a ataques y la posibilidad de tener acceso a su funcionalidad y datos.
https://es.wikipedia.org/wiki/Examen_de_penetraci%C3%B3n

3. Ventana de árbol: Muestra el árbol sitios y el árbol de scripts.
4. Ventana del área de trabajo: Muestra solicitudes, respuestas y scripts.
5. Ventana de información: Muestra detalles de las herramientas automáticas y manuales.
6. Pie de página: Muestra un resumen de las alertas encontradas y el estado de las principales herramientas automatizadas.

Figura 10. Interfaz Gráfica de Usuario de OWASP ZAP.



Fuente. Elaboración propia a partir de: https://alvasky.com/en/security/getting-started-with-owasp-zed-attack-proxy-zap-2/

El modo de funcionamiento que utiliza la herramienta OWASP ZAP para la búsqueda de vulnerabilidades de forma automática, define prácticamente la metodología a utilizar para obtener un reporte del estado actual de seguridad del aplicativo Web, mediante los siguientes pasos:

1. Identificación de todas las URL del sitio con el Spider[13].
2. Realización de un escaneo activo de todas las URLs del sitio obtenidas la herramienta Spider.
3. Análisis del contenido de cada URL y visualización de alertas dependiendo de la criticidad de la vulnerabilidad encontrada.

[13] Funcionalidad que tiene la herramienta OWASP ZAP para para analizar directorios y rutas de un sitio Web.

- **Acunetix Web Vulnerability Scanner**

Acunetix WVS[14] es una herramienta comercial automatizada para la ejecución de pruebas de seguridad en aplicaciones Web. Éste escáner de vulnerabilidades Web es ampliamente utilizado por incluir la tecnología de escaneo de caja negra XSS y la inyección SQL más avanzada. Rastrea de forma automática los sitios Web detectando vulnerabilidades peligrosas que pueden comprometer la seguridad del mismo.

Las principales características que ofrece Acunetix WVS para la solución de las vulnerabilidades Web son las siguientes:

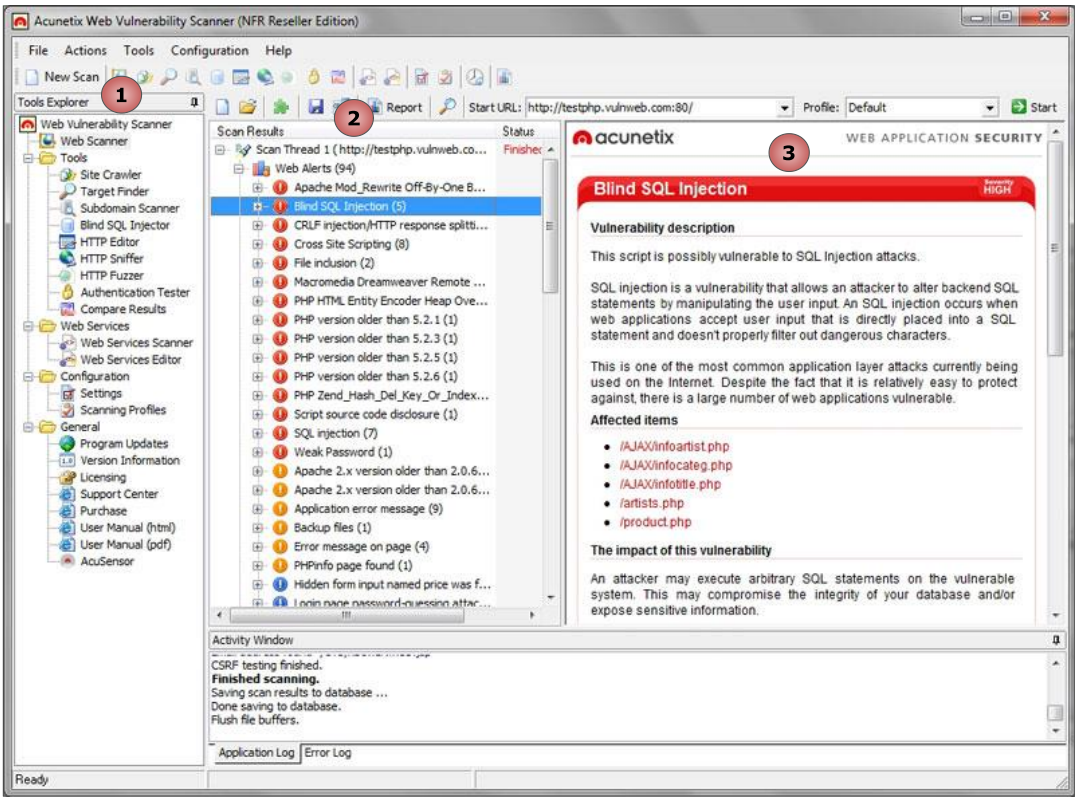
- ✓ Comprueba diferentes tipos de vulnerabilidades Web como SQL injection, Cross Site, CRLF injection, Directory traversal, Code execution, Authentication, File inclusion, entre otras.
- ✓ Escanear cualquier sitio Web accesible a través del protocolo HTTP / HTTPS.
- ✓ Incluye herramientas de pruebas de penetración manuales que contribuyen a realización de pruebas automatizadas y de vulnerabilidades lógicas.
- ✓ Realiza un escaneo y comprobación de puertos del servidor donde se tiene alojado el sitio Web en busca de potenciales vulnerabilidades.
- ✓ Ejecuta una serie de pruebas configurables por el usuario para la identificación de las vulnerabilidades, ya sea en la programación de la página o también en la configuración del servidor.
- ✓ Analiza formularios protegidos con contraseñas débilmente seguras.
- ✓ Analiza la seguridad al inicia de sesión en las webs que tienen implementado la accesibilidad mediante CAPTCHA, en el inicio de sesión único y donde se utilicen mecanismos 2FA (Doble Factor de Autenticación).
- ✓ Permite la generación de informes técnicos de seguridad.
- ✓ Capacidad de escaneo multi-hilo que permite analizar varias webs simultáneamente.
- ✓ Capacidad de detección si la Web está adaptada a dispositivos móviles.
- ✓ Capacidad de rastreo y análisis de varios sitios Web como HTML5, SOAP y AJAX.
- ✓ Cuenta con una solución Online enfocada a usuarios que necesitan escanear aplicaciones Web en Internet sin tener la aplicación local instalada (Acunetix OVS).

[14] Herramienta para auditar la seguridad de páginas Web. Se considera líder mundial en seguridad de aplicaciones Web. Su página oficial es https://www.acunetix.com/

La interfaz gráfica de la herramienta Acunetix WVS está compuesta por los siguientes paneles:

1. Panel Izquierdo: Contiene el explorador de herramientas tecnológicas.
2. Panel central: Lista las vulnerabilidades detectadas
3. Panel derecho: Muestra la descripción de la vulnerabilidad seleccionada, el detalle de ataque utilizado para explotar la vulnerabilidad.

Figura 11. Interfaz Gráfica de Usuario de Acunetix WVS.



Fuente. Elaboración propia a partir de: http://www.north-networks.com/acunetix/

El funcionamiento de la herramienta Acunetix WVS es simililar a la anterior herramienta y como ésta, también proporciona la estrategia a utilizar para el escaneo de vulnerabilidades en el aplicativo Web mediante el siguiente procedimiento:

1. Análisis y revisión de la estructura completa del sitio Web para descubrir todos los directorios y archivos con el rastreador Crawling
2. Escaneo de vulnerabilidades en donde se lanza una serie de ataques de vulnerabilidades en cada página, de forma similar a como un atacante podrían hacer para penetrar la seguridad de un sitio Web.

3. Informe detallado de vulnerabilidades encontradas que se mostrarán en la zona de alertas, en donde se describe la vulnerabilidad, ejemplos posibles para su solución, CVE[15] (Common Vulnerabilities and Exposures), CWE[16] (Common Weakness Enumeration), e información CVSS[17] (Common Vulnerability Scoring System).

4. Exportación de las vulnerabilidades detectadas en una amplia variedad de informes en diferentes formatos. La comprobación o el escaneo de alertas específicas permite probar las vulnerabilidades de forma individual evitando de nuevo la ejecución de una exploración o escaneo completo al sitio.

Descripción del Aplicativo Web. El objetivo es describir la aplicación Web objetivo blanco de ataques para tener un mayor conocimiento en cuanto a la tecnología utilizada para su desarrollo, los módulos de gestión de información y los tipos de usuarios y la interfaz gráfica de usuario.

- **Tecnología Utilizada para el Desarrollo del Aplicativo Web**

Por la información que se gestiona, el Sistema de Información Web de Gestión y Seguimiento de Novedades de Aprendices del Programa de Articulación con la Educación Media del Servicio Nacional del Aprendizaje - SENA, se puede convertir potencialmente en un activo de información valioso para la institución, por lo tanto es el aplicativo Web objetivo al cual se le ejecutarán las herramientas para el escaneo automático de vulnerabilidades. En su desarrollo no se utilizó ninguna metodología de desarrollo seguro de software, sino que se utilizó la metodología de construcción por prototipos, debido a que se requiere contar con un producto terminado lo más pronto posible debido a la necesidad de procesamiento de información que actualmente existe en el Programa de Articulación del SENA. Esta metodología de desarrollo ha permitido obtener versiones de prueba o prototipos implementadas de acuerdo al diseño de los módulos principales del sistema.

Cabe anotar que no se utilizó una plataforma de desarrollo (o framework), lo que incrementa el riesgo de seguridad en el aplicativo, pues estas plataformas habitualmente incorporan medidas de seguridad aún no implementadas. Las diferentes tecnologías y versiones utilizadas para la creación del aplicativo se describen en la siguiente tabla:

---

[15] Listado de vulnerabilidades de seguridad de la información conocidas públicamente. Es un estándar que permite identificar vulnerabilidades mediante la asignación de un código único de identificación.

[16] Listado de diferentes tipos de debilidades de software enfocada a desarrolladores y profesionales de la seguridad. Unifica la descripción de las debilidades de seguridad de un software basado en su arquitectura, diseño y código.

[17] Sistema de score para medir el impacto en caso de explotación de una vulnerabilidad.

Tabla 5. Ficha Técnica de las Tecnologías de Desarrollo utilizadas en el Aplicativo Web.

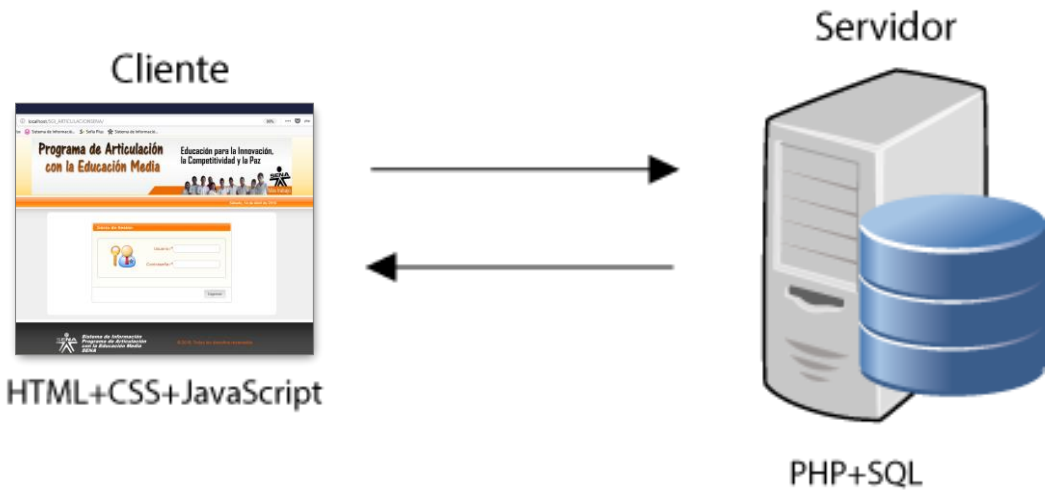
Tecnología de Desarrollo		Descripción
Tecnologías de Desarrollo del Lado del Servidor	**WampServer 2.2**	Entorno de desarrollo Web que usa Windows (sistema operativo), Apache (servidor Web), MySQL (gestor de bases de datos) y PHP (lenguaje de programación).
	Servidor Web Apache 2.2.21	Servidor Web de código abierto, para múltiples plataformas integrado en la herramienta WampServer.
	Gestor de Base de Datos MYSQL 5.5.16	Sistema gestor de base de datos relacional integrado en la herramienta WampServer.
	PHP 5.3.8	Lenguaje de programación del lado del servidor integrado en la herramienta WampServer.
Técnologías de Desarrollo del Lado del Cliente	**HTML 5**	Lenguaje empleado para la creación de las páginas Web.
	CSS3	Lenguaje de diseño gráfico utilizado para definir el estilo y presentación de la página Web.
	Javascript 5 y JQuery 1.7.1	Lenguaje de programación utilizado para agregarle interactividad y efectos visuales a la página Web, incorporando además tecnología AJAX.

Fuente. Elaboración propia.

El aplicativo Web actualmente se encuentra instalado en un servidor que cuenta con un sistema de seguridad de cuatro niveles que incluye: Firewall[18], Mod_Security[19], Antivirus[20] y CXS[21]. Es posible entonces que este mecanismo de defensa interfiera con el normal funcionamiento de las herramientas para escanear vulnerabilidades Web.

La siguiente figura muestra el esquema de ejecución del sistema de información con las tecnologías de desarrollo utilizadas para entender su modo de funcionamiento bajo la arquitectura cliente servidor y por donde se pueden ejecutar los ataques al aplicativo.

Figura 12. Esquema de Ejecución de la Aplicación Web bajo la Arquitectura Cliente Servidor.



Fuente. Elaboración propia a partir de: http://multimedia.uoc.edu/blogs/fem/es/las-aplicaciones-Web-y-las-bases-de-datos/

En general, el esquema de ejecución de una aplicación Web bajo la arquitectura cliente servidor se realiza de la siguiente manera:

- ✓ En el servidor se almacenan los archivos .php con el código necesario en PHP que contiene las instrucciones SQL para acceder a la base de datos, el código HTML y CSS

[18] Dispositivo de seguridad de red cuya función es monitorizar el tráfico entrante y saliente. Decide si debe permitir o bloquear un tráfico específico basado en un conjunto de restricciones de seguridad predefinidas.
[19] Firewall de aplicaciones Web bajo licencia GNU. Se ejecuta como módulo del servidor Web Apache, proporciona protección contra diferentes ataques hacia las aplicaciones Web, permite monitorizar tráfico HTTP, realizar análisis en tiempo real, filtrar ataques por XSS, SQL Injection, así como comportamientos anormales en protocolos, robots o troyanos.
[20] Son programas encargados de detectar o eliminar virus informáticos.
[21] Utilidad para detección de troyanos y archivos maliciosos.

para visualizar correctamente la página Web y el código en JavaScript para interactuar con el usuario.

- ✓ En el servidor se ejecutan las instrucciones PHP y las consultas SQL, se obtienen los datos y como respuesta se envía al cliente un archivo con el código en HTML, CSS, y JavaScript además de datos consultados.
- ✓ En el cliente se ejecuta el navegador que es un programa utilizado para presentar los datos al usuario enviados como respuesta una solicitud hecha al servidor.
- ✓ Mediante el navegador, el cliente envía al servidor los datos que son proporcionados el usuario y es precisamente el punto de entrada en el sistema para ejecutar los ataques más comunes a la página Web y por consiguiente a la información almacenada en el servidor.
- ✓ Finalmente el servidor recibe esa información y las procesa en la base de datos mediante consultas SQL.

- **Descripción de los módulos de Gestión de Información y los tipos de usuarios del Aplicativo Web**

El sistema de información ha sido desarrollado para uno de los tipos de programas de formación que ofrece el Servicio Nacional de Aprendizaje - SENA dentro de la amplia cobertura en formación profesional integral que ofrece ésta institución. Éste tipo de formación corresponde al Programa de Articulación con la Media, que consiste en integrar la ejecución de programas técnicos ofrecidos por el SENA, en los grados décimo y once de las instituciones educativas que ofrecen educación media.

El Sistema de Información Web de Gestión y Seguimiento de Novedades de Aprendices del Programa de Articulación con la Educación Media del Servicio Nacional del Aprendizaje - SENA, es el activo crítico de información que debe ser protegido. Es un aplicativo Web que consiste en apoyar la gestión administrativa de acuerdo con los lineamientos que exige el programa de articulación, relacionados con la gestión de información de los instructores que imparten formación, de las fichas o grupos a cargo y de la instituciones educativas que se encuentran articuladas, con el fin de proveer una información mucho más completa e inmediata, tanto para el Coordinador Académico y Líder del Programa de Articulación con la Media del SENA, como para los instructores e instituciones educativas, quienes como integrantes de la comunidad educativa SENA, se convierten en actores claves para éste proceso.

En la fase de diseñe, se definieron los módulos que conforman la arquitectura del sistema como prototipo inicial, los cuales representan los principales procesos de gestión de información en el programa de Articulación. Estos módulos son descritos en la siguiente tabla.

Tabla 6. Descripción de Módulos de Gestión de Información del Aplicativo Web

Módulo del Sistema	Descripción
Gestión de Sesiones	Este módulo es utilizado para la autenticación y seguimiento de actividades de los usuarios que acceden al sistema, mostrando información privada asociada con su perfil de usuario. Además se encarga de registrar información de inicio y cierre de sesión, facilitando las tareas de control y supervisión de accesos. También cuenta con la opción de restablecer la contraseña del usuario, ya que sea en caso de olvido o que requiera cambiarla.
Gestión de Usuarios	Éste módulo se encarga de gestionar la información de las cuentas de usuario o credenciales de acceso al sistema. La gestión de información del módulo permite listar usuarios de acuerdo a criterios de consulta, agregar cuentas de usuarios y asignar privilegios de acceso, activar o desactivar cuentas de usuarios y ver el detalle de accesos al sistema como actividad de seguimiento.
Gestión de Instructores	Éste módulo gestiona la información de los instructores que pertenecen al programa. Permite obtener el listado de instructores consultados de acuerdo a diferentes parámetros de búsqueda, importar sus datos desde el aplicativo Sofía Plus, editar su información diferente a los datos importados, ver el detalle de su información, así como las fichas asociadas y gestionar el horario de sus formaciones.
Gestión de Instituciones Educativas	Éste módulo gestiona la información de los las instituciones educativas que actualmente están articuladas con el programa. Permite obtener el listado de todas las instituciones educativas consultadas de acuerdo a criterios de búsqueda, importar sus datos desde el aplicativo Sofía Plus, editar información diferente de los datos importados, ver el detalle de su información, así como los instructores y las fichas asignadas y ver el horario general de formación.
Gestión de Fichas de Formación	Éste módulo es considerado uno de los más importantes que gestiona el aplicativo Web, pues relaciona la información de los demás módulos a las fichas que corresponden los grupos de formación que son asociadas a las instituciones educativas y a los instructores para impartir formación. A nivel general el módulo permite listar las fichas de formación de acuerdo a criterios de búsqueda, importar sus datos principales desde el aplicativo Sofía Plus y ver información detallada de las instituciones educativas e instructores asociadas y ver el horario de formación. Además a nivel específico el módulo permite la gestión de información de los aprendices asociados a la ficha y la gestión de novedades disciplinarias para la generación de informes de seguimiento y evaluación.

Fuente. Elaboración propia.

A continuación se hace una descripción de cada tipo de usuario y las principales funciones que realiza en el aplicativo Web.

Tabla 7. Descripción de Tipos de Usuarios y Funciones Relevantes en el Aplicativo Web.

Tipo de Usuario	Funciones en el Aplicativo Web	Módulos de Interacción
Instructores 	▪ Actualizar credenciales de acceso ▪ Actualizar sus datos de contacto y de contratación ▪ Actualizar datos generales de la institución educativa ▪ Gestionar los datos de los aprendices que pertenecen a las fichas de formación asociadas ▪ Gestionar y reportar novedades disciplinarias de los aprendices asociados a una ficha ante las instituciones de educación media ▪ Gestionar el horario de formación	El usuario *Instructor* interactúa con los siguientes módulos: ▪ Gestión de Sesiones ▪ Gestión de Instructores ▪ Gestión de Instituciones Gestión de Fichas de Formación
Instituciones de Educación Media 	▪ Actualizar credenciales de acceso ▪ Consultar fichas articuladas con la institución educativa ▪ Consultar datos generales de los aprendices asociados a una ficha de formación académica ▪ Consultar novedades disciplinarias de los aprendices asociados a las fichas articuladas con la institución educativa ▪ Ver horario de formación de todas las fichas	El usuario Institución interactúa con los siguientes módulos: ▪ Gestión de Sesiones ▪ Gestión de Instituciones Educativas ▪ Gestión de Fichas de Formación
Coordinador Académico y Líder del Programa de Articulación con la Media del SENA SENA SERVICIO NACIONAL DE APRENDIZAJE	▪ Actualizar credenciales de acceso ▪ Registro de datos masivos a través del importe de archivos generados por el aplicativo Sofía Plus ▪ Consultar la información de los instructores que pertenecen al programa de Articulación con la Media ▪ Gestionar la información de las instituciones de educación media articuladas con el programa ▪ Gestionar las fichas de formación pertenecientes a las instituciones de educación media y asignación de instructores	El usuario Instructor interactúa con los siguientes módulos: ▪ Gestión de Sesiones ▪ Gestión de Instructores ▪ Gestión de Instituciones Educativas ▪ Gestión de Fichas de Formación

	▪ Consultar los datos de los aprendices que pertenecen a las fichas de formación asociadas ▪ Consultar el reporte de novedades disciplinarias de los aprendices asociados a las fichas articuladas con el programa	
Administrador Web 	▪ Actualizar credenciales de acceso ▪ Gestionar los usuarios del sistema, activación de cuentas, permisos y contraseñas genéricas. ▪ Configuración de las copias de seguridad (del sitio y de la base de datos) automatizadas ▪ Robustecer las opciones de seguridad de la base de datos y del servidor Web	El usuario Instructor interactúa con los siguientes módulos: ▪ Gestión de Sesiones ▪ Gestión de Usuarios

Fuente. Elaboración propia.

- **Descripción de la Interfaz Gráfica de Usuario (GUI)**

La interfaz gráfica de usuario, muestra la organización y distribución de las diferentes zonas que conforman la estructura de contenidos del sitio Web del aplicativo Web. A continuación se hace una descripción de los elementos de la GUI y su ubicación en la página Web.

Tabla 8. Descripción de los Elementos de la Interfaz Gráfica de Usuario en el Aplicativo Web.

Elementos de la Interfaz Gráfica	Descripción
1. Encabezado	Esta zona está conformada por un banner como elemento gráfico que contiene el nombre del programa de formación del SENA, el logo de la institución y los datos del usuario que inicia sesión. Por ser una de las zonas que se visualiza con mayor frecuencia y que, por la forma tradicional de construcción del código HTML, aparece al inicio de la página Web (en la parte superior).
2. Barra de Menús	Esta zona contiene la barra de menús y submenús contextuales y muestra las distintas opciones que el usuario puede elegir para acceder a un determinado proceso de gestión de información. Esta zona representa el elemento principal de navegación y contiene los módulos principales del aplicativo Web, los cuales son visualizados dependiendo del perfil de acceso del usuario.
3. Área de Contenidos e Interacción	Esta zona muestra el área de contenidos en donde el usuario solo puede visualizar la información, como resultado de consultas. El área de interacción es la zona que permite la realización de acciones por

	partes delos usuarios de la aplicación Web, en donde se muestran los botones y se gestiona la información. Se utilizan ventanas superpuestas que aparecen encima de la página principal que ofrece la gestión de información para que el usuario tenga una mejor experiencia de uso del aplicativo.
4. Pie de Página	Esta zona muestra el pie de página que cumple la función mostrar información adicional, además de mostrar la barra de menús para ofrecer navegación del sitio desde esa ubicación. Por la forma tradicional de construcción del código HTML, aparece al final de la página Web (en la parte inferior).

Fuente. Elaboración propia.

Figura 13. Estructura de la Interfaz Gráfica de Usuario del Aplicativo Web.

Fuente. Elaboración propia.

Una vez identificados los componentes básicos del modelo de seguridad necesarios en la ejecución de pruebas de seguridad en el aplicativo Web, se definen los pasos a seguir como estrategia metodológica en la obtención de resultados que permitan analizar el estado de seguridad y determinar si el aplicativo cumple con las características de un software seguro.

Procedimiento Metodológico de Ejecución de Pruebas de Seguridad. Tomando como referencia la descripción que hacen los autores [12] acerca de los pasos utilizados para medir la efectividad de las herramientas utilizadas como escáneres para detectar vulnerabilidades Web, se definen las siguientes fases para la implementación de las pruebas de seguridad:

Fase I: Identificación de la Aplicación Web Objetivo

Como se mencionó anteriormente, la aplicación Web objetivo ya ha sido descrita en cuanto a su tecnología de desarrollo, funcionalidad e interacción con el usuario. Lo que hay que resaltar es que probablemente en su implementación incluya vulnerabilidades del informe OWASP Top 10 por los fallos de seguridad detectados en las pruebas funcionales (utilización del aplicativo en un entorno real de ejecución) y porque además no se tuvieron en cuenta buenas prácticas de codificación segura en los módulos que ya han sido implementados.

Fase II: Selección de Escáneres de Vulnerabilidades Web

[12] menciona en su artículo que los escáneres de vulnerabilidades de aplicaciones Web, son herramientas WAVS que imitan los ataques de aplicaciones Web. Estas herramientas no pueden garantizar que su uso eliminará por completo los defectos, pero pueden hacer que la aplicación sea más segura.

NIST[22] en su publicación especial sobre "Software Assurance Tools: Web Application Security Scanner Functional Specification Version 1.0", en 2008, definió una lista de requisitos que todas las WAVS deben proporcionar:

[22] NIST (National Institute of Standards and Technology). Instituto Nacional de Normas y Tecnología. Agencia del Departamento de Comercio de los Estados Unidos que se encarga de la Administración de Tecnología. Página oficial: https://www.nist.gov/

- ✓ Identificar todos los tipos de vulnerabilidades enumeradas.
- ✓ Reportar un ataque que demuestre la vulnerabilidad.
- ✓ Especificar para un ataque detectado la ubicación del script, las entradas y el contexto.
- ✓ Identificar la vulnerabilidad con un nombre semánticamente equivalente.
- ✓ Ser capaz de autenticarse en la aplicación y mantener el estado de inicio de sesión.
- ✓ Tener una tasa de falsos positivos aceptablemente baja.

Como se mencionó anteriormente, las herramientas OWASP ZAP y ACUNETIX WVS fueron seleccionadas como escáneres de vulnerabilidades Web, pues además de cumplir los requisitos que deben tener las aplicaciones WAVS, son herramientas muy populares en el ámbito de las pruebas de seguridad en aplicaciones Web. La utilización de varias herramientas permitirá obtener mayor confiabilidad en los resultados obtenidos y medir su nivel de efectividad, pues la disponibilidad del código fuente y el control sobre los resultados del servidor proporcionan una mejor evaluación sobre estas herramientas.

Fase III: Ejecución de Escáneres de Vulnerabilidades Web

Es la utilización de los escáneres de vulnerabilidades Web seleccionados para identificar posibles vulnerabilidades. La ejecución de estas herramientas viene dada por su modo de funcionamiento. En general inician con una exploración completa de la estructura de directorios y páginas, identificando las URL del sitio, con el fin de obtener la mayor cantidad de información posible necesaria para ejecutar el escaneo de vulnerabilidades Web, mediante la realización de diferentes tipos de ataques.

Fase IV: Reporte de Vulnerabilidades Web

Con la identificación de las vulnerabilidades detectadas en función de su criticidad, se tienen elementos de juicios para realizar un análisis de los resultados obtenidos y evaluar el estado de seguridad del aplicativo y la implementación de mecanismos de defensa para mejorar la seguridad del mismo. Es importar realizar unas pruebas manuales para confirmar que las vulnerabilidades identificadas por los escáneres existen (explotación de las vulnerabilidad).

Figura 14. Procedimiento Metodológico para Ejecución de Pruebas de Seguridad.

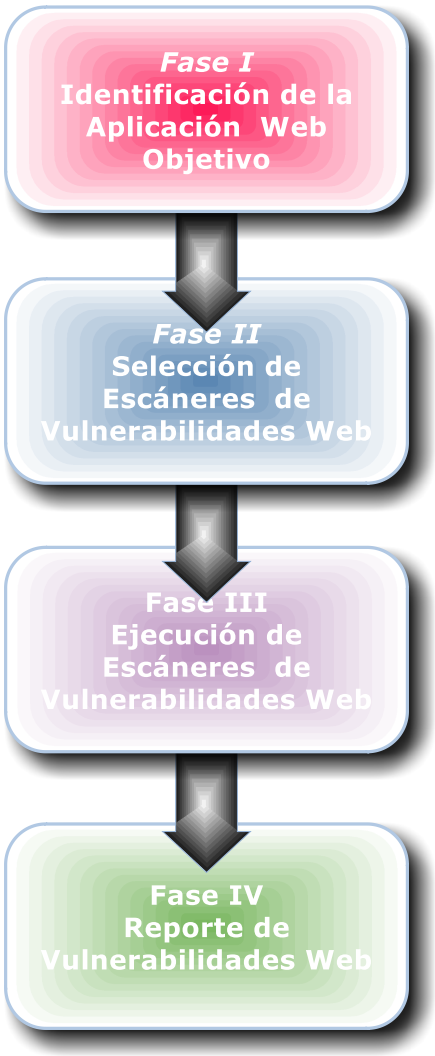


Figura 9. Elaboración propia.

Enfoque Metodológico Basado en Modelado de Amenazas

Este enfoque metodólogo se concentra en aquellas partes del sistema que son vulnerables y susceptibles de sufrir ataques. Este modelo se basa en determinar la estructura del sistema en cuanto a sus componentes, flujos de datos, activos de información y riesgos de seguridad, así como también la forma en que pueden ocurrir los ataques y la implementación de controles de seguridad que ayuden a mitigar los riesgos

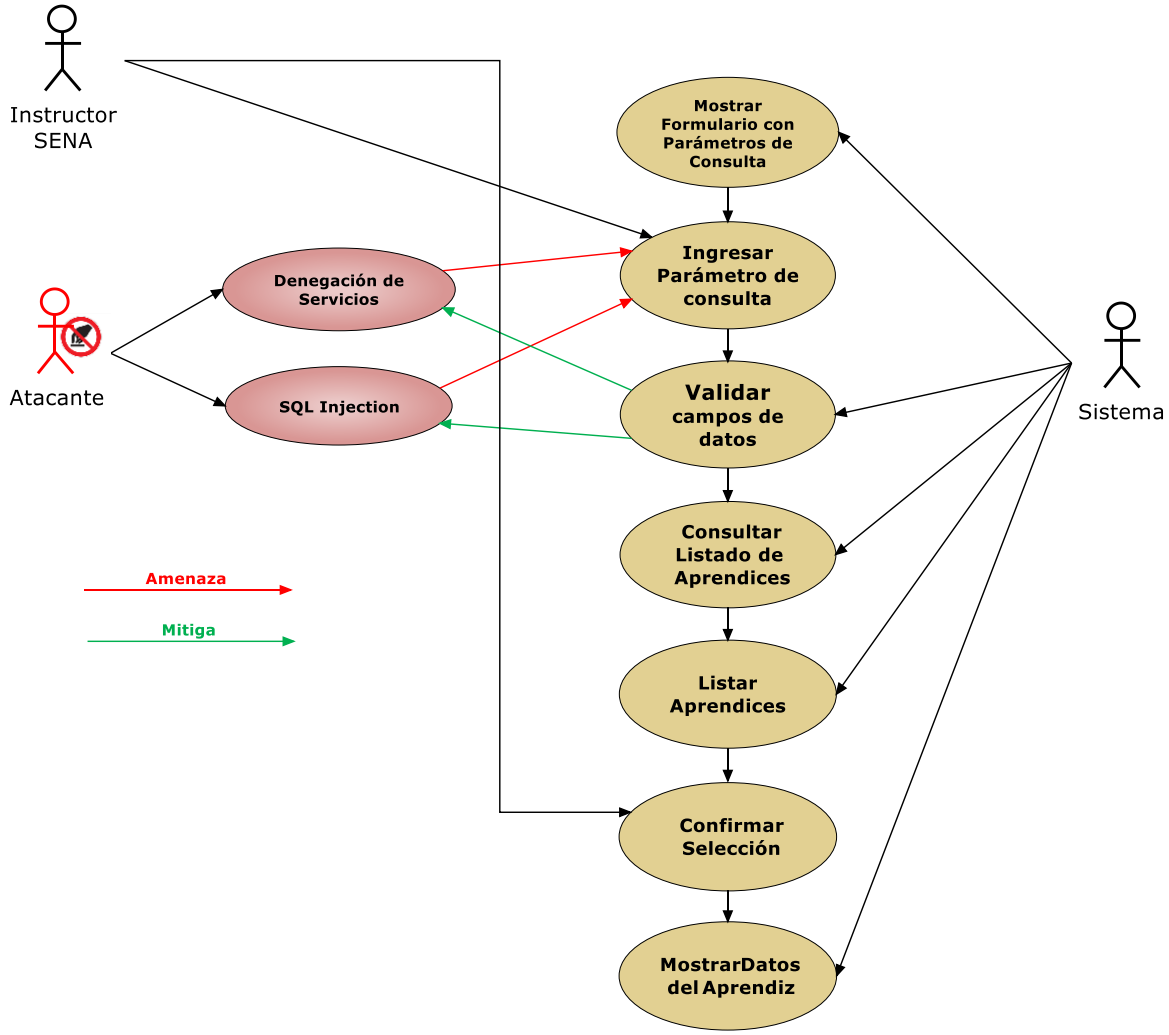
identificados o neutralizar las amenazas. A continuación se nombraran algunas técnicas utilizadas que utiliza este modelo para comprender el perfil de amenazas a las que está expuesto la aplicación Web.

Este modelo plantea la descomposición de la aplicación en varios escenarios de procesos en donde se pueden explotar vulnerabilidades potenciales. Para la aplicación Web se escogió el escenario de *Consultar Aprendices*, por contener información crítica que de ser alterada pone en riesgo la evaluación y el seguimiento disciplinario del aprendiz y por ende la integridad de la información presenta a la institución educativa.

Escenario de Consultar Aprendices
El proceso de consultar aprendices, permite a los usuarios *Coordinador Académico SENA* y *Líder SEN*A consultar los datos de identificación, de ubicación y de contacto de todos los aprendices que se encuentran actualmente en el programa de articulación independiente del estado de formación. Igualmente le permite al usuario *Instructor SENA*, consultar únicamente los datos de los aprendices de los grupos de formación de los cuales es responsable actualmente. Así mismo le permite al usuario *Institución,* consultar los datos de todos los aprendices que han pertenecido a la institución educativa sin importar su estado de formación. Este proceso se encarga de hacer una consulta del listado de aprendices teniendo en cuenta el tipo de usuario y estado para visualizar listados de aprendices de acuerdo al parámetro de consulta especificado en el formulario y posteriormente visualizar los datos del aprendiz en particular. Así mismo durante el proceso de consulta, se pueden registrar y notificar novedades disciplinarias presentadas durante la formación.

Una técnica utilizada en el modelamiento de amenazas es el diseño de casos de abuso, que para el escenario de *Consultar Aprendices*, describe el mal comportamiento o el uso indebido que puede hacer un atacante al consultar la información de los aprendices; con esto se logra descubrir posibles fallos o vulnerabilidades potenciales y tomar medidas para mitigar el impacto causado por la vulnerabilidad explotada.

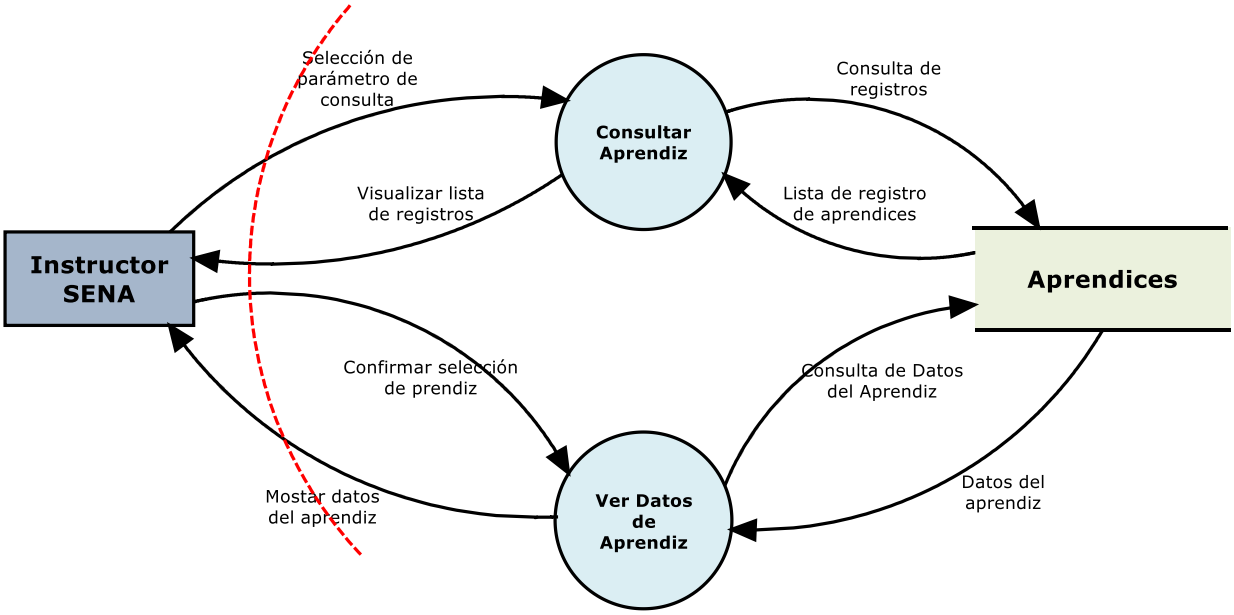
Figura 15. Caso de Abuso para el Escenario: Consultar Aprendices.



Fuente. Elaboración propia.

El modelo plantea que se debe tener una descripción de la arquitectura del sistema, para esto se hace se hace uso DFD (diagrama de flujo de datos) que incluye la lógica de los procesos, el almacenamiento de datos, el flujos de datos entre los procesos y el límite de confianza establecido, permitiendo comprender así el alcance de seguridad, que en éste caso ha sido diseñado para el escenario de *Consultar Aprendices*.

Figura 16. Diagrama de Flujo de Datos para el Escenario: Consultar Aprendices.



Fuente. Elaboración propia.

Como técnica para documentar las amenazas, se utiliza una tabla de modelamiento que describe el conjunto básico de atributos de las amenazas detectadas para el escenario de Consultar Aprendices. En esta tabla se especifica “Quién” puede ejecutar la amenaza (perfil del atacante), por “Dónde” el atacante puede ejecutar la amenaza (la puerta abierta en el sistema para ejecutar el ataque), “Qué” se pretende lograr con la amenaza (objetivo del ataque), “Cómo” el atacante puede ejecutar la amenaza (técnicas empleadas), el “Impacto” o consecuencias negativas que puede causar la amenaza y la “mitigación” para reducir la amenaza detectada.

Tabla 9. Tabla de Modelado de Amenazas para el Escenario: Consultar Aprendices.

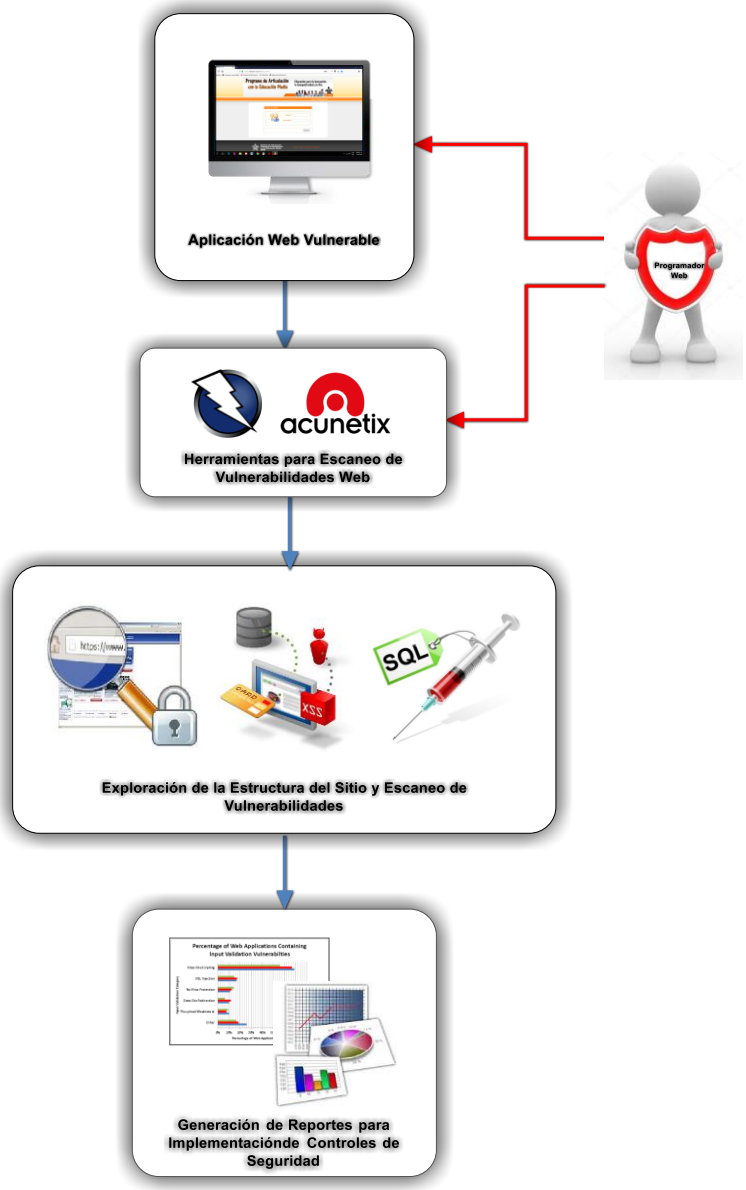
QUIÉN	DÓNDE	QUÉ	CÓMO	IMPACTO	MITIGACIÓN
La amenaza puede ser realiza por dos tipos de usuario: ✓ Un usuario Interno que puede ser un Instructor SENA que quiera consultar los datos de los demás aprendices que NO pertenecen a las fichas o grupos de formación que están a su cargo ✓ Un usuario externo que haya podido ingresar al sistema y que desee consultar los datos de todos los aprendices registrados en el sistema	La amenaza ataca directamente el módulo de Fichas que gestiona los procesos para cada grupo de formación: ✓ Actualizar los datos de la ficha ✓ Visualizar los datos de la ficha ✓ Consultar información de los aprendices de acuerdo a la ficha seleccionada ✓ Gestionar las actas de seguimiento y evaluación disciplinario de cada ficha	La amenaza tiene como objetivos: ✓ Obtener información sensible de los aprendices y alterar el contenido de la base de datos. ✓ Evitar que la aplicación Web pueda realizar su servicio con normalidad e incluso puede terminar resultando inaccesible.	Los mecanismos utilizados por el atacante para ejecutar su amenaza son: ✓ SQL Injection: A partir de información suministrada por los usuarios, se inserta código malicioso a la aplicación mediante sentencias SQL. ✓ Denegación de servicio: Consiste en el consumo exagerado de recursos de la aplicación (memoria, CPU y conexión a base de datos.	Las consecuencias que estas amenazas pueden causar son: ✓ La no integridad de la información debido a la alteración o modificación de su contenido en la base de datos. ✓ La no disponibilidad de la información debido a que el sistema no puede realizar el servicio de consulta de datos. ✓ Pérdida del buen nombre y reputación del SENA por parte de los usuarios externos.	La forma de reducir éstas amenazas son: ✓ No generar consultas SQL, por la concatenación de cadenas de caracteres ✓ Utilizar controles específicos como la parametrización de sentencias SQL. ✓ Dificultar el abuso de funcionalidad de la aplicación: paginar resultados, limitación de resultados, protección frentes a caracteres comodín.

Fuente. Elaboración propia.

Haciendo un análisis de los enfoques metodológicos presentados para la detección de vulnerabilidades y teniendo en cuenta el estado actual de desarrollo del aplicativo Web, se concluye que el enfoque metodológico orientado hacia el modelado de

amenazas, muestra el estado de vulnerabilidad del sistema desde la óptica de un atacante y debe ser aplicado desde las primeras fases de desarrollo de software. Esto sugiere que la alternativa más práctica y rápida a utilizar para la detección de vulnerabilidades en el aplicativo Web, es el enfoque metodológico orientado a las pruebas de seguridad porque comprueba el código expuesto efectivamente y puede ser aplicado en las etapas finales de desarrollo del software. La siguiente figura muestra el esquema general de los componentes fundamentales del modelo de pruebas de seguridad con el procedimiento aplicado para para la obtención de resultados.

Figura 17. Procedimiento Metodológico para la Detección de Vulnerabilidades en el Aplicativo Web.



Fuente. Elaboración propia.

DESARROLLO E IMPLEMENTACIÓN

En esta sección se presentará el desarrollo del enfoque metodológico orientado hacia la ejecución de pruebas de seguridad en el aplicativo Web, teniendo en cuenta varios escenarios de aplicación de las mismas. El objetivo es obtener información relevante que permita evidenciar nivel de riesgo o estado de seguridad del aplicativo en la fase actual de desarrollo. El procedimiento para la implementación de las pruebas de seguridad es el siguiente:

Fase I: Identificación de la Aplicación Web Objetivo

Es importante aclarar que las funcionalidades desarrolladas en el aplicativo Web aún no se encuentran implantadas de manera definitiva en un entorno real de trabajo, por lo que las pruebas de seguridad ejecutadas, no estaría comprometiendo la seguridad de la información del Programa de Articulación con la Educación Media del SENA en caso de explotar directamente las vulnerabilidades encontradas.

En este paso se explicarán los diferentes escenarios en los que el aplicativo Web estará disponible para la implementación de las pruebas de seguridad, debido a que no fue posible ejecutar las herramientas seleccionadas para el escaneo de vulnerabilidades web en las mismas condiciones. Esto de alguna manera afecta el análisis de los resultados obtenidos, pues no se pueden evaluar bajo los mismos parámetros de ejecución, pero si son considerados relevantes durante el proceso por el tipo de información que aportan.

En el escenario de ejecución de las pruebas se seguridad, se especifica el modo de acceso o lugar de instalación del aplicativo, la URL o dirección Web de acceso y las características de seguridad del servidor Web en donde se encuentra instalado el aplicativo y la base de datos.

Escenario 1 de Ejecución de Pruebas de Seguridad	
Modo de Acceso	Internet
URL	http://www.sipam.com.co
Servidor Web	El aplicativo Web se encuentra instalado en un servidor remoto bajo Linux de última generación, cuenta con un sistema de seguridad de cuatro niveles que incluye Firewall, Mod_Security, Antivirus y CXS. Posee certificación ISO/IEC 27001 (Tecnología de la información, Sistemas de Gestión de Seguridad, Técnicas de seguridad).

Escenario 2 de Ejecución de Pruebas de Seguridad	
Modo de Acceso	Local
URL	http://localhost/www.sipam.com.co
Servidor Web	El aplicativo Web se encuentra instalado en un servidor local de pruebas bajo Windows 10. Su configuración predetermina no cumplen con los requisitos de seguridad para un entorno de producción.

Fase II: Selección de Escáneres de Vulnerabilidades Web

En este paso se presenta la ficha técnica de las herramientas seleccionadas para el escaneo de vulnerabilidades Web, la cual resume las características del software en cuanto a su descarga, instalación y uso.

Tabla 10. Ficha Técnica Herramienta OWASP ZAP.

Ficha Técnica Escáner de Vulnerabilidades Web	
Nombre	OWASP Zed Attack Proxy Standard
Licencia	Open Source
Sistema Operativo	Windows 32
Idioma	Español
Versión	2.7.0
Tamaño	75 MB
Desarrollador	OWASP Foundation
URL de Descarga	https://github.com/zaproxy/zaproxy/wiki/Downloads

Fuente. Elaboración propia.

Tabla 11. Ficha Técnica Herramienta ACUNETIX WVS.

Ficha Técnica Escáner de Vulnerabilidades Web	acunetix
Nombre	Acunetix
Licencia	Versión de Prueba - Trial
Sistema Operativo	Windows 32
Idioma	Inglés
Versión	10.5
Tamaño	42 MB
Desarrollador	Acunetix
URL de Descarga	http://acunetix-Web-vulnerability-scanner.software.informer.com/10.5/

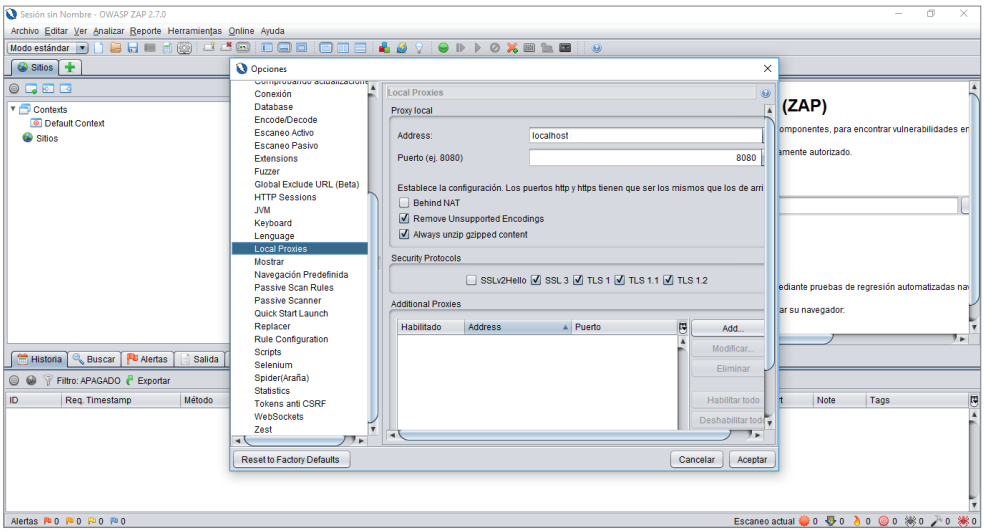
Fuente. Elaboración propia.

Fase III: Ejecución de Escáneres de Vulnerabilidades Web

En este paso se describe el proceso de ejecución de cada una de las herramientas seleccionadas para la detección de vulnerabilidades Web, teniendo en cuenta los escenarios de ejecución de pruebas, la versión de la herramienta y el modo de funcionamiento.

Ejecución de la Herramienta OWASP ZAP. Esta herramienta fue ejecutada en el escenario No. 1. Inicialmente se hizo un escaneo activo de vulnerabilidades, ingresando la URL de acceso al sitio, pero la herramienta solo hizo un reconocimiento de la estructura externa del sitio Web, es decir, no realizó una exploración de los directorios, scripts o URL que se cargan cuando el usuario inicia sesión, por lo que los ataques realizados sobre esta estructura, no generó un reporte de vulnerabilidades relevante para su análisis. Debido a esta situación, se optó por realizar un escaneo pasivo que permitió navegar por el sitio Web como un usuario normal y posteriormente, realizar ataques sobre las páginas visitadas. Para ello OWASP ZAP, se configuró para ser utilizado como un proxy[23] de interceptación en el equipo local. Finalmente, se debe configuró el navegador con el proxy localhost[24] y el puerto que se haya elegido, normalmente el 8080. La siguiente figura muestra la configuración de los parámetros del proxy en la herramienta OWASP ZAP.

Figura 18. Configuración del Proxy Local en la Herramienta OWASP ZAP.



Fuente. Elaboración propia.

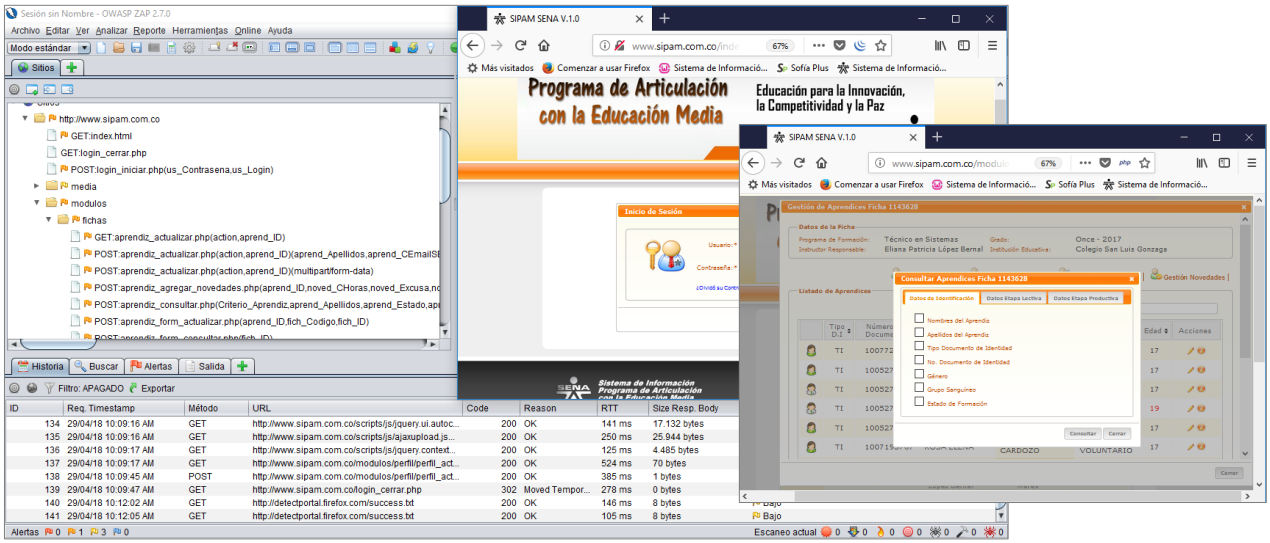
---

[23] Un proxy es un programa o dispositivo que actúa entre un computador conectado a Internet y el servidor al que está accediendo, es decir, que no se accede al servidor de manera directa sino a través del proxy quien se encarga realmente de conectarse con el servidor y devolver el resultado de la solicitud. https://desarrolloweb.com/faq/que-es-proxy.html.

[24] El localhost es el computador o dispositivo local que se está usando al cual se le asigna la dirección IP 127.0.0.1.

Una vez teniendo todo configurado, si se empieza a navegar por el sitio Web con el navegador, se observa que en la pestaña de sitios, comienza a verse el reconocimiento de la estructura completa del sitio como si se hubiese hecho un análisis activo. La ventaja es obviamente que se genera mucho menos ruido que con el ataque activo. De esta forma también se puede tener un mapeo del sitio Web. En la siguiente figura se observa la estructura del sitio Web desplegada a medida que se va interactuando con la aplicación.

Figura 19. Exploración de la Estructura del Sitio Web_Herramienta OWASP ZAP.

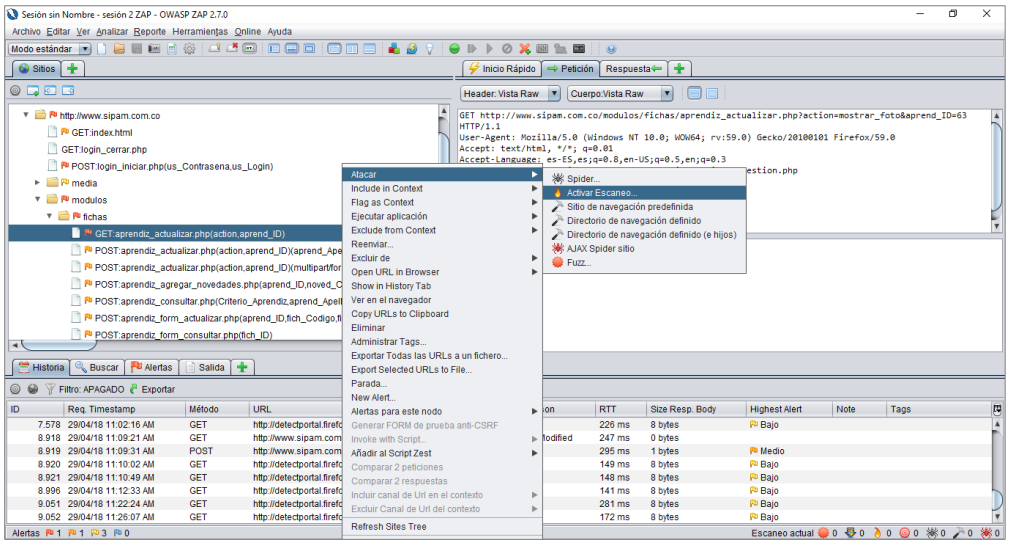


Fuente. Elaboración propia.

Finalizando el recorrido por todas las URLs del sitio a través del navegador, se empieza a seleccionar cada una de las peticiones GET o POST que se han hecho en un formulario. Pulsando con el botón derecho dentro del mismo, se abre un menú contextual donde se debe seleccionar la opción de *Atacar/Activar Escaneo* y a partir de este momento OWASP ZAP comenzará a buscar vulnerabilidades indiscriminadamente. También por esta misma opción se puede configurar por ejemplo un ataque de fuerza bruta[25] y búsqueda de XSS, seleccionado para ello, uno de los diccionarios que por defecto trae la herramienta. La siguiente figura muestra el detalle del procedimiento de activación del escaneo de forma manual para la URL seleccionada.

[25] Un ataque de fuerza bruta consiste en obtener una clave probando todas las posibles combinaciones hasta encontrar aquella que posibilita el acceso. Este tipo de ataques suele utilizar en conjunto con los ataques de diccionario, en el que se encuentran diferentes palabras para ir probando con ellas.

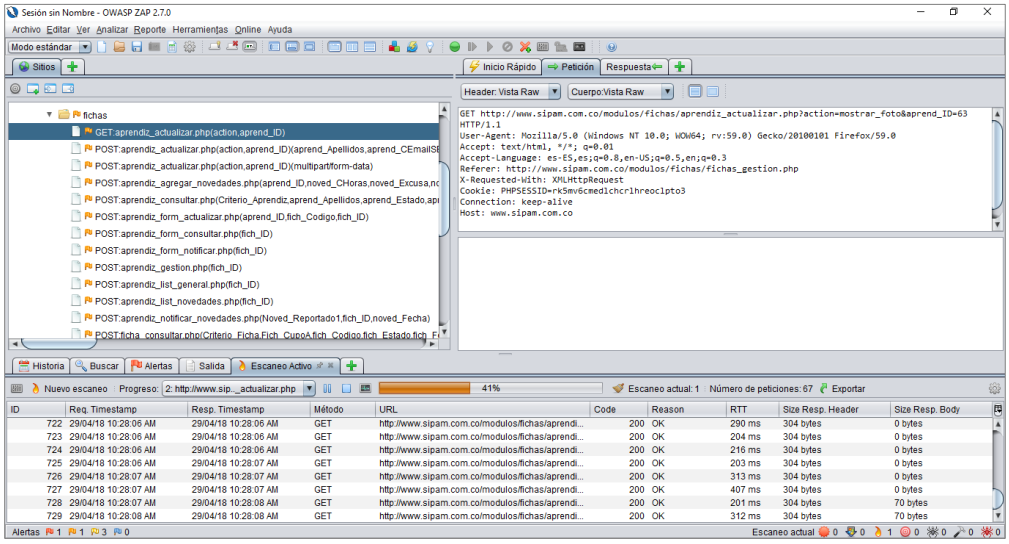
Figura 20. Activación Manual del Escaneo de Vulnerabilidades_Herramienta OWASP ZAP.



Fuente. Elaboración propia.

El *Escaneo Activo* se encarga de analizar la URL de acuerdo a una serie de criterios y evaluaciones preestablecidos por defecto en OWASP ZAP, probando todo tipo de ataques, los cuales determinan si la URL puede considerarse un riesgo, o no. En caso de que la URL esté catalogada como sospechosa, se mostrará un alerta indicando su nivel de riesgo de la vulnerabilidad, es decir, si es alto, medio, bajo o si es informativa. La siguiente figura muestra el proceso de escaneo de vulnerabilidades para la URL seleccionada.

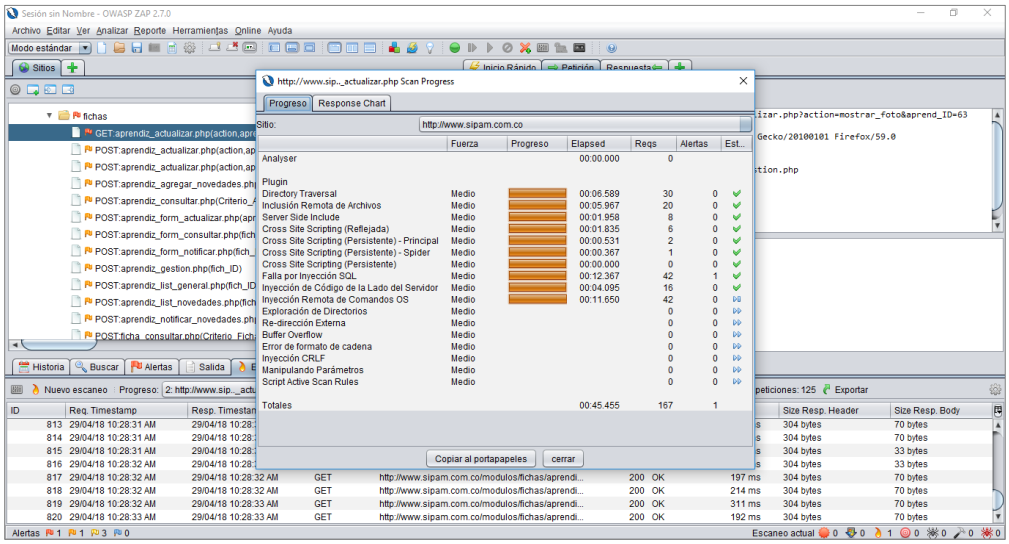
Figura 21. Proceso de Escaneo de Vulnerabilidades_Herramienta OWASP ZAP.



Fuente. Elaboración propia.

Mientras se avanza el *Escaneo Activo*, se pueden observar el detalle de las vulnerabilidades analizadas por medio del botón de consola ubicado al lado izquierdo de la barra de progreso. En la ventana emergente se observa a la herramienta comprobando vulnerabilidades como Cross Site Scripting (XSS), Falla por Inyección SQL, o Inclusión Remota de Archivos, entre otras. La siguiente figura muestra el detalle del progreso del escaneo.

Figura 22. Detalle del Proceso de Escaneo de Vulnerabilidades_Herramienta OWASP ZAP.

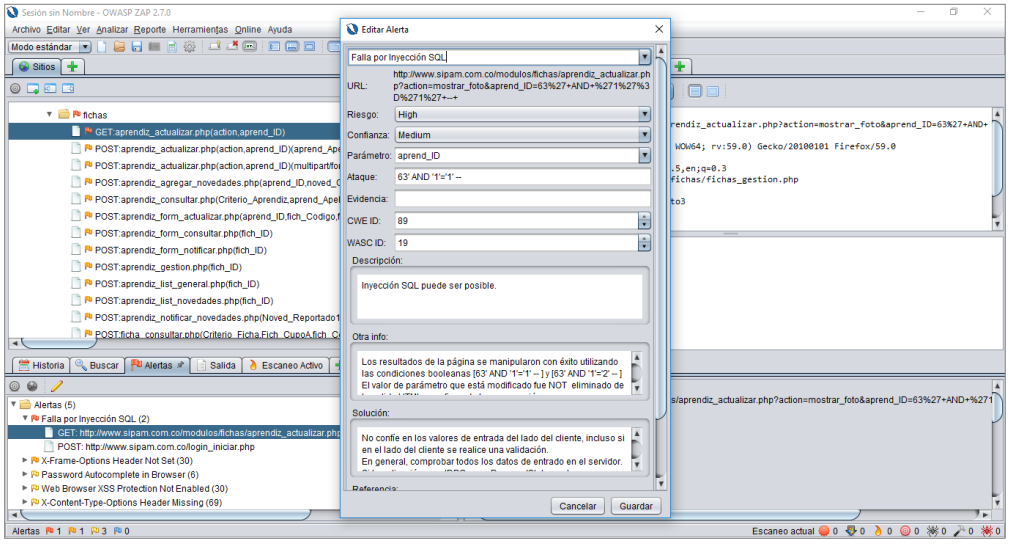


Fuente. Elaboración propia.

A medida que finaliza el escaneo de cada URL, se puede activar la pestaña de *Alertas* donde se muestran las posibles vulnerabilidades detectadas, notificándolas en forma de banderilla de diferente color. Cada color indica el nivel de riesgo o criticidad de la vulnerabilidad detectada: Rojo para indicar que es de riesgo alto, naranja para indicar que es de riesgo medio, amarilla para indicar que es de riesgo bajo y azul para indicar que es informativa.

Al dar clic en una de las URL de una determinada vulnerabilidad o alerta encontrada, se despliega una ventana emergente que muestra información detallada al respecto, indicando desde el parámetro empleado para el ataque, hasta la forma de solucionarla, lo cual es muy importante porque indica las medidas se pueden tomar para evitar que dicho fallo de seguridad pueda ser explotado. La siguiente figura muestra el detalle de una alerta escaneada.

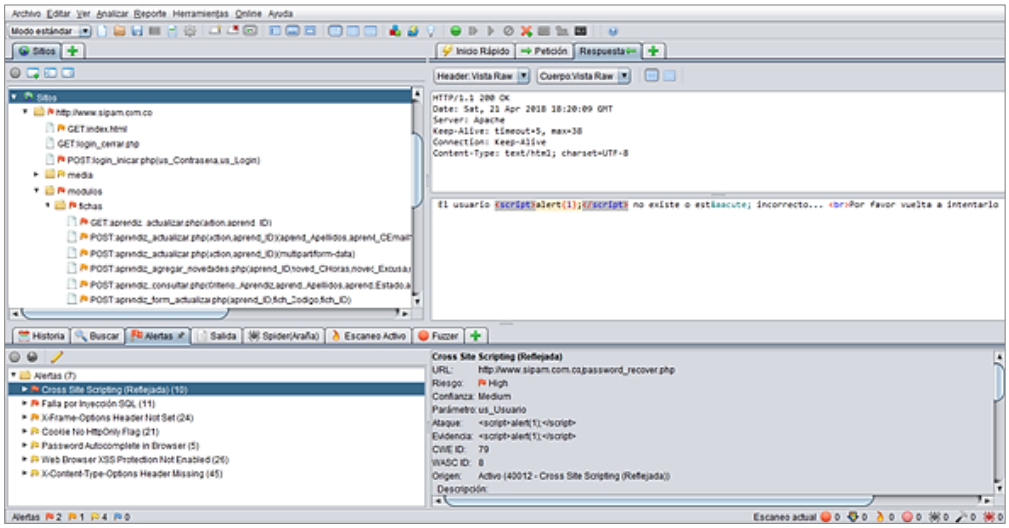
Figura 23. Detalle Alerta Escaneada_Herramienta OWASP ZAP.



Fuente. Elaboración propia.

Una vez finalizado el *Escaneo Activo* para cada URL, se puede visualizar un resumen de las vulnerabilidades detectadas, clasificadas de acuerdo a su nivel de riesgo y con el número de instancias encontradas. Estas instancias corresponden a las URL afectadas. La siguiente figura muestra el resumen de alertas detectadas por la herramienta OWASP ZAP para el aplicativo Web.

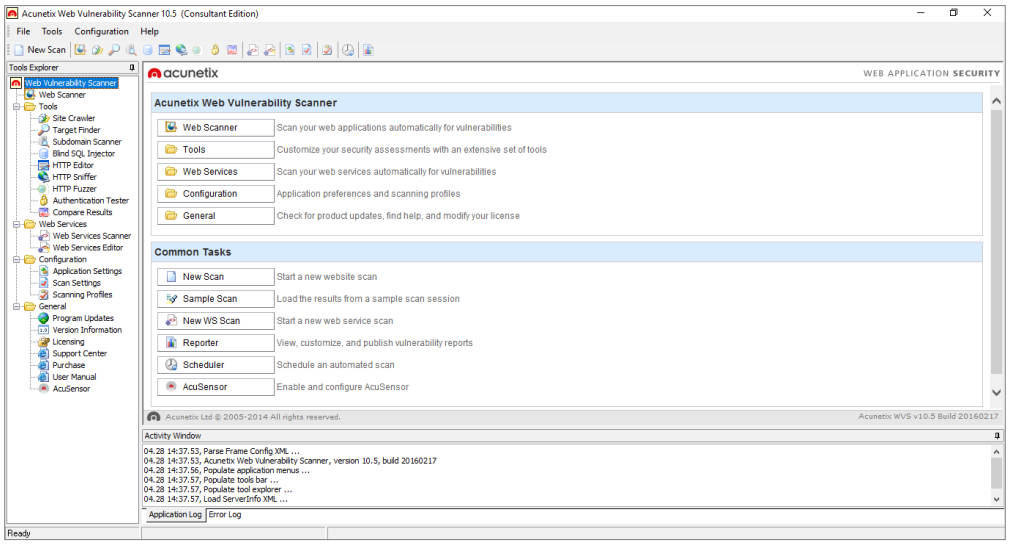
Figura 24. Resumen de Alertas Detectadas_Herramienta OWASP ZAP.



Fuente. Elaboración propia.

Ejecución de la Herramienta ACUNETIX WVS. Inicialmente se ejecutó esta herramienta en el escenario No. 1 pero el firewall del servidor, bloqueó repetidamente la IP del equipo desde el cual se estaba ejecutando ACUNETIX WVS, pues la IP 181.63.167.47 fue considerada como una amenaza. Básicamente esto ocurre como un mecanismo de defensa del servidor para proteger los sitios que se alojan en esa máquina. Por tal motivo se eligió el escenario N. 2 que no ofrece restricciones de seguridad que bloqueen su ejecución, pues es un servidor de pruebas utilizado como entorno de desarrollo de la aplicación. Cuando se ejecuta la herramienta, inicialmente aparece una pantalla, como se observa en la siguiente figura, donde se presentan las diversas opciones que ofrece esta herramienta para su utilización. Al hacer clic sobre el botón de *New Scan*, se inicia un nuevo escaneo automático del sitio Web.

Figura 25. Opciones de Escaneo de Vulnerabilidades_Herramienta ACUNETIX WVS.

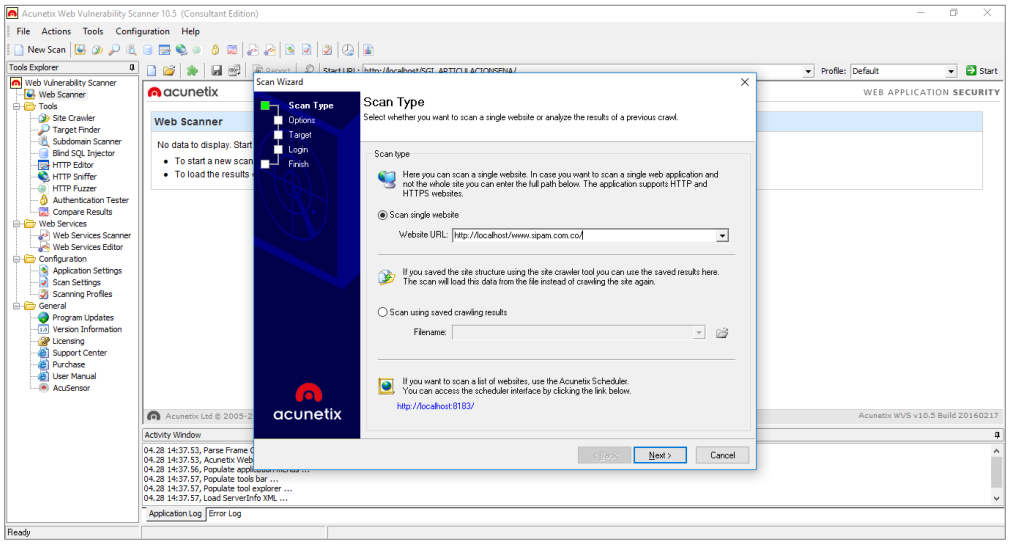


Fuente. Elaboración propia.

Inmediatamente se activa un asistente que ayuda a configurar algunos parámetros para mejorar la eficiencia del escaneo así como para hacerlo menos intrusivo. Se puede configurar para realizar un escaneo solo al sitio Web o a una inspección ya realizada anteriormente, elegir si se quiere un escaneo completo o si se quiere realizar para determinadas vulnerabilidades (SQL Injection, Cross Site Scripting, entre otras), también permite ajustar los parámetros para el Target o la dirección exacta de la URL, configurar las opciones login y password para las áreas protegidas del sitio Web y finalmente visualizar un resumen de configuración realizada. La siguiente figura muestra el inicio del asistente de configuración del escaneo, seleccionando en primera instancia la opción

Scan Single WebSite en donde se ingresa la dirección URL del sitio a escanear, en este caso http://localhost/www.sipam.com.co.

Figura 26. Asistente para la Configuración del Escaneo de Vulnerabilidades_Herramienta ACUNETIX WVS.

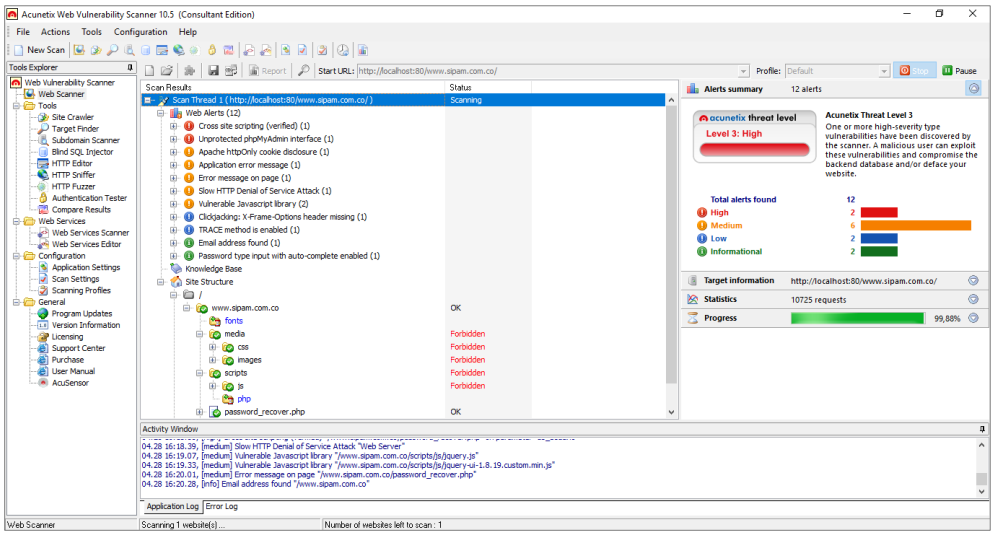


Fuente. Elaboración propia.

Una vez finalizada la configuración del escaneo, ACUNETIX WVS empieza a analizar en tiempo real el sitio Web, inspeccionando todas las carpetas, archivos y demás. En este caso, y al igual que con la anterior herramienta, ACUNETIX WVS no permitió realizar un escaneo a la totalidad de elementos que conforman en su totalidad el sitio. Solo hizo el reconocimiento de la estructura externa del sitio Web y no realizó una exploración de los directorios, scripts o URL que se cargan cuando el usuario inicia sesión. Por lo tanto el reporte de vulnerables detectadas, solo generará un informe parcial de resultados, pero con información muy importante y detallada de las vulnerabilidades encontradas a éste nivel.

En esta exploración del escaneo, la herramienta empieza a lanzar una serie de ataques de vulnerabilidades en cada página o URL, simulando lo que un hacker podría hacer para explotar las vulnerabilidades de un sitio Web. Los resultados son mostrados en el nodo de Alertas (*Web Alerts*). La siguiente pantalla muestra el progreso de avance de un escaneo automático y un resumen parcial de las alertas encontradas hasta el momento.

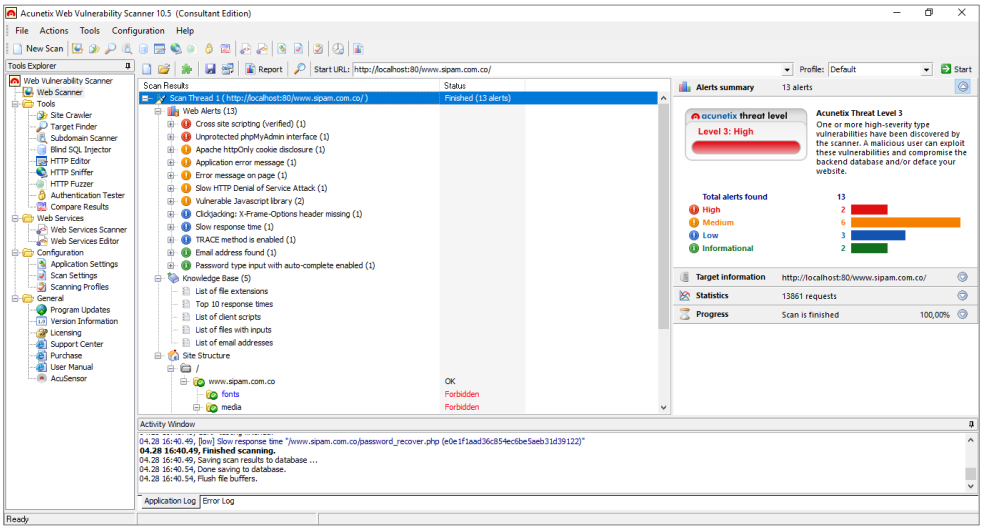
Figura 27. Proceso de Escaneo de Vulnerabilidades_Herramienta ACUNETIX WVS.



Fuente. Elaboración propia.

La siguiente figura muestra el resumen final de alertas detectadas. En ella se observa que la zona de *Scan Result* visualiza en detalle todas alertas o vulnerabilidades detectadas, junto con el número de veces que se presenta y su clasificación en niveles de riesgo bajo, medio o alto según el grado de amenaza. También se visualiza la estructura del sitio explorado. Expandiendo sobre una alerta en particular, se visualiza el archivo o recurso vulnerable. La zona de *Activity Windows* muestra el estado de finalización del escaneo y en la zona de *Alerts Summary* se observa el resumen general de alertas según su riesgo y el nivel de amenaza en el que se encuentra expuesto el aplicativo Web.

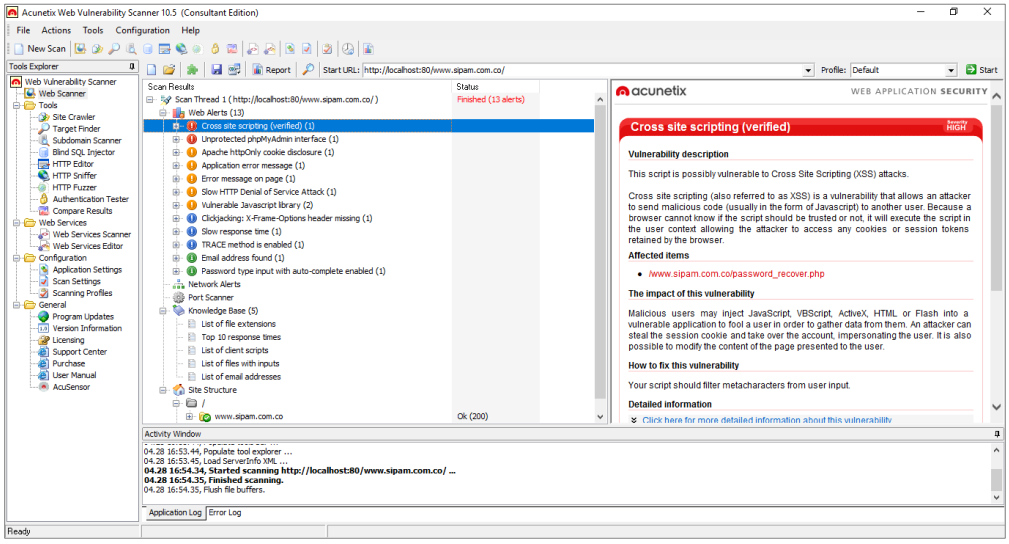
Figura 28. Resumen de Alertas Detectadas_Herramienta ACUNETIX WVS.



Fuente. Elaboración propia.

Cada alerta presenta información muy específica acerca de la vulnerabilidad detectada. Esta información contiene la descripción de la misma, el recurso o URL que se ve afectado, el impacto que causa, las recomendaciones para su mitigación y su clasificación (CVE, CWE, CVSS), entre otras. Por ejemplo en la siguiente figura se observa la descripción completa de la vulnerabilidad *Cross Site Scripting*.

Figura 29. Detalle Alerta Escaneada_Herramienta ACUNETIX WVS.



Fuente. Elaboración propia.

Fase IV: Reporte de Vulnerabilidades Web

En este paso se presente finalmente un resumen de las vulnerabilidades reportadas por las herramientas OWASP ZAP y ACUNETIX WVS como resultado de la ejecución de pruebas de seguridad realizadas al aplicativo Web. Algo muy importante que poseen estas herramientas, es la posibilidad de guardar los resultados de un proceso de escaneo de vulnerabilidades, con el fin probar individualmente cada vulnerabilidad en lugar de volverlas a ejecutar haciendo de nuevo una exploración completa del sitio o simplemente guardar los resultados de un escaneo una vez analizado el sitio. El reporte de vulnerabilidades Web es fundamental para elaborar informes técnicos y tomar medidas de control que permitan mitigar los riesgos detectados.

Reporte de Vulnerabilidades Web con la Herramienta OWASP ZAP. Las siguientes tablas presentan un resumen de las vulnerabilidades reportadas por la herramienta OWASP ZAP. La primera tabla clasifica las vulnerabilidades según su nivel de riesgo y la segunda tabla presenta la descripción de las mismas. En cada tabla se presenta el número de instancias presentadas por cada vulnerabilidad detectada.

Tabla 12. Resumen Vulnerabilidades Web por Nivel de Riesgo_Herramienta OWASP ZAP.

Nivel de Riesgo	Vulnerabilidad Detectada	No. de Instancias
Alto	Cross Site Scripting (Reflejada)	21
	Falla por Inyección SQL	
Medio	X-Frame-Options Header Not Set	24
Bajo	Cookie No HttpOnly Flag	97
	Password Autocomplete in Browser	
	Web Browser XSS Protection Not Enabled	
	X-Content-Type-Options Header Missing	

Fuente. Elaboración propia.

Tabla 13. Resumen Vulnerabilidades Web por Categoría_Herramienta OWASP ZAP.

Vulnerabilidad Detectada	Descripción	Nivel de Riesgo	No. de Instancias
Cross Site Scripting (Reflejada)	Cross-site Scripting (XSS) es un método de ataque que ejecuta el código que proporciona un atacante en una instancia del navegador del usuario, de tal forma que el código suministrado puede leer, cambiar y transmitir cualquier dato sensible al que pueda acceder el navegador. El código está escrito en HTML/JavaScript, pero también puede extenderse a cualquier otra tecnología compatible con el navegador. Un usuario de Cross-site Scripted podría tener su cuenta secuestrada (robo de cookies), redirigir su navegador a otra ubicación, o posiblemente mostrar contenido fraudulento entregado por el sitio Web que está visitando. Los ataques de scripting entre sitios comprometen la confianza entre un usuario y el sitio Web.	Alto	10
Falla por Inyección SQL	La falla por inyección SQL es ataque de inyección, que consiste en insertar comandos SQL en la entrada de datos del cliente con la finalidad de efectuar la ejecución de instrucciones SQL predefinidas en la aplicación. Un ataque por inyección SQL puede consultar información sensible de la base de datos, modificarla (Insert/Update/Delete), ejecutar operaciones de	Alto	11

	administración sobre la misma, recuperar el contenido de un determinado archivo y en otros casos ejecutar comandos al sistema operativo.		
X-Frame-Options Header Not Set	El encabezado X-Frame-Options no está incluido en la respuesta HTTP para proteger contra los ataques de ClickJacking. La mayoría de los navegadores Web aceptan el encabezado HTTP X-FrameOptions. Es necesario que esté configurado en todas las páginas Web que devuelve el sitio. El clickjacking consiste en un método de ataque que se realiza a través del navegador, tratando de engañar al usuario mediante una capa transparente que se puede conseguir con HTML e "iframe" colocada delante de un enlace o cuadro de diálogo. Su objetivo es hacer que el usuario pulse un enlace sin percatarse de ello.	Medio	24
Cookie No HttpOnly Flag	Se ha configurado una cookie sin el indicador HttpOnly. Esto indica que las cookies no se establecieron como exclusivas para HTTP, lo cual permite que las cookies sean accedidas por código JavaScript. Esto permite a un atacante inyectar código malicioso y robar las cookies del usuario. En caso de una cookie de sesión, es posible el secuestro de la sesión y con eso la ejecución de ciertas tareas o procesos con otros permisos o privilegios distintos al usuario actual con que se está ejecutando el proceso principal.	Bajo	21
Password Autocomplete in Browser	Autocompletar contraseña en el navegador. Esto significa que el atributo AUTOCOMPLETE no está deshabilitado en un elemento HTML FORM / INPUT que contiene entrada de texto de tipo de contraseña. Las contraseñas pueden ser guardadas en navegadores y recuperarse posteriormente.	Bajo	5
Web Browser XSS Protection Not Enabled	Navegador Web con protección XSS no habilitada. El encabezado de respuesta HTTP de protección X-XSS permite que el servidor Web habilite o deshabilite el mecanismo de protección XSS del navegador Web. El encabezado de respuesta HTTP X-XSS-Protection actualmente es compatible con Internet Explorer, Chrome y Safari (WebKit). Tener en cuenta que esta alerta solo se genera si el cuerpo de la respuesta podría contener una carga útil XSS (con un tipo de contenido basado en texto, con una longitud distinta de cero).	Bajo	26
X-Content-Type-Options Header Missing	El encabezado Anti-MIME-X-Content-Type-Options no se configuró en “nosniff”. Esto significa que las versiones anteriores de IE y Chrome pueden realizar el rastreo de MIME, utilizados para describir el tipo de medio del contenido en el cuerpo de la respuesta, lo que puede causar que se muestre como un tipo de contenido distinto del tipo de contenido	Bajo	45

	declarado. Las versiones actuales (principios de 2014) y heredadas de Firefox utilizarán el tipo de contenido declarado (si está configurado), en lugar de realizar el rastreo de MIME. Este problema se aplica a las páginas de tipo de error (401, 403, 500, etc.), pues son páginas que a menudo aún se ven afectadas por problemas de inyección. Si se envía la cabecera X-Content-Type-Options en la respuesta con el valor "nosniff", los navegadores que soportan esta cabecera como IE y Chrome, no cargan las CSS, ni los scripts (Javascript), cuyo Myme-type no sea el adecuado.		

Fuente. Elaboración propia.

Reporte de Vulnerabilidades Web con la Herramienta ACUNETIX WVS. Las siguientes tablas presentan un resumen de las vulnerabilidades reportadas por la herramienta ACUNETIX WVS. La primera tabla clasifica las vulnerabilidades según su nivel de riesgo y la segunda tabla presenta la descripción de las mismas, cada una con el número de instancias presentadas respectivamente. Como se puede observar, esta herramienta detectó una cantidad mayor de vulnerabilidades en comparación con la herramienta anterior.

Tabla 14. Resumen Vulnerabilidades Web por Nivel de Riesgo_Herramienta ACUNETIX WVS.

Nivel de Riesgo	Vulnerabilidad Detectada	No. de Instancias
Alto	Cross Site Scripting (XSS)	2
	Unprotected phpMyAdmin interface	
Medio	Apache httpOnly cookie disclosure	6
	Application error message	
	Error message on page	
	Slow HTTP Denial of Service Attack	
	Vulnerable Javascript library	
Bajo	Clickjacking: X-Frame-Options header missing	3
	Slow response time	
	TRACE method is enabled	
Informativo	Email address found	2
	Password type input with auto-complete enabled	

Fuente. Elaboración propia.

Tabla 15. Resumen Vulnerabilidades Web por Categoría_ Herramienta ACUNETIX WVS.

Vulnerabilidad Detectada	Descripción	Nivel de Riesgo	No. de Instancias
Cross Site Scripting (XSS)	Cross Site Scripting (también conocido como XSS) es un tipo de vulnerabilidad que permite a un atacante enviar código maligno (por lo general en forma de Javascript) a otro usuario. La secuencia de comandos se ejecutará en el contexto del usuario lo que le permite al atacante acceder a las cookies o tokens de sesión almacenados en el navegador.	Alto	1
Unprotected phpMyAdmin interface	Interfaz phpMyAdmin desprotegida. phpMyAdmin es una aplicación escrita en lenguaje PHP que proporciona una interfaz basada en Web para administrar de bases de datos MySQL. La contraseña de la cuenta raíz de MySQL inicial está vacía, por lo que cualquier usuario puede conectarse al servidor MySQL como root unicamente y se le otorgarán todos los privilegios.	Alto	1
Apache httpOnly cookie disclosure	Divulgación de cookies httpOnly de Apache. Apache HTTP Server desde la versión 2.2.x hasta la 2.2.21 no restringe acertadamente la información del encabezado en la construcción de documentos de error de Bad Request (400), por lo que los atacantes remotos pueden obtener los valores de las cookies HTTPOnly a través de vectores que implican un encabezado mal formado con un script Web diseñado.	Medio	1
Application error message	Mensaje de error de la aplicación. Informa que la aplicación escaneó una página que contiene un mensaje de error o advertencia que puede mostrar información confidencial. El mensaje también puede contener la ubicación del archivo que produjo la excepción no controlada.	Medio	1
Error message on page	Mensaje de error en la página. Informa que una página escaneada contiene un mensaje de error o advertencia que puede revelar información confidencial. El mensaje puede contener la ubicación del archivo que produjo la excepción no controlada.	Medio	1
Slow HTTP Denial of Service Attack	Ataque denegación de servicio HTTP. Informa que el servidor Web podría estar en riesgo de un ataque de denegación de servicio. Los ataques DoS se basan en el hecho de que el protocolo HTTP, requiere que las solicitudes sean completamente recibidas por el servidor antes de ser procesadas. Si una solicitud HTTP no está completa, o si la tasa de transferencia es muy baja, el servidor mantiene sus recursos ocupados esperando el resto de los datos. Si el servidor mantiene demasiados recursos ocupados, esto crea una denegación de servicio.	Medio	1

Vulnerable Javascript library	Librería Javascript Vulnerable. Se informa que se encontraron una o más vulnerabilidades para esta versión de la librería de Javascript.	Medio	2
Clickjacking: X-Frame-Options header missing	Clickjacking: Falta el encabezado X-Frame-Options. Clickjacking es una técnica maliciosa para engañar a un usuario Web para que haga clic en algo diferente de lo que el usuario percibe cuando está haciendo clic, revelando así información confidencial o tomando control de su computador haciendo clic en páginas Web aparentemente inofensivas. El servidor no devolvió un encabezado X-Frame-Options, lo que significa que este sitio Web es vulnerable a un ataque de clickjacking. El encabezado de respuesta HTTP de X-Frame-Options se puede utilizar para indicar si se debe permitir o no que un navegador muestre una página dentro de un marco o un iframe.	Bajo	1
Slow response time	Tiempo de respuesta lento. Esta página tiene un tiempo de respuesta lento. Este tipo de archivos se pueden orientar en ataques de denegación de servicio. Un atacante puede solicitar esta página repetidamente desde múltiples computadoras hasta que el servidor se sobrecargue.	Bajo	1
TRACE method is enabled	El método TRACE está habilitado El método TRACE está habilitado en este servidor Web. En presencia de otras vulnerabilidades entre dominios en los navegadores Web, la información del encabezado sensible podría leerse desde cualquier dominio que admita el método HTTP TRACE.	Bajo	1
Email address found	Dirección de correo electrónico encontrada. Se han encontrado direcciones de correo electrónico en esta página. La mayoría de los correos electrónicos no solicitados (spam) proviene de direcciones de correo electrónico cosechadas de Internet. Los spambots, son programas que buscan y registran direcciones de correo electrónico en cualquier sitio Web para luego hacer envíos masivos.	Informativo	1
Password type input with auto-complete enabled	Entrada de tipo contraseña con autocompletar. Cuando se ingresa un nuevo nombre y contraseña en un formulario y se envía, el navegador pregunta si se debe guardar la contraseña. De tal manera que cuando se visualice nuevamente el formulario, el nombre y la contraseña se completan automáticamente o se completan a medida que se ingresa el nombre. Un atacante que tenga acceso al equipo local podría recuperar la contraseña de texto claro desde el caché del navegador.	Informativo	1

Fuente. Elaboración propia.

RESULTADOS

En esta sección se presentan los resultados de la implementación del enfoque metodológico basado en la realización de pruebas de seguridad como estrategia para evaluar el estado de seguridad del Aplicativo Web de Gestión y Seguimiento de Novedades de Aprendices del Programa de Articulación con la Educación Media del Servicio Nacional de Aprendizaje - Sena. Estos resultados corresponden a un resumen de los informes o reportes técnicos que las herramientas, OWASP ZAP y ACUNETIX WVS, generaron durante el proceso de escaneo de vulnerabilidades en el aplicativo Web, dependiendo del escenario particular de ejecución y del modo de ataque de estas herramientas.

En el reporte de seguridad, se presentarán como anexos, los reportes de vulnerabilidades y amenazas que cada herramienta generó sobre el proceso de escaneo de vulnerabilidades Web. A nivel general, cada una de estas herramientas genera información relevante sobre los detalles del escaneo, el nivel de amenaza en el que se encuentra el aplicativo y la distribución de las alertas o vulnerabilidades detectadas. A nivel específico, se presenta para cada vulnerabilidad, el nivel de riesgo asociado, su descripción, el impacto que causa a la seguridad de la información, las referencias para ampliar la información, los detalles de la URL afectada y su clasificación según CWE[26] y WASC[27].

De igual forma es importante tener presente que el nivel de riesgo (Alto[28], Medio[29], Bajo[30], Informativo[31]) en el que se clasifica cada vulnerabilidad, determina el grado de amenaza o exposición ante fallos de seguridad que puede tener el aplicativo ante inminentes ataques y por consiguiente define su estado de seguridad.

---

[26] CWE (Common Weakness Enumeration). Es un estándar de enumeración de vulnerabilidades comunes.

[27] WASC (Web Application Security Consortium). Provee una lista de clasificación de amenazas de aplicaciones Web.

[28] Vulnerabilidad con riesgo de explotación alto de acceso a la información objetivo. Puede poner en peligro la disponibilidad, confidencialidad o integridad de los datos de los usuarios, o la disponibilidad de los recursos de procesamiento.

[29] Vulnerabilidad con riesgo de explotación baja con acceso a la información objetivo. Su impacto puede reducirse en gran medida ya sea mediante mediante configuraciones predeterminadas o por la dificultad propia en su explotabilidad.

[30] Vulnerabilidad sin ningún riesgo de explotación de acceso a la información objetivo. Es una vulnerabilidad difícil de explotar y con impacto es mínimo.

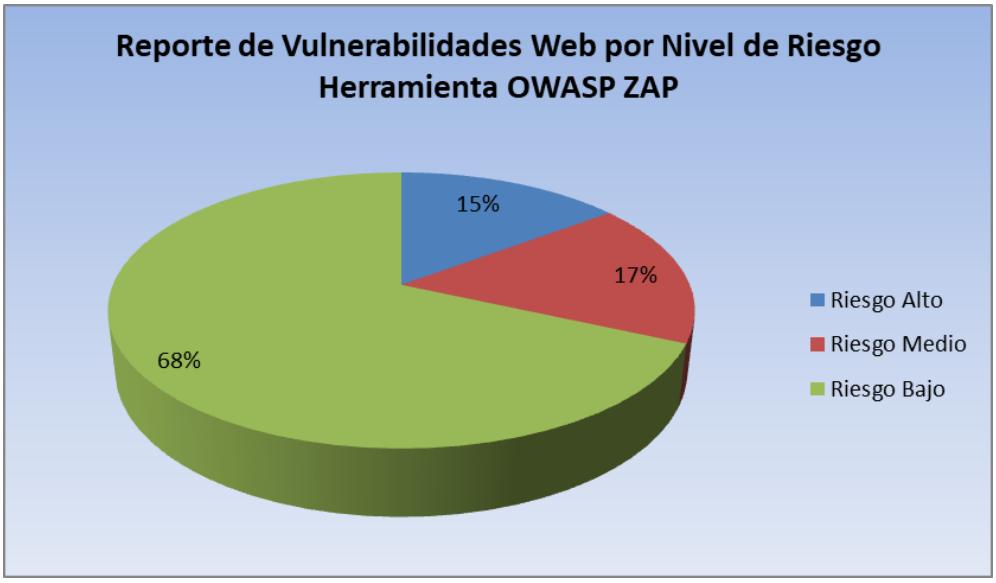
[31] No son consideradas como vulnerabilidades sino que proveen información importante sobre el servicio que es analizado.

Resultados de Ejecución de Pruebas de Seguridad con la Herramienta OWASP ZAP

Nivel de Riesgo del Aplicativo Web	**Alto**
Justificación	Existen vulnerabilidades con un nivel de riesgo posible de explotación a la confidencialidad, integridad y disponibilidad de la información que gestiona el aplicativo Web.

De acuerdo con el resultado del escaneo de vulnerabilidades Web con la Herramienta OWASP ZAP mostrado en la tabla No. 15, se determinó que a nivel general, el aplicativo Web se ubica en un nivel de riesgo alto, pues se detectaron vulnerabilidades como *Cross Site Scripting (Reflejada)* y *Falla por Inyección SQL*. En la siguiente figura se muestra el reporte de vulnerabilidades, medido por el nivel de riesgo y no por el número de instancias presentadas, pues aunque en el nivel de riesgo bajo se presentó el 68% de incidencias ocurridas con respecto al total, el daño que causan los ataques en este nivel no producen ningún impacto significativo al negocio, a diferencia de las vulnerabilidades detectadas en el nivel de riesgo alto, cuya explotabilidad busca tener acceso a la información objetivo.

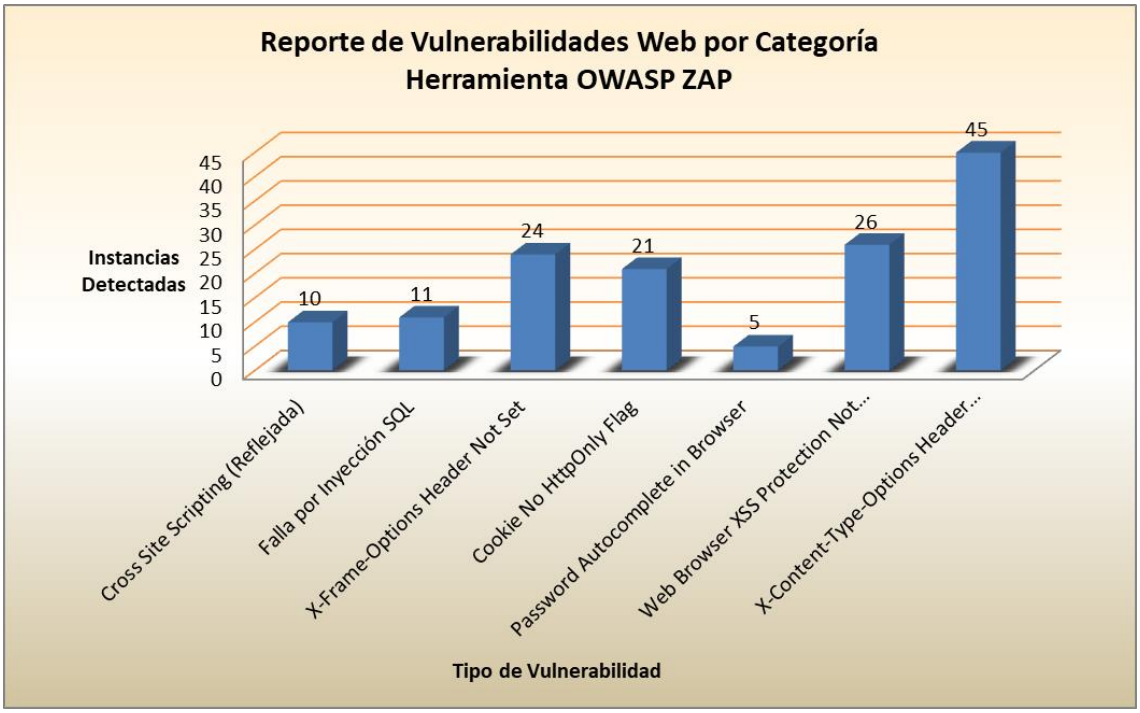
Figura 30. Gráfica de Reporte de Vulnerabilidades Web por Nivel de Riesgo_Herramienta OWASP ZAP.



Fuente. Elaboración propia.

La siguiente figura presenta en forma gráfica los resultados obtenidos en la tabla No. 16 que muestra el número de instancias presentadas por cada vulnerabilidad detectada con la herramienta OWASP ZAP.

Figura 31. Gráfica de Reporte de Vulnerabilidades Web por Categoría_Herramienta OWASP ZAP.



Fuente. Elaboración propia.

A continuación se presentará el detalle de una vulnerabilidad detectada según el reporte técnico de escaneo que genera esta herramienta, como información útil que aporta valor a los desarrolladores para explotar la vulnerabilidad y tomar medidas de seguridad a nivel de código que permitan mitigar su impacto.

Tabla 16. Detalle de Alerta Según Reporte de Vulnerabilidades_ Herramienta OWASP ZAP.

Nivel de Riesgo	Alto
Descripción	Inyección SQL puede ser posible.
URL Method: Parameter: Attack:	http://www.sipam.com.co/modulos/fichas/aprendiz_list_novedades.php Method: POST Parameter: fich_ID Attack: 3-2
Instancias	11

Solución	No confíar en los valores de entrada del lado del cliente, incluso si en el lado del cliente se realizan validaciones. Comprobar todos los datos de entrado en el servidor. Si la aplicación usa JDBC, usar CallableStatement o PreparedStatement con parámetros pasados por '?'. Si la aplicación utiliza ASP, usar ADO Command Objects con una fuerte comprobación de tipos de consultas y parámetros. Use Stored Procedures (Procedimientos Almacenados) si la Base de Datos lo permite. NO concatenar cadenas en las consultas en los procedimientos almacenados, o utilizar 'exec', 'exec immediate', o su funcionalidad equivalente. No generar consultas SQL dinámicas usando una sencilla concatenación de cadenas. Aplicar una lista blanca de caracteres permitidos, o una lista negra de caracteres no permitidos en la entrada (input) del usuario. De los privilegios usados, aplique el privilegio mínimo posible al usuario de la base de datos. Evitar el uso de los usuarios de base de datos 'sa' o 'db-owner'. Esto no elimina la inyección SQL, pero si minimiza su impacto en gran medida. Conceder el mínimo acceso de base de datos necesario para la aplicación.
Otra Información	Los resultados de las páginas originales fueron replicados con éxito utilizando la expresión [3-2] como valor del parámetro. El valor de parámetro que está modificado fue eliminado de la salida HTML para los fines de la comparación
Referencia	https://www.owasp.org/index.php/Top_10_2010-a1 https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet
CWE Id	89
WASC Id	19
Source ID	1

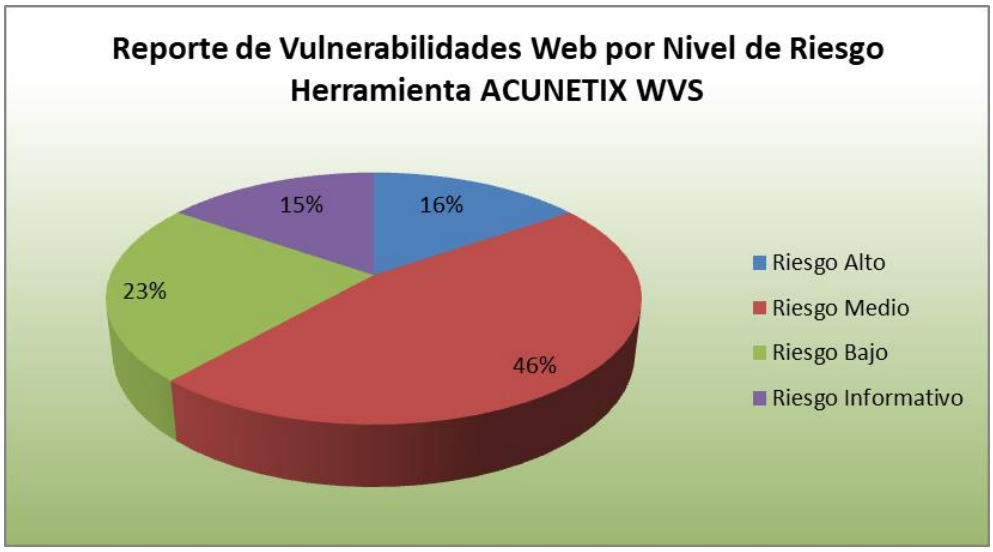
Fuente. Reporte de Escaneo ZAP.

Resultados de Ejecución de Pruebas de Seguridad con la Herramienta ACUNETIX WVS

Nivel de Amenaza del Aplicativo Web	**Alto**
Justificación	El escáner descubrió una o más vulnerabilidades de alta gravedad. Un usuario malicioso puede explotar estas vulnerabilidades y comprometer la base de datos o desconfigurar su sitio Web.

De acuerdo con el resultado del escaneo de vulnerabilidades Web con la Herramienta ACUNETIX WVS mostrado en la tabla No. 17, se determinó que el aplicativo Web se ubica en un nivel de amenaza alto, porque se detectó la vulnerabilidad *Cross site scripting (verified).* La siguiente figura se muestra el reporte de vulnerabilidades, clasificado por el nivel de riesgo y no por el número de instancias presentadas, pues se debe recordar que esta herramienta fue ejecutada en un escenario diferente al ejecutado con la herramienta OWASP ZAP. Igualmente se observa que aunque en el nivel de riesgo bajo se presentó el 46% de incidencias ocurridas con respecto al total, el daño que puede que causan los ataques en este nivel no comprometen la información del sistema y por lo tanto no causan un impacto significativo que perjudique al negocio. Caso contrario ocurre con la vulnerabilidad encontrada en el nivel de riego alto porque aunque tan solo presentó el 16% de instancias ocurridas con respecto al total, esta vulnerabilidad es de alta gravedad, ya que un usuario malintencionado puede explotarla y comprometer la información del aplicativo almacenada la base de datos.

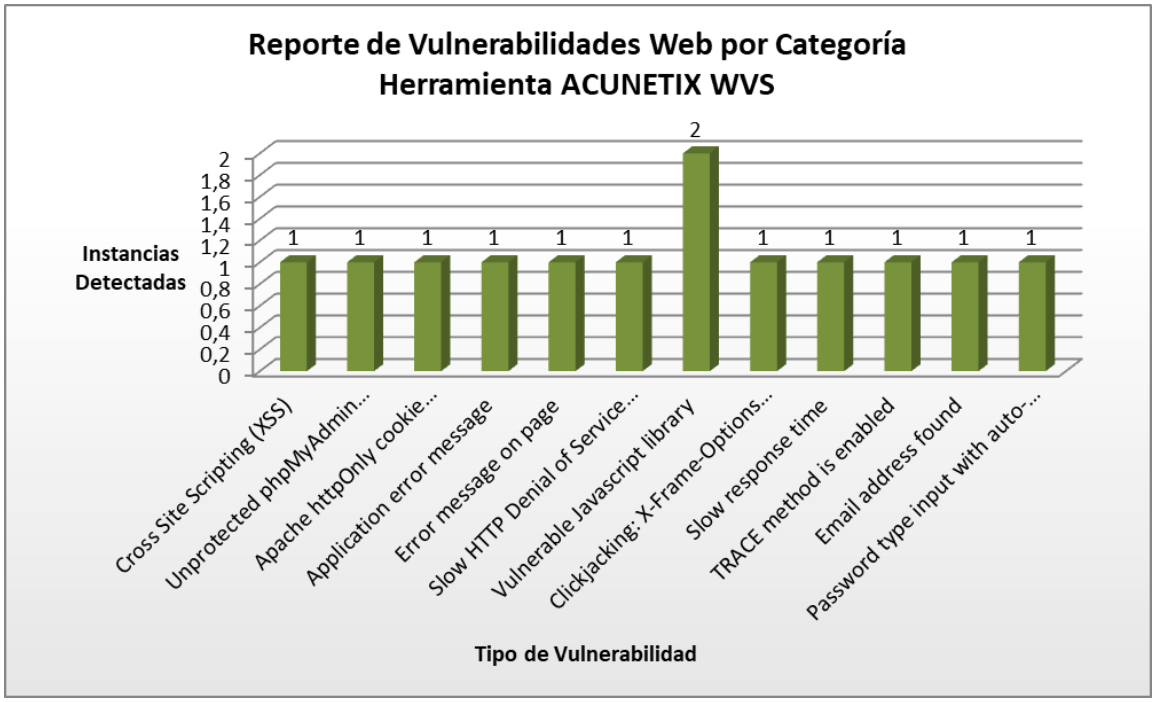
Figura 32. Gráfica de Reporte de Vulnerabilidades Web por Nivel de Riesgo_Herramienta ACUNETIX WVS.



Fuente. Elaboración propia.

La siguiente figura presenta en forma gráfica los resultados obtenidos en la tabla No. 18 que muestra el número de instancias presentadas para cada una de las vulnerabilidades detectadas por la herramienta ACUNETIX WVS. Como se observa en la figura, esta herramienta detectó una cantidad mayor de vulnerabilidades que las reportadas por la herramienta OWASP ZAP, demostrando su efectividad de escaneo en la estructura del sitio Web que pudo ser explorada.

Figura 33. Gráfica de Reporte de Vulnerabilidades Web por Categoría_Herramienta ACUNETIX WVS.



Fuente. Elaboración propia.

A continuación se presentará el detalle de una vulnerabilidad detectada según el reporte técnico de escaneo que genera esta herramienta, como información útil que aporta valor a los desarrolladores para explotar la vulnerabilidad y tomar medidas de seguridad a nivel de código que permitan mitigar su impacto.

Tabla 17. Detalle de Alerta Según Reporte de Vulnerabilidades_ Herramienta ACUNETIX WVS.

Alerta	Cross site scripting (verified)
Gravedad	Alta
Tipo	Validación
Informado por Módulo	Scripting (XSS.script)
Descripción	Este script es potencialmente vulnerable a los ataques de Cross Site Scripting (XSS). Cross site scripting (también conocido como XSS) es una vulnerabilidad que permite a un atacante enviar código malicioso, por lo general en Javascript, a otro usuario. Debido a que un navegador no puede detectar si la secuencia de comandos es confiable o no, ejecutará la secuencia de comandos en el lado del usuario permitiendo al atacante acceder a las cookies o tokens de sesión almacenados en el navegador.

Impacto	Un atacante puede inyectar código JavaScript, VBScript, ActiveX, HTML o Flash en una aplicación vulnerable para engañar a un usuario con el fin de recopilar sus datos. Un atacante puede robar la cookie de sesión y hacerse cargo de la cuenta, haciéndose pasar por el usuario. También es posible alterar el contenido de la página presentada al usuario.
Recomendación	La secuencia de comandos se debe ser filtrada mediante metacaracteres desde la entrada del usuario.
Referencias	VIDEO: How Cross-Site Scripting (XSS) Works How To: Prevent Cross-Site Scripting in ASP.NET The Cross Site Scripting Faq OWASP Cross Site Scripting XSS Annihilation XSS Filter Evasion Cheat Sheet Cross site scripting OWASP PHP Top 5 Acunetix Cross Site Scripting Attack
Items Afectados	/www.sipam.com.co/password_recover.php

Fuente. Reporte del Desarrollador. Auditoría del sitio Web Acunetix.

DISCUSIÓN Y CONCLUSIONES

Dentro de las discusiones planteadas en torno al análisis de la literatura revisada y a la ejecución de la estrategia metodológica utilizada en la evaluación de los controles técnicos de seguridad de la información en el aplicativo Web de Gestión y Seguimiento de Novedades de Aprendices del Programa de Articulación con la Educación Media del Servicio Nacional del Aprendizaje – Sena, surgen las siguientes:

- Actualmente existen investigaciones científicas en torno a medir la efectividad en la capacidad de detección y evaluación del rendimiento de escáneres de vulnerabilidades en aplicaciones, en donde los resultados obtenidos no describen los escenarios particulares de ejecución, pues aunque estas herramientas pueden escanear de forma automática la estructura de un sitio Web y encontrar rápidamente vulnerabilidades, también hay que tener en cuenta el modo de ataque, es decir, no todas las pruebas de seguridad pueden ser realizadas automáticamente, por lo que en ocasiones se deben configurar para realizar pruebas de seguridad de forma manual. De esto depende que se puedan obtener información más confiable o precisa o de lo contrario obtener información muy general que no aporte resultados concretos.

- Otro aspecto importante que influyó en la obtención de resultados, fue el nivel de seguridad aplicado a la configuración del servidor Web en donde se encontraba instalada la aplicación para la ejecución de las pruebas tantos funcionales como de seguridad, pues como mecanismo de defensa, por las múltiples peticiones y solicitudes hechas al servidor web en la ejecución de los escáneres de seguridad, se bloqueó la dirección IP de la máquina de donde provenían estos ataques. Este inconveniente hizo que se debiera utilizar otro escenario de realización de pruebas de seguridad, donde la configuración por defecto del servidor, fuera insuficiente para la ejecución normal de estas herramientas y ser más efectivas en la detección de vulnerabilidades en el aplicativo Web.

- Se comprobó que los escáneres de vulnerabilidades utilizados fueron efectivos al detectar dos de las vulnerabilidades con mayor riesgo para las aplicaciones Web, que se encuentran en el Top 10 de OWASP en su última publicación. Esto indica que el nivel de riesgo frente a una amenaza es alto, pues estas fallas pueden ser explotadas por los atacantes para comprometer la confidencialidad, disponibilidad e integridad de la información sensible objeto del ataque e inclusive a la indisponibilidad del sistema, entre otras consecuencias, lo que eventualmente conlleva a una afectación de la reputación de la entidad y la confiabilidad por parte de sus usuarios. En el caso del

Sena por ser una entidad de orden nacional, estas consecuencias serían supremamente graves. Lo anterior sugiere al grupo de desarrolladores, que se requiere de una revisión del código existente para una inmediata solución.

- El nivel de cumplimiento de los objetivos fue satisfactorio en un 100%. Los resultados obtenidos indican que el enfoque metodológico orientado hacia las pruebas de seguridad de caja negra, resultó pertinente aplicarlo en la fase de implementación del aplicativo web. Esto permitió la detección temprana de fallos o vulnerabilidades en los controles técnicos de seguridad de la información implementados hasta el momento, antes de que la aplicación fuera puesta en marcha en su entorno real de producción, dando solución a la situación problémica planteada inicialmente y aportando información relevante para la construcción de un software de calidad, pues la mejor forma de prevenir vulnerabilidades en las aplicaciones es escribir código seguro.

Las conclusiones respecto a las implicaciones de los resultados el área de desarrollo seguro de software dentro de la ingeniería de sistemas y sobre las futuras investigaciones que se sugieren emprender según los hallazgos encontrados, son las siguientes:

- Se utilizaron estrategias para la toma de decisiones que propendan a la protección del aplicativo web como activo de información, frente a todo tipo de amenazas, ya sean internas o externas, que comprometan la seguridad de su información, cumpliendo así con uno de los objetivos del Sistema de Gestión de la Seguridad de la Información adoptado por el SENA.

- Es posible disminuir las vulnerabilidades en una aplicación Web si durante el proyecto de desarrollo del software son consideradas desde el análisis y definición de requerimientos, diseño, implementación y pruebas, para asegurarse que han sido eliminadas de forma correcta. Es fundamental asignar recursos para las actividades de seguridad en la gestión de productos de software, el desarrollo de una política de desarrollo seguro, realización de pruebas de intrusión y técnicas para la revisión manual del código fuente.

- Los escáneres de vulnerabilidad ayudaron a comprobar las vulnerabilidades del Top 10 de OWASP en el aplicativo Web. Su importancia radica en que fueron utilizadas para descubrir vulnerabilidades antes de ser expuestas para su explotación por parte de algún usuario malintencionado. Con esto se pretende implementar medidas necesarias para su remediación y evitar su exposición en un entorno de producción.

- El reporte técnico de seguridad que se genera como producto de esta investigación, es una herramienta fundamental que sirve de directriz para conocer el nivel de riesgo de amenazas, tomar medidas para solucionar las vulnerabilidades encontradas y proporcionar medidas de defensa que permitan asegurar la aplicación Web de manera significativa como objeto de un posible ataque.

- Se recomienda analizar los hallazgos informados en el reporte técnico de seguridad, en lo referente a la explotación de las vulnerabilidades identificadas según su criticidad, para comprobar que son verdaderas y dimensionar el daño que podría causar a la entidad, en función de la información que gestiona el aplicativo.

- En el caso de que el aplicativo web llegase a ser incluido como parte del software institucional, se recomienda tomar medidas de aseguramiento (hardening) sobre la infraestructura necesaria para el funcionamiento del aplicativo en la Web, adoptando políticas de seguridad que reduzcan de manera efectiva la superficie de ataque sobre dicha infraestructura.

REFERENCIAS

[1] D. Wichers, “OWASP Top-10 2013,” 2013.

[2] S. C. Romaniz, “Seguridad de aplicaciones web: vulnerabilidades en los controles de acceso.”

[3] H. T. Quinche, René Guamán, “Seguridad en Entornos Web para Sistemas de Gestión Académica,” no. January, pp. 1–47, 2011.

[4] C. Arbeláez Salazar, M. Aguirre, F. Alejandro, C. Osorio, and J. Andrés, “Herramientas para el Desarrollo Rápido de Aplicaciones Web,” *Sci. Tech. Año XVII*, vol. 47, no. 47, pp. 254–258, 2011.

[5] C. E. Gómez Montoya, C. Andrés, C. Uribe, L. Eduardo, and S. Rodríguez, “Seguridad en la configuración del Servidor Web Apache * Security in the Apache Web Server Configuration,” *Rev. Inge CuC*, vol. 9, no. 2, pp. 31–38, 2013.

[6] E. D. Barbosa, R. De, and O. Castro, “Desenvolvimento de Software Seguro: Conhecendo e Prevenindo Ataques Sql Injection e Cross-site Scripting(XSS).”

[7] F. Asteasuain, “17_Aplicación de la POA como Solución a los Problemas de la Seguridad en el Software.”

[8] M. Castellaro, S. Romaniz, J. Ramos, and P. Pessolani, “Hacia la Ingeniería de Software Seguro,” *Fac. Reg. St. Fe - Univ. Tecnológica Nac.*, vol. 610, p. 10, 2009.

[9] C. Brito, “Metodologías para Desarrollar Software Seguro,” *Univ. Autónoma Zacatecas*, vol. 2, no. 3, pp. 1–16, 2013.

[10] M. Castellaro, S. Romaniz, J. C. Ramos, C. Feck, and I. Gaspoz, “Aplicar el Modelo de Amenazas para incluir la Seguridad en el Modelado de Sistemas.”

[11] G. E. Barba Olivares, “16_Modelado de Amenazas, una Técnica de Análisis y Gestión de Riesgo Asociado a Software y Aplicaciones.”

[12] C. Joshi and U. K. Singh, “Performance Evaluation of Web Application Security Scanners for More Effective Defense,” vol. 6, no. 6, pp. 660–667, 2016.

[13] S. M. D. Diaz, “Pruebas de Seguridad en Aplicaciones Web como Imperativo en la Calidad de Desarrollo del Software.”

[14] S. M. Diaz Diaz, “Pruebas de seguridad en aplicaciones web como imperativo en la calidad de desarrollo del software.”

[15] A. L. Hernández Saucedo and J. Mejia Miranda, “Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones web,” *ReCIBE*, vol. 4, no. 1, p. 17, 2015.

[16] F. R. Muñoz, I. Israel, S. Cortés, L. Javier, and G. Villalba, “Capacidades de Detección de las Herramientas de Análisis de Vulnerabilidades en Aplicaciones Web.”