OWASP COMO ELEMENTO ESTRATÉGICO EN LA IDENTIFICACIÓN DE VULNERABILIDADES Y LA VALIDACIÓN DE SEGURIDAD EN EL DISEÑO, PROGRAMACIÓN Y OPERACIÓN DE APLICACIONES SEGURAS EN LAS ORGANIZACIONES DESARROLLADORAS DE SOFTWARE EN COLOMBIA.

OMAR CAMILO SANTIAGO GARCÍA.

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA – UNAD
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA - ECBTI
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
VILLAVICENCIO.
2021

OWASP COMO ELEMENTO ESTRATÉGICO EN LA IDENTIFICACIÓN DE VULNERABILIDADES Y LA VALIDACIÓN DE SEGURIDAD EN EL DISEÑO, PROGRAMACIÓN Y OPERACIÓN DE APLICACIONES SEGURAS EN LAS ORGANIZACIONES DESARROLLADORAS DE SOFTWARE EN COLOMBIA.

.

OMAR CAMILO SANTIAGO GARCÍA.

Proyecto de Grado – Monografía presentado para optar por el título de
ESPECIALISTA EN SEGURIDAD INFORMÁTICA

YENNY STELLA NUÃ‘EZ
Directora de proyecto

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA – UNAD
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA - ECBTI
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
VILLAVICENCIO.
2021

NOTA DE ACEPTACIÓN

Firma del Presidente de Jurado

Firma del Jurado

Firma del Jurado

Ciudad., Fecha sustentación

DEDICATORIA

Dedico este trabajo a mi madre, quien es la que me ayudo a cumplir este proyecto de vida, a mi hermano quien me colaboro en los momentos difíciles y al ingeniero Daniel Cubillos, quien es mi mentor e inspiración para seguir creciendo como ingeniero y ahora como especialista.

AGRADECIMIENTOS

Agradezco primero a Dios por permitirme realizar esta etapa de mi vida académica, a mi señora madre, quien es la persona que me inspira para realizar este proyecto y continuar mi proyecto de vida adelante, a mi hermano por ayudarme cuando lo necesite, y a todas estas personas que me acompañaron durante este proceso, como es el Ingeniero Daniel Cubillos, quien fue mi mentor durante mi carrera.

CONTENIDO

Pág.

LISTA DE FIGURAS

Pág.

GLOSARIO

APLICACIÓN: Software informático que está diseñado para la realización de operaciones con funciones específicas dentro de equipos compatibles con este tipo de software.

INFORMÁTICA: Rama de la ingeniería que se encarga del estudio del hardware y software necesarios para el tratamiento de información, designando un conjunto de conocimientos teóricos y prácticos que son relativos en el área de la ciencia y de la tecnología, mediante sistemas informáticos o computadoras.

OWASP: metodología para la identificación de vulnerabilidades dentro de aplicaciones web, desarrollado de forma libre y sus siglas significa Open Web Application Security Project, y promueve el desarrollo de software seguro en el ámbito del Backend

SEGURIDAD: la seguridad es el proceso de prevención y detección de acceso o uso no autorizado de un sistema, esto previene la inserción de software malicioso dentro de las organizaciones con fines perjudiciales dentro de las organizaciones.

TIC´S: Las tecnologías de información y comunicación son el conjunto de herramientas relacionadas con la transmisión, procesamiento y almacenamiento digitalizado de la información.

WEB: Palabra inglesa que significa red o telaraña. Se designa como “LA WEB”, al sistema de gestión de información más popular para la transmisión de datos a través de internet.

RESUMEN

El presente trabajo se abordará en la metodología OWASP, como elemento estratégico para la identificación de vulnerabilidades y su posterior validación de seguridad en el diseño, programación y operación de aplicaciones web seguras dentro de las organizaciones dedicadas al desarrollo de software en Colombia.

Esto se logra con el estudio y entendimiento de la metodología para realizar una auditoría y posterior mejoramiento de las aplicaciones, adicional a ello estudiando las posibles vulnerabilidades que se puedan presentar dentro de ellas.

Palabras claves: Aplicación, Informática, OWASP, Seguridad, TIC´S, Web.

ABSTRACT

This work will be addressed in the OWASP methodology, as a strategic element for the identification of vulnerabilities and their subsequent security validation in the design, programming and operation of secure web applications within organizations dedicated to software development in Colombia.

This is achieved by studying and understanding this methodology for auditing and subsequently improving applications, in addition to this by studying the possible vulnerabilities that may arise within them.

Keywords: Application, Computing, OWASP, Security, ICT'S, Web

INTRODUCCIÓN

De la misma forma en que las innovaciones tecnológicas avanzan, y más en el ámbito de las aplicaciones web, como medio de transmisión y comunicación de la información para las organizaciones que las implementan para su constante comunicación y transferencia de información.

Recordemos que para las organizaciones la información que se manipulan y se trabajan dentro y fuera de ella es un activo de suma importancia, puesto que allí contienen información valiosa de sus clientes, usuarios y de la misma organización. Con esto presente dichas aplicaciones están mayormente expuestas a ataques de intrusos informáticos.

La mayoría de las organizaciones no prestan la atención debida a estos ataques, debido a que tampoco sus propios usuarios lo prestan, ya sea por ingenuidad o desconocimiento de los mismos. Por lo anterior sobresale la necesidad de contar con redes seguras.

Para ello se debe conocer cuáles son sus falencias de seguridad(vulnerabilidades) y sus posibles salvaguardas que permitan mejorar el nivel de seguridad de las aplicaciones, estas detenciones se logran por medio de herramientas que escanean dichas aplicaciones para encontrar sus puntos débiles de seguridad.

La seguridad informática establece mecanismos mínimos y necesarios de seguridad para las organizaciones, estableciendo estrategias, políticas, procedimientos y aplicación de medidas de manera preventiva para salvaguardas la información que se maneja dentro de las organizaciones[1].

En este trabajo se estudiara la metodología OWASP como elemento estratégico para la identificación de vulnerabilidades en aplicaciones web o sitios web, cuya finalidad es ver como esta metodología nos permite realizar auditorías dentro de estos sistemas de manera práctica y sencilla, esto a su vez nos lleva a que tengamos conocimientos de los lenguajes de programación enfocados en esta área del desarrollo web para entender más fácilmente las acciones y funciones tanto de OWASP como de las mismas aplicaciones[2].

[1] **Marcela, Caucali Beltrán Diana. 2020.** Repositoriy.unad.edu.co. [En línea] 2020. https://repository.unad.edu.co/bitstream/handle/10596/38709/dmcaucalib.pdf?sequence=1&isAllowed=y.

[2] **Edwin, Melgarejo Martinez. 2018.** repositorio unad. [En línea] 2018. [Citado el: 18 de Mayo de 2021.] https://repository.unad.edu.co/bitstream/handle/10596/21360/80125726.pdf?sequence=1&isAllowed=y

1 DEFINICIÓN DEL PROBLEMA

Actualmente el uso de aplicaciones web ha crecido de manera exponencial, causando un impacto significativo en el uso de las mismas y del acceso a internet, puesto que se ha convertido en uno de los mayores medios de transmisión de información y de servicios, causando que los desarrolladores avances en el mejoramiento de la seguridad de la información, pero así mismo como los desarrolladores mejoran ese aspecto los ciberdelincuentes también mejoran en su desarrollo para explotar fallas que por x o y motivo los desarrolladores de las aplicaciones web no toman en cuenta o se les pasa por alto, adicional a ello las organizaciones por la carrera de tener una aplicación web funcional, llamativa, responsiva y multiplataformas, han dejado a un lado la seguridad de las mismas causando que estas aplicaciones tenga muchas más falencias de explotación de vulnerabilidades por cuenta de los atacantes[3].

las infiltraciones a los datos y las aplicaciones web donde alojamos la información, por parte de los ciberdelincuentes han crecido de gran manera volviéndose uno de los principales problemas en la seguridad de las aplicaciones web y de la información que se guarda en ellas.

Debido a ello las aplicaciones web son el medio preferido por los delincuentes informáticos debido a que estas aplicaciones contienen acceso a la gran mayoría de las infraestructuras organizacionales, por ello la gran mayoría de las organizaciones optan por tecnologías de seguridad basadas en software como antivirus, cortafuegos y sistemas de detección de intrusos.

El método para lograr mitigar estos problemas son las pruebas de penetración, también conocidas como Pentesting o Hacking ético, estas prácticas son reconocidas a nivel internacional como pruebas de testeo de seguridad, esto con el fin de medir la posibilidad de evadir las defensas y acceder a la infraestructura interna y datos de almacenamiento dentro de las aplicaciones web[4].

No cabe duda que la seguridad de estas aplicaciones web es el auge actual, causando que en muchas ocasiones estas aplicaciones tengan diversas restricciones de acceso a los usuarios, impidiendo el libre acceso a la información no autorizada de las organizaciones o usuarios, en donde se crean mecanismos de salvaguardas para mejorar la seguridad de la aplicaciones, adicional a ello esto

3 **Girald, Luis Fernando Garcés, Sepúlveda Aguirre, Jovany Arley y Melguizo Múnera, Daniela. 2020.** americana.edu.co. [En línea] 2020. https://americana.edu.co/medellin/wp-content/uploads/2020/12/Pra%CC%81cticas-y-resultados-en-formacio%CC%81n-investigativa.-Semilleros-de-investigacio%CC%81n-generando-conocimiento-completo.pdf#page=272.

4 **Brito, Henry Raúl Gonzáles y Montesino Perurena, Raydel. 2018.** http://scielo.sld.cu/. [En línea] 2018. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992018000400005.

también genera que actualmente el estudio de la realización de Pentesting este en auge debido a que los creadores o desarrolladores de las aplicaciones web quieran identificar y conocer cuáles son las posibles falencias que pueden tener las aplicaciones que ellos crean, es decir tanto la parte de desarrollo como la parte de la seguridad web está muy correlacionado entre sí para dar una aplicación final con mejores estándares de calidad y de seguridad[5].

1.1 FORMULACIÓN DEL PROBLEMA

De lo anterior surge la siguiente pregunta:

¿Cómo OWASP puede ser un apoyo estratégico en la detección de vulnerabilidades y en aseguramiento de aplicaciones web dentro de las organizaciones desarrolladoras de SW?

[5] **Girald, Luis Fernando Garcés, Sepúlveda Aguirre, Jovany Arley y Melguizo Múnera, Daniela. 2020.** americana.edu.co. [En línea] 2020. https://americana.edu.co/medellin/wp-content/uploads/2020/12/Pra%CC%81cticas-y-resultados-en-formacio%CC%81n-investigativa.-Semilleros-de-investigacio%CC%81n-generando-conocimiento-completo.pdf#page=272.

2 JUSTIFICACIÓN

Gracias a este auge de las aplicaciones web se vuelve importante y obligatorio la protección de las mismas sobre todo de la información tanto personal como empresarial que se guarda en estas aplicaciones web, ya que por ser una plataforma de alojamiento de grandes cantidades de información y de intercambio de la misma, los ciberdelincuentes estarán siempre a la espera de encontrar cualquier vulnerabilidad debido a fallas informáticas para extraer esta información y hacer delitos como lo son: suplantación, robo de activos, malversación de activos, y demás delitos cibernéticos que se encuentran actualmente, recordando que uno de los preferidos en la actualidad para las organizaciones es el secuestro y extorsión de la información.

Como se aclarara anteriormente las aplicaciones web son un medio de alojamiento y transferencia de información dentro de una organización y estas aplicaciones son utilizadas desde los empleados, supervisores, encargados y gerentes, por poner un caso de ejemplo el envío masivo de correspondencia electrónica de un punto A un punto B, dentro de una misma organización ya sea por Hotmail, o Gmail que son las aplicaciones más comúnmente utilizadas para transferir información, en ese proceso los delincuentes informáticos aprovechan para instalar software malicioso en donde pueden acceder a sus cuentas o el secuestro de las mismas, esto se debe a la falta de mejoramiento de la seguridad tanto de los equipos informáticos como de la misma infraestructura informática, causando daños y "huecos" de la seguridad de la información[6].

Con base en lo anteriormente dicho cabe aclarar que en la actualidad dichas organizaciones presentan múltiples fallas de la seguridad de la información, que día tras día, desmejoran su rendimiento, debido a que desconocen las vulnerabilidades que tiene dentro de las mismas aplicaciones que utilizan para el manejo de la información[7]

En la actualidad el problema no tener un software inseguro es el mayor de los retos para los desarrolladores, con OWASP se brinda a las organizaciones unos estándares o planes de mejoras para las aplicaciones que trabajan y funcionan dentro de las mismas, mostrando un nivel de madurez con el que cuenta sus tecnologías informáticas[8].

---

[6] **Edgardo, Bernardis, y otros. 2017.** [En línea] Abril de 2017. http://sedici.unlp.edu.ar/handle/10915/62726.

[7] **David, Múnera Álvarez Jesús y Uribe Arango, Christian David. 2020.** dspace.tdea.edu.co. [En línea] 28 de Agosto de 2020. https://dspace.tdea.edu.co/handle/tdea/1081.

[8] **Fernández Mahecha, Edwin Neyid y Llano Ruiz, Anderson Julian. 2018.** [En línea] Junio de 2018. https://repository.ucatolica.edu.co/bitstream/10983/16045/1/TrabajoDeGA.

3 OBJETIVOS

3.1 OBJETIVO GENERAL

- Analizar la metodología OWASP como elemento estratégico en la identificación de vulnerabilidades tomando como referencia el Estándar de Verificación de Seguridad en Aplicaciones (ASVS) en el diseño, programación y operación de aplicaciones seguras en las organizaciones desarrolladoras de software en Colombia.

3.2 OBJETIVOS ESPECÍFICOS

- Estructurar la información sobre el panorama actual de la aplicabilidad de OWASP en el aseguramiento de aplicaciones web desde su desarrollo hasta su utilización.

- Establecer los tipos de riesgos más críticos y los controles de seguridad que deben contemplarse a la hora de realizar tareas de auditoria en las aplicaciones web basado en OWASP

- Proponer las mejores prácticas usando el estándar de Verificación de Seguridad en Aplicaciones de OWASP y los requisitos de seguridad más importantes para el desarrollo de proyectos de SW y sus entornos.

4 MARCO REFERENCIAL

4.1 MARCO TEÓRICO

La evolución que ha tenido las TIC´S, y el uso frecuente de las aplicaciones web, ha concentrado que la mayor transferencia de información se realice por medio de estas plataformas web, al tener mayor uso de estos aplicativos, crece la necesidad de realizar el estudio de las vulnerabilidades más frecuentes que se puedan encontrar dentro de las mismas, por tanto es de suma importancia conocer qué tipo de información necesita mucha más protección que otra es decir un mayor control de toda la información que se agregue a las aplicaciones web[9].

Por lo anterior mencionado es importante realizar pruebas de pentesting antes de producir la app a mayor escala o en etapa de funcionalidad total, esto se hace necesario para poder tener presente cuales son las falencias que se tienen en el aplicativo ayudando a ver qué nivel de seguridad se tiene, esto con el fin de poder llevar el aplicativo a la parte de producción.

Actualmente las amanezcas de atacantes cibernéticos son más recurrentes hoy en día, y las organizaciones están con el mayor grado de probabilidad de ser atacados por los ciberdelincuentes, dichos ataques van desde la inyección de software maliciosos como lo son la encriptación de la información y posterior secuestro también conocido como ataques de Ransoware, hasta la suplantación de identidad conocida como ingeniería social para beneficios propios, para lo cual se requiere mayor implementación de controles que gestión enfocada en las seguridad de la información[10].

La seguridad de la información es asociada a la gestión de TIC, teniendo como propósito mantener los estándares de seguridad aceptables de riesgo[11]
El proyecto OWASP tiene una comunidad libre y abierta a nivel mundial, donde se

[9] **Nemury, Silega Martínez y García Rodríguez, Ana Marys. 2018.** repositorio.uci.cu. [En línea] 2018. https://repositorio.uci.cu/handle/123456789/7916.

[10] **WELIVESECURITY BY ESET. 2015.** [En línea] 25 de Febrero de 2015. https://www.welivesecurity.com/la-es/2015/02/25/que-es-un-0-day/.

[11] **Javier, Valencia Duque Francisco y Orozco Alzate, Mauricio. 2017.** [En línea] 1 de Marzo de 2017. https://www.researchgate.net/profile/Francisco-Valencia-Duque/publication/318204876_A_methodology_for_implementing_an_information_security_management_system_based_on_the_family_of_ISOIEC_27000_standards/links/5fd9d2ea299bf1408811f7b3/A-methodology-for-imp.

enfocan en el mejoramiento de la seguridad del software en aplicaciones web o sitios web, contienen diversas herramientas, documentos, y foros donde solo se habla del tema, el objetivo principal de OWASP es como se menciona anteriormente el mejoramiento de la seguridad, pero también la recopilación de diversas técnicas de extracción de información para hacer pruebas de penetración[12].

OWASP tiene dos fases, las cuales se conocen como pasivas y activas, cuyo enfoque es la famosa "caja negra" partiendo de esto se puede decir que por lo general ninguna aplicación web es complementa mente segura y que esté libre de ataques de seguridad, OWASP permite mejorar de cierto modo el nivel de seguridad de estas aplicaciones[13].

La metodología OWASP nos permite ver de una manera fácil y práctica la forma de hacer una auditoria en aplicaciones web, en cuanto a la parte de comprobación y pruebas de seguridad OWASP permite realizarla de forma confiable y segura.

OWASP contiene un framework que es utilizado libremente el cual sirve como punto de referencia donde parte de actividades y tareas específicas a realizar sobre todas las fases del ciclo del desarrollo del software a desplegar[14].

Hoy en día el monitoreo permanente de las aplicaciones en el ámbito de la seguridad de las mismas demuestra que es una parte clave al hablar de seguridad para cualquier organización brindando mayor confianza en el software que se crea y se implementa[15].

Este monitoreo tiene como características los siguientes aspectos:

1. La integridad de los datos: la cual se basa en que cualquier modificación que se haga sobre los datos debe conocerse y ser autorizada por la organización.
2. Disponibilidad de los sistemas: permite tener la productividad de la organización y así mismo permite tener mayor credibilidad de la misma.

[12] **Muñoz Mayorga, Andrés Felipe y Pérez Solarte, Santiago Alejandro. 2017.** Repositorio.unicauca.edu.co. [En línea] Septiembre de 2017. http://repositorio.unicauca.edu.co:8080/bitstream/handle/123456789/1773/PENTESTING%20SOBRE%20APLICACIONES%20WEB%20BASADO%20EN%20LA%20METODOLOG%c3%8dA%20OWASP%20UTILIZANDO%20SBC%20DE%20BAJO%20COSTO.pdf?sequence=1&isAllowed=y.

[13] **Rendón Tacle, Jean Carlos y Raza Rivas, José Steven. 2019.** [En línea] 30 de Agosto de 2019. http://repositorio.ug.edu.ec/handle/redug/45164.

[14] **Muñoz Mayorga, Andrés Felipe y Pérez Solarte, Santiago Alejandro. 2017.** Repositorio.unicauca.edu.co. [En línea] Septiembre de 2017. http://repositorio.unicauca.edu.co:8080/bitstream/handle/123456789/1773/PENTESTING%20SOBRE%20APLICACIONES%20WEB%20BASADO%20EN%20LA%20METODOLOG%c3%8dA%20OWASP%20UTILIZANDO%20SBC%20DE%20BAJO%20COSTO.pdf?sequence=1&isAllowed=y.

[15] **Edwin, Melgarejo Martinez. 2018.** repositorio unad. [En línea] 2018. [Citado el: 18 de Mayo de 2021.] https://repository.unad.edu.co/bitstream/handle/10596/21360/80125726.pdf?sequence=1&isAllowed=y.

3. Confidencialidad: toda información divulgada debe ser autorizadas y protegida contra posibles ataques que violen el principio de confidencialidad[16].

4.2 MARCO CONCEPTUAL

La seguridad informática está basada en políticas y normas tanto internas como externas de una empresa, permitiendo proteger la integridad de la información que se encuentre en la empresa, mitigando amenazas y riesgos tanto físicos como lógicos que se encuentren expuestos[17].

El pentesting es la metodología implementada para la identificación de las vulnerabilidades de los sistemas informáticos, o de una aplicación web. Esta metodología es diseñada para la clasificación y medición de los alcances de los fallos de seguridad, en donde se puede realizar la evaluación de dichas fallas para así mismo realizar el mejoramiento de seguridad requerido[18].

Kali Linux es una distribución basada en Debian, el cual permite realizar la práctica de Pentesting, encargada de analizar y búsqueda de vulnerabilidades informáticas en los sistemas en este caso lo utilizaremos para la identificación de vulnerabilidades de las aplicaciones web[19].

Vulnerabilidades informáticas, se conoce una vulnerabilidad informática a una falla o debilidad dentro de un sistema informático o equipo informático, en la que puede ser explotada por ciberdelincuentes obteniendo acceso no autorizados o la realización de acciones no autorizadas para perjudicar a las organizaciones o a las personas[20].

Las aplicaciones web funcionan con la ayuda de un servidor web, en la que usuarios pueden acceder por medio de programas llamados navegadores web conectados a

[16] **Rendón Tacle, Jean Carlos y Raza Rivas, José Steven. 2019.** [En línea] 30 de Agosto de 2019. http://repositorio.ug.edu.ec/handle/redug/45164.

[17] **Rendón Tacle, Jean Carlos y Raza Rivas, José Steven. 2019.** [En línea] 30 de Agosto de 2019. http://repositorio.ug.edu.ec/handle/redug/45164.

[18] **Yucenid, Vanegas Romero Alfonso. 2019.** repository.unipiloto.edu.co. [En línea] 2019.

[19] **Andrés, Rubén. 2016.** [En línea] 2016. https://computerhoy.com/paso-a-paso/software/que-es-kali-linux-que-puedes-hacer-41671.

[20] **Angel, Eulises Ortiz. 2020.** [En línea] 22 de Junio de 2020. https://www.hostdime.com.pe/blog/que-es-una-vulnerabilidad-en-seguridad-informatica-ejemplos/.

internet, un ejemplo de navegador web es Google Chrome, en estas aplicaciones web las organizaciones suben su información valiosa para que se pueda acceder a ella desde cualquier lugar del mundo, esto con el fin de transportar dicha información de manera más eficaz y más rápida, con lo anterior se infiere que estas aplicaciones están basadas en una arquitectura cliente/servidor[21].

Así mismo las aplicaciones web han evolucionado causando así que también los ciberdelincuentes lo hagan ya que al momento de desarrollar una aplicación web no se le da la importancia suficiente a la seguridad de las mismas, es por esto que se hace necesario aplicar una metodología que permita mejorar esta falencia y unas de las metodologías más utilizadas y de más reconocimiento es OWASP[22].

OWASP busca realizar pruebas de seguridad en los sitios y aplicaciones web para la identificación de vulnerabilidades que se llegaran a presentar en este tipo de sitios web, a esta metodología se le conoce como código abierto en la cual permite delimitar y mejorar la seguridad de estas aplicaciones[23].

OWASP además propone unas métricas o preguntas que sirven para establecer el nivel de impacto de las amenazas, lo cual permite que esta metodología sea mucho más eficiente ya que permite identificar los recursos necesarios para el mejoramiento de las aplicaciones web o sitios web[24].

Esta metodología indica los riesgos principales de la seguridad en aplicaciones web, permitiendo identificar las vulnerabilidades más comunes y esto a su vez nos proporciona el nivel de impacto que pueden tener estas vulnerabilidades para encontrar el mejor plan para evitar tener ataques de las misma[25].

Las auditorias hoy en día son fundamentales ya que permite realizar un seguimiento el cual ayuda a disminuir las falencias de seguridad en las aplicaciones web, para garantizar que la seguridad de la información y datos alojados en dichas aplicaciones[26].

---

[21] **Desarrollo Web. 2020.** [En línea] 24 de Agosto de 2020. https://profile.es/blog/desarrollo-aplicaciones-web/.

[22] **CHAVARRIA GONZALEZ, Victor. 2018.** [En línea] 2018. Disponible en: https://dspace.uib.es/xmlui/handle/11201/151259?show=full.

[23] **Marcela, Caucali Beltrán Diana. 2020.** Repositoriy.unad.edu.co. [En línea] 2020. https://repository.unad.edu.co/bitstream/handle/10596/38709/dmcaucalib.pdf?sequence=1&isAllowed=y.

[24] **Víctor, González Chavarría. 2018.** [En línea] 2018. https://dspace.uib.es/xmlui/bitstream/handle/11201/151259/Memoria_EPSU0643.pdf?sequence=1&isAllowed=y.

[25] **López Sevilla, Galo Mauricio y Gambia Safla, Diego Leonardo. 2021.** [En línea] 2021. https://repositorio.pucesa.edu.ec/handle/123456789/3175.

[26] **EDWIN, Melgarejo Martinez. 2018.** [En línea] 2018. Disponible en: https://repository.unad.edu.co/bitstream/handle/10596/21360/80125726.pdf?sequence=1&isAllowed=y.

4.3 MARCO LEGAL

4.3.1 Constitución Política de 1991.

- Artículo 209. La función administrativa está al servicio de los intereses generales y se desarrolla con fundamento en los principios de igualdad, moralidad, eficacia, economía, celeridad, imparcialidad y publicidad, mediante la descentralización, la delegación y la desconcentración de funciones. Las autoridades administrativas deben coordinar sus actuaciones para el adecuado cumplimiento de los fines del Estado. La administración pública, en todos sus órdenes, tendrá un control interno que se ejercerá en los términos que señale la ley[27].

- Artículo 269. En las entidades públicas, las autoridades correspondientes están obligadas a diseñar y aplicar, según la naturaleza de sus funciones, métodos y procedimientos de control interno, de conformidad con lo que disponga la ley, la cual podrá establecer excepciones y autorizar la contratación de dichos servicios con empresas privadas colombianas[28].

Leyes informáticas en Colombia.

- Ley estatutaria 1266 de 2008. Por la cual se dictan las disposiciones generales del hábeas data y se regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia, comercial, de servicios y la proveniente de terceros países y se dictan otras disposiciones[29].
- “Ley 1273 del 5 de enero de 2009. Delitos informáticos: se crea un nuevo bien jurídico tutelado - denominado "de la protección de la información y de los datos" y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones[30].
- Artículo 269C. Interceptación de datos informáticos: El que, sin orden judicial previa intercepte datos informáticos en su origen, destino o en el interior de un sistema informático, o las emisiones electromagnéticas provenientes de

---

[27] **Constitución Política de Colombia.** constitucioncolombia.com. [En línea] https://www.constitucioncolombia.com/titulo-7/capitulo-5/articulo-209.

[28] **Constitucón Política de Colombia.** constitucioncolombia.com. [En línea] https://www.constitucioncolombia.com/titulo-10/capitulo-1/articulo-269.

[29] **LEY ESTATUTARIA 1266 DE 2008. 2008.** [En línea] Diciembre de 2008. https://tic.bogota.gov.co/node/137.

[30] **LEY ESTATUTARIA 1266 DE 2008. 2008.** [En línea] Diciembre de 2008. https://tic.bogota.gov.co/node/137.

un sistema informático que los trasporte incurrirá en pena de prisión de treinta y seis (36) a setenta y dos (72) meses[31].

- Artículo 269D. Daño informático: El que, sin estar facultado para ello, destruya, dañe, borre, deteriore, altere o suprima datos informáticos, o un sistema de tratamiento de información o sus partes o componentes lógicos, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 salarios mínimos legales mensuales vigentes[32].
- Artículo 269G. Suplantación de sitios web para capturar datos personales: El que con objeto ilícito y sin estar facultado para ello, diseñe, desarrolle, trafique, venda, ejecute, programe o envíe páginas electrónicas, enlaces o ventanas emergentes, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 salarios mínimos legales mensuales vigentes, siempre que la conducta no constituya delito sancionado con pena más grave[33].
- En la misma sanción incurrirá el que modifique el sistema de resolución de nombres de dominio, de tal manera que haga entrar al usuario a una IP diferente en la creencia de que acceda a su banco o a otro sitio personal o de confianza, siempre que la conducta no constituya delito sancionado con pena más grave. la pena señalada en los dos incisos anteriores se agravará de una tercera parte a la mitad, si para consumarlo el agente ha reclutado víctimas en la cadena del delito"[34].

[31] **Superintendencia de insdustria y comercio. 2009.** Sic.gov.co. [En línea] 5 de Enero de 2009. [Citado el: 18 de Mayo de 2021.] https://www.sic.gov.co/recursos_user/documentos/normatividad/Ley_1273_2009.pdf.

[32] **Superintendencia de insdustria y comercio. 2009.** Sic.gov.co. [En línea] 5 de Enero de 2009. [Citado el: 18 de Mayo de 2021.] https://www.sic.gov.co/recursos_user/documentos/normatividad/Ley_1273_2009.pdf.

[33] **Superintendencia de insdustria y comercio. 2009.** Sic.gov.co. [En línea] 5 de Enero de 2009. [Citado el: 18 de Mayo de 2021.] https://www.sic.gov.co/recursos_user/documentos/normatividad/Ley_1273_2009.pdf.

[34] **Superintendencia de insdustria y comercio. 2009.** Sic.gov.co. [En línea] 5 de Enero de 2009. [Citado el: 18 de Mayo de 2021.] https://www.sic.gov.co/recursos_user/documentos/normatividad/Ley_1273_2009.pdf.

5 DESARROLLO DE LOS OBJETIVOS

5.1 PANORAMA ACTUAL DE LA APLICABILIDAD DE OWASP EN EL ASEGURAMIENTO DE APLICACIONES WEB DESDE SU DESARROLLO HASTA SU UTILIZACIÓN.

Antes de empezar a hablar de OWASP para la seguridad en aplicaciones web, debemos saber ¿Cómo se define una aplicación web? Teniendo claro este concepto podemos saber el ¿Por qué se utiliza OWASP?

OWASP ASVS es el proyecto de verificación de seguridad en aplicaciones de entornos web, cuyo objetivo es la revisión segura del código implementado en el desarrollo de estas aplicaciones web, implementando estándares vigentes en el sector del desarrollo.

Una aplicación web es: un tipo de software orientando a la web, basado en lenguajes de programación como lo son HTML, CSS, JavaScript, PHP, y demás lenguajes de programación implementado para la creación de aplicaciones web, esta aplicación se ejecuta por medio de un navegador web, esto nos permite inferir que la estructura básica de la aplicación es, servidor web – navegador = página/ aplicación web.

Las aplicaciones web son aplicadas en diversos entornos como lo son: tiendas online, web logs, wikis, web mail[35].

La metodología Open Web Application Security Project (OWASP) es la metodología implementada para la toma de decisiones en cuanto a la seguridad de aplicaciones web, ya que esta permite evidenciar vulnerabilidades existentes dentro de las aplicaciones y sus riesgos, esta metodólogo presenta una secuencia de instrucciones organizadas que abarca distintas áreas de las vulnerabilidad en las aplicaciones esto lo realiza de manera sistémica dentro de las mismas, donde analiza el porcentaje de riesgo que tienen las organizaciones, esto a su vez facilita a los especialistas en seguridad la realización de auditorías de los vectores que tengan mayor incidencia de ser atacados por los ciberdelincuentes.

[35] **Rojas Osorio, Jorge Armando. 2018.** [En línea] 2018. [Citado el: 15 de Mayo de 2021.] http://repository.unipiloto.edu.co/handle/20.500.12277/8654.

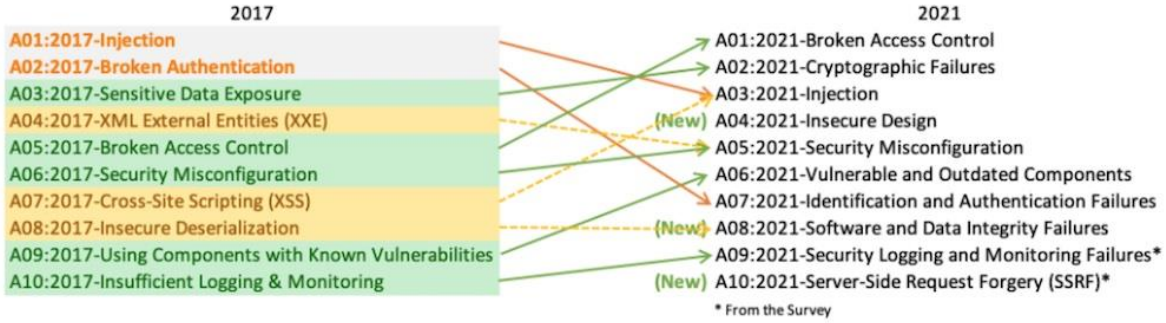
OWASP permite extraer información para la realización de informes de las vulnerabilidades presentes dentro de las aplicaciones web, en lo cual permite emplearlas al momento de realizar una auditoría para posteriormente realizar los respectivos ajustes de seguridad de las aplicaciones con el fin de mejorar la seguridad de las mismas[36].

La metodología OWASP se basa en el top ten de vulnerabilidades para la realización de Pentesting centrado en el mejoramiento de la seguridad de sitios web.

OWASP implemente una lista de vulnerabilidades conocidas como el Top Ten basándose en la ciberseguridad más comúnmente que hay en la actualidad, este top está diseñado para explicar las vulnerabilidades que los desarrolladores deben tener presente al desarrollar las aplicaciones web o sitios web.

Los procesos de desarrollo de estas aplicaciones o sitios web seguros son importantes, pero a su vez también es de suma importancia realizar el registro y la supervisión de eventos dentro de las mismas, para detectar anomalías que se puedan prestar, al no realizar este proceso las aplicaciones se van con el tiempo volviéndose menos seguras[37].
Este top ten se refleja en la siguiente imagen.

Ilustración 1. OWASP 2017 CON OWASP 2021



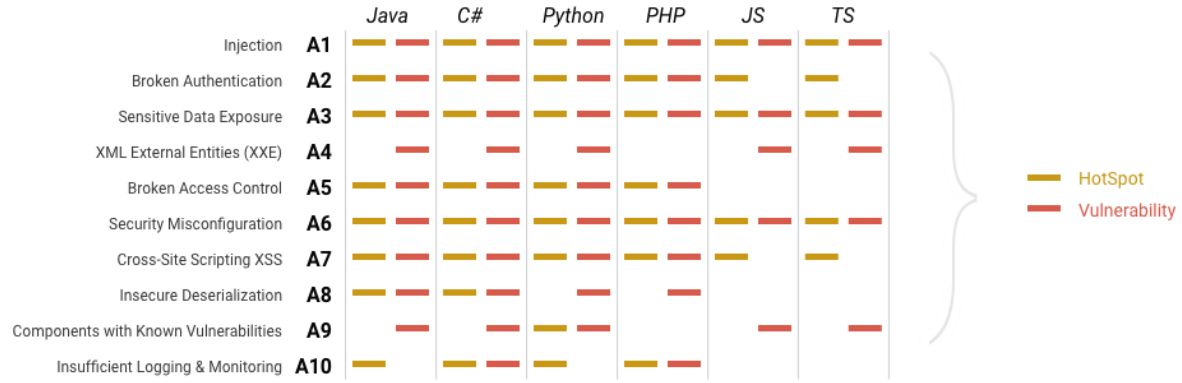
Fuente 1: Helpnetsecurity. [sitio web]. OWASP Top 10 2021: Los riesgos de seguridad de aplicaciones web más graves. [consulta: 03 de octubre de 2021]. Disponible en: https://www.helpnetsecurity.com/2021/09/24/owasp-top-10-2021/

[36] **Moisés, Delgado Basurto Jonathan. 2020.** [En línea] Enero de 2020. https://repositorio.uleam.edu.ec/handle/123456789/2068.

[37] **M., Elhadi, Shakshuki y Ansar Yasar. 2019.** sciencedirect.com. [En línea] Noviembre de 2019. https://www.sciencedirect.com/science/article/pii/S1877050920323589.

Para tener un panorama más claro del uso de OWASP para los diferentes lenguajes de programación y medición de puntos de acceso / vulnerabilidades se refleja en la siguiente imagen.

Ilustración 2 OWASP EN LENGUAJES DE PROGRAMACIÓN



FUENTE 1 SonarQube. 2021. https://www.sonarqube.org/features/security/owasp/?gads_campaign=South-America-OWASP&gads_ad_group=OWASP&gads_keyword=owasp%20top10&gclid=Cj0KCQjwwY-LBhD6ARIsACvT72NujlizGy5TmtBk825YCLcjoaNlIN16KXxB3MwfXMkH2bl8S3BpEflaAvtHEALw_wcB.

Función de la metodología OWASP:

El principal objetivo de esta metodología o herramienta para la identificación de vulnerabilidades para el mejoramiento de la seguridad en aplicaciones o sitios web, es precisamente la detección de anomalías o fallas de seguridad existentes dentro de las mismas aplicaciones o sitios web, realizando un análisis para la evaluación de dichas fallas.

Las características generales de la metodología OWASP tienen el fundamento de la búsqueda de riesgos informáticos, buscando el objetivo de ayudar para el mejoramiento del nivel de seguridad en la web, adicional a ello cuenta con una gran comunidad de desarrolladores de código abierto, para el soporte y apoyo desde la plataforma GitHub, cuenta con una aplicación Juice Shop para banco de pruebas para la detección de vulnerabilidades.

¿Para qué sirve realizas pruebas con la metodología OWASP?

Una prueba OWASP realiza pruebas de verificación y corrección de falencias de (bugs) de seguridad dentro de las aplicaciones web o sitios web, el principal factor de estas pruebas es informar a las organizaciones o personas que implementen estas aplicaciones de los problemas de seguridad.

Ventajas y desventajas de la metodología OWASP.

Ventajas:

- Puede ser aplicadas a todo tipo de aplicaciones web o sitios web.
- Ayuda a fomentar el trabajo en equipo para el mejoramiento de la seguridad.
- Es flexible.

Desventajas:

- Requiere conocimientos, para ser más efectivo.
- Requiere mucho tiempo para la realización de pruebas[38].

El modelo de pruebas para la metodología OWASP se divide en:

- Auditor: son todas aquellas personas que van a realizar las pruebas de comprobación de seguridad de las aplicaciones.
- Herramientas y metodologías: se trata de las herramientas que se van a implementar para la realización de la auditoría junto con la metodología a implementar en este caso OWASP.
- Aplicación: más comúnmente conocida como la caja negra o banco de pruebas.

Estas pruebas se dividen en dos fases las cuales corresponden en:

1. Modo pasivo: Las personas que realizan la auditoría se colocan en la tarea de entender la lógica de la aplicación a evaluar, interactúan con ella y recopilan toda la información necesaria como lo es el proxy HTTP, esto se realiza con el fin de comprender y conocer cuáles son los puntos de acceso.
2. Modo activo: Aquí es donde se aplica la información recolectada anteriormente con la ayuda de herramientas de pentesting las cuales hacen pruebas de gestión de la configuración, pruebas lógicas de la aplicación, pruebas de autenticación y de acceso a sesiones, validación de los datos alojados en la base de datos, pruebas con el servidor y pruebas AJAX[39].

La metodología OWASP, es preferida para realizar pruebas y auditorías de seguridad debido a que OWASP presenta una mayor seguridad de la información, mayor seguridad de los procesos, mayor seguridad en las tecnologías de internet, mejor seguridad en las comunicaciones, mayor estabilidad en la seguridad inalámbrica y seguridad física[40].

---

[38] **Tatiana, Tapia Bastidas y Rodriguez Zambrano, Stalyn Fabrico. 2021.** [En línea] 22 de Febrero de 2021. http://repositorio.itb.edu.ec/handle/123456789/2679.

[39] **e Creative Commons.** OWASP.ORG. [En línea] https://owasp.org/www-pdf-archive/Gu%C3%ADa_de_pruebas_de_OWASP_ver_3.0.pdf.

[40] **Altamirano Ruiz, Marco Vinicio y Bonilla Vaca, Carolina Anabel. 2017.** Repositorio Universidad Técmica de Ambato. [En línea] 2017. http://repositorio.uta.edu.ec/handle/123456789/24534.

5.2 ASEGURAMIENTO CON LA METODOLOGÍA OWASP

Esta metodología analiza todas las amenazas desdés dos puntos los cuales son: la probabilidad en que se pueda ocurrir y la segunda es el nivel de impacto que pueda producir dicha falencia o vulnerabilidad encontrada; estos dos puntos de vista se subdividen como lo muestra a continuación.

OWASP 2017 tiene una lista de vulnerabilidades de recursos tomados de la ciberseguridad más conocidos y más aplicados, la lista está diseñada para mejorar la seguridad de las aplicaciones web o sitios web, en esta lista se encuentran las siguientes vulnerabilidades.

1. Ataques de inyección.
2. Ataques de autenticación Broken.
3. Ataques de exposición sensible de datos.
4. Entidades externas XML (XXE).
5. Ataques de control de acceso remoto.
6. Configuración incorrecta de la seguridad.
7. Ataques de scripting entre sitios (XXS).
8. Deserialización insegura.
9. Uso de componentes con vulnerabilidades conocidas.
10. Registro y monitoreos insuficientes[41]."

A1. Inyección: consiste en realizar un ataque de inyección SQL, LDAP, código SSI, este ataque de instrucciones de forma maliciosa que manipula las bases de datos dentro de las aplicaciones o sitios web.

A2. Ataques de autenticación Broken: está perdida de autenticación es relacionada a usuarios/contraseñas/cuentas administrativas, en donde realiza ataques de fuerza bruta con el fin de explotar y atacar la identidad de los usuarios y sus contraseñas en las aplicaciones web logrando suplantar la identidad de un usuario, de un sitio web.

A3. Exposición de datos sensibles: la exposición de los datos hace referencia a evitar el robo de contraseñas.

A4. Entidades externas XML (XXE): con las identidades externas se logra revelar archivos internos mediante la URI o archivos internos que se encuentran dentro de servidores con permisos no autorizados, escanea puertos LAN para ejecutar

[41] **M., Elhadi, Shakshuki y Ansar Yasar. 2019.** sciencedirect.com. [En línea] Noviembre de 2019. https://www.sciencedirect.com/science/article/pii/S1877050920323589.

código de forma remota para la realización de ataques que generan denegación de servicios conocidos como (DoS).

A5. Ataque de control de acceso: administrador de aplicaciones que relacionan derechos de acceso, modificaciones de datos y de permisos para usuarios ejecutando explotación de manera maliciosa.

A6. Configuración de seguridad incorrecta: se refiere a las actualización y configuración de aplicaciones.

A7. Secuencia de comandos de sitios cruzados: más conocido como un hueco de seguridad, este ataque es en la suplantación de sitios web, o aplicaciones para que los usuarios den información valiosa para luego realizar actos malintencionados.

A8. Deserialización insegura: ataque relacionado a la recepción de objetos seriados dañados, manipulando el sistema mediante inyecciones, privilegios o ejecución remota.

A9. Componentes vulnerables: filtración de ataques mediante componentes como bibliotecas.

A10. Registro y monitores insuficientes: hace referencia al monitoreo y el registro continuo de las aplicaciones, al no realizarse de manera continua se puede perder el control de las mismas logrando extraer información, destrucción de la misma.

Todo lo anteriormente mencionado hace referencia al top ten de la metodología OWASP[42].

Dentro de este top ten podemos establecer el siguiente borrador donde se expone las vulnerabilidades más críticas que se pueden encontrar en el desarrollo de aplicaciones web o páginas web

- Interfaz web segura: Busca la evaluación de HTTPS, para el aseguramiento de la información transmitida, adicional a ello busca evaluar la interfaz web.
- Autenticación o autorización insuficiente: Evalúa las contraseñas junto con el mecanismo de recuperación de contraseñas esto con el fin de mejorar dichas fallas.
- Servicios de red inseguros: Dar solución a los puertos de prueba inseguros, también dar solución a los servicios que no funcionan de manera correcta.
- Falta de cifrado de transporte: Principalmente da solución a los firewalls.

[42] **Marcela, Caucali Beltrán Diana. 2020.** Repositoriy.unad.edu.co. [En línea] 2020. https://repository.unad.edu.co/bitstream/handle/10596/38709/dmcaucalib.pdf?sequence=1&isAllowed=y.

- Problemas de privacidad: Dar mayor confianza a los usuarios finales en la información tanto suministrada como buscada, por medio de los criterios de aseguramiento de protección adecuados.
- Configuración de seguridad insuficiente: Evalúa las alertas y notificaciones de seguridad para dar solución a los eventos de inseguridad presentados.
- Software y Firmware inseguro: Busca el mejoramiento de las actualizaciones y archivos cifrados, adicional a ello valida los archivos antes de ser instalados.
- Mala seguridad física: Evalúa los dispositivos donde se instalar todo lo necesario para el correcto funcionamiento de las aplicaciones[43].

Esta metodología analiza todas las amenazas desdés dos puntos los cuales son: la probabilidad en que se pueda ocurrir y la segunda es el nivel de impacto que pueda producir dicha falencia o vulnerabilidad encontrada; estos dos puntos de vista se subdividen como lo muestra a continuación.

OWASP tiene en cuenta los vectores de ataque para asi mismo analizar las vulnerabilidades de las aplicaciones web, dichos vectores son: agentes de amenazas, vector de ataque, debilidad de seguridad, controles de seguridad, nivel de impacto técnicos e impactos a las organizaciones, adicional a ello la metodología OWASP hace el análisis partiendo del levantamiento de la información, el análisis de las vulnerabilidades encontradas, definición de los objetivos es decir cuales vulnerabilidades tiene mayor impacto para organizarlos y asi mismo empezar a mejorar y mitigar dichas vulnerabilidades, ataques sufridos y posibles ataques que se puedan sufrir y el análisis de los resultados teniendo presente la cantidad de veces que se ha sufrido el ataque como de las mejoras establecidas para mitigar dichos ataques[44].

Partiendo de lo anterior, en la actualidad OWASP 2021 ha modificado estos criterios teniendo presente las nuevas amenazas y de los principales factores de ataques más comunes, quedando como lista actual:

- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- A03:2021-Injection
- A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components

[43] **Marcela, Perez Sanches Laura. 2019.** Repositorio Universidad Francisco de Paula Santander. [En línea] 8 de Agosto de 2019. http://repositorio.ufpso.edu.co/jspui/handle/123456789/1022.

[44] **Rodríguez Rodríguez , Rafael Enrique y Sánchez, Andrés Felipe. 2018. [En línea] 2018. https://core.ac.uk/download/pdf/213560272.pdf.**

- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures
- A10:2021-Server-Side Request Forgery[45].

Con esto se genera tres nuevas etapas como lo son: diseño inseguro, fallas de integridad de datos y software, y por ultima etapa fallas de monitoreo y registro de seguridad.

En el año 2021 OWASP se centra en los riesgos que están relacionados con el diseño y las fallas arquitectónicas de uso modelado de amenazas y patrones de diseño seguro con arquitectura de referencia[46].

La lista OWASP 2021 Es sumamente beneficiosa para la evaluación de vulnerabilidades, los desarrolladores pueden basarse en el top ten para la definición de pautas seguras en su desarrollo garantizando la seguridad y estándares del desarrollo seguro[47].

6 RIESGOS Y CONTROLES DE SEGURIDAD QUE DEBEN CONTEMPLARSE A LA HORA DE REALIZAR TAREAS DE AUDITORIA EN LAS APLICACIONES WEB BASADO EN OWASP.

Los ciberdelincuentes al momento de encontrar falencias y poner en riesgo la seguridad de las aplicaciones usan diferentes rutas del sitio web, dichas rutas pueden tener o no tener consecuencias graves dentro de la aplicación y posterior a la organización colocando en riesgo la misma aplicación. OWASP con su top 10 de los riesgos más comunes dentro de las aplicaciones web nos da a conocer cuáles son y sus consecuencias para luego dar sus controles[48].

La evaluación de los riesgos y de los controles que se deben contemplar al momento de realizar la auditoria de las aplicaciones web basándonos en la metodología

[45] **Zorz, Zeljka. 2021. En línea] 2021 Disponible en: https://www.helpnetsecurity.com/2021/09/24/owasp-top-10-2021/.**

[46] **Zorz, Zeljka. 2021. En línea] 2021 Disponible en: https://www.helpnetsecurity.com/2021/09/24/owasp-top-10-2021/.**

[47] **Maury, Julien. 2021.[En línea] 2021 Disponible en: https://www.esecurityplanet.com/applications/owasp-list-gets-a-new-top-vulnerability/.**

[48] **HAMNER, BRIONES PINCAY GERSON y HERNANDEZ PEÑAHERRERA, ERIKA BELÉN. 2018.** [En línea] 2018. Disponible en: http://repositorio.ug.edu.ec/bitstream/redug/26837/1/B-CINT-PTG-N.249%20Briones%20Pincay%20Gerson%20Hamner.%20Hern%c3%a1ndez%20Pe%c3%b1a%20herrera%20Erika%20Bel%c3%a9n.pdf.

OWASP consiste en 6 etapas las cuales se deben realizar en el mismo orden para mayor eficacia en la evaluación de los riesgos y de controles, dichas etapas son:

1. La identificación de los riesgos. la identificación de los riesgos consiste en identificar los agentes de amenazas que puedan ser perjudiciales junto con el nivel de impacto si se llega a materializar la amenaza.
2. La estimación de factores de probabilidad: consiste en una vez encontrado todos los riesgos y vulnerabilidades de la aplicación se procede a estimar el porcentaje de probabilidad en que sea descubiertas y posteriormente explotada, esto se realiza de manera cualitativa combinado con los análisis cuantitativos.
3. Estimar factores de impacto: esto se realiza para la verificación del daño que puede ocasionar las vulnerabilidades encontradas en la aplicación.
4. Determinación de la severidad del riesgo: esta determinación se realiza con una matriz de severidad junto con la probabilidad y de impacto de la misma.
5. Priorizar los planes de acción: después de clasificar todos los riesgos se procede a realizar un listado de riesgos de mayor prioridad con los de menos, para así mismo dar una respuesta inmediata de acuerdo al orden de prioridad.
6. Personalizar el modelo de clasificación del riesgo: es primordial realizar esta etapa para la creación de un modelo de clasificación de los riesgos donde se tenga en cuenta los factores de riesgos, para hacer la ponderación de dichos factores.

De acuerdo a la metodología OWASP hablamos de los siguientes controles:

1. A.10.4.1. Controles contra software malicioso. Consiste en aplicar controles a los softwares maliciosos que por alguna razón se logra filtrar dentro de la aplicación.
2. A.10.6.1. Controles de red. Consiste en la implementación de controles de tráfico de la red, y esta a su vez del uso de la misma, para garantizar una red más confiable y seguridad para los usuarios y para el despliegue de la aplicación.
3. A.12.3.1. Políticas sobre el uso de controles criptográficos. Estas políticas se basan en el uso de software de criptografía para el traspaso de información de forma segura y confiable tanto para los usuarios como para la organización
4. A.12.4.1. Control de software operacional. En este control se habla de todos los softwares que se utilicen para la generación de aplicaciones tales pueden ser (Sistema operativo, editores de código).
5. A.12.4.3. Control de acceso al código fuente del programa. Al momento del desarrollo de las aplicaciones se ven involucrados personales de Front end y de Backend este personal son los encargados de llevar a cabo la aplicación por ende ellos tienen el acceso total del código fuente y son los responsables

de que dicho código no se filtre código malicioso y perjudicial para la organización.[49]

A la hora de resumir estos riesgos y estos controles se puede hablar del modelado de amenazas el cual nos dice que hay tres tipos dentro de este modelado los cuales corresponden en (Descomponer la aplicación – dar jerarquía a las amenazas de la aplicación – mitigar dichas amenazas).

- Descomponer la aplicación: se realiza una exploración y barrido de la aplicación, para estudiar y entenderla como se hace los procesos de relación y de interacción con los usuarios, al igualmente la entrada y salida de información, esto se realiza para hacer el proceso de categorizar y dar jerarquía a las mismas.
- Dar jerarquía a las amenazas de la aplicación: una vez realizado el anterior ítem se procese a dar la importancia de las amenazas que está expuesta la aplicación por medio de diferentes métodos y metodologías como lo es OWASP.
- Mitigar dichas amenazas: después de que se da jerarquía a las amenazas de la aplicación se procede a hacer una mitigación de las mimas por medio de acciones y herramientas que permitan realizar dicha mitigación y posteriormente mejoramiento de la seguridad[50].

También se debe tener en cuenta que cada desarrollador debe conocer las políticas de seguridad en caso tal de que los usuarios requieran alguna asesoría o tenga inquietudes al respecto, periódicamente se debe actualizar estas políticas y depurar las que se consideren obsoletas, realizar acuerdos de privacidad de los datos de usuarios, definición de los roles en la política de seguridad, un correcto manejo de los activos de la organización y su posterior gestión, controles de los equipos de trabajo aplicando seguridad perimetral como CCTV y sensores de protección[51].

6.1. HERRAMIENTAS PARA LA IDENTIFICACIÓN DE VULNERABILIDADES

[49] **TITUAÑA VILLA, Milton José. 2017.** [En línea] 28 de Julio de 2017.Disponible en: https://bibdigital.epn.edu.ec/handle/15000/17543.

[50] **DAVID, Rodriguez Parra Jesus. 2018.** [En línea] 2018. Disponible en: https://www.researchgate.net/profile/David_Parra14/project/Implementation-of-The-OWASP-Insurance-Mobile-Development-

[51] **AGUIRRE, MANUEL FERNANDO MARULANDA y DIAZ MONTES, JACOBO. 2018.** [En línea] 2018. Disponible en: https://repository.unad.edu.co/bitstream/handle/10596/20479/1060648494.pdf?sequence=3&isAllowed=y#page=150&zoom=100,148,270.

6.1.1 HERRAMIENTAS OPEN SOURCE

Las herramientas para la identificación de vulnerabilidades web nos proporcionan un escaneo de falencias en la seguridad de los sitios web con el fin de hacer un seguimiento de las mismas, analizar y evaluar el nivel de impacto para las organizaciones, entre las herramientas que encontramos para la identificación de vulnerabilidades web tenemos:

Kali Linux
Kali Linux es un sistema operativo que derivado del sistema operativo DEBIAN, su orientación es a la seguridad informática para la realización de pruebas de penetración (pentesting), cuenta con herramientas que nos permiten realizar estas pruebas de seguridad.

Características:

- Es completamente gratuito.
- Código abierto
- Kali Linux, es una de las distribuciones de Linux, más utilizada por los informáticos, ya que es por medio de ella donde se realizan los mejores procesos de auditoría y ciberseguridad. Su soporte de mantenimiento es de Offesive Security Ltd.
- Esta, es la versión mejorada y fortificada del sistema conocido como BackTrack. Ofrece hoy en día, más de 600 herramientas incluyendo entre ellas, el famoso Nmap y gran descifrador de claves Aircrack-ng. Entre las nuevas actualizaciones recibidas, Kali puede ser usado desde un Live CD, o USB-Live, y también como un sistema instalado directamente al disco duro.
- Metasploit es una de las herramientas que tiene el sistema operativo Kali Linux, este programa o herramienta realiza pentesting por medio de explotación de vulnerabilidades.
- Usuarios root
- Herramientas para pruebas de penetración
- Actualizaciones de seguridad
- Código abierto
- Servicios de red deshabilitados
- Personalización del Kernel Linux
- Todo el sistema es completamente configurable
- Alta seguridad y confiabilidad en el Sistema
- Cuenta con una gran cantidad de utilidades ARM
- Gran utilidad para análisis forense[52]

Vega Vulnerability Scanner

[52] Internetpasoapaso. [sitio web]. Kali Linux ¿Qué es, para qué sirve y cuáles son sus características especiales?. [Consulta: 22 de octubre de 2021. Disponible en: https://internetpasoapaso.com/kali-linux/

es una herramienta de código libre de escaneo y testeo de la seguridad de aplicaciones web, ayudando a encontrar y arreglar XSS, SQL injection y otras vulnerabilidades[53]

Características:

- Herramienta gráfica de auditoría web gratuita
- De código abierto desarrollada por la empresa de seguridad Subgraph.
- Esta herramienta contiene varias características interesantes como un escáner proxy.
- Permite pruebas de seguridad automatizadas para encontrar y validar la inyección de SQL, la secuencia de comandos en sitios cruzados (XSS), la información confidencial revelada inadvertidamente y muchas otras vulnerabilidades[54]

Nessus

Es un software de que tiene dos versiones uno de pago y otro gratuito, el cual utiliza una interfaz gráfica el cual realiza una búsqueda de vulnerabilidades[55].

Es el escáner de vulnerabilidad web más utilizado de la industria. Nessus está respaldado por un equipo de expertos de investigación de vulnerabilidades y una de las bases de datos más grandes del mundo[56].

Características:

- La operación normal, nessus comienza escaneando los puertos con nmap o con su propio escaneador de puertos para buscar puertos abiertos y después intentar varios exploits para atacarlo.
- Opcionalmente, los resultados del escaneo pueden ser exportados en reportes en varios formatos como texto plano, XML, HTML y LATEX.
- Escanea el servidor con la dirección IP que necesite.
- Se escoge el nombre del análisis, escaneo interno y los IP de los hosts que se quieren analizar.
- En la opción HOSTS muestra las vulnerabilidades en porcentajes clasificadas en 5: Críticas, Altas, Medias, Bajas y de información.
- Se puede ingresar a cada vulnerabilidad para una descripción más detallada[57].

Nmap

[53] Incibe. [Sitio web]. Vega Vulnerability Scanner. [consulta: 10 de octubre de 2021]. Disponible en: https://www.incibe.es/protege-tu-empresa/catalogo-de-ciberseguridad/listado-soluciones/vega-vulnerability-scanner

[54] DiseñoWeb.[Sitio web]. Encontrar vulnerabilidades web con Vega. [Consulta: 10 de octubre de 2021]. Disponible en: https://www.disenowebwordpress.com/encontrar-vulnerabilidades-web-con-vega/

[55] **Alejandro, Parra Tapia Erik y Díaz Ortiz , Daniel Giovanny. 2020.** dspace.ups.edu.ec. [En línea] Febrero de 2020. [Citado el: 7 de Mayo de 2021.] http://dspace.ups.edu.ec/handle/123456789/18395.

[56] Seaq.[sitio web]. Nessus. [Consulta: 12 de octubre de 2021]. Disponible en: https://www.seaq.co/nessus.html

[57] Ibid..

Es un escáner de puertos técnica empleada para la identificación de puertos, el cual permite visualizar los equipos activos de red para la obtención de información principalmente de los puertos abiertos, la información que suministra esta herramienta brinda a los ciberdelincuentes información en la emplean para la ruptura de la seguridad de un sistema web[58].

Características:

- Nmap es gratuita y de código abierto
- Permite descubrir redes y host, así como realizar auditoría de seguridad.
- Este programa es compatible con sistemas operativos Linux, Windows y también macOS, pero en todos ellos se utiliza a través de la línea de comandos.
- Brinda la posibilidad de instalar ZenMap que es la utilidad gráfica de Nmap para hacer los escaneos de puertos a través de la interfaz gráfica de usuario.
- Permite detectar hosts de una red local, y también a través de Internet, de esta forma, se puede saber si dichos hosts (ordenadores, servidores, routers, switches, dispositivos IoT) están actualmente conectados a Internet o a la red local.
- Esta herramienta también permite realizar un escaneo de puertos a los diferentes hosts, ver qué servicios están activos en dichos hosts gracias a que indica el estado de sus puertos, de esta manera se saber qué sistema operativo está utilizando un determinado equipo, e incluso podremos automatizar diferentes pruebas de pentesting para comprobar la seguridad de los equipos[59].

ACUNETIX

Es un escáner capaz de encontrar vulnerabilidades de sitios web dichas vulnerabilidades son de tipo 8 este tipo de vulnerabilidades son: Inyección SQL – XSS – y vulnerabilidades a nivel de código.

Características:

- Las herramientas de testeo de inyección SQL y de Cross site scripting más avanzadas y profundas de la industria.
- Herramientas para fácil aseguramiento de formularios web y contraseñas.
- Facilidad de generación de informes amplios, incluyendo informes de cumplimiento PCI.
- Acunetix escanea y analiza sitios web incluyendo contenido flash, SOAP y AJAX.

[58] **Lizeth, Santillán Mosquera Ángela. 2019.** repository.unad.edu.co. [En línea] 15 de Diciembre de 2019. [Citado el: 7 de Mayo de 2021.] https://repository.unad.edu.co/bitstream/handle/10596/31771/alsantillanm.pdf?sequence=1&isAllowed=y.

[59] Redeszone. [Sitio web]. Realiza escaneos de puertos con Nmap a cualquier servidor o sistema. [Consulta: 17 de octubre de 2021]. Disponible en: https://www.redeszone.net/tutoriales/configuracion-puertos/nmap-escanear-puertos-comandos/

- Un cliente de analizador automático de secuencia de comandos que permite realizar pruebas de seguridad de Ajax y aplicaciones Web 2.0.
- Herramientas avanzadas de penetración, como HTTP Editor y HTTP Fuzzer.
- Soporte para páginas con CAPTCHA, single sign-on y mecanismos con factor de autenticación.
- El escaneo inteligente detecta el tipo de servidor web y lenguaje de la aplicación.
- Acunetix escanea y analiza sitios web incluyendo contenido flash, SOAP y AJAX[60].

BetterCap

Herramienta potente pero flexible y portable creada para el análisis y búsqueda de vulnerabilidades MITM y manipulación de protocolos HTTP, HTTPS, y de tráficos TCP en tiempo real.

Características:

- Escáner de redes WiFi, permite hacer ataques de desautenticación, también permite realizar ataques sin clientes a asociaciones PMKID, permite capturar handshakes de clientes que usan protocolo WPA y WPA2.
- Escáner de dispositivos BLE (Bluetooth Low Energía) para leer y escribir información.
- Escáner de dispositivos inalámbricos que usen la banda de 2.4GHz, como los ratones inalámbricos, también permite realizar ataques MouseJacking con inyección de datos.
- Permite hacer ataques pasivos y activos a redes IP
- Permite realizar ataques MitM basados en ARP, DNS y también DHCPv6, con el objetivo de capturar toda la información.
- Permite crear un proxy HTTP/HTTPS para levantar el tráfico seguro HTTPS, y facilita enormemente el uso de scripts.
- Sniffer de red muy potente para recolección de credenciales de usuario.
- Escáner de puertos muy rápido
- Tiene una potente API REST para realizar ataques fácilmente.
- Incorpora una interfaz gráfica de usuario para facilitar los ataques, aunque el terminal de comandos es muy potente.
- Proporciona gran cantidad de módulos de diferentes categorías para ampliar funcionalidades[61]

[60] Seaq. [Sitio web]. Acunetix. [Consulta: 17 de octubre de 2021]. Disponible en : https://www.seaq.co/acunetix.html

[61] Hacker.net. [Sitio web]. bettercap: la navaja suiza del tráfico de red. [Consulta: 12 de octubre de 2021]. Disponible en: https://blog.elhacker.net/2021/05/bettercap-la-navaja-suiza-del-trafico-analizar-red.html

DVWA
Esta es una herramienta de pruebas es decir es utilizada como un entorno seguro para la realización de pruebas de las herramientas mencionada para encontrar vulnerabilidades, no es propiamente una herramienta de detección sino más bien una herramienta de entorno seguro de pruebas[62].

Características:

- Entorno de entrenamiento en explotación de seguridad web escrito en PHP y MySQL cuyo objetivo principal es permitir a programadores y técnicos estudiar e investigar sobre las diferentes temáticas involucradas en dicho campo en un entorno completamente legal.
- Identifica vulnerabilidades de sitios web y aplicaciones web
- Permite usar técnicas de explotación e intrusión,
- Facilita la implementación de métodos de corrección para asegurar mejor los sistemas.
- Detecta vulnerabilidades web disponibles en la aplicación DVWA
- Permite realizar ataques de fuerza bruta , de inyección SQL , XSS y de CSRF
- Ejecución de comandos vía shell_exec en PHP
- Cargar vulnerabilidad[63]

HTTrack
Herramienta que permite descargar un sitio WORLD WIDE WEB, para obtener contenidos HTML, imagines permitiendo realizar la duplicación de sitios web.

Características:
- Puede copiar, editar y compartir el contenido del sitio web descargado.
- Fácil proceso de instalación.
- Puede actualizar el contenido del sitio web descargado con conexión a Internet.
- No encontrará ninguna diferencia entre la navegación en línea y la navegación fuera de línea

Wireshark
Más que una herramienta de detención de vulnerabilidades es un analizador de tráfico y de protocolos de la red, permitiendo obtener información confidencial del tráfico de la red y de quienes este conectados en ella, esta herramienta también es útil para hacer la auditoria para la identificación de contraseñas inseguras[64].

---

[62] **LLERENA, Alain Eduardo Rodríguez. 2020.** [En línea] 2020. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-18592020000100116.

[63] Pchardwarepro. [Sitio web]. DVWA: Ponga a prueba sus habilidades de piratería informática. [consulta: 10 de octubre de 2021].Disponible en: https://www.pchardwarepro.com/dvwa-ponga-a-prueba-sus-habilidades-de-pirateria-informatica/

[64] **IVAN, CORONEL SUÁREZ y GARCÍA PERERO, FREDDY GIANCARLO. 2021.** [En línea] 2021. Disponible en: https://repositorio.upse.edu.ec/handle/46000/5917.

Características:

- Disponible para Linux y Windows
- Captura de paquetes en vivo desde una interfaz de red
- Muestra los paquetes con información detallada de los mismos
- Abre y guarda paquetes capturados
- Importar y exportar paquetes en diferentes formatos
- Filtrado de información de paquetes
- Resaltado de paquetes dependiendo el filtro
- Crear estadísticas
- Entre sus cualidades nos encontramos con una enorme versatilidad que le lleva a soportar más de 480 protocolos distintos, además de la posibilidad de trabajar tanto con datos capturados desde una red durante una sesión con paquetes previamente capturados que hayan sido almacenados en el disco duro.
- Wireshark soporta el formato estándar de archivos tcpdump, es capaz de reconstruir sesiones TCP, y está apoyado en una completa interfaz gráfica que facilita enormemente su uso[65].

6.1.2 HERRAMIENTAS OWASP

Zapattack de OWASP

Herramienta gratuita de análisis de sitios web que permite hacer una evaluación del sitio y cuenta con un manual de utilización[66]. OWASP ZAP (Zed Attack Proxy) es el escáner web de vulnerabilidades más utilizado en todo el mundo, es completamente gratuito y de código abierto. Este programa es mantenido activamente por una comunidad internacional de voluntarios, los cuales trabajan para ir mejorando la herramienta poco a poco y también incorporando nuevas características.

Características:

- Es completamente gratuita y de código abierto
- Es una herramienta multiplataforma, siendo compatible con sistemas operativos Windows (de 32 y 64 bits), Linux, MacOS, e incluso se puede descargar en un contenedor Docker que incorporará todo lo necesario para ejecutarlo correctamente.
- Este programa es muy sencillo de instalar, tan solo se requiere tener Java instalado en el equipo para poder ejecutarlo.
- Esta herramienta está traducida en más de 12 idiomas entre los que se incluye el español, además, gracias a la comunidad se dispone de una gran

[65] Ecured. [Sitio web]. Wireshark. [Consulta:10 de octubre de 2021].disponible en: https://www.ecured.cu/Wireshark

[66] **CASTAÑEDA SUÁREZ, Andrés Fernando. 2017.** [En línea] 2017. Disponible en:https://repository.unimilitar.edu.co/handle/10654/16513.

cantidad de documentación, tutoriales de ayuda y foros donde puede poner un problema para ayudar a solucionarlo.

- Los usuarios de esta herramienta forense de seguridad podrán auditar diferentes aplicaciones web con una serie de funciones y análisis específicos.
- Se puede comprobar todas las peticiones y respuestas entre cliente y servidor, levantando un proxy que se encargará de capturar todas las peticiones para su posterior estudio.
- Se puede localizar recursos en un servidor, realizar análisis automáticos, análisis pasivos, posibilidad de lanzar varios ataques a la vez, e incluso usar certificados SSL dinámicos.
- Permite usar tarjetas inteligentes como el DNIe, e incluso certificados personales, también es capaz de trabajar con sistemas de autenticación
- Proporciona una tienda de extensiones (plugins) para aumentar aún más las funcionalidades de esta gran herramienta[67].

OWASP ZED ATAQUE PROXY (ZAP)

El escáner de aplicaciones web más utilizado del mundo. Libre y de código abierto. Mantenido activamente por un dedicado equipo internacional de voluntarios. Es una herramienta libre escrita en Java proveniente del Proyecto OWASP para realizar, en primera instancia, tests de penetración en aplicaciones web, aunque también puede ser usado por desarrolladores en su trabajo diario. Al día de hoy se encuentra en su versión 2.1.0 y necesita Java 7 para ejecutarse, aunque yo lo uso en Debian GNU/Linux bajo OpenJDK 7[68].

características

- Proxy de interceptación Ideal para los que somos newbies en este campo de la seguridad, configurado de la manera correcta.
- Permite ver todo el tráfico entre el navegador y el servidor web de turno, dejando ver de forma sencilla las cabeceras y cuerpo de los mensajes HTTP sin importar el método usado (HEAD, GET, POST, etc).
- Se puede modificar el tráfico HTTP a nuestro antojo en ambas direcciones de la comunicación (entre el servidor web y el navegador).
- Spider: Es una característica que ayuda a descubrir nuevas URL's en el sitio auditado. Una de las maneras que realiza esto es analizando el código HTML de la página para descubrir etiquetas <a> y seguir sus atributos href.
- Forced Browsing: Intenta descubrir directorios y archivos no indexados en el sitio como pueden ser páginas de inicio de sesión. Para lograrlo cuenta por

[67] Redeszone. [sitio web]. OWASP ZAP, audita la seguridad de webs y evita vulnerabilidades. [consulta: 12 de octubre de 2021]. Disponible en: https://www.redeszone.net/tutoriales/seguridad/owasp-zap-auditar-seguridad-web/

[68] Zaproxy. [sitio web]. Owasp zed ataque proxy (zap). [Consulta: 13 de octubre de 2021]. Disponible en: https://www.zaproxy.org/

defecto con una serie de diccionarios que utilizará para realizar peticiones al servidor esperando status code de respuesta 200[69].

- Active Scan: Genera de manera automatizada diferentes ataques web contro el sitio como CSRF, XSS, Inyección SQL entre otros.
- Y muchas otras:
- Soporte para web sockets desde las versiones 2.0.0, AJAX Spider, Fuzzer, y otras cuantas.

SECURITY KNOWLEDGE FRAMEWORK

El marco de conocimientos de seguridad de OWASP es una aplicación web de código abierto que explica los principios de codificación segura en varios lenguajes de programación. El objetivo de OWASP-SKF es contribuir en el aprendizaje e integración de la seguridad por diseño en su desarrollo de software y crear aplicaciones que sean seguras por diseño. OWASP-SKF hace esto a través de proyectos de desarrollo de software manejables con listas de verificación (usando OWASP-ASVS / OWASP-MASVS o listas de verificación de seguridad personalizadas) y laboratorios para practicar la verificación de seguridad (usando SKF-Labs, OWASP Juice-shop y ejemplos de códigos de mejores prácticas de SKF y OWASP-Cheatsheets).

Características

El nivel actual de seguridad de las aplicaciones web no es suficiente para garantizar la seguridad. Esto se debe principalmente a que los desarrolladores web simplemente no son conscientes de los riesgos y peligros que acechan a la espera de ser explotados por piratas informáticos. Debido a esto, se desarrolla un marco para crear un sistema de guía disponible para todos los desarrolladores para que puedan desarrollar aplicaciones seguras por diseño desde el principio.

OWASP AntiSamy

El proyecto OWASP AntiSamy es unas cuantas cosas. Técnicamente, es una API para asegurarse que las entradas HTML/CSS del usuario estén en cumplimiento con las reglas de la aplicación. Otra forma de decirlo podría ser: es una API que le ayuda a asegurarse que los clientes no provean código malicioso en el HTML que proveen para su perfil, comentarios, etc. Que se quedan almacenados en el servidor. El termino código malicioso en términos de aplicaciones Web es generalmente relacionado solo con JavaScript. Hojas de estilo en cascada (CSS) son solo consideradas maliciosas cuando invocan a JavaScript. Sin embargo, hay

[69] Ibid..

muchas situaciones donde HTML y CSS “normales” pueden ser usados de una forma maliciosa[70].

OWASP CAL9000

Es una colección de herramientas de prueba de seguridad de aplicaciones web que complementan el conjunto de web proxies y escáners automáticos actuales. CAL9000 le da la flexibilidad y funcionalidad que necesita para aumentar la eficacia de las pruebas manuales. Funciona mejor cuando se usa con Firefox o Internet Explorer. CAL9000 está escrito en JavaScript, así que se tiene completo acceso al código fuente. Se puede modificar para que se adapte mejor a sus necesidades particulares. CAL9000 tiene algunas características poderosas (como ejecutar cross-domain xmlHttpRequests y escribir a disco)[71].

OWASP CLASP

CLASP (Proceso de seguridad en aplicación completo y ligero) proporciona un enfoque bien organizado y estructurado para mover las inquietudes de seguridad a las fases iniciales del ciclo de vida de desarrollo de software, cuando esto sea posible. CLASP es en realidad un conjunto de piezas de proceso que puede ser integrado en cualquier proceso de desarrollo de software. Está diseñado para ser fácil de adoptar y efectivo a la vez. Toma un enfoque prescriptivo, documentando las actividades que las organizaciones debería estar haciendo. Y proporciona una amplia riqueza de recursos de seguridad que hacen razonable implementar esas actividades[72].

OWASP DirBuster

DirBuster es una aplicación Java multi hilo diseñada para obtener por fuerza bruta los nombres de directorios y archivos en servidores Web/de aplicación. A menudo ocurre que lo que ahora parece un servidor Web en una fase de instalación por omisión no lo es, y tiene páginas y aplicaciones ocultas. DirBuster trata de encontrar estos. Sin embargo, las herramientas de esta naturaleza a menudo son solo tan buenas como la lista de archivos y directorios con los que vienen. Un enfoque diferente fue usado para generar esto. ¡La lista fue generada desde cero, rastreando en Internet y colectando los directorios y archivos que son realmente usados por los desarrolladores! DirBuster viene con un total de 0 listas diferentes, esto hace a DirBuster extremadamente efectivo encontrando esos archivos y directorios ocultos. Y si eso no fuera suficiente, DirBuster también tiene la opción de realizar fuerza

[70] Dragonjar. [Sitio web]. OWASP AntiSamy. [Consulta: 15 de octubre de 2021]. Disponible en: https://www.dragonjar.org/proyectos-owasp-en-espanol.xhtml

[71] Dragonjar. [Sitio web]. OWASP CAL9000. [Consulta: 15 de octubre de 2021]. Disponible en: https://www.dragonjar.org/proyectos-owasp-en-espanol.xhtml

[72] Dragonjar. [Sitio web]. OWASP CLASP. [Consulta: 15 de octubre de 2021]. Disponible en: https://www.dragonjar.org/proyectos-owasp-en-espanol.xhtml

bruta pura, lo que no les deja lugar para esconderse a los archivos y directorios ocultos Si tiene el tiempo[73].

OWASP Encoding

Las aplicaciones web enfrentan un gran número de amenazas; una de ellas es cross-site scripting y ataques de inyección relacionados. El 90% de todas las aplicaciones web contiene ataques de cross-site scripting porque son fáciles de introducir, y las herramientas adecuadas no están siempre disponibles para prevenirlos. La biblioteca Reform proporciona un conjunto sólido de funciones para codificar la salida para los objetivos de contexto más comunes en aplicaciones web (por ejemplo: HTML, XML, JavaScript, etc.). La biblioteca también tiene una opinión conservadora de cuáles son los caracteres permitidos basado en vulnerabilidades históricas y técnicas actuales de inyección.

OWASP Enterprise Security API

La ESAPI es una colección gratis y abierta de todos los métodos de seguridad que un desarrollador necesita para construir una aplicación Web segura. Se puede usar solo las interfases y construir su propia implementación usando la infraestructura de su compañía. O, puede usar la implementación de referencia como un punto de inicio. En concepto, la API es independiente del lenguaje. Sin embargo, los primeros entregables del proyecto son una API Java y una referencia de implementación Java. Esfuerzos para construir ESAPI en .NET y PHP están en marcha. Desafortunadamente, las plataformas disponibles, y herramientas (Java EE, Struts, Spring, etc.…) simplemente no proporcionan protección suficiente. Esto deja a los desarrolladores con la responsabilidad de diseñar y construir mecanismos de seguridad. Este reinventado de la rueda para cada aplicación lleva a una pérdida de tiempo y agujeros de seguridad masivos[74].

OWASP Insecure Web App

InsecureWebApp es una aplicación que incluye vulnerabilidades comunes en aplicaciones Web. Es un objetivo de pruebas de penetración automatizadas y manuales, análisis de código fuente, evaluaciones de vulnerabilidades y modelado de amenazas. InsecureWebApp es ante todo una ayuda para desafiar y mejorar las habilidades de diseño y codificación segura. Arquitectos y desarrolladores necesitan aprender a identificar las vulnerabilidades en una aplicación Web real. Los objetivos de esta herramienta son de tres tipos: 1) demostrar lo peligrosas que pueden ser

---

[73] Dragonjar. [Sitio web]. OWASP DirBuster. [Consulta: 15 de octubre de 2021]. Disponible en: https://www.dragonjar.org/proyectos-owasp-en-espanol.xhtml
Dragonjar. [Sitio web]. OWASP Encoding. [Consulta: 15 de octubre de 2021]. Disponible en: https://www.dragonjar.org/proyectos-owasp-en-espanol.xhtml

[74] Dragonjar. [Sitio web]. OWASP Enterprise Security API. [Consulta: 15 de octubre de 2021]. Disponible en: https://www.dragonjar.org/proyectos-owasp-en-espanol.xhtml

las vulnerabilidades de las aplicaciones, 2) cerrar la brecha que existe entre la teoría de seguridad en aplicaciones y el código que realmente se está diseñando y escribiendo, 3) aprender como estas vulnerabilidades pueden ser arregladas[75].

6.1.3 Cuadro Top Owasp asociado a las de herramientas de identificación de vulnerabilidades

TOP TEN OWASP	HERRAMIENTAS
A01:2021 – BROKEN ACCESS CONTROL	1. KALI LINUX. 2. NMAP. 3. DVWA. 4. ZAPATTACK DE OWASP. 5. OWASP ZED ATAQUE PROXY. 6. OWASP CAL9000. 7. OWASP CLASP. 8. OWASP ENCODING. 9. OWASP INSECURE WEB APP.
A02: 2021 – CRYPTOGRAPHIC FAILURES	1. KALI LINUX. 2. DVWA. 3. BETTERCAP. 4. ZAPATTACK DE OWASP. 5. SECURITY KNOWLEDGE FRAMEWORK. 6. OWASP CLASP. 7. OWASP ENCODING. 8. OWASP ENTERPRISE SECYRITY API. 9. OWASP INSECURE WEB APP.
A03: 2021 – INJECTION	1. KALI LINUX. 2. VEGA VULNERABILITY SCANNER. 3. ACUNETIX. 4. DVWA. 5. ZAPATTACK DE OWASP. 6. OWASP ZED ATAQUE PROXY. 7. OWASP CLASP. 8. OWASP ENCODING. 9. OWASP INSECURE WEB APP.
A04: 2021 – INSECURE DESING	1. KALI LINUX. 2. ACUNETIX,

[75] Dragonjar. [Sitio web]. OWASP Insecure Web App . [Consulta: 15 de octubre de 2021]. Disponible en: https://www.dragonjar.org/proyectos-owasp-en-espanol.xhtml

	3. DVWA. 4. HTTRACK. 5. ZAPATTACK DE OWASP. 6. SECURITY KNOWLEDGE FRAMEWORK. 7. OWASP ANTISAMY. 8. OWASP ENCODING. 9. OWASP CLASP. 10. OWASP DIRBUSTER. 11. OWASP INSECURE WEB APP.
A05: 2021 – SECURITY MISCONFIGURATION.	1. KALI LINUX. 2. NMAP. 3. BETTERCAP. 4. DVWA. 5. ZAPATTACK DE OWASP. 6. OWASP ZED ATAQUE PROXY. 7. OWASP CLASP. 8. OWASP DIRBUSTER 9. OWASP INSECURE WEB APP.
A06: 2021 – VULNERABLE AND OUTDATED COMPONENTS.	1. KALI LINUX. 2. NESSUS. 3. NMAP. 4. DVWA. 5. ZAPATTACK DE OWASP. 6. OWASP CLASP. 7. OWASP DIRBUSTER. 8. OWASP ENTERPRISE SECYRITY API. 9. OWASP INSECURE WEB APP.
A07: 2021 – IDENTIFICATION AND AUTHENTICATION FAILURES	1. KALI LINUX. 2. DVWA. 3. ZAPATTACK DE OWASP. 4. OWASP ZED ATAQUE PROXY. 5. OWASP CLASP. 6. OWASP INSECURE WEB APP.
A08: 2021 – SOFTWARE AND DATA INTEGRITY FAILURES	1. KALI LINUX. 2. VEGA VULNERABILITY SCANNER. 3. DVWA. 4. ZAPATTACK DE OWASP. 5. OWASP CLASP. 6. OWASP DIRBUSTER. 7. OWASP INSECURE WEB APP.

A09: 2021 – SECURITY LOGGING AND MONITORING FAILURES	1. KALI LINUX. 2. NESSUS. 3. DVWA. 4. ZAPATTACK DE OWASP. 5. WIRESHARK. 6. OWASP CLASP. 7. OWASP ENTERPRISE SECYRITY API. 8. OWASP INSECURE WEB APP.
A10: 2021 – SERVER-SIDE REQUEST FORGERY	1. KALI LINUX. 2. DVWA. 3. NESSUS. 4. ZAPATTACK DE OWASP. 5. OWASP ZED ATAQUE PROXY. 6. OWASP CLASP. 7. OWASP ENTERPRISE SECYRITY API. 8. OWASP DIRBUSTER. 9. OWASP INSECURE WEB APP.

6.2. MEJORAS DE SEGURIDAD CON ESTÁNDARES OWASP

Teniendo presente que con las herramientas adecuadas podemos dañar un aplicativo, por medio de ataques cibernéticos, generando una perdida valiosa a las organizaciones, podemos establecer unas posibles mejoras a estas aplicaciones partiendo del historial de incidentes que se puedan encontrar tanto en las organizaciones como en proyectos aplicados para el estudio del mejoramiento de seguridad en aplicaciones web, para ello podemos hacer:

1. Creación de bases de datos con las direcciones IP´S de cada uno de los intrusos detectados, esta base de datos se debe crear en orden de impacto y el daño causando, esto con el fin de identificar la principal fuente de deseo del atacante.
2. Realización de una valoración de ataques y un posterior seguimiento de los mismos esto con el fin de lograr identificar los puntos más débiles de la seguridad.
3. Realizar un diagrama con el plan de acción de respuesta generado a partir de la base de datos donde se evidencia cada IP y tomando en cuenta la valoración de ataques[76].

[76] **Pérez Barrera, Dennis, González Brito, Henry Raúl y Sánchez Borrell, Yailin. 2019.** [En línea] 22 de Enero de 2019. https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/306.

Por otro lado, podemos también hablar de:

- Estudio de las guias de pruebas, las cuales permiten enfocar el desarrollo de software de manera seguro, acoplando las pruebas de seguridad de las mismas antes de dar por terminado la aplicación y entrega de la misma a la organización.
- Acatar el top 10 de OWASP los cuales son documentos con información de los ataques más frecuentes en la seguridad de las aplicaciones, como las aplicaciones web como las aplicaciones de escritorio.
- Implementar el conocimiento de los ataques más frecuentes en las aplicaciones basándonos en el top 10 de OWASP para el mejoramiento de la seguridad informática[77].

También se puede realizar una serie de pruebas apuntando a procesos específicos de la organización basados en la metodología OWASP estas pruebas son:

1. Recopilación de información: es la forma correcta de iniciar la serie de pruebas basadas en esta metodología estudiada ya que permite buscar la mayor cantidad de información posible acerca de la aplicación o sitio web a analizar y auditar, en esta serie se debe buscar nombre de la aplicación dominio establecido, enfoque de la aplicación, y demás información relevante para tener una mayor capacidad de entender y analizar la aplicación web.
2. Realizar pruebas de gestión de la configuración: en esta serie consiste en el análisis de la arquitectura de la aplicación como lo es el código fuente, los permisos que tiene la aplicación para los usuarios, los métodos HTTP, métodos de autenticación entre otros.
3. Realización de pruebas para la autenticación: se realiza para tener la verificación de cómo se realizar la autenticación de los sistemas o aplicaciones de la compañía, una prueba sería las pruebas de fuerza bruta.
4. Pruebas de autorización: aquí es importante que la persona encargada de realizar la auditoria debe contar con el conocimiento necesario y el entendimiento de la arquitectura de la aplicación para hacer pruebas que le permitan saltar dicha arquitectura que le permita de una u otra forma acceder a los recursos e incluso hacer manipulación completa de la aplicación.
5. Realizar pruebas de denegación de servicios: se realiza una gran cantidad de peticiones dentro de la aplicación en búsqueda de “caer” el servidor debido a que no tiene el soporte necesario para dar respuesta a un gran volumen de peticiones, esto se hace para medir la capacidad de resistencia de los equipos y aplicaciones.

[77] Hernández Peñaherrera, Erika Belén y Briones Pincay, Gerson Hammer. 2018. [En línea] 2018. http://repositorio.ug.edu.ec/handle/redug/26837.

6. Ataques SQL a servicios web: esto se hace para ver qué tan segura es la base de datos dentro del servicio web, partiendo que estos servicios deben contar con protocolos SSL, HTTP & FTP[78].
7. Identificación de vulnerabilidades del lado del servidor: permite realizar acciones preventivas y correcciones requeridas para el correcto funcionamiento de la aplicación web.
8. Realización de respaldo y restauraciones: se realiza un respaldo de los procesos y datos alojados en la aplicación esto sirve para que llegado el caso falle la aplicación se pueda hacer una restauración sin pérdida de información ni de configuración[79].
9. En entornos de desarrollado basados en gestores de contenido como WordPress la utilización de plugin como BackWPup el cual permite realizar una copia de seguridad que posteriormente se guarda en el servidor FTP, ayuda a que en caso tal de falla se pueda hacer una restauración de la misma.
10. También para gestores de contenidos hacer la instalación del plugin Themes Security, el cual permite proteger el sitio web ante intrusos y para identificación de vulnerabilidades de la misma asi mismo como contraseñas débiles[80].
11. Agregar certificaciones SSL que garanticen asegurar la correcta comunicación de los usuarios y de los datos que se trasmiten dentro de la aplicación[81].
12. Realizar periódicamente los análisis de seguridad utilizando la metodología OWASP.
13. Implementación de estándares de seguridad como lo son las ISO/IEC 20000 y la familia que trate de seguridad de la información[82].

[78] **Juliana, García Zapata. 2018.** [En línea] 2018. https://repository.unad.edu.co/bitstream/handle/10596/28466/1033648651.pdf?sequence=1&isAllowed=y.

[79] **ROBAYO BAUTISTA, Eliana Catherine. 2021.** [En línea] 2021. Disponible en: https://repository.ucatolica.edu.co/bitstream/10983/25731/1/Trabajo%20de%20Grado_Eliana_robayo_Gu%2b%c2%a1a%20de%20principios_28-11-2020.pdf#page=73&zoom=100,148,314.

[80] **GARCÍA, Luz Angela Robayo. 2021.** [En línea] 2021. Disponible en: https://repository.unad.edu.co/handle/10596/40343.

[81] **FERNÁNDEZ MIRANDA, Henry Armando. 2019.** [En línea] 2019. Disponible en: Fernández Miranda, Henry Armando.

[82] **GONZÁLEZ MENDOZA, Dewin Fernando. 2017.** [En línea] 2017. Disponible en: https://repository.unad.edu.co/handle/10596/17397.

7. MEJORES PRÁCTICAS USANDO EL ESTÁNDAR DE VERIFICACIÓN DE SEGURIDAD EN APLICACIONES DE OWASP EN EL DESARROLLO DE PROYECTOS DE SOFTWARE

Al momento de hablar de las mejores prácticas usando el estándar de verificación con la metodología OWASP se debe tener presente ¿Para qué sirve hacer una prueba con la metodología OWASP? partiendo de este interrogante se puede hablar que sirve para la verificación y posterior corrección de falencias y error de seguridad (bugs) dentro de las aplicaciones web, con el fin de que la organización y sus desarrolladores conozcan sobre dichos problemas de seguridad[83].

Los requisitos para tener una buena práctica de desarrollo para software seguro se pueden categorizar en tres ítems.

1. Integridad: se establece marcos de trabajo para evitar ataques de integridad de los datos como lo son ataques XSS, ataques CSRF, ataques de inyección, adicional a ello la implementación de controles SQL para evitar fugas de datos, validación de datos tanto de entrada como de salida y realizar cifrado de datos y este cifrado se debe realizar de acuerdo a su complejidad y sensibilidad de los datos.
2. Confidenciadla: protección de conexiones, evitar la elevación de privilegios de los usuarios también deshabilitar el almacenamiento cache de los datos sensibles que se puedan manejar dentro de la aplicación.
3. Disponibilidad: en este ítem se habla de la realización de estudios de las vulnerabilidades basándonos en la metodología OWASP, la utilización de tecnologías seguras de desarrollo, controlar la conexión de la base de datos, el análisis de todos los riesgos que se puedan tener en la aplicación[84].

OWASP ASVS

OWASP ASVS estándar de verificación parte de dos objetivos los cuales son:

1. Ayudar a los desarrolladores y organizaciones en el desarrollo de las aplicaciones seguras y su posterior mantenimiento.
2. La alineación basada en las necesidades y ofertas de los servicios de seguridad junto con las herramientas de seguridad y sus consumidores.

[83] **FABRICIO, Rodríguez Zambrano Stalyn. 2019.** [En línea] 2019. Disponible en: https://repositorio.itb.edu.ec/bitstream/123456789/2679/1/PROYECTO%20DE%20GRADO%20DE%20RODR%c3%8dGUEZ%20ZAMBRANO%20STALYN%20FABRICIO.pdf.

[84] **YISEL, BENITEZ NIÑO y SILEGA MARTÍNEZ , NEMURY. 2018.** [En línea] 2018. Disponible en: http://scielo.sld.cu/scielo.php?pid=S2227-18992018000500015&script=sci_arttext&tlng=pt.

Basados en estos dos objetivos tiene tres niveles de verificación:

- ASVS nivel 1: dirigido a todo tipo de software.
- ASVS nivel 2: dirigido a las aplicaciones que contengan datos sensibles que requieres niveles de protección.
- ASVS nivel 3: dirigido a las aplicaciones con características de seguridad más críticas tales como las que impliquen transacciones de alto valor, datos médicos de pacientes, en resumidas cuentas, es dirigido a todas las aplicaciones que requieran un nivel de confidenciales de su información[85].

Es necesario que todos los desarrolladores conozcan las mejores prácticas de desarrollo acompañado de los estándares que respalden estas prácticas como lo son ITIL, CMMI, COBIT & ISO/IEC 27000 entre otras[86].

- ISO/IEC 27000

El estándar ISO/IEC 27000 es una norma internacional que proporciona directrices de seguridad de la información donde incluye procedimientos físicos de la seguridad dentro de esta norma encontramos también toda su familia de normas ISO como lo son (ISO 27001, ISO 27002 e ISO 27005) cuya variación es el alcance de cada una y su finalidad.

La norma es implementada tanto en el desarrollo como en la auditoria, esta norma como se menciona anteriormente asegura la información por ende mejora el nivel de seguridad de la organización, dicho esto proporciona el conocimiento de la infraestructura de TI relevante para el entorno físico de la organización, junto con las amenazas y contramedidas correspondientes[87].

- ITIL

ITIL es un estándar de enfoque administrativo de servicios y de ciclo de vida del programa, la estrategia del ITIL es la guía del diseño, desarrollo e implementación de la administración de los servicios desde un enfoque de las capacidades organizativas y de sus activos, esto permite conocer quiénes son realmente los

[85] **OWASP. 2017.** [En línea] Abril de 2017. Disponible en: https://owasp.org/www-pdf-archive/Est%C3%A1ndar_de_Verificaci%C3%B3n_de_Seguridad_en_Aplicaciones_3.0.1.pdf.

[86] **Aliaga, JUAN JOSÉ Romero. 2019.** [En línea] 2019.Disponible en: https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/2078/Juan%20Romero_Trabajo%20de%20Investigacion_Maestria_2019.pdf?sequence=1&isAllowed=y#page=31&zoom=100,81,736.

[87] **MERIAH, INES y Arfa Rabai, Latifa Ben. 2019.** [En línea] 2019.Disponible en: https://doi.org/10.1016/j.procs.2019.09.447.

clientes del servicio, logrando satisfacer las necesidades del cliente junto con las capacidades del sistema y recursos empleados[88].

- COBIT.

COBIT es un estándar cuyo objetivo es el control de las tecnologías informáticas y tecnologías conexas, este estándar es desarrollado principalmente para la satisfacción de necesidades administrativas creando espacios de información de los riesgos empresariales, el control y los problemas técnicos.

Proporciona un marco exitoso por sus cuatro pilares los cuales hacen cumplir el objetivo de este estándar.

1. Planificación y organización.
2. Adquisición e implementación.
3. Entrega y soportes.
4. Supervisión y evaluación[89].

OWASP recomienda unos ítems de prueba los cuales se dividen en diez subcategorías las cuales son (Pruebas de gestión de la configuración, compilación de la información obtenida de la organización junto con la información del uso de la aplicación a auditar, pruebas lógicas, pruebas de autenticación, pruebas de gestión de sesiones, pruebas de autorización, pruebas de verificación y validación de los datos, pruebas de denegación de servicio, pruebas basadas en servicios web y pruebas de AJAX)[90].

Una buena práctica al momento del mejoramiento de seguridad en las aplicaciones web teniendo presente a OWASP se debe partir de lo anteriormente mencionado, pero también con algunas prácticas comunes como lo son: la aplicación de actualizaciones de seguridad y de corrección de gestión de vulnerabilidades, priorización de las amenazas y su impacto, proporcionar un seguimiento de las amenazas y de los ataques sufridos dentro de la aplicación[91].

---

[88] **BARRA, AL FARUQ, y otros. 2020.** [En línea] 2020.Disponible en: http://www.warse.org/IJATCSE/static/pdf/file/ijatcse157932020.pdf.

[89] **JOHANES FERNADES, Andry y HENNY, Hartono. 2017.** [En línea] 2017.Disponible en: https://www.researchgate.net/profile/Johanes-Andry/publication/320254270_Performance_Measurement_of_IT_Based_on_COBIT_Assessment_A_Case_Study/links/59d84f18a6fdcc2aad065a7c/Performance-Measurement-of-IT-Based-on-COBIT-Assessment-A-Case-Study.pdf.

[90] **REVISTA VINCULOS. 2019.** [En línea] 1 de Julio de 2019.Disponible en: https://web-b-ebscohost-com.bibliotecavirtual.unad.edu.co/ehost/pdfviewer/pdfviewer?vid=7&sid=5c013c36-8b99-4369-9bd5-223dbdc233f1%40pdc-v-sessmgr03.

[91] **MOLINA GARCÍA, Jorge Alberto. 2019.** [En línea] 2019. Disponible en: http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/6754/LA%20IMPORTANCIA%20DE%20LA%20GESTI%c3%93N%20DE%20RIESGOS%20Y%20SEGURIDAD%20EN%20EL%20INTERNET%20DE%20LAS%20COSAS%20%28IOT%29-%20Jorge%20Molina%20Garc%c3%ada%20ESI41.pdf?sequence=1&isAll.

Un ejemplo de al momento de realizar una buena práctica es la realización de: recopilación de información – realizar pruebas de seguridad a la configuración y despliegue de la aplicación – realizar pruebas de administración de identidad – pruebas de autenticación – pruebas de inicio de sesión – la validación de acceso – pruebas lógicas de la aplicación y pruebas del lado del cliente.

- Recopilación de información: se hace un barrido de toda la información que se pueda obtener de la aplicación, lenguajes de programación, funcionalidad, contenido de la aplicación y demás información crucial de la aplicación web.
- Realización pruebas de seguridad a la configuración y despliegue de la aplicación: su objetivo es recopilar la información de infraestructura que se encuentre directamente relacionados al correcto funcionamiento de la aplicación.
- Pruebas de administración de la identidad: se realizan pruebas de seguridad relacionadas a la gestión de credenciales de los usuarios, para identificar los roles que tiene cada usuario dentro de la aplicación.
- Pruebas de autenticación: se hace una evaluación del proceso de autenticación para determinar las credenciales de acceso de los usuarios.
- Pruebas de inicio de sesión: se realiza una serie de procedimientos los cuales se basan en encontrar cualquier falencia para evadir el mecanismo de gestión de sesiones.
- Validación de acceso: en este ítem encontramos la mayor cantidad de vulnerabilidades tales como peticiones HTTP, código SQL, y ataques de inyección de códigos, por eso es importante realizar esta validación para encontrar estas falencias y posteriormente hacer las correcciones necesarias.
- Pruebas lógicas de la aplicación: se realiza pruebas para evadir flujo de trabajo, si es posible hacer la manipulación de los parámetros de la aplicación, de datos tanto de entrada como de salida.
- Pruebas del lado del cliente: en este apartado se realiza pruebas para encontrar la manera de hacer manipulación de los recursos por medio del DOM y de inyección de código HTML, CSS, JavaScript y demás lenguajes de programación frontend[92].
- Ejecución de pruebas de explotación de vulnerabilidades: se debe realizar en entornos seguros y controlados ejecutando errores de manera intencional para el análisis dinámico y encontrar las falencias más débiles de la seguridad.

[92] **FERNÁNDEZ MAHECHA, Edwin Neyid y Llano Ruiz, Anderson Julian. 2018.** [En línea] Junio de 2018. Disponible en: https://repository.ucatolica.edu.co/bitstream/10983/16045/1/TrabajoDeGA.

- Implementación del método fuzzing: el cual valida entradas y los límites que tienen valorando su nivel de tolerancia, esto más que todo se aplica para emails, conexiones de red, objetos ActiveX[93].
- Mejoramiento de la seguridad en el hardware: esto se debe tener muy en cuenta ya que esto se ve en segundo plano actualmente, pero olvidamos lo fundamental y primordial que esta protección ya que el hardware como lo son los equipos de trabajo es utilizado para el desarrollo de las aplicaciones y si no contamos con un buen esquema de seguridad por ese mismo lado se puede filtrar información valiosa de la organización o perdida de la misma. [94]

7.1. METODOLOGÍAS DE DESARROLLO SEGURO Y AGIL.

También tenemos Secure Software Development Life Cycle (S-SDLC) consiste en la verificación de requisitos de seguridad durante todas las fases de construcción del software correspondientes al análisis, desarrollo, pruebas y mantenimiento, ya que permite mayor atención al detalle, que favorece la identificación oportuna e inmediata de las vulnerabilidades y posterior mejora[95].

S-SDLC se asegura de implementar códigos seguros para el desarrollo de software ya que verifica y analiza los factores más críticos para las aplicaciones es decir donde se pueden encontrar falencias de seguridad de a las mismas, convirtiéndose en una herramienta fundamental para el desarrollo.

Para tener presente otras metodologías que hacen referencia a las mejores prácticas para el desarrollo seguro y ágil de software encontramos: Kanban cuyo origen es japonés donde es ágil ya que hace uso de las tarjetas para gestionar de manera visual la realización y procesos determinados durante el desarrollo, donde permite la visualización oportuna de lo que se hace y los procesos terminados.

Kanban tiene unos fundamentos los cuales permite la correcta implementación de esta metodología, dentro de estos fundamentos encontramos:

- La visualización de flujo de trabajo.
- Limitación de cantidad de trabajo en proceso.
- Monitorea miento de seguimiento del tiempo de trabajo.

---

[93] **ASDRUBAL|, Guayara Rubio. 2019.** Respositorio UNAD. [En línea] 22 de Abril de 2019. Disponible en:https://repository.unad.edu.co/bitstream/handle/10596/25233/%20%09aguayarar.pdf?sequence=1&isAllowed=y.

[94] **LONDOÑO, JEFFERSON GONZALEZ. 2020.** [En línea] 2020. Disponible en: https://repository.unad.edu.co/bitstream/handle/10596/36669/jgonzalezlon.pdf?sequence=1&isAllowed=y#page=48&zoom=100,148,186.

[95] **IT SOLUTIONS DE BETWEEN. 2020. [En línea] Disponible en: https://impulsate.between.tech/tecnicas-desarrollo-seguro-software.**

eXtreme Programming es otra metodología que facilitan las prácticas de desarrollo seguro y ágil de software ya que es la metodología más sencilla y fácil para implementar para clientes o proyectos ambiguos o de diversidad de variables presentadas dentro del mismo, permitiendo la interacción del desarrollo de un día para otro siendo más reflexionado el trabajo en cuanto al diseño y documentación del código a medida de su mismo desarrollo[96].

Por otro lado, CbyC es una metodología para el desarrollo de software con un alto nivel de seguridad y que se pueda demostrar esa seguridad, donde su principal beneficio o objetivo es tener una taza de defectos mínimos, CbyC busca la producción del software sea fiable desde un principio, con correctos requerimientos de desarrollo, altos niveles de seguridad donde se visualiza el comportamiento del sistema[97].

SPIRAL, toma las ventajas del modelo de desarrollo en cascada añadiendo el concepto del análisis de riesgos, para lo cual define cuatro actividades para dar cumplimiento.

1. La planificación: recolección de información.
2. Análisis de riesgos: basados en los requisitos se determina la capacidad de desarrollo o no del software.
3. Ingeniería: diseño de prototipos con relación a los requisitos de la primera actividad.
4. Evaluación del cliente: aquí el cliente comenta todo lo relacionado con el software donde se evalúa si cumple o no con lo solicitado[98].

7.2. CICLO DE VIDA.

Cuando se habla del ciclo de vida de desarrollo de software, se habla de la actividad más importante en la ingeniería de software, ya que presenta una secuencia o estructura definida para el correcto desarrollo del software, esto con el fin de producir un producto eficiente y funcional.

---

96 **Maldonado Manuel** [En línea]. - 2018. Dispobible en: https://www.digital55.com/desarrollo-tecnologia/mejores-metodologias-agiles-creacion-software/.

97 **Brito Abundis Carlos Joaquín** [En línea]. - 2013.Dispoble en: https://www.redalyc.org/pdf/5122/512251564005.pdf.

98 **Maida Esteban Gabriel y Pacienzia Julián** [En línea]. - 2015. Disponible en: https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf.

El ciclo de vida se basa en el desarrollo – control de versiones – pruebas de integración – el despliegue del software – operación y monitoreo del software.

a. Desarrollo: en este primer ciclo se habla de la ejecución de requisitos donde se recopilan la información se analiza la misma y se formula los requerimientos que debe cumplir basado en los requerimientos del cliente, también se habla de los lenguajes de programación a implementar teniendo presente que todos los lenguajes tienen sus limitaciones y sus orientaciones partiendo de esta se busca los lenguajes de programación adecuados para el desarrollo del software, de igual forma se debe tener claro que el código que se desarrolla debe ser legible y claro con una buena estructuración, debe contar con una lógica correcta para una mayor claridad y entendimiento del mismo y por ultimo se debe comentar el código de manera que se pueda "dividir cada función del código".
b. Control de versiones: se maneja por medio de un repositorio de código como puede ser GitHub, lo cual permite tener un mayor control de las actualizaciones que se hagan del software para también facilitar el proceso de manteamiento y de las copias de seguridad.
c. Pruebas de integración: permite la identificación de errores por medio de la unión de diversos componentes que van a conformar el sistema para ver cuales son compatibles y medir el nivel de adaptación de los mismo permitiendo la funcionalidad total del diseño del software. posteriormente se realiza un test interno del software donde se pone a prueba cada componente del sistema su correcto funcionamiento colocando el sistema al limite para la medición de las limitaciones del mismo.
d. Despliegue del software: se le conoce como el ciclo donde el software entra en funcionamiento, aquí es donde los usuarios cumplen un importante papel ya que ellos son los que van a dar el visto bueno o malo del mismo, pueden dar una apreciación del mismo o proponer mejoras.
e. Operación y monitoréamelo del software: cuando se habla de esta ultima etapa del ciclo se conoce como el control constante del funcionamiento del sistema, en donde se puede eliminar posibles fallas sin afectar a los usuarios de manera definitiva, se realiza mantenimientos periódicos del mismo[99].

Para la empresa Microsoft el ciclo de vida se puede componer en cinco fases las cuales contempla lo siguiente:

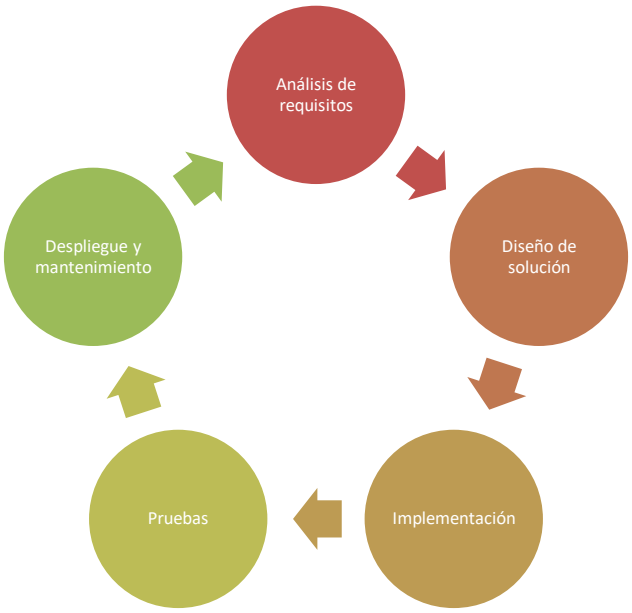
1. Requisitos del software y evaluación de amenazas: en donde se define las amenazas que puede tener el sistema.

[99] **INDIRA, BACH. Huamanchumo Becerra Heily. 2021.** [En línea] 2021.Disponible en: https://repositorio.uss.edu.pe/bitstream/handle/20.500.12802/7850/Huamanchumo%20Becerra%20Heily%20Indira.pdf?sequence=1&isAllowed=y.

2. Diseño y modelos de amenazas: es la incorporación de código para la prevención de ataques de desarrollo.
3. Desarrollo y análisis estático de seguridad: se revisa el código desarrollado de manera escalable para una codificación segura.
4. Pruebas de funcionamiento y análisis dinámico de seguridad: se comprueba la ejecución del desarrollo con el fin de verificar el funcionamiento.
5. Producción control y evaluación de seguridad: se verifica las vulnerabilidades contempladas en la evaluación de riesgos basándonos en OWASP.[100]

Resumiendo, se puede simplificar el ciclo de vida con la siguiente imagen, se puede observar que componentes o acciones se deben tener al momento de realizar el ciclo de vida de un SW con relación a la seguridad y prevención de la misma.

Ilustración 3 Ciclo de vida

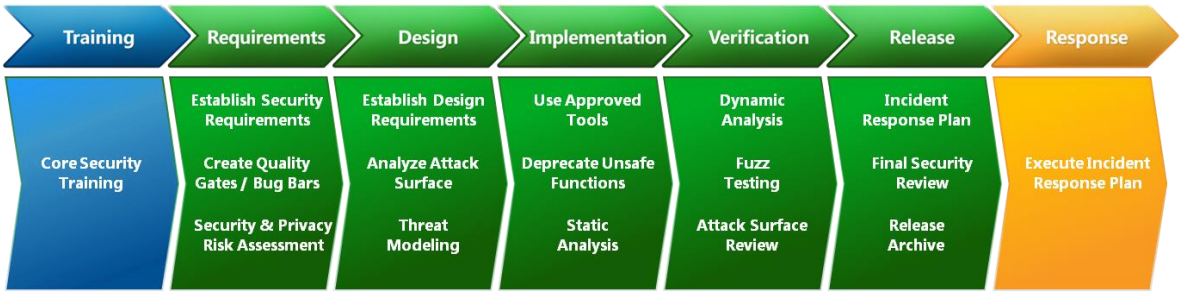


FUENTE 3: DÍAZ AYALA, Santiago Elías y Castaño Castaño, Diego Adrian. 2020. [En línea] 2020. Disponible es: https://dspace.tdea.edu.co/bitstream/handle/tdea/926/Desarrollo%20Software.pdf?sequence=1&isAllowed=y

Adicional a ello podemos apoyarnos con la metodología SDL en conjunto a la de OWASP la cual nos permite tener un óptimo ciclo de vida

[100] **DÍAZ AYALA, Santiago Elías y Castaño Castaño, Diego Adrian. 2020.** [En línea] 2020. Disponible es: https://dspace.tdea.edu.co/bitstream/handle/tdea/926/Desarrollo%20Software.pdf?sequence=1&isAllowed=y.

Ilustración 3: Implementación apoyados en los lineamientos OWASP



Fuente3: Gutierrez, Antonio Froufe. 2018.[En línea] Disponible en: https://betabeers.com/blog/microsoft-ciclos-vida-desarrollo-seguro-350/.

En la ilustración se puede observar la forma correcta que Microsoft implementa el ciclo de vida del software, en donde parte de la formación y respuesta, con el antecedente que se tiene presente las fases de requisitos del software, el diseño del mismo, la implementación y comprobación del correcto funcionamiento de la aplicación y su posterior lanzamiento, teniendo claro todas las fases anteriores y comportamiento de las mismas, en relación al OWASP se observa que tiene presente el TOP Ten que OWASP establece para la comprobación de falencias de seguridad.

7.3. CICLOS DE VIDA CON RELACIÓN A OWASP.

- CLASP es un proyecto relacionado con OWASP la cual establece una serie de actividades de buenas prácticas de desarrollo dirigidas a la coordinación de procesos seguros de software, con objetivos claros como son la distribución de funciones, valoración de procesos aplicables, implementación de estos procesos y el listo de los problemas presentados durante el proceso para la detección de vulnerabilidades.

- El Secure Software Development Framework (SSDF) provee indicaciones a la organización para la implementación de la seguridad informática cuyo objetivo es proteger el uso habitual del software facilitando la detección de ataques y su posterior solución de manera eficaz y rápida[101].

[101] **IT SOLUTIONS DE BETWEEN. 2020. [En línea] Disponible en: https://impulsate.between.tech/tecnicas-desarrollo-seguro-software.**

8.CONCLUSIONES

Se analizo diferentes documentos referentes a la metodología OWASP para el mejoramiento de la seguridad en aplicaciones o sitios web como elemento estratégico para la identificación de vulnerabilidad con el estándar ASVS se logra identificar que esta metodología permite a las organizaciones desarrolladoras de estas aplicaciones o website mejoren el nivel de seguridad desde el código como desde la publicación, evidenciando que la metodología OWASP permite generar estrategias significativas de seguridad para este tipo de aplicaciones; esto se logra por medio del estudio de los riesgos más comunes actualmente en el desarrollo de aplicaciones web.

En el presente trabajo se identifico que la metodología OWASP es un referente para el mejoramiento de la seguridad en las aplicaciones web, permitiendo así aplicar procedimientos que garanticen el aseguramiento de la información y ataques cibernéticos que puedan generarse, lo cual proporciona un seguimiento y posterior documentación de antecedentes para luego realizar una retroalimentación de las fallas presentadas durante los incidentes de seguridad.

Se estableció que las organizaciones en la actualidad optan por tener sitios o aplicaciones web generadas a partir de gestores de contenido, sin tener presente la seguridad de las mismas, ya que al momento de crear utilizando estos gestores se generan códigos innecesarios volviendo a las aplicaciones web mucho más vulnerables de lo habitual, lo cual permite a los ciberdelincuentes se aprovechen de estas falencias de seguridad.

Es por estos gestores de contenido que es importante tener en cuenta todos los riesgos que tienen y que están expuestos estos sitios web junto con las organizaciones ya que al no tener presente estos riesgos se encuentran mayormente expuestos a tener fallas catastróficas, por esto OWASP da un TOP TEN de las fallas de seguridad más comunes en las cuales dan una forma de como evaluarlas y como mitigarlas al mismo tiempo permitiendo que al momento de realizar una auditoria esta se pueda hacer más fluidamente y adicional a ello permite a las organizaciones empezar a mejorar dichas falencias para quitarlas dentro de los sistemas, logrando así un mayor porcentaje de confiabilidad a los usuarios.

Por concerniente OWASP nos permite crear planes de mitigación y de autoría periódica lo cual genera mayor uso de herramientas de auditoria y mayor seguridad de las aplicaciones que se implementen dentro de las organizaciones, causando que los usuarios estén menos expuestos a delincuentes y posteriormente que la organización no se vea afectada por estas falencias de seguridad.

OWASP es la metodología más adecuada al momento de identificar y catalogar las vulnerabilidades que tienen las aplicaciones web, convirtiéndose asi en un apoyo enorme para los desarrolladores junto con las organizaciones, esto se debe a que OWASP y el TOP TEN de las misma, permite tener conocimiento y identificación de las vulnerabilidades mayormente atacadas por los delincuentes, por otro lado OWASP permite crear un plan de acción y mitigación de las mismas ya que presta primero un amplio circulo de información dentro de foros con personas expertas en el tema de seguridad y segundo permite tener presente como se puede mitigar y corregir falencias de seguridad.

9.RECOMENDACIONES

Partiendo que la carta de prestación de las organizaciones hablando a nivel de internet es su sitio web crece la necesidad de tener un aplicaciones con buenos estándares de seguridad esto con el fin de generar una mayor confianza a los usuarios y a sus empleados al manejar dichas plataformas web para tener contacto con los clientes por ello se recomienda al personal encargado de la seguridad de la información hacer uso de esta metodología para hacer la auditoria de las aplicaciones o sitios web que estén funcionando y que sean de vital importancia para las organismos, con el fin de identificar y adaptar las necesidades que tienen para la mitigación de los posibles riesgos y falencias que se presenten en los sistemas destinados a entornos web.

Al nivel de herramientas idóneas para la identificación y realización de pruebas de penetración de seguridad, se recomienda analizarlas y catalogarlas en función de una correcta aplicación de las mismas, teniendo presente que no todas las herramientas se aplican para la identificación de vulnerabilidades, ya que hay herramientas de extracción y otras de atacas, que, aunque se vean que se enfocan para lo mismo, pero cada una de ellas funcionan de forma diferente.

Por otro lado, se recomienda al profesional de seguridad de la información realizar pruebas periódicas de pentests en búsqueda de vulnerabilidades en las aplicaciones o sitios web basándose en la metodología OWASP ASVS, teniendo presente el top ten que sugiere la metodología.

También se debe generar políticas dentro de la organización y para los usuarios para la utilización de contraseñas seguras y cambio periódico de las mismas,

seguido de una revisión de cookies del sitio web junto con la inclusión de formularios de inicio de sesión tanto para los empleados como para los usuarios externos.

Por concerniente, las organizaciones deben hacer la implementación de recursos o herramientas que permitan detectar las posibles falencias/vulnerabilidades que se puedan tener dentro de las aplicaciones web, y que por medio de estas puedan vulnerar los sistemas de información de las organizaciones, al hacer uso de estos recursos las organizaciones pueden prevenir amenazas que puedan afectarlas.

10.DIVULGACIÓN

El desarrollo del presente proyecto de grado será dado a conocer en colaboración de la biblioteca de la Universidad Nacional Abierta y a Distancia – UNAD, a través de su aplicativo en línea, en donde se publicará un archivo PDF correspondiente al documento final presentado ante los jurados, posterior a la sustentación de este (Si es informe técnico por seminario o créditos de maestría, no tiene jurado); con el fin de que todos los estudiantes de la Universidad que se encuentren interesados en el tema de OWASP COMO ELEMENTO ESTRATÉGICO EN LA IDENTIFICACIÓN DE VULNERABILIDADES Y LA VALIDACIÓN DE SEGURIDAD EN EL DISEÑO, PROGRAMACIÓN Y OPERACIÓN DE APLICACIONES SEGURAS EN LAS ORGANIZACIONES DESARROLLADORAS DE SOFTWARE EN COLOMBIA., puedan acceder al documento.

11.BIBLIOGRAFÍA

AGUIRRE MANUEL FERNANDO MARULANDA y DIAZ MONTES JACOBO [En línea]. - 2018. - https://repository.unad.edu.co/bitstream/handle/10596/20479/1060648494.pdf?sequence=3&isAllowed=y#page=150&zoom=100,148,270.

Alejandro Parra Tapia Erik y Díaz Ortiz Daniel Giovanny dspace.ups.edu.ec [En línea]. - Febrero de 2020. - 7 de Mayo de 2021. - http://dspace.ups.edu.ec/handle/123456789/18395.

Aliaga JUAN JOSÉ Romero [En línea]. - 2019. - https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/2078/Juan%20Romero_Trabajo%20de%20Investigacion_Maestria_2019.pdf?sequence=1&isAllowed=y#page=31&zoom=100,81,736.

Altamirano Ruiz Marco Vinicio y Bonilla Vaca Carolina Anabel Repositorio Universidad Técmica de Ambato [En línea]. - 2017. - http://repositorio.uta.edu.ec/handle/123456789/24534.

ANDRÉS OSORIO GUTIÉRREZ DIEGO [En línea]. - 2020. - https://repository.udistrital.edu.co/handle/11349/25722.
Andrés Rubén [En línea]. - 2016. - https://computerhoy.com/paso-a-paso/software/que-es-kali-linux-que-puedes-hacer-41671.

Angel Eulises Ortiz [En línea]. - 22 de Junio de 2020. - https://www.hostdime.com.pe/blog/que-es-una-vulnerabilidad-en-seguridad-informatica-ejemplos/.

Asdrubal| Guayara Rubio Respositorio UNAD [En línea]. - 22 de Abril de 2019. - https://repository.unad.edu.co/bitstream/handle/10596/25233/%20%09aguayarar.pdf?sequence=1&isAllowed=y.

BARRA AL FARUQ [y otros] [En línea]. - 2020. - http://www.warse.org/IJATCSE/static/pdf/file/ijatcse157932020.pdf.

Beltrán Caucali Diana Marcela [En línea]. - 2020. - https://repository.unad.edu.co/handle/10596/38709.

Brito Abundis Carlos Joaquín [En línea]. - 2013. - https://www.redalyc.org/pdf/5122/512251564005.pdf.

Brito Henry Raúl Gonzáles y Montesino Perurena Raydel http://scielo.sld.cu/ [En línea]. - 2018. - http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992018000400005.

Castañeda Suárez Andrés Fernando [En línea]. - 2017. - https://repository.unimilitar.edu.co/handle/10654/16513.

CHAVARRIA GONZALEZ Victor [En línea]. - 2018. - https://dspace.uib.es/xmlui/handle/11201/151259?show=full.

Constitución Política de Colombia constitucioncolombia.com [En línea]. - https://www.constitucioncolombia.com/titulo-7/capitulo-5/articulo-209.

Constitucón Política de Colombia constitucioncolombia.com [En línea]. - https://www.constitucioncolombia.com/titulo-10/capitulo-1/articulo-269.

David Múnera Álvarez Jesús y Uribe Arango Christian David dspace.tdea.edu.co [En línea]. - 28 de Agosto de 2020. - https://dspace.tdea.edu.co/handle/tdea/1081.

David RODRIGUEZ PARRA Jesus [En línea]. - 2018. - https://www.researchgate.net/profile/David_Parra14/project/Implementation-of-The-OWASP-Insurance-Mobile-Development-Guidelines/attachment/58e7971f1042bf08bcde4368/AS:480504093908992@1491572511159/download/Implementation+of+The+OWASP+Insurance+Mobile+Devel.

Desarrollo Web [En línea]. - 24 de Agosto de 2020. - https://profile.es/blog/desarrollo-aplicaciones-web/.

Días Montes Jacobo y Marulanda Aguirre Manuel Fernando [En línea]. - 07 de Septiembre de 2018. - https://repository.unad.edu.co/handle/10596/20479.

Díaz Ayala Santiago Elías y Castaño Castaño Diego Adrian [En línea]. - 2020. - https://dspace.tdea.edu.co/bitstream/handle/tdea/926/Desarrollo%20Software.pdf?sequence=1&isAllowed=y.

e Creative Commons OWASP.ORG [En línea]. - https://owasp.org/www-pdf-archive/Gu%C3%ADa_de_pruebas_de_OWASP_ver_3.0.pdf.

Edgardo Bernardis [y otros] [En línea]. - Abril de 2017. - http://sedici.unlp.edu.ar/handle/10915/62726.

Edwin MELGAREJO Martinez [En línea]. - 2018. - https://repository.unad.edu.co/bitstream/handle/10596/21360/80125726.pdf?sequence=1&isAllowed=y.

Edwin Melgarejo Martinez repositorio unad [En línea]. - 2018. - 18 de Mayo de 2021. - https://repository.unad.edu.co/bitstream/handle/10596/21360/80125726.pdf?sequence=1&isAllowed=y.

Esquivel Cabezas Harold Alfredo y Lozano Olivares Jesús Herney [En línea]. - 25 de Septiembre de 2020. - https://repository.unad.edu.co/handle/10596/36704.

Fabricio Rodríguez Zambrano Stalyn [En línea]. - 2019. - https://repositorio.itb.edu.ec/bitstream/123456789/2679/1/PROYECTO%20DE%20GRADO%20DE%20RODR%c3%8dGUEZ%20ZAMBRANO%20STALYN%20FABRICIO.pdf.

Fernández Mahecha Edwin Neyid y Llano Ruiz Anderson Julian [En línea]. - Junio de 2018. - https://repository.ucatolica.edu.co/bitstream/10983/16045/1/TrabajoDeGA.

Fernández Miranda Henry Armando [En línea]. - 2019. - Fernández Miranda, Henry Armando.

GARCÍA LUZ ANGELA ROBAYO [En línea]. - 2021. - https://repository.unad.edu.co/handle/10596/40343.

Girald Luis Fernando Garcés, Sepúlveda Aguirre Jovany Arley y Melguizo Múnera Daniela americana.edu.co [En línea]. - 2020. - https://americana.edu.co/medellin/wp-content/uploads/2020/12/Pra%CC%81cticas-y-resultados-en-formacio%CC%81n-investigativa.-Semilleros-de-investigacio%CC%81n-generando-conocimiento-completo.pdf#page=272.

González Mendoza Dewin Fernando [En línea]. - 2017. - https://repository.unad.edu.co/handle/10596/17397.

Gutierrez Antonio Froufe [En línea]. - 2018. - https://betabeers.com/blog/microsoft-ciclos-vida-desarrollo-seguro-350/.

HAMNER BRIONES PINCAY GERSON y HERNANDEZ PEÑAHERRERA ERIKA BELÉN [En línea]. - 2018. - http://repositorio.ug.edu.ec/bitstream/redug/26837/1/B-CINT-PTG-N.249%20Briones%20Pincay%20Gerson%20Hamner.%20Hern%c3%a1ndez%20Pe%c3%b1a%20herrera%20Erika%20Bel%c3%a9n.pdf.

Hernández Peñaherrera Erika Belén y Briones Pincay Gerson Hammer [En línea]. - 2018. - http://repositorio.ug.edu.ec/handle/redug/26837.

Indira Bach. Huamanchumo Becerra Heily [En línea]. - 2021. - https://repositorio.uss.edu.pe/bitstream/handle/20.500.12802/7850/Huamanchumo%20Becerra%20Heily%20Indira.pdf?sequence=1&isAllowed=y.

INFOSEC Blog.segu-info.com.ar [En línea]. - 1 de Febrero de 2021. - https://blog.segu-info.com.ar/2021/02/owasp-top-10-2021-propuesta-y.html.

IT SOLUTIONS DE BETWEEN [En línea]. - 2020. - https://impulsate.between.tech/tecnicas-desarrollo-seguro-software.

IVAN CORONEL SUÁREZ y GARCÍA PERERO FREDDY GIANCARLO [En línea]. - 2021. - https://repositorio.upse.edu.ec/handle/46000/5917.

Javier Valencia Duque Francisco y Orozco Alzate Mauricio [En línea]. - 1 de Marzo de 2017. - https://www.researchgate.net/profile/Francisco-Valencia-Duque/publication/318204876_A_methodology_for_implementing_an_information_security_management_system_based_on_the_family_of_ISOIEC_27000_standards/links/5fd9d2ea299bf1408811f7b3/A-methodology-for-imp.

JOHANES FERNADES Andry y HENNY Hartono [En línea]. - 2017. - https://www.researchgate.net/profile/Johanes-Andry/publication/320254270_Performance_Measurement_of_IT_Based_on_COBIT_Assessment_A_Case_Study/links/59d84f18a6fdcc2aad065a7c/Performance-Measurement-of-IT-Based-on-COBIT-Assessment-A-Case-Study.pdf.

Juliana García Zapata [En línea]. - 2018. - https://repository.unad.edu.co/bitstream/handle/10596/28466/1033648651.pdf?sequence=1&isAllowed=y.

LEY ESTATUTARIA 1266 DE 2008 [En línea]. - Diciembre de 2008. - https://tic.bogota.gov.co/node/137.

Lizeth Santillán Mosquera Ángela repository.unad.edu.co [En línea]. - 15 de Diciembre de 2019. - 7 de Mayo de 2021. - https://repository.unad.edu.co/bitstream/handle/10596/31771/alsantillanm.pdf?sequence=1&isAllowed=y.

Llerena Alain Eduardo Rodríguez [En línea]. - 2020. - http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-18592020000100116.

LONDOÑO JEFFERSON GONZALEZ [En línea]. - 2020. - https://repository.unad.edu.co/bitstream/handle/10596/36669/jgonzalezlon.pdf?sequence=1&isAllowed=y#page=48&zoom=100,148,186.

López Sevilla Galo Mauricio y Gambia Safla Diego Leonardo [En línea]. - 2021. - https://repositorio.pucesa.edu.ec/handle/123456789/3175.

M. Elhadi, Shakshuki y Ansar Yasar sciencedirect.com [En línea]. - Noviembre de 2019. - https://www.sciencedirect.com/science/article/pii/S1877050920323589.

Maida Esteban Gabriel y Pacienzia Julián [En línea]. - 2015. - https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf.

Maldonado Manuel [En línea]. - 2018. - https://www.digital55.com/desarrollo-tecnologia/mejores-metodologias-agiles-creacion-software/.

Marcela Caucali Beltrán Diana repositorio UNAD [En línea]. - 2020. - 18 de Mayo de 2021. - https://repository.unad.edu.co/bitstream/handle/10596/38709/dmcaucalib.pdf?sequence=1&isAllowed=y.

Marcela Caucali Beltrán Diana Repositoriy.unad.edu.co [En línea]. - 2020. - 2021. -

https://repository.unad.edu.co/bitstream/handle/10596/38709/dmcaucalib.pdf?sequence=1&isAllowed=y.

Marcela Perez Sanches Laura Repositorio Universidad Francisco de Paula Santander [En línea]. - 8 de Agosto de 2019. - http://repositorio.ufpso.edu.co/jspui/handle/123456789/1022.

Maury Julien [En línea]. - 2021. - https://www.esecurityplanet.com/applications/owasp-list-gets-a-new-top-vulnerability/.

MERIAH INES y Arfa Rabai Latifa Ben [En línea]. - 2019. - https://doi.org/10.1016/j.procs.2019.09.447.

Moisés Delgado Basurto Jonathan [En línea]. - Enero de 2020. - https://repositorio.uleam.edu.ec/handle/123456789/2068.

Molina García Jorge Alberto [En línea]. - 2019. - http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/6754/LA%20IMPORTANCIA%20DE%20LA%20GESTI%c3%93N%20DE%20RIESGOS%20Y%20SEGURIDAD%20EN%20EL%20INTERNET%20DE%20LAS%20COSAS%20%28IOT%29-%20Jorge%20Molina%20Garc%c3%ada%20ESI41.pdf?sequence=1&isAll.

Muñoz Mayorga Andrés Felipe y Pérez Solarte Santiago Alejandro Repositorio.unicauca.edu.co [En línea]. - Septiembre de 2017. - http://repositorio.unicauca.edu.co:8080/bitstream/handle/123456789/1773/PENTESTING%20SOBRE%20APLICACIONES%20WEB%20BASADO%20EN%20LA%20METODOLOG%c3%8dA%20OWASP%20UTILIZANDO%20SBC%20DE%20BAJO%20COSTO.pdf?sequence=1&isAllowed=y.

Nemury Silega Martínez y García Rodríguez Ana Marys repositorio.uci.cu [En línea]. - 2018. - https://repositorio.uci.cu/handle/123456789/7916.

NEYID FERNÁNDEZ MAHECHA EDWIN y LLANOS RUIZ ANDERSON JULIÁN [En línea]. - 2018. - https://repository.ucatolica.edu.co/handle/10983/16045.

OWASP [En línea]. - Abril de 2017. - https://owasp.org/www-pdf-archive/Est%C3%A1ndar_de_Verificaci%C3%B3n_de_Seguridad_en_Aplicaciones_3.0.1.pdf.

Pérez Barrera Dennis, González Brito Henry Raúl y Sánchez Borrell Yailin [En línea]. - 22 de Enero de 2019. - https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/306.

Rendón Tacle Jean Carlos y Raza Rivas José Steven [En línea]. - 30 de Agosto de 2019. - http://repositorio.ug.edu.ec/handle/redug/45164.

REVISTA VINCULOS [En línea]. - 1 de Julio de 2019. - https://web-b-ebscohost-com.bibliotecavirtual.unad.edu.co/ehost/pdfviewer/pdfviewer?vid=7&sid=5c013c36-8b99-4369-9bd5-223dbdc233f1%40pdc-v-sessmgr03.

REVISTA VINCULOS [En línea]. - 1 de Julio de 2019. - https://web-b-ebscohost-com.bibliotecavirtual.unad.edu.co/ehost/pdfviewer/pdfviewer?vid=7&sid=5c013c36-8b99-4369-9bd5-223dbdc233f1%40pdc-v-sessmgr03.

Robayo Bautista Eliana Catherine [En línea]. - 2021. - https://repository.ucatolica.edu.co/bitstream/10983/25731/1/Trabajo%20de%20Grado_Eliana_robayo_Gu%2b%c2%a1a%20de%20principios_28-11-2020.pdf#page=73&zoom=100,148,314.

Rodríguez Rodríguez Rafael Enrique y Sánchez Andrés Felipe [En línea]. - 2018. - https://core.ac.uk/download/pdf/213560272.pdf.

Rojas Osorio Jorge Armando [En línea]. - 2018. - 15 de Mayo de 2021. - http://repository.unipiloto.edu.co/handle/20.500.12277/8654.

Rubio Guayara Asdrubal [En línea]. - 2019. - https://repository.unad.edu.co/handle/10596/25233.

SonarQube [En línea]. - 2021. - https://www.sonarqube.org/features/security/owasp/?gads_campaign=South-America-OWASP&gads_ad_group=OWASP&gads_keyword=owasp%20top10&gclid=Cj0KCQjwwY-LBhD6ARIsACvT72NujIizGy5TmtBk825YCLcjoaNlIN16KXxB3MwfXMkH2bl8S3BpEfIaAvtHEALw_wcB.

Superintendencia de insdustria y comercio Sic.gov.co [En línea]. - 5 de Enero de 2009. - 18 de Mayo de 2021. -

https://www.sic.gov.co/recursos_user/documentos/normatividad/Ley_1273_2009.pdf.

Tatiana Tapia Bastidas y Rodriguez Zambrano Stalyn Fabrico [En línea]. - 22 de Febrero de 2021. - http://repositorio.itb.edu.ec/handle/123456789/2679.

TITUAÑA VILLA Milton José [En línea]. - 28 de Julio de 2017. - https://bibdigital.epn.edu.ec/handle/15000/17543.

Valentino Jaramillo Quirama Yeisson [En línea]. - 19 de Octubre de 2019. - https://repository.unad.edu.co/handle/10596/28259.

Víctor González Chavarría [En línea]. - 2018. - https://dspace.uib.es/xmlui/bitstream/handle/11201/151259/Memoria_EPSU0643.pdf?sequence=1&isAllowed=y.

WELIVESECURITY BY ESET [En línea]. - 25 de Febrero de 2015. - https://www.welivesecurity.com/la-es/2015/02/25/que-es-un-0-day/.

YISEL BENITEZ NIÑO y SILEGA MARTÍNEZ NEMURY [En línea]. - 2018. - http://scielo.sld.cu/scielo.php?pid=S2227-18992018000500015&script=sci_arttext&tlng=pt.

Yucenid Vanegas Romero Alfonso repository.unipiloto.edu.co [En línea]. - 2019. - http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/6286/00005220.pdf?sequence=1&isAllowed=y.

Zorz Zeljka [En línea]. - 2021. - https://www.helpnetsecurity.com/2021/09/24/owasp-top-10-2021/.

Zúñiga Mosquera Jhonnier Yesid [En línea]. - 16 de Noviembre de 2018. - https://repository.unad.edu.co/handle/10596/21574.

6.1 ANEXOS

Anexo A. Sustentación.
https://drive.google.com/file/d/1TyPnO1J1fahDojFjx5_fkxDHrITGAJmN/view?usp=sharing

Anexo B. RAE

Fecha de Realización:	30/07/2021
Programa:	Especialización en Seguridad Informática
Línea de Investigación:	Cadena de formación en electrónica, telecomunicaciones y redes
Título:	OWASP COMO ELEMENTO ESTRATÉGICO EN LA IDENTIFICACIÓN DE VULNERABILIDADES Y LA VALIDACIÓN DE SEGURIDAD EN EL DISEÑO, PROGRAMACIÓN Y OPERACIÓN DE APLICACIONES SEGURAS EN LAS ORGANIZACIONES DESARROLLADORAS DE SOFTWARE EN COLOMBIA.
Autor(es):	Santiago García Omar Camilo
Palabras Claves:	Aplicación, Informática, OWASP, Seguridad, TIC´S,
Descripción:	El presente trabajo se abordará en la metodología OWASP, como elemento estratégico para la identificación de vulnerabilidades y su posterior validación de seguridad en el diseño, programación y operación de aplicaciones web seguras dentro de las organizaciones dedicadas al desarrollo de software en Colombia. Esto se logra con el estudio y entendimiento de la metodología para realizar una auditoría y

	posterior mejoramiento de las aplicaciones, adicional a ello estudiando las posibles vulnerabilidades que se puedan presentar dentro de ellas.
Fuentes bibliográficas destacadas: **AGUIRRE, MANUEL FERNANDO MARULANDA y DIAZ MONTES, JACOBO. 2018.** [En línea] 2018. Disponible en: https://repository.unad.edu.co/bitstream/handle/10596/20479/1060648494.pdf?sequence=3&isAllowed=y#page=150&zoom=100,148,270. **ALEJANDRO, PARRA TAPIA Erik y Díaz Ortiz , Daniel Giovanny. 2020.** dspace.ups.edu.ec. [En línea] Febrero de 2020. Disponible en: http://dspace.ups.edu.ec/handle/123456789/18395. **ALIAGA, Juan José Romero. 2019.** [En línea] 2019. Disponible en: https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/2078/Juan%20Romero_Trabajo%20de%20Investigacion_Maestria_2019.pdf?sequence=1&isAllowed=y#page=31&zoom=100,81,736. **ALTAMIRANO RUIZ, MARCO VINICIO Y BONILLA VACA, Carolina Anabel. 2017.** Repositorio Universidad Técmica de Ambato. [En	
Contenido del documento:	Portada, sub portada, introducción, definición del problema, formulación del problema, justificación, objetivos, marco referencial, desarrollo de los objetivos, conclusiones, recomendaciones, bibliografías y anexos
Marco Metodológico:	No aplica
Conceptos adquiridos:	Después del desarrollo del trabajo de grado
Conclusiones:	Al analizar la documentación de la metodología OWASP para el mejoramiento de la seguridad en aplicaciones o sitios web como elemento estratégico para la identificación de vulnerabilidad con el estándar ASVS se logra identificar que esta metodología permite a las organizaciones desarrolladoras de estas aplicaciones o website mejoren el nivel de seguridad desde el código como desde la publicación, evidenciando que la metodología OWASP permite generar estrategias significativas de seguridad para este tipo de aplicaciones; esto se logra por medio del estudio de los riesgos más comunes actualmente en el desarrollo de aplicaciones web.

	Los buenos resultado que se tienen con la metodología OWASP como referente para el mejoramiento de la seguridad en las aplicaciones web permite aplicar procedimientos que garantiza el aseguramiento de la información y de ataques cibernéticos, permitiendo generar un seguimiento y posterior documentación de antecedentes y realizar una retroalimentación de las mismas fallas en la actualidad. Actualmente las organizaciones optan por tener sitios web o aplicaciones creados a partir de gestores de contenido como lo pueden ser WordPress, Joomla o cualquier otro gestor de contenido, y en aplicaciones de escritorio, partiendo de allí es donde mayormente se genera problemas y falencias de seguridad dentro de las mismas, debido a que estos gestores de contenido crean código innecesarios en muchas ocasiones lo cual permite que los ciberdelincuentes se aprovechen de esto, en la actualidad casi todos los sitios web están basados en estos gestores de contenido. Es por estos gestores de contenido que es importante tener en cuenta todos los riesgos que tienen y que están expuestos estos sitios web junto con las organizaciones ya que al no tener presente estos riesgos se encuentran mayormente expuestos a tener fallas catastróficas, por esto OWASP da un TOP TEN de las fallas de seguridad más comunes en las cuales dan una forma de como evaluarlas y como mitigarlas al mismo tiempo permitiendo que al momento de realizar una auditoria esta se pueda hacer más fluidamente y adicional a ello permite a las organizaciones empezar a mejorar dichas falencias para quitarlas dentro de los sistemas, logrando así un mayor porcentaje de confiabilidad a los usuarios. Por otro lado, las organizaciones deben hacer la implementación de recursos o herramientas que

	permitan detectar las posibles falencias/vulnerabilidades que se puedan tener dentro de las aplicaciones web, y que por medio de estas puedan vulnerar los sistemas de información de las organizaciones, al hacer uso de estos recursos las organizaciones pueden prevenir amenazas que puedan afectarlas. Por concerniente OWASP nos permite crear planes de mitigación y de autoría periódica lo cual genera mayor uso de herramientas de auditoria y mayor seguridad de las aplicaciones que se implementen dentro de las organizaciones, causando que los usuarios estén menos expuestos a delincuentes y posteriormente que la organización no se vea afectada por estas falencias de seguridad. OWASP es la metodología más adecuada al momento de identificar y catalogar las vulnerabilidades que tienen las aplicaciones web, convirtiéndose asi en un apoyo enorme para los desarrolladores junto con las organizaciones, esto se debe a que OWASP y el TOP TEN de las misma, permite tener conocimiento y identificación de las vulnerabilidades mayormente atacadas por los delincuentes, por otro lado OWASP permite crear un plan de acción y mitigación de las mismas ya que presta primero un amplio circulo de información dentro de foros con personas expertas en el tema de seguridad y segundo permite tener presente como se puede mitigar y corregir falencias de seguridad.