INSTITUTO SUPERIOR UNIVERSITARIO BOLIVARIANO DE TECNOLOGÍA

FACULTAD DE CIENCIAS ECONÓMICAS Y EMPRESARIALES

Proyecto de Investigación previo a la obtención del título de:

TECNÓLOGO EN ANÁLISIS DE SISTEMAS

TEMAS:

IMPLEMENTACIÓN DE LA HERRAMIENTA OWASP UTILIZANDO LA APLICACIÓN SPIDER PARA EL ESCANEO Y ATAQUES PASIVOS DE VULNERABILIDADES EN EL SISTEMA DE GESTIÓN ACADÉMICO DEL "ITB"

Autor: Rodríguez Zambrano Stalyn Fabricio

Tutora: Ph.D. Tapia Bastidas Tatiana

Guayaquil, Ecuador

2019

DEDICATORIA

El presente trabajo de titulación va dedicado a Dios como parte fundamental en mi formación espiritual, mis padres que siempre han estado apoyándome en cada momento de mi vida, mi familia que siempre está pendiente de mi progreso y sobretodo dándome aliento en toda circunstancia que la vida me otorga.

Stalyn Fabricio Rodríguez Zambrano

AGRADECIMIENTOS

A Dios por las bendiciones concedidas en todo este tiempo de viaje por la vida y de todo el conocimiento adquirido.

A mis padres por ser la fuente principal de mi educación, los grandes mentores en mi vida, los cuales me inculcaron buenas prácticas profesionales y humanísticas, de innovar constantemente y no quedarse sin hacer nada, con los brazos cruzados ante la adversidad.

A mi familia por estar siempre motivándome a ser mejor cada día.

A mis docentes por cada enseñanza transmitida y por ser parte integral del conocimiento adquirido a lo largo del transcurso de mi paso estudiantil por el ITB.

A Mi tutora de Tesis Ph.d Tatiana Tapia por la constate guía y asesoría brindada en el desarrollo de mi trabajo de titulación.

Al Instituto Superior Universitario Bolivariano por fomentar mi formación como un profesional y permitirme efectuar mi trabajo de titulación y otorgarme todas las facilidades para el desarrollo final de la misma.

Stalyn Fabricio Rodríguez Zambrano

INSTITUTO SUPERIOR UNIVERSITARIO BOLIVARIANO DE TECNOLOGÍA

FACULTAD DE CIENCIAS ECONÓMICAS Y EMPRESARIALES

Proyecto de Investigación previo a la obtención del título de:

TECNÓLOGO EN ANÁLISIS DE SISTEMAS

TEMA: IMPLEMENTACIÓN DE LA HERRAMIENTA OWASP UTILIZANDO LA APLICACIÓN SPIDER PARA EL ESCANEO Y ATAQUES PASIVOS DE VULNERABILIDADES EN EL SISTEMA DE GESTIÓN ACADÉMICO DEL "ITB"

Autor: Rodríguez Zambrano Stalyn Fabricio

Tutora: PhD. Tapia Bastidas Tatiana

Resumen

El Instituto Superior Tecnológico Bolivariano utiliza el Sistema de Gestión Académico “SGA” al ser una aplicación web puede presentar vulnerabilidades creando inestabilidad, inseguridad y miedo en los propietarios y usuarios que conforman el ecosistema informático del “SGA”. El propósito de la investigación estuvo orientado en la implementación de la herramienta OWASP utilizando la aplicación Spider para el escaneo y ataques pasivos de Vulnerabilidades en el Sistema de Gestión Académico "SGA". Los tipos de investigación utilizados fueron de tipo exploratoria, descriptivo, de campo y la investigación bibliográfica. La conclusión más relevante a la que se llegó, fue aplicar el uso de la herramienta OWASP para el escaneo de vulnerabilidades del “SGA”. Se propuso como alternativa a solución debe implementar políticas de desarrollo seguro utilizando las normativas OWASP.

Aplicación Web | Vulnerabilidades | Desarrollo Seguro

INSTITUTO SUPERIOR UNIVERSITARIO BOLIVARIANO DE TECNOLOGÍA

FACULTAD DE CIENCIAS ECONÓMICAS Y EMPRESARIALES

Proyecto de Investigación previo a la obtención del título de:

TECNÓLOGO EN ANÁLISIS DE SISTEMAS

TEMA: IMPLEMENTACIÓN DE LA HERRAMIENTA OWASP UTILIZANDO LA APLICACIÓN SPIDER PARA EL ESCANEO Y ATAQUES PASIVOS DE VULNERABILIDADES EN EL SISTEMA DE GESTIÓN ACADÉMICO DEL "ITB"

Autor: Rodríguez Zambrano Stalyn Fabricio

Tutora: PhD. Tapia Bastidas Tatiana

Abstract

The Instituto Superior Tecnológico Bolivariano uses the Academic Management System "SGA" as it is a web application, it can present vulnerabilities creating instability, insecurity and fear in the owners and users that make up the "SGA" computing ecosystem. The purpose of the research was oriented in the implementation of the OWASP tool using the Spider application for scanning and passive attacks of Vulnerabilities in the Academic Management System "SGA". The types of research used were exploratory, descriptive, field and bibliographic research. The most relevant conclusion that was reached was to apply the use of the OWASP tool to scan the "SGA" vulnerabilities. It was proposed as an alternative solution must implement secure development policies using OWASP regulations.

Web Applications | Vulnerabilities | Secure Development

ÍNDICE GENERAL

Contenidos **Páginas:**

CAPÍTULO I

EL PROBLEMA

PLANTEAMIENTO DEL PROBLEMA

CAPÍTULO II

MARCO TEÓRICO

FUNDAMENTACIÓN TEÓRICA

CAPÍTULO III

METODOLOGÍA

CAPÍTULO IV

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

CAPÍTULO V
CONCLUSIONES Y RECOMENDACIONES

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA.

1.1 Diagnostico

El Gobierno de la República del Ecuador el día 11 de abril del 2019 toma la decisión quitar el asilo político al fundador de Wikileaks Julian Assage, esta acción emprendida por el gobierno ecuatoriano causo consecuencias, tal fue el descontento que Hactivistas de todo el mundo, provocaron que la nación ecuatoriana sufra ataques cibernéticos de forma masivo a nivel nacional en todas las plataformas Web tanto instituciones públicas como privadas, quedando así expuesta muchas brechas de seguridad informática. Este incidente marca un precedente la cual radica en la falta de cultura y la poca socialización que tienen muchos profesionales en el ámbito informático con respecto a la seguridad de la información.

La presente investigación trata sobre Implementación de la herramienta OWASP utilizando la aplicación Spider para el escaneo y ataques pasivos de Vulnerabilidades en el Sistema de Gestión Académico "SGA" perteneciente al Instituto Superior Tecnológico Bolivariano.

En un primer análisis investigativo se llevó a cabo una prueba, como parte de un estudio de vulnerabilidades en la plataforma del SGA, para tal efecto se utilizó la herramienta de auditoria de seguridad llamada OWASP ZAP 2.8.0, el análisis se llevó acabo el día 4 de septiembre del 2019, cuyos resultados de las pruebas efectuadas me dieron un índice de muestreo considerable con respecto a una vulnerabilidad de alta peligrosidad denominada falla por Inyección SQL.

Una inyección SQL se basa en la introducción de código SQL a través del navegador web para que este a su vez llegue a la base de datos y pueda modificar estructura del código fuente. Es necesario implementar la herramienta OWASP ZAP como una aplicación de análisis de vulnerabilidades de uso primario en el entorno de desarrollos Web

1.2 Ubicación en su contexto

Desde la creación de la World Wide Web por el Científico Británico Tim Berners-Lee (Berners-Lee, 1990) la Internet ha tenido un desarrollado de forma exponencial, en la actualidad tenemos una gran variedad de contenidos digitales basados en Sitios Web, en la cual están inmersas instituciones académicas a nivel global que cuentan con un sistema de gestión académico.

Los diseños de sitios Web son una fuente de información inmediata que tiene como objetivo ayudar a establecer soluciones inmediatas a muchos de los problemas existentes, en mucho de los casos es dar el uso correcto a la toda la masa de datos que procesan las Empresas, Bancos, Universidades, Institutos Tecnológicos etc. Por ende, la creación de estos sitios Web en mucho de los casos se crea omitiendo varios protocolos de desarrollo seguro y dejan en un segundo plano la seguridad.

El diseño de sitios web pueden tener estructuras en su código de programación, que presenten vulnerabilidades que a su vez puede ser el causal de una explotación de recursos informáticos, las misma que puede ser propiciado por usuarios inconscientes que gestionan el sistema o por usuarios externos como Hackers. Un Hacker puede ser una ayuda para corregir vulnerabilidades de seguridad o puede ser de gran perjuicio creando inestabilidad, inseguridad y miedo en los propietarios y usuarios que son parte de un sistema gestión informático.

Existen herramientas para el Ethical Hacking como es el OWASP Open Web Application Security Project, la cual permite hacer escaneos para identificar vulnerabilidades. Zap cuenta con una araña o spider para la detección automática y descubrimiento de nuevos recursos o URLs en el sitio web a auditor (Pablo Gonzalez, 2016).

1.3 Situación

El Instituto Superior Tecnológico Bolivariano de Tecnología, tiene un campus matriz ubicada en la ciudad de Guayaquil en las calles Víctor Manuel Rendón y Pedro Carbo, además tiene otros campus anexos: Campus Boyacá, Atarazana, y Rocafuerte. Todas estas infraestructuras físicas dan la facilidad para capacitar en estudios tecnológicos a un gran número de personas en sus distintas especialidades que oferta.

El Sistema de Gestión Académica del “SGA” se ha vuelto indispensable en muchos ámbitos de la vida estudiantil, está disponible para su acceso en cualquier momento a todos los usuarios y por supuesto puede estar expuesto algún tipo de ataque con código malicioso.

Una vulnerabilidad por un ataque de código malicioso puede exponer información importante de los usuarios, administradores y otras entidades que confían en el Sistema de Gestión Académica.

En la Actualidad el Sistema de Gestión Académica puede ser vulnerable a una validación de la entrada y salida de datos, para un hacker poder manipular la información necesita cumplir dos parámetros obligatorios: insertar un dato no seguro y manipular la página web utilizando ese dato. Un dato no seguro podría darse como ejemplos: la Inyección de código Sql, Cross-site scripting, Inyección de comandos.

1.4 Factibilidad de la implementación

El Instituto Superior Tecnológico Bolivariano de Tecnología tiene una infraestructura adecuada, el Departamentos de Tecnologías de la Información y la Comunicación (TIC’S), va a brindar apoyo a la presente investigación por que representa una oportunidad, para profundizar en el conocimiento de buenas prácticas OWASP para el desarrollo seguro de aplicaciones web.

1.5 Delimitación

1.5.1 **Campo:** Tecnología de la información y comunicación (Tic's)

1.5.2 Áreas: Seguridad Informática

1.5.3 Aspectos: Sitio Web Institucional

1.5.4 Tiempo: 2019

1.6 Formulación

¿Cómo incide la detección de vulnerabilidades de manera temprana en la seguridad de la data del sistema de gestión académico "SGA"?

1.7 Definición de variables

1.7.1 Independiente

Vulnerabilidades en el sistema de gestión del ITB.

1.7.2 Dependiente

Implementación de software abierto de seguridad de aplicaciones web.

1.8 Objetivos

1.8.1 Objetivo General

Implementar la herramienta OWASP utilizando la aplicación Spider para el escaneo y ataques pasivos de Vulnerabilidades en el Sistema de Gestión Académico del "ITB".

1.8.2 Objetivos específicos

- Identificar la información científica disponible con respecto a la implementación de la herramienta OWASP y su aplicación integrada spider.
- Diagnosticar las vulnerabilidades del sistema de gestión académica a través del escaneo y ataque pasivos de la herramienta OWASP y su aplicación spider.
- Proponer incluir en los protocolos de seguridad informática del ITB la implementación del escaneo y ataques pasivos mediante la herramienta OWASP y su aplicación integrada spider.

1.9 Justificación

El presente estudio tiene como justificación realizar una valoración del sistema de gestión académico “SGA” ya que en la actualidad esta página web proporciona servicios tales como consultar registros académicos, Finanzas, Consultas generales etc. Todos estos datos son sensibles ya que por cada usuario existe una cuenta creada y asociada al servidor web del sistema de gestión académico.

Según la publicación del diario El Universo (Anonimo, EL UNIVERSO, 2019) indica que:

Información personal de casi "cada ciudadano ecuatoriano" se ha encontrado expuesta en línea, un hallazgo realizado por la compañía de seguridad vpnMentor, asegura un reporte de la BBC.

Esta información mostrada por el diario de mayor circulación del país “EL UNIVERSO” expone la vulnerabilidad de un sitio web. Por ello se recalca la necesidad imperativa de realizar un escaneo y ataque pasivo al sistema de gestión académico “SGA” con la herramienta OWASP con el fin de diseñar un modelo adecuadas y eficiente para la detección oportuna de vulnerabilidades y evitar riesgo o fallos informáticos a nivel de seguridad el cual tenga un acto consecuente para todos los usuarios que usan este sistema de gestión académico “SGA”.

1.9.1 Conveniencia

A nivel mundial en la actualidad existe un índice elevado de código malicioso creado por personas con un alto grado de conocimiento y preparación para realizar ataques a infraestructuras de software, los cuales toman ventaja de códigos de programación mal implementados por los desarrolladores de aplicaciones. Implementar una herramienta como OWASP que ayude al desarrollo seguro de aplicaciones web, podría beneficiar en mejorar aspectos, atributos y características de suma importancia en la creación de sistemas informáticos menos propensos a vulnerabilidades de códigos maliciosos.

1.9.2 Relevancia social

La relevancia social es fortalecer el conocimiento con el uso de herramientas de hacking ético en detección temprana de vulnerabilidades, encontrar los fallos de seguridad en el desarrollo de código en aplicaciones web y poder establecer políticas seguras en las futuras implementaciones de código en la plataforma web del Instituto Superior Tecnológico de Tecnología “SGA”.

1.9.3 Implicaciones prácticas

En este caso de estudio la implementación de esta Herramienta OWASP ayudara acrecentar el conocimiento, en técnicas de desarrollo seguro de códigos de programación y procesos de almacenamiento de datos, con el fin tener una visión global de todas las operaciones que realiza el sistema, es decir tener una plataforma web automatizada y las decisiones que se tomen de forma oportuna, en caso de un suceso inesperado sea este causada por uno o varios entes externos al sistema como los Hackers.

1.9.4 Utilidad metodológica

Esta investigación con la herramienta OWASP ayudara a crear nuevos procesos en la recolección de datos, los cuales serán tomados como un muestreo para la presentación de resultados a través de representaciones estadísticas y gráficas. Estos resultados pueden ser la base de estudio para otros investigadores en otras instituciones u entidades que deseen aprender la utilización de esta herramienta tecnológica para la detección de vulnerabilidades en desarrollo Web, en el marco teórico y práctico de un sistema de gestión académico.

CAPITULO II

2. MARCO REFERENCIAL

2.1 Fundamentación Teórica

2.1.1 Antecedentes históricos

A finales del año 1943 inmerso en la Segunda Guerra Mundial el matemático Británico Alan Turing crea una maquina llamada Bombé y logra descifrar los códigos de la maquina Enigma construida por los alemanes – Nazis, con lo cual logra salvar muchas vidas y dar inicio para que finalice la guerra. Históricamente se considera al Matemático Alan Turing como padre de la Informática, precursor de la Inteligencia Artificial y el primer Hacker en la historia de la humanidad.

Los fundamentos básicos para la seguridad informática fueron creados por primera vez el 15 de abril de 1980 por James P. Anderson cuyo estudio se titula " **Computer Security Threat Monitoring and Surveillance** ". En este documento se expone por primera vez los términos y conceptos de:

- Acceder a la información
- Manipular la información
- Convertir un sistema en utilizable o no utilizable

Además, se exponen otras terminologías como:

- Riesgo o exposición accidental
- Vulnerabilidad o falla conocida
- Ataque o amenaza especifica
- Penetración o ataque exitoso

Ciberseguridad en Ecuador según el Ministerio de Telecomunicaciones y la sociedad de la información comunican que Ecuador ocupa el sexto puesto de 19 países en Latinoamérica como uno de los países con mejores índices en Ciberseguridad. (Anonimo, https://www.telecomunicaciones.gob.ec/, 2017)

El último trimestre del 2019 los navegadores web sufrieron el 13.4% de ataques informáticos, en este informe no detalla el tipo de aplicaciones web que tuvieron problemas (Anonimo, precisesecurity, 2019).

Precisesecurity.com es un sitio web dedicado a dar una solución eficaz a la eliminación de virus y resolver problemas moderados a complejos. Utilizando técnicas de eliminación manual para usuarios expertos y herramientas para eliminar amenazas como una ayuda a los usuarios principiantes (Mathew, 2019)

Según de acuerdo a los diferentes estudios realizado por los investigadores académicos y científicos que conforman el grupo de trabajo del sitio web de seguridad informática precisesecurity.com, hacen referencia a que la mayoría de aplicaciones Web que existen a nivel mundial, tienen vulnerabilidades que son sensibles a la perdida de datos, con lo cual crea una inseguridad en los usuarios que utilizan estas plataformas web.

Según el estudio realizado por (Ilic, 2019) indica que:

Los ciberataques se han dirigido a casi el 75 por ciento de las grandes empresas en Europa y América del Norte en los últimos doce meses. Según la investigación por Precise Security casi el 40 por ciento de los ataques se efectuaron mediante secuencias de comando Cross-Site Scripting que es el vector de ataque favorito de los piratas informáticos a nivel mundial.

Firewall o Cortafuegos

Introducción

A partir del punto evolutivo de las redes de comunicación y de la seguridad informática, comprendemos que estamos cada vez más inmersos en un mundo tecnológico que gira en torno al internet, con millones de usuarios que utilizan redes sociales, sitios web de información, aplicaciones web etc. En la actualidad tenemos una idea vaga y poco realista del internet en pleno siglo XXI ¿porque razón?, Porque aún no sabemos cuál será el alcance, el crecimiento y los posibles límites que el internet tendrá en los próximos años. La innovación que hoy

en día piensa el hombre es llevar el internet al espacio profundo para comunicarnos, el viaje espacial que supone será el hito más grande de la humanidad es llegar al planeta marte.

Así como el hombre tiene grandes ambiciones en dominar otros conceptos tecnológicos, de la misma manera pretende dominar los conceptos y nuevas teorías relacionadas con las redes de comunicación a nivel general, también se encuentra en la obligación de plantear, investigar e innovar, modernas metodologías que conformen el desarrollo de nuevas normativas, arquitecturas físicas, y protocolos de seguridad, en el ámbito de hardware y software en el uso correcto del Firewall o Cortafuegos.

Los cortafuegos pueden ser un medio eficaz de protección de un sistema o red local frente a las amenazas de seguridad provenientes de la red, mientras que al mismo tiempo proporciona acceso al exterior mediante redes de área ancha e internet (Stalling, 2004)

El firewall actúa como un filtro de paquetes. Inspecciona todos y cada uno de los paquetes entrantes y salientes. Los paquetes que cumplen cierto criterio descrito en reglas formuladas por el administrador de la red se reenvían en forma normal. Los que fallan la prueba simplemente se descartan (Andrew S. Tanenbaum, 2012).

La creación de Firewall se inicia en la década de los 80 específicamente en el año 1988 con el desarrollo del Firewall de filtrado de paquetes diseñado y publicado por la compañía Digital Equipment Corporation (DEC) su función consistía en filtrar los paquetes enviados por la red, estableciendo control de puertos con simples reglas definidas en el Firewall. Este desarrollo es conocido como la primera generación de Firewall (Guerra, 2016)

El Firewall sigue su carrera al desarrollo un año más tarde en 1989 con las investigaciones de AT&T Bell Laboratories dieron como resultado los Firewall de estados, conocidos como la segunda generación o Firewall a nivel de circuitos. Su función consistía en su capacidad para almacenar un conjunto grandes de paquetes y determinar el estado de conexión de

red, es decir, las direcciones IP origen y destino, los puertos, el número de secuencia etc. (Guerra, 2016).

Posteriormente en el ano de 1992 los científicos Bob Braden y su compañera DeSchon Annette ambos de la Universidad del sur de California (USC) diseñaron el primer Firewall con un interfaz gráfico los cuales fueron compatibles con Windows y MacOs. Dos años después en el año 1994 la empresa Israelita Check Point Software Technologies Ltd. Compra la patente y finalmente lo terminan denominando como Firewall 1 (Wikipedians).

En el año de 1998 el Científico Informático Marty Roesch que actualmente es el Chief Architect of Cisco's Security Business Group, Desarrollo un IDS, Intrusion detection system, llamado Snort de código abierto el cual realiza filtrado de red por medio de monitoreo en busca de código malicioso, este software es un complemento del Firewall basado en búsqueda de anomalías (Meak, 2018).

En el año 2003 la compañía Gartner concibe el termino Next Generation Firewall, el cual este concepto lo basa en la masificación de servicios operativos en el control y filtrado de red por medio de software.

En el año 2006 la tendencia de Firewalls para fortalecer las aplicaciones Web se pone en auge en el mercado de Software.

Los Firewalls o Cortafuegos son herramientas para denegar o filtrar el acceso a la información proveniente de internet. En la actualidad existen dos tipos de Firewalls de hardware en equipos físicos o de software como sistemas lógicos. La función específica del Firewall es cumplir con criterios y normativas de seguridad establecida para impedir que ciertos datos no deseados lleguen a su destino las cuales pueden ser un centro de servidores o una red local.

2.1.2 Antecedentes del Problema

El Instituto Superior Tecnológico Bolivariano de Tecnología actualmente cuenta con un Sistema de Gestión Académico llamado SGA su función es el almacenamiento de información de cada uno de los usuarios que interactúan con la plataforma, por medio del sistema los estudiantes

pueden conocer sus registros académicos de cada materia asignada con su respectivo tutor. Esta plataforma también permite realizar transacciones en línea con respecto a pago de mensualidades.

El SGA utiliza varias integraciones de software para el desarrollo y mantenimiento de protocolo de datos, estas herramientas son de código abierto el cual es necesario mantener actualizado para no tener inconvenientes con alguna vulnerabilidad conocida.

El Sistema Académico SGA utiliza el siguiente software:

1. Base de datos: PostgreSql Versión 9.4
2. Lenguaje de Programación Python 2.7
3. Framework: Django 1.3

El software utilizado es software libre y se encuentra instalado en los servidores del instituto protegido por un firewall Hillstone T2860 de máxima disponibilidad.

La aplicación OWASP se encarga de efectuar la identificación de vulnerabilidades críticas en la etapa de desarrollo y en proceso de producción, está orientado a todo tipo de aplicaciones móviles o web. Nos permite conocer de manera oportuna el tipo de impacto que tenemos y de las probabilidades que podamos ser vulnerados. OWASP cuenta con un top 10 de vulnerabilidades las que sirven como guía para maximizar la seguridad de nuestras aplicaciones.

Con OWASP identificaremos las principales vulnerabilidades existentes en la plataforma Web SGA.

Riesgos actuales en la plataforma SGA según CVE:

¿Qué es un CVE?

CVE (Common Vulnerabilities and Exposures) Vulnerabilidades y Exposiciones Comunes. El CVE es una lista de vulnerabilidades de seguridad conocidas que afectan a los sistemas operativos y software que utilizamos a menudo. Esta lista se expone al público mediante la página web https://cve.mitre.org/data/downloads/index.html.

Por año 1999 al 2009	Por año 2010 al 2020
CVE-2009-xxxx entradas	Entradas CVE-2020-xxx Entradas
CVE-2008-xxxx entradas	CVE-2019-xxxxxx Entradas
CVE-2007-xxxx entradas	CVE-2018-xxxxxx Entradas
CVE-2006-xxxx entradas	CVE-2017-xxxxxx Entradas
CVE-2005-xxxx entradas	CVE-2016-xxxxxx Entradas
CVE-2004- Entradas xxx Entradas	CVE-2015-xxxxxx Entradas
CVE-2003-xxxx Entradas	CVE-2014-xxxx
CVE-2002-xxxx Entradas	CVE-2013-xxxx Entradas
CVE-2001-xxxx Entradas	CVE- Entradas 2012-xxxx
CVE-2000-xxxx Entradas	CVE-2011-xxxx entradas
CVE-1999-xxxx	CVE-2010-xxxx entradas

Tabla 1: Lista de publicación de CVE.
Fuente: https://cve.mitre.org/data/downloads/index.html

PostgreSql

PostgreSql gestor relacional de base de datos con fuente de código abierto, fue creado en los años 80 en la Universidad de Berkely California por el científico Michael Stonebraker. Ingres en el año de 1982 fue el primer desarrollo de un potente motor de datos, que luego en el año de 1986 evolucionaria a Postgre. El 3 de junio de 1994 es publicado Posgret con un lenguaje de consulta Postquel bajo licencia MIT, con código fuente libre para que desarrolladores de todo el mundo puedan modificar y utilizar libremente.

Dos estudiantes de la Universidad de Berkely Andrew en el año de 1994 Andrew Yu y Jolly Chen reemplazan el lenguaje de consulta Postquel por el lenguaje de consulta SQL motivo por el cual nace Postgres95.

Para 1996, quedó claro que el nombre " Postgres95 " no resistiría la prueba del tiempo. Elegimos un nuevo nombre, PostgreSQL, para reflejar la relación entre los POSTGRES originales y las versiones más recientes con capacidad SQL, al mismo tiempo, configuramos la numeración de la versión para que comience en 6.0, volviendo a colocar los números en la secuencia originalmente comenzada por el proyecto Berkeley POSTGRES (Anonimo, Postgresql.org, 1996).

Características

La principal ventaja de usar el modelo MVCC de control de concurrencia en lugar de bloquear es que en MVCC los bloqueos adquiridos para consultar (leer) datos no entran en conflicto con los bloqueos adquiridos para escribir datos, por lo que la lectura nunca bloquea la escritura y la escritura nunca bloquea la lectura (Anonimo, postgresql.org/, 1996).

El MVCC en concreto indica que mientras un proceso registra datos en una tabla, otro proceso puede leer la misma tabla sin que ocurra un bloqueo.

- Postgresql presenta Json como una herramienta con grandes ventajas para buscar e indexar elementos.
- Capacidad de almacenar transacciones en tiempo real sin hacer Backups, esta tecnología se basada en una operación de respaldo continuo WAL Write-Ahead Logging.
- Postgresql cumple con el protocolo ACID que significa Atomicidad, Consistencia, Aislamiento y Durabilidad.

Ventajas

- Gratis de código abierto cuenta con desarrolladores de todo el mundo.
- Es multiplataforma compatible con todos los sistemas operativos.
- Implementa lenguaje SQL.
- Posee escalabilidad y confiabilidad en 20 años de desarrollo nunca tuvo caída de su base de datos.
- **pgAdmin**: Se trata de una herramienta gráfica con la que podemos administrar nuestras bases de datos de forma fácil e intuitiva. Podemos ejecutar sentencias SQL, e incluso crear copias de seguridad o realizar tareas de mantenimiento (Anonimo, todopostgresql.com, 1996).

Desventajas

- Es relativamente lento en inserciones y actualizaciones en bases de datos pequeñas, PostgreSQL está diseñado para ambientes de alto volumen. Esto hace que la velocidad de respuesta pueda

parecer lenta en comparación con bases de datos de pequeño tamaño (Anonimo, todopostgresql.com, 1996).

- A nivel de comandos su sintaxis no es muy intuitiva se requiere un nivel de conocimiento avanzado para el uso del lenguaje SQL.

En mención a la versión de PostgreSQL 9.4 el cual actualmente utiliza el sistema “SGA” es una versión acumulativa se debe detener en cuenta que dejara de recibir nuevas actualizaciones a partir del 13 de febrero del 2020. PostgreSQL 9.4.

Actualmente PostgreSQL lanzo su versión 12.1 el 14 de noviembre del 2019, la cual tiene cambios significativos en rendimiento e Inserción y actualización de tablas.

Python

Python es un lenguaje de programación flexible y de código abierto, es considerado un lenguaje de alto nivel, por ser semejante o parecido al lenguaje humano. Python es considerado actualmente por desarrolladores como uno de los lenguajes de programación más fuertes y de fácil aprendizaje.

Python fue creado por el Científico matemático Holandés Guido van Rossum a finales de los años 90, empezó como un proyecto en base a un lenguaje llamado ABC el cual necesitaba ser sustituido por un lenguaje más intuitivo y flexible. En el mes de febrero del año 1991 el científico matemático Guido Van Rossum decide liberar la versión 0.9 de Python, luego en enero del año 1994 se libera la versión 1.0, la siguiente versión 2.0 fue liberada en el año 2000 y la última versión se hizo pública en diciembre del año 2008.

En 12 de julio del 2018 el científico matemático Guido Van Rossum después de 30 años de arduo trabajo como desarrollador del núcleo de Python toma la decisión de abandonar todos los proyectos relacionados a Python. Los proyectos de Python quedan en manos de la organización Python Software Foundation, que se encargara de seguir mejorando y avanzando con todos los proyectos creados por los miles desarrolladores que Python tiene a nivel mundial.

Figura 1: Mensaje de agradecimiento de Guido Van Rossum
Fuente: Twitter @gvanrossum

Características

- Python es fuertemente tipado significa que para poder cambiar un valor hay que realizar una determinada conversión. Por ejemplo, una cadena de caracteres como un String no se puede convertir directamente en un número entero, para poder llevar a cabo esta conversión se necesita utilizar el método str "str(total)".
- Tipado dinámico en Python no se necesita establecer el tipo al que pertenece una variable y además se puede cambiar su valor en cualquier parte del código.
- Multiplataforma: Python funciona en cualquier sistema operativo.

Ventajas

- Python es un lenguaje de programación de simplificado rápido esto quiere decir que necesita pocas líneas de código para resolver un problema.
- En Python no es necesario declarar cada parámetro como en otros leguajes programación como por ejemplo Java. Por esa razón Python es considerado como un lenguaje de programación flexible y elegante.
- Es un lenguaje portable ya que se puede utilizar las librerías y los códigos en Windows, Mac, Android y Linux.

- Es un lenguaje limpio y ordenado debido a su estructura de diseño, cualquier programador puede leer y escribir de manera rápida cualquier rutina de código.
- Python tiene una comunidad muy grande de desarrolladores en todo el mundo los que se encargan de revisar y hacer implementaciones necesarias al código, de acuerdo a las necesidades.

Desventajas

- Muchos servicios hosting no son compatibles con Python y en el caso de ser compatibles existen complicaciones al momento de configurarlos.
- La mayoría de librerías en Python son de terceros, quiere decir que son de otros desarrolladores por lo cual estas librerías con sus respectivos módulos hay que descargar para su integración.

Versiones

Python desde su creación se ha caracterizado por tener versiones principales y cada una de ellas fue mejorada con el tiempo con cada nueva versión publicada.

Python 2.7 dejo de recibir actualizaciones el 1 de enero del 2020 esto quiere decir que ya no cuenta con soporte de actualización para cada uno de sus módulos y librerías de desarrollo.

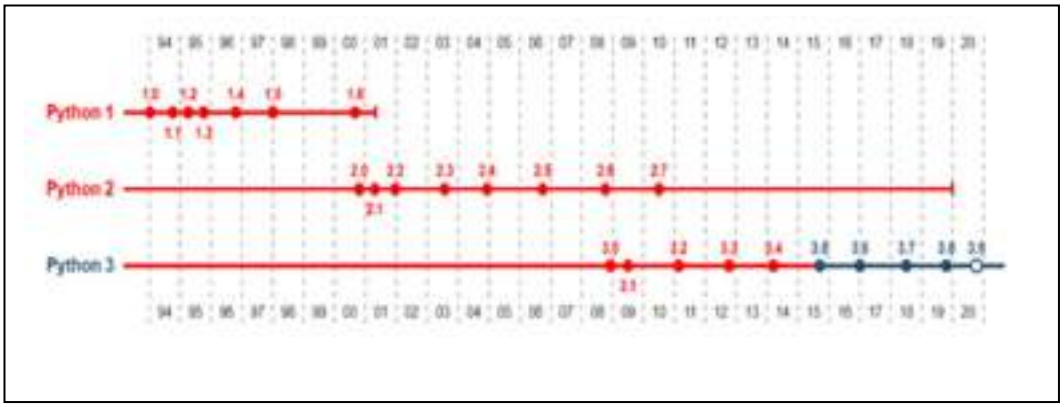

Figura 2: Versiones de Python Obsoletas con rojo y Versiones que siguen publicando actualizaciones con Azul.
Fuente: *https://www.mclibre.org/consultar/python/otros/historia.html#versiones*

Django

Django es un framework con estructura de código abierto para el desarrollo de aplicaciones Web, escrito por completo en Python y cumple con las normativas modelo vista controlador. Django fue desarrollado por Adrián Holovay y Simón Wilson en el año 2003, trabajabando en el periódico Lawrence Journal-World publicado en Kansas, Estados Unidos. Debido a las exigencias que todo desarrollador tiene al trabajar en la actualización de datos sobre las plataformas o aplicaciones web, con la que funciona internet, y las exigencias de la continua información actualizada que debe obligatoriamente tener la prensa digital en sus periódicos, los desarrolladores de contenido web Adrián Holovay y Simón Wilson decidieron escribir varias rutinas de código en el lenguaje Python para optimizar en tiempo de desarrollo y mantenimiento del sitio web.

En julio del 2005 dan el nombre al framework desarrollado como Django en honor al músico de Jazz Django Reinhardt, finalmente es liberado el código al público bajo una licencia BSD. En 2008 Django Software Foundation anuncia que se hará cargo del mantenimiento y desarrollo de este framework.

Como un dato a destacar Django fue un código extraído y generado de la vida real, no fue desarrollado por científicos universitarios o producto de una empresa de software conocida.

Características

Dentro de las características principales Django hereda todas las funcionalidades que Python tiene, por consiguiente, ser fácil aprendizaje, de rápida codificación, robustez y elegancia estructural. A continuación, destaco otras características elementales:

- Utiliza expresiones regulares para reducir el código, basa esta implementación en los llamados a las URl, para definir, encontrar patrones rápidamente de un conjunto de datos ejemplo: podría utilizarse para encontrar todas las direcciones de correo electrónico de una determinada página web.

- ORM Mapeo Relacional de Objetos: Esta es una metodología utilizada en el desarrollo de registro de datos en aplicaciones web, utilizada, para la creación de base de datos relacionales orientada a objetos virtuales. Django permite efectuar consultas a una base de datos SQL, sin la necesidad de utilizar código Sql. Ejemplo ilustrativo:

  Utilizando una llamada normal a la base de datos:

  Select * from Mascotas Where edad = 2

  Utilizando la técnica ORM:

  Mascotas.objects.filter(edad = 2)
- Interfaz Admin: Django utiliza una interfaz de administración, permite realizar cualquier acción sobre el contenido de algún sitio web, no es necesario previa configuración ya que es un paquete denominado django.contrib.
- Django utiliza el principio Dry se refiere a la filosofía de no reutilizar el del código. Redundancy is bad. Normalization is good. The framework, within reason, should deduce as much as possible from as little as possible (Anonimo, docs.djangoproject, 2005-2020) .

Ventajas

- Permite tener código ágil.
- Cuenta con una gran comunidad personas a nivel del mundo orientadas a desarrollar, actualizar y optimizar el Framework de Django constantemente.

Desventajas

- La mayor parte de documentación existente es en idioma Ingles y algo confusa.
- Poner Django en producción es algo difícil.
- En algunos casos es difícil conseguir hosting por no tener soporte con Python.

Django con Python

Actualmente se recomienda trabajar el Framework de Django con Python versión 3.

Adjunto tabla de lanzamientos y fin de soporte de cada una de las distribuciones de Django.

Serie de lanzamiento	Último lanzamiento	Fin del soporte principal [1]	Fin del soporte extendido [2]
3.0	3.0.3	Agosto 2020	Abril 2021
2.2 LTS	2.2.10	2 de diciembre de 2019	Abril 2022
2.1	2.1.15	1 de abril de 2019	2 de diciembre de 2019
2.0	2.0.13	1 agosto 2018	1 de abril de 2019
1.11 LTS [3]	1.11.28	2 de diciembre de 2017	Abril 2020
1.10	1.10.8	4 de abril de 2017	2 de diciembre de 2017
1.9	1.9.13	1 de agosto de 2016	4 de abril de 2017
1.8 LTS	1.8.19	1 de diciembre de 2015	1 de abril de 2018
1.7	1.7.11	1 de abril de 2015	1 de diciembre de 2015
1.6	1.6.11	2 de septiembre de 2014	1 de abril de 2015
1.5	1.5.12	6 de noviembre de 2013	2 de septiembre de 2014
1.4 LTS	1.4.22	26 de febrero de 2013	1 de octubre de 2015
1.3	1.3.7	23 de marzo de 2012	26 de febrero de 2013

Figura 3: Lanzamientos y Series soportadas Django.
Fuente: https://www.djangoproject.com/download/

Como vemos en el grafico tomaremos como referencia el fin de soporte extendido para saber, hasta qué punto nuestro framework tendrá una vigencia que garantiza el desarrollo seguro de nuestra aplicación web.

Por lo expuesto anteriormente, el autor de esta investigación vio una Oportunidad para brindar una posible solución al problema descrito con la implementación del sistema OWASP para estandarizar el mantenimiento y desarrollo seguro de la plataforma web SGA.

2.1.3 Antecedentes Referenciales

En esta investigación efectué un estudio de 3 compañías que desarrollan software en el ámbito nacional las cuales utilizan las normas OWASP para el desarrollo seguro de aplicaciones Web a continuación un breve análisis de estas empresas:

Agrosoft S.A.

Esta compañía con 20 años de experiencia en el Ecuador se ha dedicado a brindar soluciones tecnológicas en el área del sector de Agro Industrias siendo pioneros en temas de producción agrícola. Utilizan metodologías seguras para el desarrollo de aplicaciones en las cuales tienen como una de sus normativas principales el uso de la herramienta OWASP. La visión de esta compañía es implementar soluciones sostenibles de la industria 4.0 (Anonimo, https://agrosoftcomec.wordpress.com/nosotros/).

Viamatica S.A.

Es una empresa fundada en febrero del 2005 en la ciudad de Sevilla (España) en agosto de 2005 deciden realizar la apertura de una sucursal principal en la Ciudad de Guayaquil (Ecuador). En el ámbito laboral de esta compañía es el desarrollo de aplicaciones Web para Smartphone, aplicar normativas de Seguridad Informática, brinda asesorías y Capacitaciones. Entre sus clientes tienen empresas como Banco del Pacifico, Banco Guayaquil, Dipaso, Labra que Labra entre otras (Anonimo, http://viamatica.com/nosotros/).

Uno de los requisitos fundamentales para trabajar con esta compañía es tener una certificación o conocimiento en normativas OWASP para el desarrollo seguro de aplicaciones.

Eclisoft S.A.

Esta es una compañía experta en dar soluciones tecnológicas a dispositivos móviles además brindan soluciones tecnológicas sobre cualquier Plataforma o lenguaje de desarrollo. Los servicios que ofrecen son orientados a la mensajería de textos SMS. Sus desarrolladores tienen un amplio marco de conocimiento en aplicar metodologías agiles de desarrollo seguro de aplicaciones y una de las normativas básicas aplicadas es OWASP (Anonimo, http://www.eclipsoft.com/).

En esta etapa de mi investigación empresas como Eclisoft S.A., Viamatica S.A. y Agrosoft S.A. Considero que es fundamental la utilización de normativas, procedimientos, planificación y organización en cada una de las etapas del desarrollo seguro de un Software. La implementación y uso

de las normativas OWASP ha sido fundamental para estas empresas, debido a las exigencias que el mundo tecnológico hoy demanda, seguridad de la información.

2.2 Firewall

El uso más frecuente de un firewall es filtrar el tráfico de entrada de una red pública (normalmente Internet) hacia una red privada, con el objeto de evitar accesos no autorizados a la red privada. Lógicamente, para que el nivel de seguridad que proporciona un firewall sea efectivo, todo el tráfico de entrada debe pasar por el firewall (Santos Gonzalez Manuel, 2014, p. 305).

El proceso de inspección de paquetes intercambiados entre dos redes para permitir o denegar el propio intercambio se le denomina generalmente filtrado. La función de filtrado se puede implementar tanto en software como en hardware (Santos Gonzalez Manuel, 2014, p. 304).

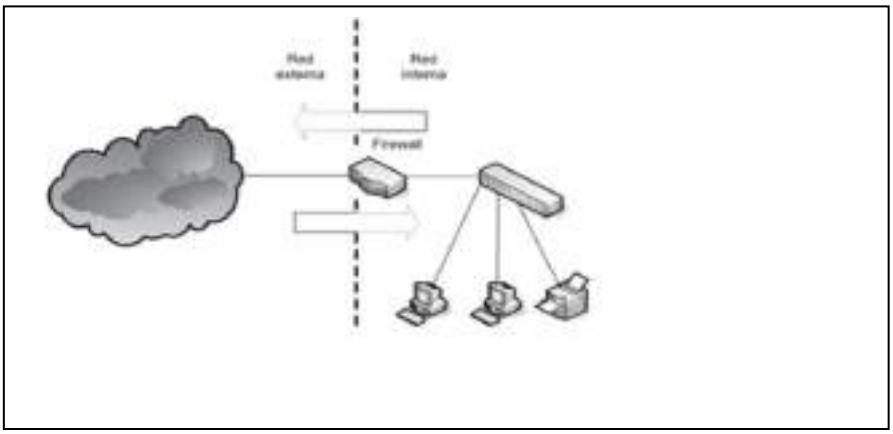

Figura 4: Firewall (Santos Gonzalez Manuel, 2014, pág. 304)

La protección fundamental que ofrece un firewall es bloquear los intentos de intrusión a nuestra red u ordenador personal desde internet. Hay que tener en cuenta que una posible táctica de ataque es entrar en cualquier ordenador de la red Wi-Fi desde internet para conseguir la información necesaria y posteriormente pueda ser utilizada para lograr el acceso inalámbrico (Carballar A. Jose, 2014, p. 212).

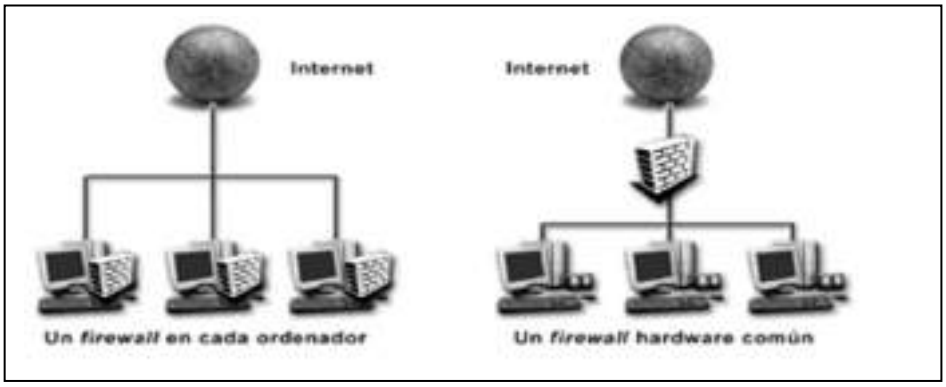

Figura 5: Firewall en red Local (Carballar A. Jose, 2014, p. 215)

Los firewalls más sencillos, llamados de primera generación, básicamente se colocan en la tercera capa, que corresponde al nivel de red, del modelo OSI, donde se envían “paquetes” de información (Urbina Baca Gabriel, 2016, p. 204).

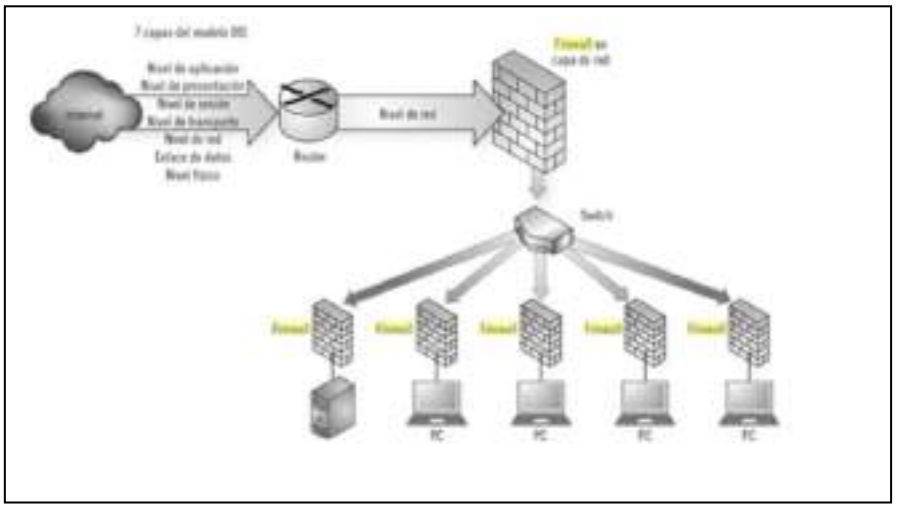

Figura 6: Ubicación de un firewall en una LAN de primera generación (Urbina Baca Gabriel, 2016, p. 205)

2.2.1 Características Firewall

Las características que el investigador encontró de firewall tomo referencia de varios contextos:

- Previenen la exposición de los host y las aplicaciones sensibles a usuarios no confiables (Ariganello Ernesto, 2014, p. 192).
- Hace que la aplicación de una política de seguridad se torne simple, escalable y robusta (Ariganello Ernesto, 2014, p. 192).

- Realiza informes de actividades sospechosas en el Firewall. Esto permite saber cosas como: datos sobre el intruso que intenta acceder a la red o que empleado intenta acceder a lugares inapropiados de Internet. El registro de actividad de un firewall es imprescindible para analizar un posible aguajero de seguridad y actuar en consecuencia (Carballar A. Jose, 2014, p. 216)

Los firewalls o corta fuegos son prácticamente la primera barrera de seguridad que tiene nuestro conjunto de red de datos para minimizar los riesgos informáticos, previendo ataques de código maliciosos o que intrusos accedan a nuestra red sin autorización.

2.2.2 Tipos de Firewall

En la actualidad existen cinco tipos clásicos de seguridad Firewall:

Cortafuego de paquete de filtrado (Packet Filtering Firewalls) la función que realizan es analizar la información que pasa por el router por medio de un mecanismo de criterios establecidos, es decir en el caso de un envió o reenvió de paquete este se direcciona por el número de puerto permitido por la dirección IP.

Pasarela de nivel de circuito (Circuit Level Gateways) la función que ejecuta es la de supervisar el intercambio de los protocoles de transmisión (TCP) entre la comunicación de los host remotos y locales con el cual pueden determinar si se inició una sesión legitima, es decir si la conexión remota a la con la cual tiene el intercambio de datos es Fiable.

Cortafuegos de inspección de estado (Stateful Inspection Firewalls) su función es examinar exhaustivamente cada uno de los paquetes que interactúan en la sesión establecida (TCP), es decir que nos brinda mayor seguridad al filtrar y monitorizar cada paquete por sí solo, la desventaja que tiene es la perdida de rendimiento de la red.

Pasarelas a nivel de aplicación (Application Level Gateways) este tipo de pasarela combina atributos lógicos al igual que los Firewalls de filtrado de paquetes y tiene uno que otro atributo de la denominada pasarela a nivel de circuito, el filtrado de paquetes no solo se da a nivel de puertos específicos sino también a las cadenas de petición HTTP. Si bien esta

pasarela a nivel de aplicación es muy robusta con respecto al envío y recepción de datos, tiene la desventaja de disminuir considerablemente el desempeño de nuestra red.

Cortafuegos de inspección multicapa (Multiplayer Inspection Firewalls) este cortafuego es la combinación de filtrado y monitorización de paquetes circuitos, permitiendo conexiones directas entre otros hosts sean estos locales o remotos. Este tipo de red nos otorga una serie de medidas a nivel de protocolos para la retención y asignación de paquetes en la red.

Los cortafuegos limitan cualquier acción de amenaza en la red, las empresas de acuerdo al giro del negocio deben de incorporar un sistema de cortafuegos con respecto a la seguridad de la información que le brinde integridad en aplicaciones, integridad en sus datos, confidencialidad y la autenticación de los datos.

Un cortafuego mal configurado en muchos de los casos, se debe a una inadecuada preparación técnica de un determinado profesional, este error humano puede ser el causal de que nuestros equipos estén expuestos ante amenazas en la red, a su vez también nos da una falsa impresión de tener una seguridad eficiente.

Firewall como diseño de red

En términos de seguridad las redes deben de mantener una política en la cual se establezcan reglas o normas de acceso a un determinado servicio en este contexto se realiza un breve análisis a la Zona Desmilitarizada “DMZ”.

Zona desmilitarizada

Una red perimetral o zona desmilitarizada (DMZ, por sus siglas en inglés; Demilitarized Zone) constituye una zona segura ubicada entre la red interna de una organización y una red externa, que suele ser internet. La zona desmilitarizada permite la conexión entre las dos redes, la interna y la externa a la DMZ (Urbina Baca Gabriel, 2016, p. 209).

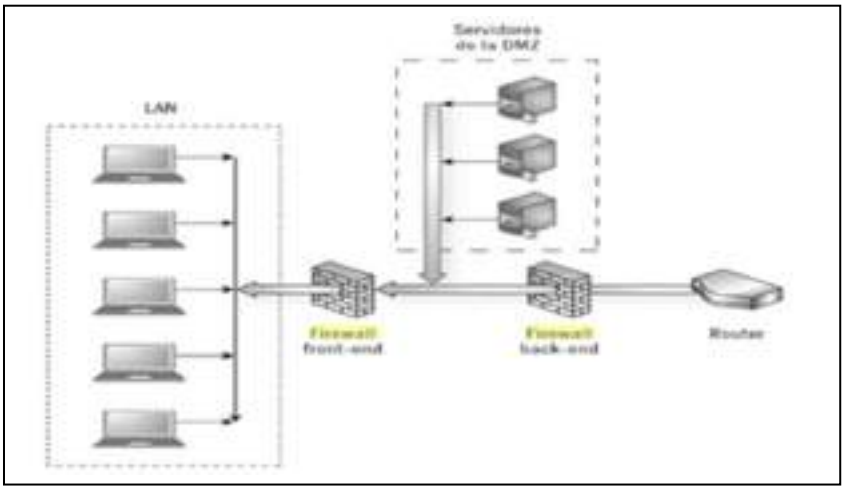

Figura 7: DMZ con Front-End y Back-End (Urbina Baca Gabriel. 2016. p. 210)

Por lo tanto, si se instalan dos firewalls, la configuración será mucho más segura, ya que esto ayuda a prevenir el acceso desde la rede externa hacia la red interna. En este caso el DMZ funciona con dos firewalls; el primero recibe el nombre de Front-End y solo permite que pase la información del exterior al DMZ y el segundo firewall se denomina Back-End que permite que la información pase del DMZ a la red interna (Urbina Baca Gabriel, 2016, p. 210).

2.2.3 Limitaciones del Firewall

La limitación del firewall radica en que solo sirven para filtrar la información que pasa atreves de la red.

Si los ataques informáticos traspasan el firewall; esto es porque el filtrado de la información no es muy estricto, por ejemplo, al utilizar puertos TCP abiertos, o porque la información no utiliza una red. Un firewall tampoco puede proteger de ataques internos a la organización o de las amenazas que provocan los usuarios descuidados o negligentes (Urbina Baca Gabriel, 2016, p. 218).

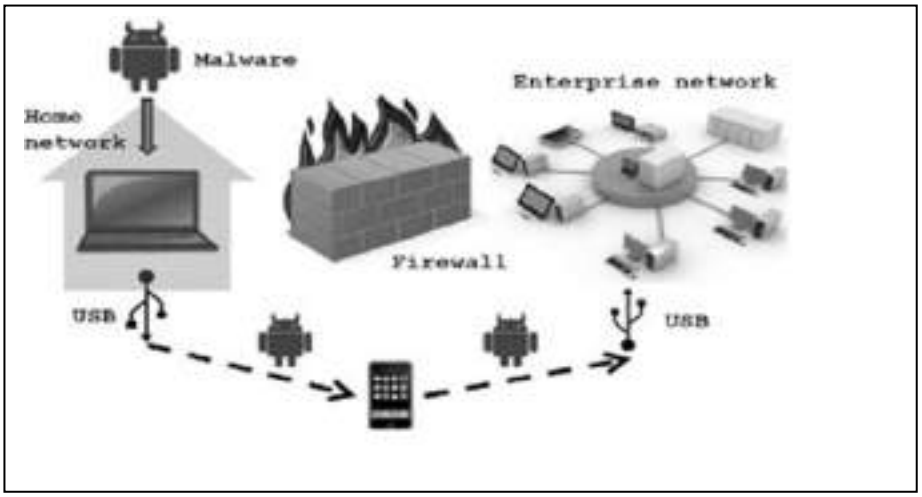

Figura 8: Móvil y USB evitan firewall.
Fuente:https://www.researchgate.net/figure/Mobility-USB-circumventing-corporate-firewalls_fig23_323332207

En el grafico número 5 podemos visualizar la manera en la que un usuario negligente puede evadir el firewall sea con un dispositivo Usb o un teléfono Inteligente, ambos dispositivos ya tienen una infección latente la cual puede es un Malware. Es un hecho que en la mayoría de empresas a nivel global se da esta situación, por falta de normativas de seguridad a nivel de usuarios.

Entre las diferentes limitaciones del Firewall tenemos casos puntuales en la que usuarios pueden limitar la capacidad de reacción de defensa del Firewall:

- El Firewall tiene un sitio de operación física y lógica, por lo tanto, fuera de ese perímetro de operaciones el firewall es ineficaz contra los ataques o penetración de intrusos.
- El Firewall no puede establecer una prohibición de seguridad contra los intrusos corporativos que realicen copias o editen datos sensibles de forma ilegal en medios extraíbles: Usb, discos duros externos o CD-ROM y sustraigan la información fuera del perímetro de una determinada empresa.
- El Firewall no puede darnos protección contra los ataques de manipulación psicológica conocida como la ingeniería social que se utiliza a menudo para perpetrar ciberataques. Por ejemplo, una

llamada telefónica de un espía a un representante de negocios de una compañía, el espía se hace pasar por un ente legítimo del banco para saber las claves de acceso de la compañía.

- El Firewall no puede protegernos de ataques internos a la red, de los usuarios que manipulan códigos maliciosos instalando malware o virus en nuestros equipos. En este caso específicamente se necesitará la instalación de un sistema de antivirus y anti-malware.

2.2.4 Políticas de Firewall

Las políticas se definen como reglas para el tráfico de red en la cual podemos determinar el tipo de conexión autorizada, las políticas tenemos que se dividen en dos; Políticas Restrictivas y Políticas Permisivas.

Política Restrictiva

Es aquella que rechaza el paso de cualquier información, excepto la que esta explícitamente autorizada y que consiste en forma principal de servicios por internet y de proveedores, por lo que es una política que en general adoptan los organismos gubernamentales y empresariales. Aquí se supone que el firewall puede obstruir todo el tráfico y que cada uno de los servicios o las aplicaciones que necesita la organización deberá ser analizado y aceptado, caso por caso (Urbina Baca Gabriel, 2016, p. 219).

Política Permisiva

Es aquella que autoriza el paso de todo tipo de información, excepto aquella para la cual el tránsito esta negado. Toda la información que la organización considere que es potencialmente peligrosa se aísla y se analiza, en tanto que el resto pasa sin ser filtrada. Las organizaciones que normalmente adoptan esa política son las universidades, los centros de investigación y los servicios públicos con acceso a Internet (Urbina Baca Gabriel, 2016, p. 219).

2.3 OWASP

El Proyecto Abierto de Seguridad de Aplicaciones Web (OWASP), es una comunidad abierta dedicada a permitir a las organizaciones realizar el desarrollo, adquisición y mantenimiento de aplicaciones fiables. Todas las herramientas, documentos, foros y delegaciones del OWASP son libres y

abiertas a cualquiera interesado en mejorar la seguridad de las aplicaciones (OWASP, https://www.owasp.org, 2008).

2.3.1 Función del OWASP

El objetivo principal de la herramienta OWASP es detectar todas los fallos o anomalías existentes de una aplicación web, esto quiere decir que primero efectuará un análisis y luego una evaluación de la aplicación para identificar cada uno de los riesgos existentes, iniciando con los parámetros de baja prioridad, luego riesgos medios y finalmente riesgos críticos. La finalidad de medir los riesgos es proporcionar la información adecuada sobre la probabilidad y el impacto que representan cada riesgo en nuestra aplicación.

OWASP anualmente entrega una lista de vulnerabilidades el cual se denomina el top 10 de riesgo de seguridad más importante en las aplicaciones, en el cual muestra detalla todos los riesgos y las vulnerabilidades que usuarios malintencionados usan para violar todas las políticas y restricciones de seguridad que poseen las aplicaciones.

2.3.2 Características Generales

OWASP tiene características especiales que son fundamentales para la búsqueda de riesgos informáticos, y buscan como objetivo ayudar con nuevas prácticas y políticas seguras al desarrollo de aplicaciones web de acuerdo a mi investigación detallo las siguientes:

- Cuenta con mucho desarrollador de todo el mundo por ser de código abierto.
- Es una aplicación que puede ser usada en cualquier sistema operativo sea este Windows, MacOs y Linux.
- Cuentan con el soporte de apoyo de la plataforma GitHub en la cual se alojan proyectos creado por entusiastas desarrolladores de software de todo el mundo.
- Tienen una aplicación como Juice Shop que sirve como un banco de pruebas para la detección de vulnerabilidades.

- Usan el Proyecto OWASP Cheat Sheet es una hoja de truco en la cual contiene información de temas específicos con respecto a la implementación de seguridad de aplicaciones web.

Características del Software

- Es una aplicación de código abierto (Open Source).
- Esta aplicación puede realizar una conexión escucha man in the middle con la aplicación Fuzz.
- Fuzz también puede realizar un ataque de fuerza bruta para poder autentificarse en páginas web con usuarios y contraseñas de débil cifrado.
- Realiza predicción de valores Idénticos de sesión que gestiona la autenticación de cada usuario, con el fin de poder pasar como un usuario autentico de una determinada cuenta de la aplicación web.
- Puede realizar prueba de caja negra para medir el riesgo de vulnerabilidad que tiene la aplicación auditada.
- Realiza pruebas de desbordamientos de bufer, Desbordamiento de pila, inyección de código Sql.
- Realiza múltiples pruebas con respecto a denegación de servicios.

2.3.3 Para qué sirve una prueba OWASP

Una prueba OWASP representa una metodología de pruebas para verificar y corregir, defectos o errores (bugs) de seguridad en el software. Todo este proceso efectuarlo en el menor tiempo posible para optimizar el costo de la empresa y evitar los fallos. El factor principal es hacerle conocer a las organizaciones sobre los problemas de seguridad en el entorno de desarrollo y calidad del software que diseñan.

En la presente investigación se toma como referencia 4 pruebas de comprobación que realiza OWASP como técnicas seguras en la construcción de un programa de pruebas contra fallos de software:

Inspecciones y revisión de manuales

Las revisiones manuales son particularmente buenas para comprobar si las personas comprenden el proceso de seguridad, si han sido concienciadas de la existencia de una política, y si tienen los conocimientos apropiados para diseñar y/o implementar una aplicación segura. Otras actividades, incluyendo la revisión manual de documentación, políticas de programación segura, requerimientos de seguridad y diseños estructurales, deberían ser efectuadas usando revisiones manuales (OWASP, https://www.owasp.org, 2008, p. 24).

Ventajas:

- No requiere tecnología de apoyo (OWASP, https://www.owasp.org, 2008, p. 25).
- Puedes ser aplicada a una variedad de situaciones (OWASP, https://www.owasp.org, 2008, p. 25).
- Fomenta el trabajo en equipo (OWASP, https://www.owasp.org, 2008, p. 25).
- Flexible (OWASP, https://www.owasp.org, 2008, p. 25).

Desventajas:

- Puede consumir mucho tiempo (OWASP, https://www.owasp.org, 2008, p. 25)
- Material de apoyo no siempre disponible (OWASP, https://www.owasp.org, 2008, p. 25)
- Precisa de bastantes conocimientos, reflexión y competencia para ser efectiva (OWASP, https://www.owasp.org, 2008, p. 25).

Modelado de Amenazas

Los modelos de amenaza deberían ser creados tan pronto como sea posible en el ciclo de vida de desarrollo del software, y deberían ser revisados a medida que la aplicación evoluciona y el desarrollo va progresando. El modelado de amenazas es esencialmente la evaluación del riesgo en aplicaciones. Se recomienda que todas las aplicaciones tengan un modelo de amenaza desarrollado y documentado (OWASP, https://www.owasp.org, 2008, p. 25).

Ventajas:

- Visión practica del sistema desde el punto de vista de un atacante (OWASP, https://www.owasp.org, 2008, p. 26).

Desventajas:

- Unos buenos modelos de amenaza no significan un buen software (OWASP, https://www.owasp.org, 2008, p. 26).

Revisión de código fuente

La revisión de código fuente es el proceso de comprobar manualmente el código fuente de una aplicación web en busca de incidencias de seguridad. Muchas vulnerabilidades de seguridad serias no pueden ser detectadas con ninguna otra forma de análisis o prueba (OWASP, https://www.owasp.org, 2008, p. 26).

Ventajas:

- Eficacia e Integridad (OWASP, https://www.owasp.org, 2008, p. 26)

Desventajas:

- Requiere desarrolladores de seguridad altamente competentes (OWASP, https://www.owasp.org, 2008, p. 27).
- No puede detectar errores en tiempo de ejecución con facilidad (OWASP, https://www.owasp.org, 2008, p. 27).
- El código fuente realmente en uso puedes ser diferente del que está siendo analizado (OWASP, https://www.owasp.org, 2008, p. 27).

Pruebas de intrusión

También son conocidos comúnmente como pruebas de caja negra o hacking ético. Las pruebas de intrusión son esencialmente el ``arte´´ de comprobar una aplicación en ejecución remota, sin saber el funcionamiento interno de la aplicación, para encontrar vulnerabilidades de seguridad. Generalmente, el equipo de prueba de intrusión tendría acceso a una aplicación como si fuesen usuarios. Los probadores actúan como un atacante, e intentan encontrar y explotar vulnerabilidades (OWASP, https://www.owasp.org, 2008, p. 27).

Ventajas:

- Requiere un conocimiento relativo menor que una revisión de código fuente (OWASP, https://www.owasp.org, 2008, p. 27).
- Puede ser rápido y por lo tanto, barato (OWASP, https://www.owasp.org, 2008, p. 27).

Desventajas:

- Pruebas solo de impactos frontales (OWASP, https://www.owasp.org, 2008, p. 27).

Que limitaciones tiene OWASP

De acuerdo a la investigación efectuada la única desventaja existente, radica que en el mercado tenemos aplicaciones como OpevAS, Nessus, Burp Suite entre otras aplicaciones que ofrecen mejoras en el escaneo de vulnerabilidades con respecto al OWASP. Por ejemplo:

PortSwigger Burp es una herramienta que ofrece mejores aplicativos en torno a la herramienta de gestión de escaneo de vulnerabilidades, pero esta aplicación no es de software libre es de pago.

¿OWASP trabaja con algún otro software como herramienta de complemento?

OWASP al ser un software de código abierto tiene una gran comunidad de entusiastas desarrolladores que con iniciativa propia han creado herramientas y documentos como una extensión de OWASP que brindan otros campos interesantes entorno a la seguridad de la información en aplicaciones web.

En la dirección web: https://owasp.org/projects/ encontraremos un inventario de proyectos como:

- **Flagship Projects:** la misma que tiene 14 desarrollos para seguridad de aplicaciones web.
- **Lab Projects:** consta con 10 desarrollos que sirven para experimentar problemas y tener varias perspectivas para una solución viable a cualquier fallo de una aplicación web.
- **Incubator Projects:** tienes un total de 16 desarrollos es una incubadora de proyectos, estos proyectos están en etapa inicial de

experimentación buscando nuevas formas metodológicas en el desarrollo de aplicaciones seguras.

- **Projects Needing Website Update:** Este es el proyecto más completo que tiene OWASP con un total de 105 desarrollos estas son implementaciones futuras que añadirán a OWASP entre una de las principales tenemos la guía Anti-Ransomware.

En conclusión, OWASP tiene un futuro prometedor manejando las nuevas tendencias en desarrollo de aplicaciones web, podría decirse que en un futuro será una herramienta indispensable para los desarrolladores de aplicaciones web.

3. Fundamentación Legal

De acuerdo a la normativa vigente del Código Orgánico Integral Penal (COIP) de la República del Ecuador, Registro Oficial Suplemento 180 de 10 de febrero del 2014. Establece en el Capítulo Tercero: Delitos del Buen Vivir, Sección Tercera: Delitos contra la seguridad de activos de los sistemas de información y comunicación.

Se detalla el marco legal:

Art. 229.- Revelación ilegal de base de datos. - La persona que, en provecho propio o de un tercero, revele información registrada, contenida en ficheros, archivos, bases de datos o medios semejantes, a través o dirigidas a un sistema electrónico, informático, telemático o de telecomunicaciones; materializando voluntaria e intencionalmente la violación del secreto, la intimidad y la privacidad de las personas, será sancionada con pena privativa de libertad de uno a tres años.

Si esta conducta se comete por una o un servidor público, empleadas o empleados bancarios internos o de instituciones de la economía popular y solidaria que realicen intermediación financiera o contratistas, será sancionada con pena privativa de libertad de tres a cinco años.

Art. 230.- Interceptación ilegal de datos. - Será sancionada con pena privativa de libertad de tres a cinco años:

1. La persona que sin orden judicial previa, en provecho propio o de un tercero, intercepte, escuche, desvíe, grabe u observe, en cualquier forma

un dato informático en su origen, destino o en el interior de un sistema informático, una señal o una transmisión de datos o señales con la finalidad de obtener información registrada o disponible.

2. La persona que diseñe, desarrolle, venda, ejecute, programe o envíe mensajes, certificados de seguridad o páginas electrónicas, enlaces o ventanas emergentes o modifique el sistema de resolución de nombres de dominio de un servicio financiero o pago electrónico u otro sitio personal o de confianza, de tal manera que induzca a una persona a ingresar a una dirección o sitio de internet diferente a la que quiere acceder.

3. La persona que a través de cualquier medio copie, clone o comercialice información contenida en las bandas magnéticas, chips u otro dispositivo electrónico que esté soportada en las tarjetas de crédito, débito, pago o similares.

4. La persona que produzca, fabrique, distribuya, posea o facilite materiales, dispositivos electrónicos o sistemas informáticos destinados a la comisión del delito descrito en el inciso anterior.

Art. 231.- Transferencia electrónica de activo patrimonial. - La persona que, con ánimo de lucro, altere, manipule o modifique el funcionamiento de programa o sistema informático o telemático o mensaje de datos, para procurarse la transferencia o apropiación no consentida de un activo patrimonial de otra persona en perjuicio de esta o de un tercero, será sancionada con pena privativa de libertad de tres a cinco años.

Con igual pena, será sancionada la persona que facilite o proporcione datos de su cuenta bancaria con la intención de obtener, recibir o captar de forma ilegítima un activo patrimonial a través de una transferencia electrónica producto de este delito para sí mismo o para otra persona.

Art. 232.- Ataque a la integridad de sistemas informáticos. - La persona que destruya, dañe, borre, deteriore, altere, suspenda, trabe, cause mal funcionamiento, comportamiento no deseado o suprima datos informáticos, mensajes de correo electrónico, de sistemas de tratamiento de información, telemático o de telecomunicaciones a todo o partes de

sus componentes lógicos que lo rigen, será sancionada con pena privativa de libertad de tres a cinco años.

Con igual pena será sancionada la persona que:

1. Diseñe, desarrolle, programe, adquiera, envíe, introduzca, ejecute, venda o distribuya de cualquier manera, dispositivos o programas informáticos maliciosos o programas destinados a causar los efectos señalados en el primer inciso de este artículo.

2. Destruya o altere sin la autorización de su titular, la infraestructura tecnológica necesaria para la transmisión, recepción o procesamiento de información en general.

Si la infracción se comete sobre bienes informáticos destinados a la prestación de un servicio público o vinculado con la seguridad ciudadana, la pena será de cinco a siete años de privación de libertad.

Art. 233.- Delitos contra la información pública reservada legalmente. - La persona que destruya o inutilice información clasificada de conformidad con la Ley, será sancionada con pena privativa de libertad de cinco a siete años.

La o el servidor público que, utilizando cualquier medio electrónico o informático, obtenga este tipo de información, será sancionado con pena privativa de libertad de tres a cinco años.

Cuando se trate de información reservada, cuya revelación pueda comprometer gravemente la seguridad del Estado, la o el servidor público encargado de la custodia o utilización legítima de la información que sin la autorización correspondiente revele dicha información, será sancionado con pena privativa de libertad de siete a diez años y la inhabilitación para ejercer un cargo o función pública por seis meses, siempre que no se configure otra infracción de mayor gravedad.

Art. 234.- Acceso no consentido a un sistema informático. - telemático o de telecomunicaciones. – La persona que sin autorización acceda en todo o en parte a un sistema informático o sistema telemático o de telecomunicaciones o se mantenga dentro del mismo en contra de la voluntad de quien tenga el legítimo derecho, para explotar ilegítimamente

el acceso logrado, modificar un portal web, desviar o re direccionar de tráfico de datos o voz u ofrecer servicios que estos sistemas proveen a terceros, sin pagarlos a los proveedores de servicios legítimos, será sancionada con la pena privativa de la libertad de tres a cinco años.

CAPITULO III

3. METODOLOGÍA

3.1 Diseño de la Investigación

La metodología de la investigación, por supuesto, comprende desde la búsqueda o nacimiento de la idea para investigar, hasta la redacción del informe de la investigación (Carlos Muñoz Rocha, 2015).

El método científico otorga a las afirmaciones intelectuales orden, coherencia, claridad, precisión y exactitud, ya que son producto de las relaciones objetivas de la realidad, esto es, no dependen de nuestros deseos o voluntad (Carlos Muñoz Rocha, 2015).

El método científico es la sucesión de pasos que debemos dar para descubrir nuevos conocimientos, o en otras palabras, para comprobar hipótesis que implican o predican conductas de fenómenos desconocidos hasta el momento (Ernesto Rodriguez Moguel, 2005).

En la presente investigación científica permitirá conocer y entender mejor la política y estructura de los lenguajes de programación que utiliza el área de Tic's del ITB, en el desarrollo y mantenimiento del sistema académico institucional SGA, permitiendo a sus desarrolladores tener una herramienta fundamental como OWASP en la prevención de códigos mal implementados que afecten la seguridad de datos de la aplicación web SGA.

Este proyecto es factible para el área de Tic's del Instituto Superior Tecnológico Bolivariano de tecnología, implementar la guía de aseguramiento contra vulnerabilidades de OWASP, que contiene los principios básicos de la seguridad informática en el desarrollo de aplicaciones web.

3.2 Tipos de Investigación

3.2.1 Investigación exploratoria

Los estudios exploratorios sirven para familiarizarnos con fenómenos relativamente desconocidos, obtener información sobre la posibilidad de llevar a cabo una investigación más completa respecto de un contexto

particular, indagar nuevos problemas, identificar conceptos o variables promisorias, establecer prioridades para investigaciones futuras, o sugerir afirmaciones y postulados (Sampieri, 2014).

La investigación explotaría servirá para conocer las operaciones fundamentales que realizan los diseñadores web y relacionar las incidencias de riesgos informáticos.

3.2.2 Investigación descriptiva

Con los estudios descriptivos se busca especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis (Samperi, 2014).

Así como los estudios exploratorios sirven fundamentalmente para descubrir y prefigurar, los estudios descriptivos son útiles para mostrar con precisión los ángulos o dimensiones de un fenómeno, suceso, comunidad, contexto o situación (Samperi, 2014).

La investigación descriptiva busca como objetivo en el presente proyecto, conocer los riesgos de codificación estructurados en el sistema “SGA” cuya finalidad es poder efectuar un análisis y a su vez diseñar un plan de mitigación de las brechas de seguridad encontradas.

3.2.3 Investigación de Campo

En el estudio de campo se trata de estudiar una comunidad o grupo específico, tomando en cuenta las interrelaciones que se establecen entre aspectos de la estructura y la interacción social que se produce (Gómez-Peresmitré & Martínez).

Las técnicas del trabajo de investigación de campo comúnmente empleados son: El Experimento, La Observación y exploración de terreno, La Observación participante, El cuestionario, La entrevista (Rodriguez L. d., 2011).

La investigación de campo no permite recolectar toda la información que genera la plataforma “SGA” por el uso de los usuarios, la idea es saber el comportamiento que tiene la plataforma ante la demanda de un número x de usuarios accediendo al sistema de gestión académico “SGA”.

3.2.4 Investigación bibliográfica-documental

La investigación documental, se caracteriza por la utilización de documentos; recolecta, selecciona, analiza y presenta resultados coherentes; porque utiliza los procedimientos lógicos y mentales de toda investigación; análisis, síntesis, deducción, inducción, etc., porque realiza un proceso de abstracción científica, generalizando sobre la base de lo fundamental; porque supone una recopilación adecuada de datos que permiten redescubrir hechos, sugerir problemas, orientar hacia otras fuentes de investigación, orientar formas para elaborar ins-trumentos de investigación y elaborar hipótesis (Rodriguez M. L., 2013).

En este tipo de investigación bibliográfica se usarán todas las fuentes de textos disponibles del departamento de Tic's para conocer su estructura de funcionamiento, riesgos informáticos y el giro del negocio.

3.3 Población y Muestra

3.3.1 Que es población

Una investigación puede tener como propósito el estudio de un conjunto numeroso de objetos, individuos e incluso documentos. A dicho conjunto se le denomina población (Arias, 2012).

La población, o en términos más precisos población objetivo, es un conjunto finito o infinito de elementos con características comunes para los cuales serán extensivas las conclusiones de la investigación. Esta queda delimitada por el problema y por los objetivos del estudio (Arias, 2012).

3.3.1.1 Tipos de Población:

Población Finita: agrupación en la que se conoce la cantidad de unidades que la integran. Además, existe un registro documental de dichas unidades (Arias, 2012).

Población Infinita: es aquella en la que se desconoce el total de elementos que conforman, por cuanto no existe un registro documental de estos debido a que su elaboración sería prácticamente imposible (Arias, 2012).

Población accesible: también denominada población muestreada, es la porción finita de la población objetivo a la que realmente se tiene acceso y de la cual se extrae una muestra representativa. El tamaño de la población accesible depende del tiempo y de los recursos del investigador (Ary, Jacobs, & Razavieh, 1989).

Para este proyecto la población será finita, se trabajará con el departamento de Tecnologías de la Información “Tic’s”.

Instituto Superior Tecnológico Bolivariano Departamento de Tic's	
Universo	Muestra Poblacional
Desarrolladores	4
Administrador de Base de Datos	1
Web Master	1
Diseñadora Web	1
Experto en Seguridad Informática	1
Soporte Técnico	6
Total	14

Tabla 2: Elaborado por: Rodríguez Zambrano Stalyn

3.3.2 Muestra

De acuerdo a varios autores; la muestra es una porción o subconjunto del universo que representa datos de información, el cual tiene un interés sobre una determinada población en la que se establece una base de estudio. Los datos de la muestra son un subconjunto de información que será finito, para así poder representar una población específica de todo el conjunto universo de la muestra.

En el presente proyecto la muestra poblacional de datos se extrae del departamento de desarrollo de software que pertenece al área de Tic’s del Instituto Tecnológico Bolivariano.

3.3.2.1 Universo y como se determina la muestra

El universo es un conjunto infinito y a su vez finito de datos que se representan con una cantidad “x” con sus características y elementos que forman parte de una muestra poblacional.

Para determinar el universo de la muestra poblacional se utilizan dos tipos técnicas de muestreo, el muestreo probabilístico y muestreo no probabilístico.

3.3.3 Tipos de muestreo

Muestreo probabilístico

El muestreo probabilístico es el conjunto de datos que pueden ser elegidos de forma aleatoria o al azar simple, con el objetivo de poder determinar una situación poblacional. El muestreo probabilístico está conformado por varios tipos: Aleatorio simple, Aleatorio estratificado, Sistemático, Por conglomerados.

Muestreo no probabilístico

Aquí el procedimiento no es mecánico ni se basa en fórmulas de probabilidad, sino que depende del proceso de toma de decisiones de un investigador o de un grupo de investigadores y, desde luego, las muestras seleccionadas obedecen a otros criterios de investigación (Sampieri, 2014).

El tipo de muestreo que se va a utilizar en la presente investigación va ser no probabilístico, porque sabemos el grupo poblacional con el que vamos a trabajar de manera mecánica y aleatoria y la técnica de muestreo será intencional.

3.4 Técnicas e instrumentos de la Investigación

Técnicas

Se entenderá por técnica de investigación, el procedimiento o forma particular de obtener datos o Información (Arias, 2012).

En la presente investigación se ha seleccionado las siguientes técnicas e instrumentos de la investigación:

La observación

La observación es una técnica que consiste en visualizar o captar mediante la vista, en forma sistemática, cualquier hecho, fenómeno o situación que se produzca en la naturaleza o en la sociedad, en función de unos objetivos de investigación preestablecidos (Arias, 2012).

La técnica de observación aplicado en el presente proyecto ayudara a tener un visón global, para conocer las prácticas o técnicas de programación utilizada por los desarrolladores del sistema de gestión académico “SGA”

Observación Directa:

La observación directa es un método utilizado para recolectar datos que forman parte de una estadística, por lo cual el investigador tomo como referencia un escenario ubicado en el tiempo y el espacio de un determinado fenómeno para su análisis e interpretación.

Esta etapa de la observación se tomará todos los datos del comportamiento evolutivo del desarrollo y mantenimiento de la plataforma “SGA” por parte del equipo de desarrolladores.

Observación Indirecta:

Se puede decir que la observación indirecta es el fenómeno que estudia el investigador de fuentes que ya están desarrollada por otros investigadores, el fin que busca es emitir uno o varios criterios diferentes de acuerdo al caso de estudio de un fenómeno.

Para este proyecto se revisará el manual de diseño del desarrollo del sistema de gestión académico “SGA” y conocer las diferentes etapas de su modelación en la estructura de sus datos.

Entrevista

La entrevista, más que un simple interrogatorio, es una técnica basada en un diálogo o conversación "cara a cara", entre el entrevistador y el entrevistado acerca de un tema previamente determinado, de tal manera que el entrevistador pueda obtener la información requerida (Samperi, 2014).

En la presente investigación se realiza la entrevista al director general de desarrollo de software de la plataforma “SGA” con el objetivo de conocer los diferentes tópicos en base al diseño seguro de la aplicación web “SGA”.

Encuesta

El diseño encuesta es exclusivo de las ciencias sociales y parte de la premisa de que, si queremos conocer algo sobre el comportamiento de las personas, lo mejor, lo más directo y simple, es preguntárselo directamente a ellas. Se trata por tanto de requerir información a un grupo socialmente significativo de personas acerca de los problemas en estudio para luego, mediante un análisis de tipo cuantitativo, sacar las conclusiones que se correspondan con los datos recogidos (Sabino, 1992).

Encuesta Oral:

La encuesta oral se fundamenta en un interrogatorio "cara a cara" o por vía telefónica, en el cual el encuestador pregunta y el encuestado responde. Contraria a la entrevista, en la encuesta oral se realizan pocas y breves preguntas porque su duración es bastante corta (Samperi, 2014).

Encuesta Escrita:

La encuesta escrita es la que se realiza mediante un cuestionario (Samperi, 2014).

En este proyecto se utilizará la encuesta escrita, analítica de respuesta cerrada.

Estructura del Departamento de Tic's del ITB

El departamento de Tic's del Instituto Superior Tecnológico Bolivariano de Tecnología actualmente hay 4 Desarrolladores de Sotfware, 1 Administrador de Base de Datos, 1 Web Master, 1 Diseñadora Web, 1 Experto en Seguridad Informática y 6 técnicos en soporte.

En la presente investigación se trabajará con el personal de desarrollo de software.

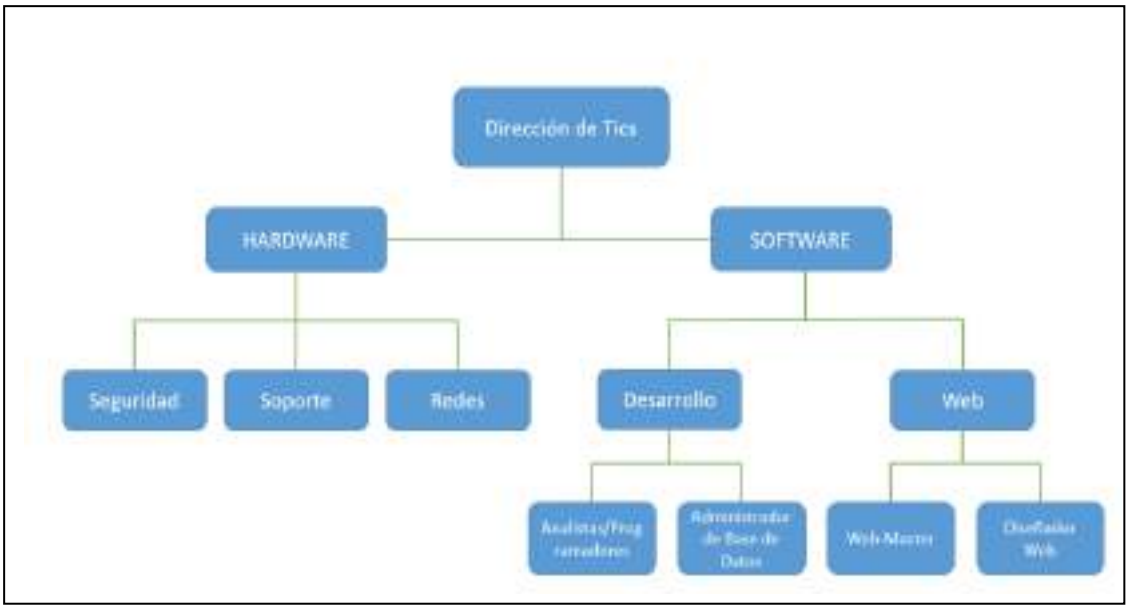

Figura 9: Organigrama del Área de Tic's Fuente: ITB.

3.5 Procedimiento de la Investigación

La investigación se llevará a cabo con la técnica de la encuesta y entrevista.

Entrevista: constara de 10 preguntas la misma que se efectuara al jefe de área de desarrollo de la plataforma web “SGA”.

Encuesta: constara de 10 preguntas objetivas las mismas que se presentaran ante los desarrolladores de software del área de Tic’s del Instituto Superior Tecnológico Bolivariano, para identificar posibles factores de riesgo en la seguridad de datos en la plataforma informática SGA.

La entrevista se realiza al jefe encargado del área de desarrollo de software del área de Tic’s.

Entrevista Preguntas:

1.- ¿Por qué utilizan el lenguaje de programación Python para el desarrollo web del “SGA”?

Bueno desarrollar con el lenguaje Python es una elección acertada en nuestro equipo de desarrolladores, porque contiene una serie de módulos

que nos ayudan y enseñan a que un sitio web sea más dinámico. La mayoría de sitios web conocidos utilizan la tecnología del lado del servidor con frameworks adecuados como Django.

2.- ¿Trabajan con algún Framework y cuál es su función en el sistema "SGA"?

En realidad, nosotros usamos como herramienta de desarrollo Django, es un Framework web sencillo, rápido y con un índice alto de reutilización de código que basa su estructura en el Modelo de vistas – controlador.

Django trabaja en un servidor de desarrollo con las especificaciones WSGI PEP 3333, con este estándar tenemos la posibilidad de trabajar con una amplia gama de servidores y a su vez tener una gran escalabilidad.

3.- ¿Cuál es el motor de base de datos del "SGA" y que tan frecuente son sus respaldos de datos?

En la actualidad utilizamos como base de registros de información, la base de datos relacional PostgreSQL. Los respaldos y control de información se efectúan tal como dictan las normativas COBIT y la ISO 27001.

4. Sobre qué sistema operativo trabajan en el desarrollo de la plataforma "SGA".

Trabajamos sobre CentOS basada en una distribución de RetHat Linux Enterprise, es un sistema operativo flexible nos brinda muchos beneficios a nuestro servidor, como la funcionalidad, estabilidad, velocidad, seguridad este conjunto de parámetros ayudan a que nuestra plataforma del sistema de gestión académico "SGA" sea robusto y confiable.

5.- ¿Qué estándar Web utilizan para el desarrollo de procesos del sistema de gestión académica "SGA"?

Bueno, el estándar lo rige la World Wide Web Consortium, son un consorcio internacional dedicado al crecimiento de la World Wide Web. El objetivo de estos estándares son crear una web orientada a la accesibilidad a más personas y en cualquier tipo de dispositivo con acceso a internet.

6.- ¿Usan protocolos de seguridad para evitar vulnerabilidades del sistema "SGA"?

Bueno, si hablamos de protocolos te diría que son varios lo que utilizamos en el ITB para proteger la integridad de nuestros datos, el Protocolo Tcp/ip, Protocolo Http, Protocolo Ftp, Protocolo Ssh y Protocolo DNS.

7.- ¿Cómo equipo de desarrollo, ustedes efectúan revisión de código?

Generalmente, si lo realizamos.

8.- ¿En este mundo de tecnología ustedes actualmente hacen teletrabajo y que normativas de seguridad aplican?

Actualmente por la complejidad que está suscitándose en el país y a nivel mundial por la pandemia generada por el SARS-CoV 2, efectuamos teletrabajo. Nos enfocamos en aplicar varias normas de conexión segura para el mantenimiento de nuestros sistemas en línea, utilizamos la tecnología de redes privadas a través del uso de un VPN "Virtual Private Network".

9.- ¿Tienen alguna restricción por usar otro tipo de software de manera eventual en el desarrollo del "SGA", por decir hoy use Photoshop para editar imágenes de los banners y mañana uso Gimp?

Realmente no tenemos restricciones de ese tipo con ningún de software.

10.- ¿Qué tan seguro piensa usted que es el "SGA"?

Muy seguro, ya que nos preocupamos hasta en el mínimo detalle para tener los sistemas funcionales y seguros.

Análisis de Encuesta realizada al equipo de desarrollo de software del área de Tic's del Instituto Tecnológico Bolivariano.

Encuesta Preguntas:

1.- Usted conoce las guías de pruebas OWASP (Open Web Application Security Project).

Tabla 3: Encuesta Pregunta 1

Respuesta	Desarrolladores	Porcentaje
Si	1	25%
No	3	75%
No tengo conocimiento	0	0%
Total	**4**	**100%**

Elaborado por: Rodríguez Zambrano Stalyn

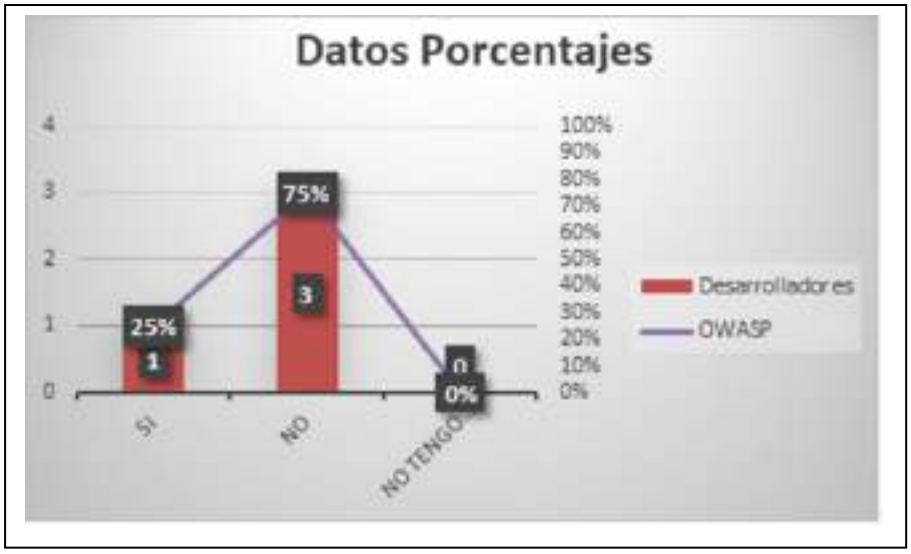

Figura 10: Estadísticas Pregunta 1
Elaborado por: Rodríguez Zambrano Stalyn

Objetivo: Valorar si la población encuestada tiene el conocimiento de técnicas seguras de desarrollo web.

Interpretación: Con el resultado de los datos encuestados podemos deducir que del 100% de la población, el 75% dice no conocer la OWASP como guía para diseño seguro de aplicaciones Web, sin embargo, el 25% admite conocer y utilizar la guía de OWASP.

Análisis: En conclusión los encuestados están de acuerdo que la inclusión de la Guía OWASP debe de ser incluido como parte del protocolo de desarrollo seguro del sistema de gestión académico “SGA”.

2.- Piensa usted que el sistema de gestión académico "SGA" necesita de revisiones permanentes contra fallos de código malicioso, que vulneren la integridad de los datos en el servidor.

Tabla 4: Encuesta Pregunta 2

Respuesta	Desarrolladores	Porcentaje
De acuerdo	2	50%
En desacuerdo	2	50%
Poco importante	0	0%
Total	**4**	**100%**

Elaborado por: Rodríguez Zambrano Stalyn

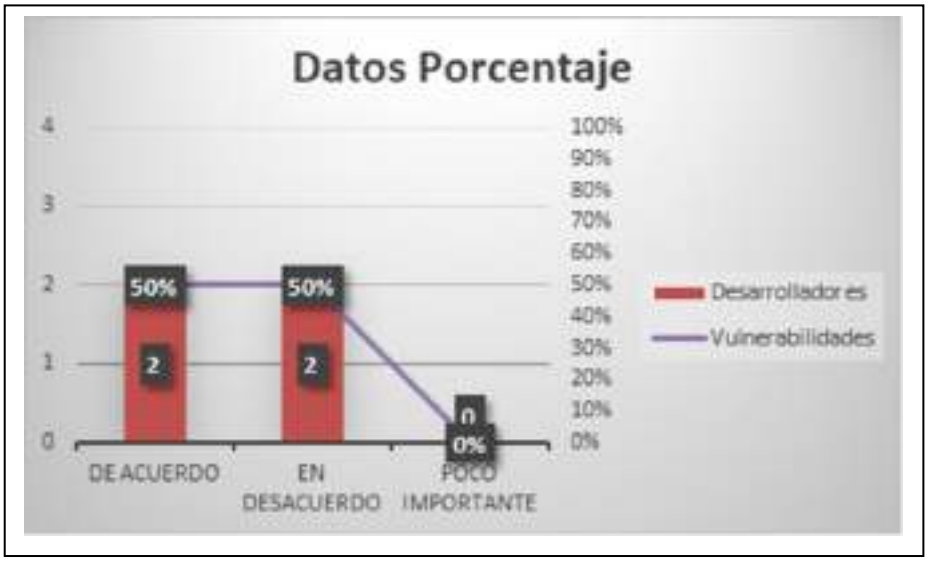

Figura 11: Estadísticas Pregunta 2
Elaborado por: Rodríguez Zambrano Stalyn

Objetivo: Determinar si la población encuestada está de acuerdo con revisiones del "SGA" y evitar vulnerabilidades en el servidor.

Interpretación: Con el resultado de los datos encuestados podemos deducir que del 100% de la población, el 50% está de acuerdo que es necesario realizar revisiones permanentes de código malicioso en el servidor, sin embargo, el otro 50% está en desacuerdo.

Análisis: En conclusión los encuestados admiten tener un ecosistema seguro y confiable pero también indican no tener el recurso humano para realizar revisiones periódicas por el tiempo que este conlleva.

3.- Piensa usted que el sistema de gestión académico "SGA", tiene vulnerabilidades en su diseño estructural del software.

Tabla 5: Encuesta Pregunta 3

Respuesta	Desarrolladores	Porcentaje
Es posible	2	50%
No es posible	0	0%
Definitivamente no	1	25%
Otros	1	25%
Total	**4**	**100%**

Elaborado por: Rodríguez Zambrano Stalyn

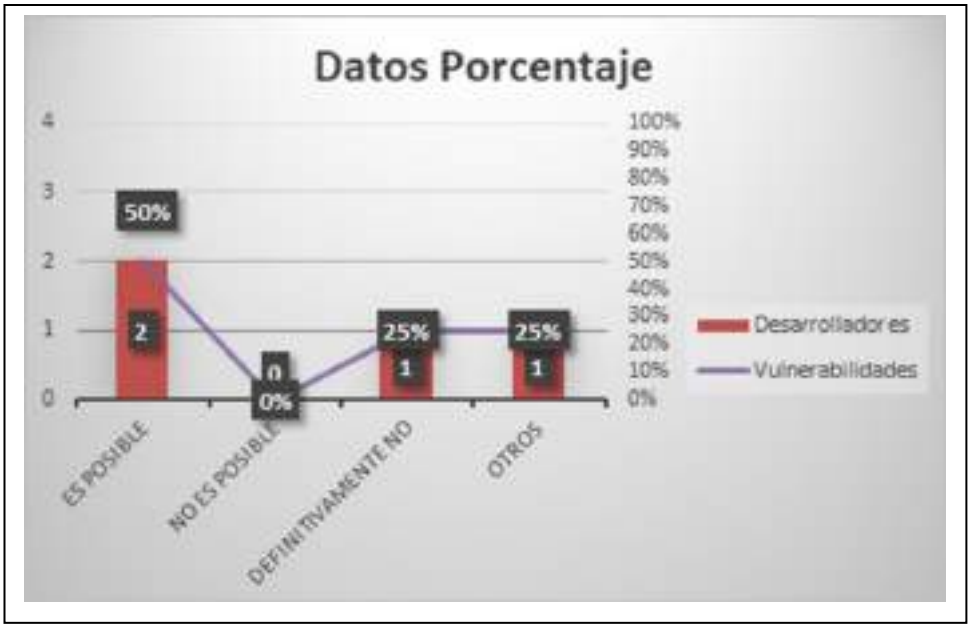

Figura 12: Estadísticas Pregunta 3
Elaborado por: Rodríguez Zambrano Stalyn

Objetivo: Determinar si la población encuestada conoce sobre fallos en la programación del código fuente de la plataforma del "SGA".

Interpretación: Con el resultado de los datos encuestados podemos deducir que del 100% de la población, el 100% está de acuerdo que el sistema de gestión académico "SGA" no tiene problemas en su estructura de código fuente.

Análisis: En conclusión los encuestados admiten tener una eficiente técnica de diseños de software.

4.- Usted conoce de algún incidente o sustracción de información que haya sido perpetrado al web server del sistema de gestión académico "SGA"

Tabla 6: Encuesta Pregunta 4

Respuesta	Desarrolladores	Porcentaje
Si	0	0%
No	4	100%
Total	**4**	**100%**

Elaborado por: Rodríguez Zambrano Stalyn

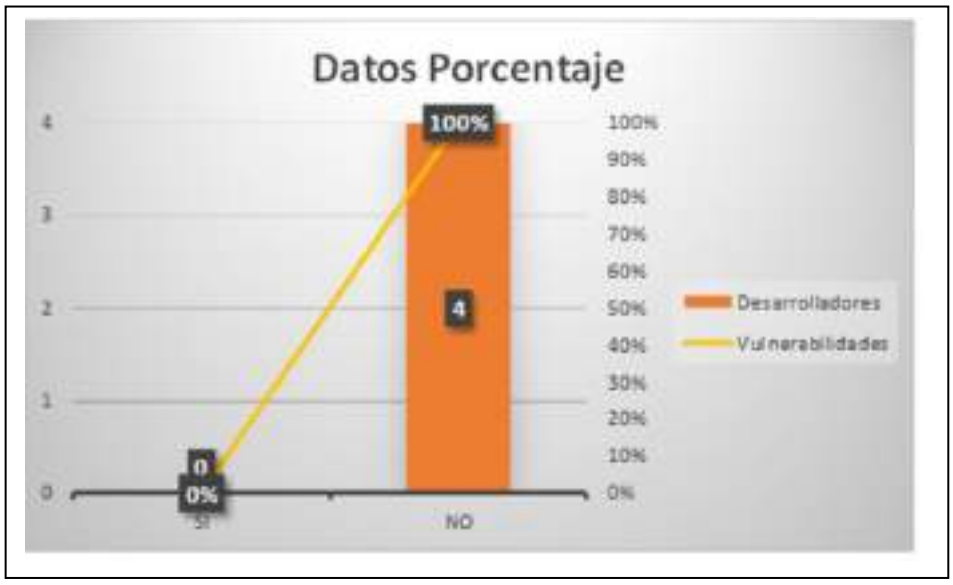

Figura 13: Estadísticas Pregunta 4
Elaborado por: Rodríguez Zambrano Stalyn

Objetivo: Determinar si la población encuestada conoce incidente de sustracción de datos en el web server del sistema de gestión académico "SGA".

Interpretación: Con el resultado de los datos encuestados podemos deducir que del 100% de la población, el 100% está de acuerdo que el sistema de gestión académico "SGA" no tienen conocimiento sobre incidentes en el web server del sistema de gestión académico "SGA".

Análisis: En conclusión el sistema de gestión académico "SGA" es seguro para sus usuarios, pero no es un sistema infalible contra riesgos informáticos.

5.- Una amenaza informática tiene un horario en específico para atacar y afectar al sistema de gestión académico "SGA".

Tabla 7: Encuesta Pregunta 5

Respuesta	Desarrolladores	Porcentaje
Horarios de Oficina	1	25%
Fuera de Horarios de Oficina	1	25%
Fines de Semana	0	0%
En Cualquier momento	2	50%
Total	**4**	**100%**

Elaborado por: Rodríguez Zambrano Stalyn

Figura 14: Estadísticas Pregunta 5
Elaborado por: Rodríguez Zambrano Stalyn

Objetivo: Determinar si la población encuestada conoce un horario específico en que puede ocurrir un incidente informático en la plataforma "SGA".

Interpretación: Con el resultado de los datos encuestados podemos deducir que del 100% de la población, el 50% está de acuerdo que el sistema de gestión académico "SGA" puede ser perpetrado un ataque de código malicioso en cualquier momento, sin embargo, el otro 50% indica que pude ser en horarios de oficina y fuera de horarios de oficina.

Análisis: En conclusión el sistema de gestión académico "SGA" puede ser vulnerable en cualquier momento si no se tienen las debidas políticas de seguridad.

6.- Considera usted que el personal del área de desarrollo de software está preparado para prevenir y reaccionar ante un ataque informático.

Tabla 8: Encuesta Pregunta 6

Respuesta	Desarrolladores	Porcentaje
Si	3	75%
No	0	0%
Tal vez	1	25%
Total	**4**	**100%**

Elaborado por: Rodríguez Zambrano Stalyn

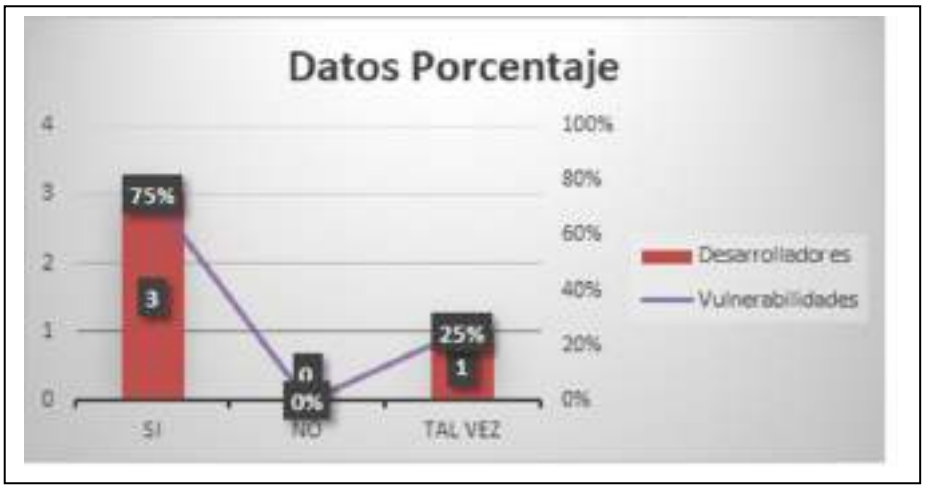

Figura 15: Estadísticas Pregunta 6
Elaborado por: Rodríguez Zambrano Stalyn

Objetivo: Determinar si la población encuestada está preparada para reaccionar ante un riesgo informático.

Interpretación: Con el resultado de los datos encuestados podemos deducir que del 100% de la población, el 75% está preparado para reaccionar ante un riesgo informático, sin embargo, el 25% considera que no está seguro.

Análisis: En conclusión los desarrolladores de software necesitan más preparación en el ámbito de seguridad informática.

7.- La responsabilidad del diseño y la implementación de nuevos códigos al sistema de gestión "SGA", corresponde a:

Tabla 9: Encuesta Pregunta 7

Respuesta	Desarrolladores	Porcentaje
Jefe de Tic's	0	0%
Área de Redes	0	0%
Desarrolladores	4	100%
Jefe de Seguridad Informática	0	0%
Total	**4**	**100%**

Elaborado por: Rodríguez Zambrano Stalyn

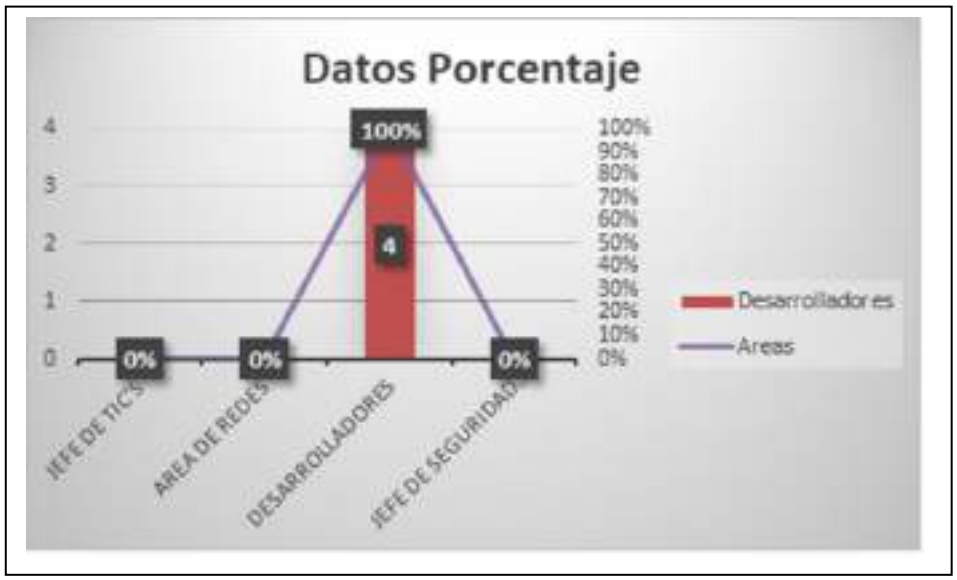

Figura 16: Estadísticas Pregunta 7
Elaborado por: Rodríguez Zambrano Stalyn

Objetivo: Determinar si la población encuestada elige los nuevos diseños estructurales de código de la plataforma web "SGA".

Interpretación: Con el resultado de los datos encuestados podemos deducir que del 100% de la población se encarga de estructurar los nuevos códigos para un diseño óptimo y seguro de la plataforma web "SGA".

Análisis: En conclusión los desarrolladores de software del área de Tic's, son los encargados en mantener y estructurar nuevo código fuente en la aplicación web "SGA".

8.- La responsabilidad y mantenimiento de la integridad de los datos en el servidor de la plataforma "SGA" corresponde a:

Tabla 10: Encuesta Pregunta 8

Respuesta	Desarrolladores	Porcentaje
Web Master	0	0%
Seguridad Informática	2	50%
Dba	2	50%
Operadores informáticos	0	0%
Total	**4**	**100%**

Elaborado por: rodríguez Zambrano Stalyn

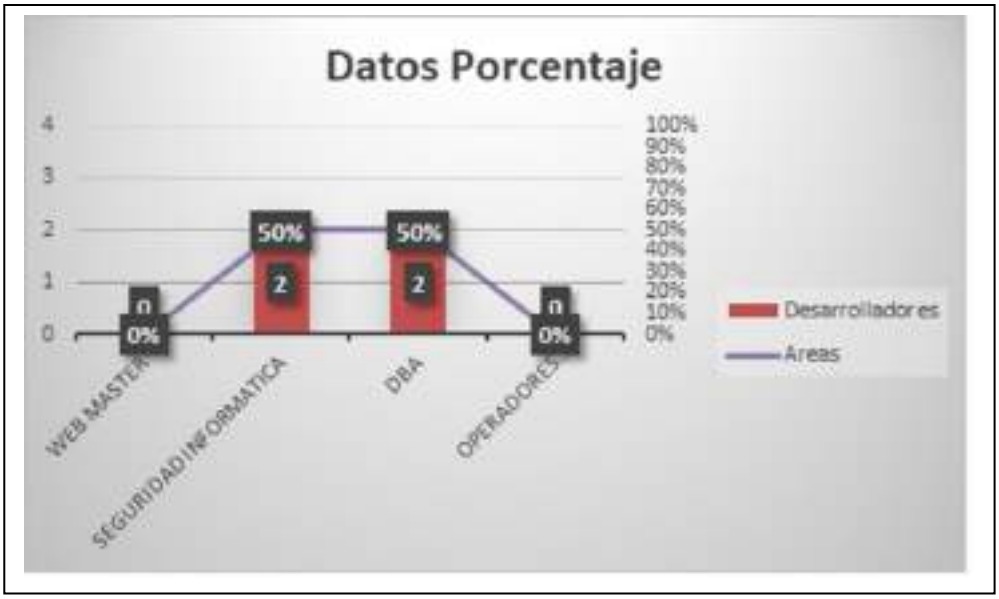

Figura 17: Estadísticas Pregunta 8
Elaborado por: Rodríguez Zambrano Stalyn

Objetivo: Determinar si la población encuestada tiene conocimiento sobre las responsabilidades de operación en mantenimiento e integridad de datos del servidor de la plataforma web "SGA".

Interpretación: Con el resultado de los datos encuestados podemos deducir que del 100%, el 50% por ciento indica que el encargado de la protección de la data es el DBA (Administrador de la base de datos) y el otro 50% indica que el experto en seguridad marca las pautas en la protección de la data.

Análisis: En conclusión todos son participes de la protección de la data del servidor web del sistema de gestión académica "SGA".

9.- En la actualidad el departamento de Tic's tiene políticas de desarrollo seguro de software.

Tabla 11: Encuesta Pregunta 9

Respuesta	Desarrolladores	Porcentaje
Si	1	25%
No	1	25%
En proceso	1	25%
Tal vez	1	25%
Total	**4**	**100%**

Elaborado por: Rodríguez Zambrano Stalyn

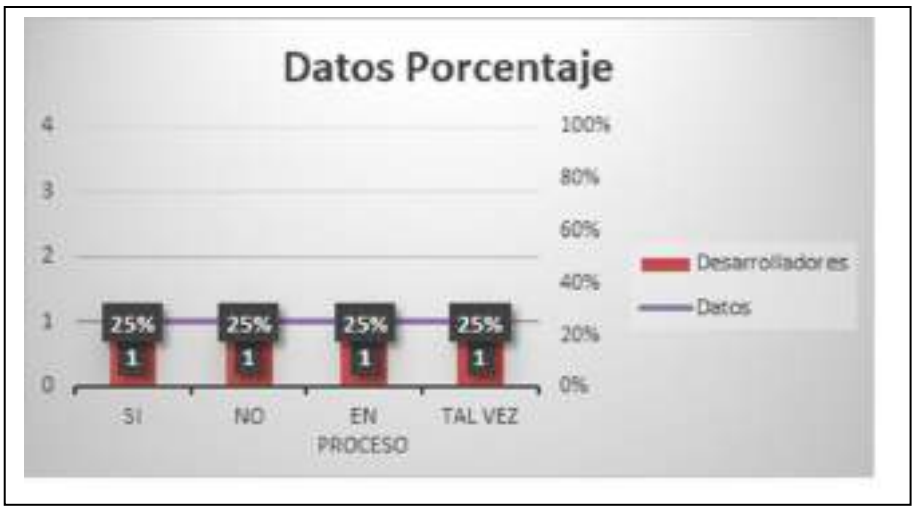

Figura 18: Estadísticas Pregunta 9
Elaborado por: Rodríguez Zambrano Stalyn

Objetivo: Determinar si la población encuestada tiene conocimiento sobre políticas de desarrollo seguro de software.

Interpretación: Con el resultado de los datos encuestados podemos deducir que del 100%, un 25% indica que si conoce conocimiento de políticas de desarrollo seguro de software, 25% manifiesta que no conoce del tema, 25% manifiesta que está en proceso de implementación y un 25% restante indica que tal vez conoce algo del tema.

Análisis: En conclusión de acuerdo a los datos encuestados se indica una tendencia de inconformidad y aceptación con respecto a políticas de desarrollo seguro de software.

10.- Considera usted que el software OWASP puede ser de gran utilidad para el control de vulnerabilidades de la plataforma "SGA".

Tabla 12: Encuesta Pregunta 10

Respuesta	Desarrolladores	Porcentaje
Importante	2	50%
Indispensable	2	50%
Poco importante	0	0%
No se toma en cuenta	0	0%
Total	4	100%

Elaborado por: Rodríguez Zambrano Stalyn

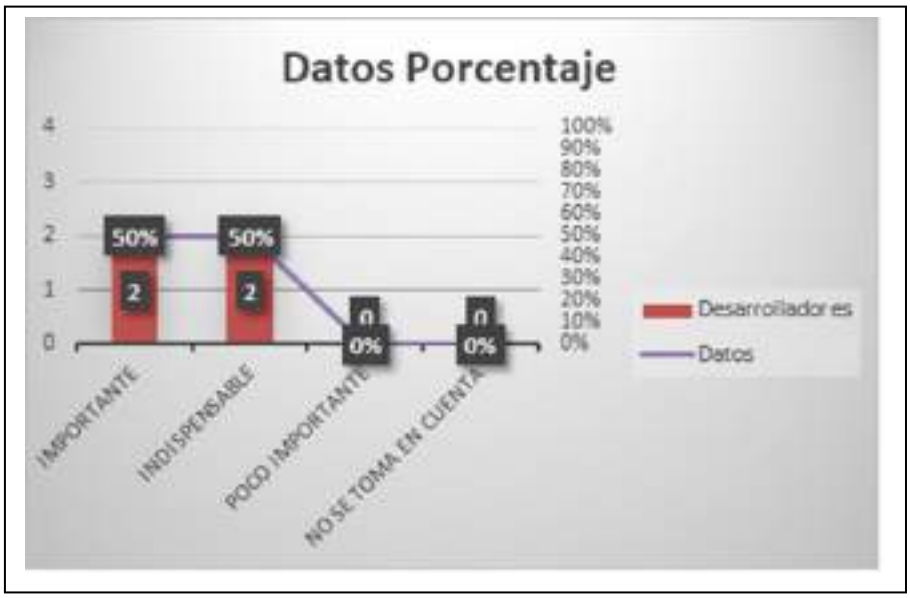

Figura 19: Estadísticas Pregunta 10
Elaborado por: Rodríguez Zambrano Stalyn

Objetivo: Determinar si la población encuestada considera OWASP útil para controlar las vulnerabilidades del sistema de gestión académico "SGA".

Interpretación: Con el resultado de los datos encuestados podemos deducir que del 100%, el 50% indica que es importante aplicar OWASP y el otro 50% indica que es indispensable.

Análisis: En conclusión de acuerdo a los datos encuestados podemos definir como OWASP una herramienta de utilidad indispensable para su aplicación en el sistema de gestión académico "SGA".

3.6 Modelo De Desarrollo Del Sitio Web

El desarrollo rápido de aplicaciones (RAD) es una metodología orientada a objetos para el desarrollo de sistemas, la cual incluye un método de desarrollo así como herramientas de software (Kendall, 2005).

Fases de RAD

Hay tres amplias fases para RAD en las que se involucra tanto a los usuarios como a los analistas en la evaluación, el diseño y la implementación (Kendall, 2005).

Fase de planeación de los requerimientos: En esta fase, los usuarios y analistas se reúnen para identificar los objetivos de la aplicación o el sistema, y para identificar los requerimientos de información que surgen a partir de estos objetivos (Kendall, 2005).

Taller de diseño RAD: Se trata de una fase de diseño y refinación que se puede caracterizar mejor como un taller (Kendall, 2005).

Fase de Implementación: En esta fase los analistas trabajan intensivamente con los usuarios para diseñar los aspectos de negocio o los aspectos no técnicos del sistema. Tan pronto como se llega a un consenso sobre estos aspectos, y se crean y refinan los sistemas, se prueban los nuevos sistemas o las nuevas partes de los mismos y después se introducen a la organización (Kendall, 2005).

RAD utilizado como una metodología para el desarrollo ágil de aplicaciones, ya que involucra nuevos esquemas diferentes a los tradicionales. El área de Tic’s lo utiliza en el desarrollo de su motor de base de datos para web, el cual se basa en código libre en este caso Postgresql, además utiliza versiones modernas de framework como es Django, todo esto conlleva a mejorar su infraestructura de software de manera constante, lo cual facilita un desarrollo de su aplicación web "SGA".

Todo desarrollo basado en metodología RAD significa un cambio radical en la infraestructura en el ámbito gerencial administrativo y de software ya que ayuda a economizar recursos presupuestarios, evita incumplimiento

de fechas de entrega de un producto en desarrollo y permite ciclos de desarrollo a corto plazo.

3.7 Presupuesto Económico

Se detalla la factibilidad económica del proyecto de investigación, insumos de oficina, valor por horas trabajadas, hardware y software:

Gasto de Oficina:

Tabla 13: Gastos de Oficina

Gasto Operacional de Oficina		
Descripción	**Costo**	
Hojas	$	5.00
Impresiones	$	10.00
Movilización	$	25.00
Carpeta	$	2.00
Alimentación	$	35.00
Total	**$**	**77.00**

Elaborado por: Rodríguez Zambrano Stalyn

Gasto de Talento Humano:

Tabla 14: Talento Humano

Talento Humano					
Descripción	**Valor horas**		**Horas**	**Total**	
Analista de sistemas	$	5.83	80	$	466.40

Elaborado por: Rodríguez Zambrano Stalyn

Gastos de Infraestructura Tecnológica:

Tabla 15: Infraestructura Tecnológica

Infraestructura Tecnológica		
Descripción		**Costo**
Acer Aspire 5 Slim	$	374.00
Software OWASP	$	-
Ubuntu 20.04 Lts	$	-
Tarjeta Ram 8 Gb	$	45.00
Disco Duro Interno 1 TB	$	56.00
Total	**$**	**475.00**

Elaborado por: Rodríguez Zambrano Stalyn

La compra tecnológica de la laptop Acer Aspire 5 Slim, es un equipo de cómputo imprescindible por ser parte de la estrategia de análisis de vulnerabilidades del sistema de gestión académico “SGA”.

Estará equipado con software adecuado para el desarrollo de este proyecto, sistema operativo principal Ubuntu versión 20.04.4 Lts Focal Fossa, para realizar el levantamiento informático de datos con el software OWASP. La instalación del sistema operativo y del software puede ser efectuado por un técnico calificado y con experiencia en soporte Linux y software libre, que pertenezca al departamento de sistemas del ITB.

Con este equipo de cómputo se busca usar un sistema operativo minimalista, rápido y con gran capacidad de procesos de datos para el análisis de código fuente generado por el software OWASP.

Gastos Total:

Tabla 16: Gasto Total

Coste del Proyecto		
Descripción		**Costo**
Gasto Operacional de Oficina	$	77.00
Talento Humano	$	466.40
Infraestructura Tecnológica	$	475.00
Total	$	1,018.40

Elaborado por: Rodríguez Zambrano Stalyn

CAPITULO IV

4.1 Preparación del hardware

En esta fase de estudio se consideran varios requisitos que debe de cumplir el equipo de cómputo previo a la instalación del sistema operativo y sus respectivas aplicaciones.

El equipo es una laptop del fabricante Acer.

Figura 20: Acer Aspire 5 Slim
Elaborado por: Rodríguez Zambrano Stalyn

Especificaciones técnicas del equipo:

Tabla 17: Componentes de laptop Acer Aspire 5 Slim

Procesador	AMD Ryzen 3 – 3200U
Gráficos	Radeon Vega Mobile Gfx
Velocidad	3.5 Ghz
Memoria Ram	DDR4 SDRAM 8 Gb
Disco Duro	SSD 128 Gb
BIOS	V1.06
BIOS Fabricante	Insyde Corp.
BIOS Tamaño	128 KIB
BIOS Capacidad	4608 KIB

Elaborado por: Rodríguez Zambrano Stalyn

Las características de este equipo de cómputo son de tecnología avanzada, con un procesador de gama baja de decima generación de la serie 3000 de AMD, sin embargo este procesador actualmente es el de mayor rendimiento y robustez entre los procesadores de gama baja.

Entre las características importantes de este procesador tenemos los siguientes datos de acuerdo al fabricante “AMD”:

Tabla 18: Características del procesador AMD 3 3200U

AMD Ryzen™ 3 3200U	
Arquitectura	Núcleo AMD ZEN
Familia	AMD Ryzen
Fecha de Lanzamiento	Primer Cuarto del 2019
Núcleos del CPU	2
Núcleos del GPU	3
Reloj Base	2.6Ghz
Reloj Máximo	3.5Ghz
Cache L2	1MB
Cache L3	3MB

Elaborado por: Rodríguez Zambrano Stalyn

El procesador AMD Ryzen 3 3200U cuenta con la arquitectura Zen, que tiene un alto rendimiento y capacidad, permitiendo realizar múltiples procesos a equipos de cómputo de escritorio y portátiles, el cual es aprovechado por los creadores de contenido, Video jugadores y demás profesionales del ámbito informático.

Desde el lanzamiento en 2017 de la arquitectura Zen los procesadores AMD, han tenido un accenso exitoso en la distribución de sus núcleos, llegando a producir una cantidad muy respetable de 260 millones de núcleos “Zen” en todo el planeta, cubriendo de esta manera la gran demanda del público, y llevando a sus rivales como la compañía Intel ha innovar nuevas arquitectura para satisfacer la gran demanda de los usuarios por tener equipos más potentes y rápidos.

Figura 21: Arquitectura ZEN
Fuente:https://www.anandtech.com/show/15595/amd-shipped-260-million-zen-cores-by-2020

El sistema operativo que el fabricante nos entrega por defecto en este equipo de cómputo es Windows 10 home, el mismo que está instalado en una memoria SSD M.2 que tiene 128 Gb de almacenamiento. Este dispositivo de almacenamiento al ser de poca capacidad, se implementa un disco duro adicional con una capacidad de 500 Gb de almacenamiento, con el fin de implementar la instalación un sistema operativo adicional para el presente proyecto.

El sistema operativo que implementara este caso de estudio será una Distribución de Linux llamada Ubuntu 20.04 lts Focal Fossa con la misma se busca optimizar todos los recursos que nos pueda ofrecer el equipo de cómputo Acer Aspire 5 Slim.

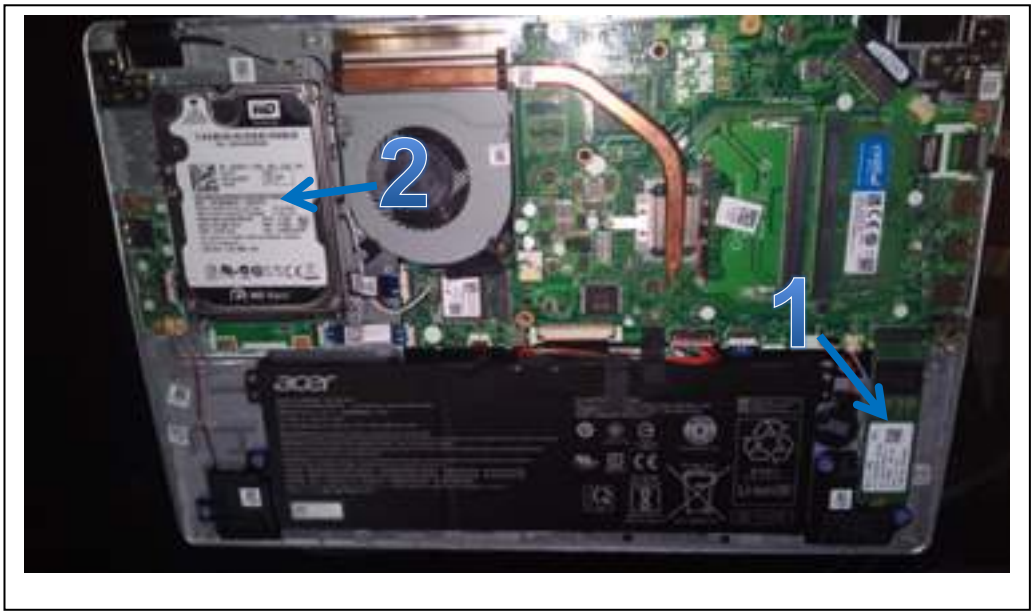

Figura 22: Estructura de Hardware Acer Aspire 5 Slim
Elaborado por: Rodríguez Zambrano Stalyn

En el Gráfico número 21 tenemos implementado el disco duro de 500 Gb de almacenamiento en la que instalaremos el sistema operativo Ubuntu 20.04Lst y la tarjeta SSD M.2 de 128 Gb con el sistema operativo Windows 10.

Figura 23: Conectores Flexibles HDD
Elaborado por: Rodríguez Zambrano Stalyn

Para la conexión del disco duro se utiliza un cable flexible hdd para sata el cual va conectado a un slot jhdd1.

Este equipo de cómputo está diseñado para trabajar en el área de Tecnologías, y realizar todo tipo de tareas y operaciones bajo un rendimiento estándar de acuerdo a las indicaciones del fabricante.

4.1.1 configuración de equipo

El equipo de cómputo Acer Aspire 5 Slim se debe realizar la configuración del BIOS para validar el sistema con el uso del hardware y cumplir con las especificaciones del fabricante, esto quiere decir que no se llevara a efecto ningún cambio que comprometa la integridad del hardware.

En la BIOS nosotros encontraremos todas las configuraciones que ayudan a optimizar el rendimiento del hardware, cabe recalcar que una mala configuración puede ocasionar fallos críticos en el funcionamiento del equipo de cómputo.

En este caso de estudio explicare brevemente la configuración adecuada para habilitar el disco duro con el que funcionara el sistema operativo Ubunto 20.04 Lts Focal Fossa.

Como primer paso en la BIOS debemos habilitar el disco duro ingresar a la opción de **Security** y proceder habilitar la opción **Password on Boot**, para luego configurar la unidad de boteo en la opción **Secure Boot Mode**.

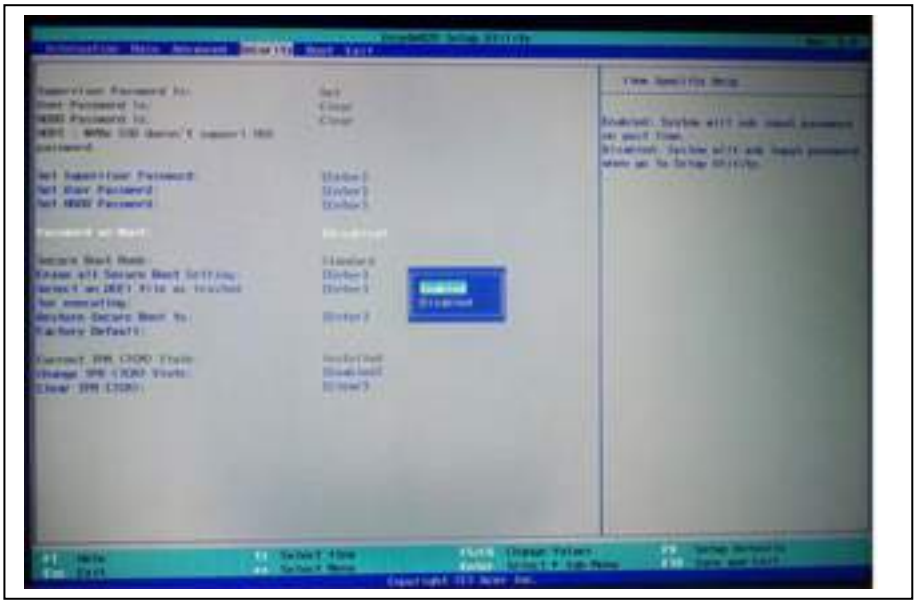

Figura 24: BIOS opción para habilitar disco duro.
Elaborado por: Rodríguez Zambrano Stalyn

El siguiente paso a realizar es ir a la opción de **Boot,** para asignarle una prioridad en el orden de boteo a la unidad física de almacenamiento. En este caso la unidad física de almacenamiento que iniciara primero será la tarjeta SSD M.2 con el sistema operativo Windows 10 y la unidad secundaria física de almacenamiento será el disco duro con el sistema operativo Ubuntu 20.04 Lst Focal Fossa.

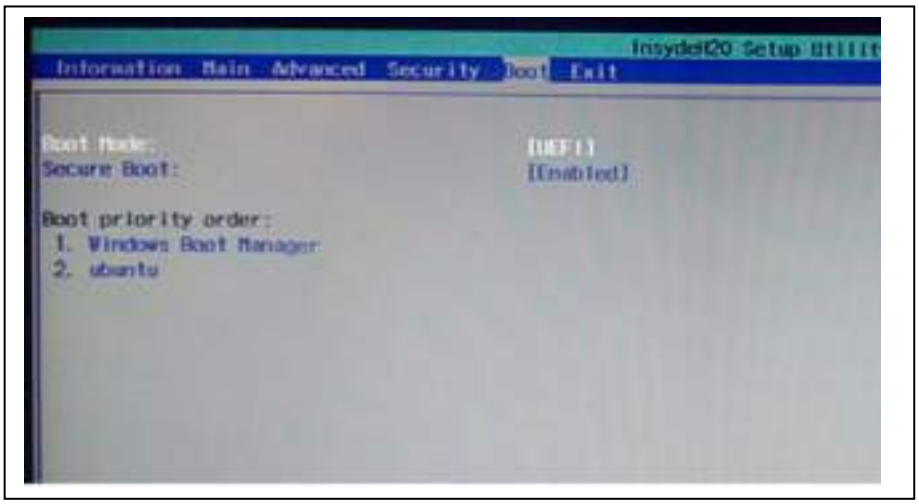

Figura 25: Sistema BIOS modificación al Boot manager
Elaborado por: Rodríguez Zambrano Stalyn

Finalmente para ingresar al sistema operativo Ubuntu 20.04Lts, al iniciar el equipo presionamos la tecla F12 para activar el Boot Manager y seleccionar la opción 2.

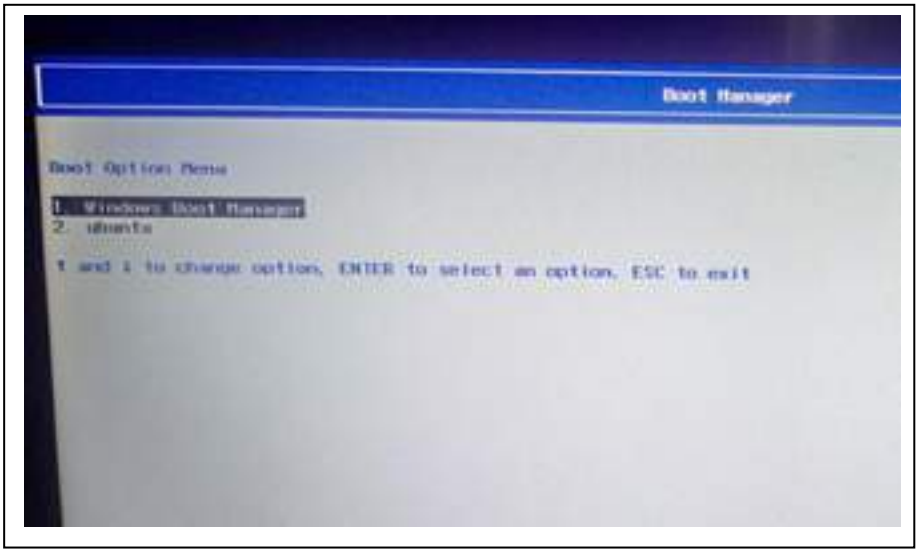

Figura 26: Menú del sistema para ejecución de Boot Manager
Elaborado por: Rodríguez Zambrano Stalyn

4.1.2 Pruebas del equipo

El equipo de cómputo Acer Aspire 5 Slim se efectuaron varias pruebas de rendimiento (Benchmark) para conocer el alcance que tendrá al implementar varias herramientas de Ethical Hacking para el presente caso de estudio con la aplicación OWASP.

Todas estas pruebas realizadas nos darán una métrica de las ventajas y desventajas que tendremos al utilizar el equipo de cómputo Acer Aspire 5 Slim y de acuerdo a las características encontradas, optimizaremos los recursos para la puesta en ejecución del equipo.

Ubuntu al ser un sistema operativo de software libre, encontramos múltiples herramientas aplicativas para efectuar pruebas de software y hardware a nuestro equipo de cómputo, para este caso de estudio utilizare la herramienta Phoronix Text Suit v9.6.0.

Phoronix Text Suit v.9.6.0: Es un software libre que nos permite por medio del uso de la terminal del sistema operativo Ubuntu, conocer el estado del equipo de cómputo que estemos utilizando, con el objetivo de realizar comparaciones con otros dispositivos, y medir las capacidades para obtener resultados con respecto al rendimiento de sus componentes y sacar conclusiones.

Medir la capacidad y determinar la eficiencia de un equipo de cómputo es importante, porque nos otorga un patrón a seguir para poder desarrollar nuestro caso de estudio.

Figura 27: Terminal Ubuntu aplicación Phoronix Text Suite v9.6.0
Elaborado por: Rodríguez Zambrano Stalyn

4.1.3 Consideraciones a tomar en cuenta

Como todo sistema operativo antes de su instalación, hay aspectos importantes que considerar.

- **Linux y sus distribuciones:** Linux al ser un software libre tiene una gran cantidad de entusiastas desarrolladores que permiten tener constantes actualizaciones a cada una de estas distribuciones, sin embargo se debe tener en cuenta las características que tiene nuestro hardware y si la distribución que tenemos lista para instalar, no tendrá repercusiones en el rendimiento de nuestro equipo. En general estas distribuciones de Linux tienen muchos instaladores genéricos para dispositivos de audio, video etc.
- **Compatibilidad de Hardware:** Cuando nos referimos a la compatibilidad del hardware, consideramos de forma específica

cada módulo, socket, componente integrado y otros dispositivos que conforman la Mainboard, con todos estos elementos debemos de conocer si el fabricante recomienda el sistema operativo que deseamos instalar, debido a que la mayoría de fabricantes recomiendan el uso del sistema operativo Windows.

- **Arquitectura de almacenamiento de procesos en bits:** En este contexto se hace referencia a la arquitectura de 32 bits y 64 bits. Debemos conocer si la arquitectura de nuestro procesador nos permitirá registrar una estructura de datos en un espacio de almacenamiento de 32 bist o 64 bits a su vez conocer si nuestro procesador gestionara una mayor o menor cantidad de datos entre los 32 o 64 bits.
- **Compatibilidad de Software:** El único problema que un usuario tendría con Linux es instalar aplicaciones de licenciamiento corporativo de las compañías Microsoft y Apple. Existen aplicaciones como Playlinux o UniOS que permiten emular este tipo de aplicaciones de estas dos compañías, sin embargo consumen el doble de recursos que normalmente usarían, haciendo que el sistema opere en con sobre carga en la gestión de procesos.
- **Recomendaciones de requerimientos del sistema operativo Ubuntu 20.04 Lts:**
    - Procesador dual core 2GHz o superior.
    - 4 Gb de memoria Ram.
    - 25 Gb libres como mínimo de espacio en el disco duro.
    - Un dispositivo DVD o un USB para realizar la instalación.
    - Acceso a Internet para las actualizaciones en línea.

4.2. Instalación de software

¿Porque elegí Ubuntu como sistema operativo?

Ubuntu es una distribución de Linux con fundamentos basados en Debian, lo cual me permite hacer un uso total y radical del Kernel con el fin de

personalizar todos los recursos que dispongo del hardware y software, estas configuraciones se efectúan de acuerdo a mi necesidad.

Para este caso de estudios las modificaciones fueron en el ámbito, de seguridad y virtualización para arquitecturas x86 de Intel y AMD.

Seguridad: La seguridad de la información es muy importante entorno al manejo y la trata de la data de información, porque nos ayudan a prevenir, fallos críticos a nuestros sistemas de informáticos. Específicamente se actualizo el Kernel de Ubuntu 20.04 Lts por tener un fallo de alta vulnerabilidad según NVD ”National Vulnerability Database” CVE-2020-11884 (Database, 2020).

Figura 28: Vulnerabilidad Crítica de seguridad Kernel Ubuntu 20.04 Lts
Fuente: https://nvd.nist.gov/vuln/detail/CVE-2020-11884#vulnCurrentDescriptionTitle

Virtualización: La virtualización se refiere al uso de kernel basado en la utilización de máquinas virtuales que se ejecuten sobre el sistema operativo Ubuntu 20.04 Lts. Para este caso de estudio se escogió al software Vmware Workingstation 15 player.

¿Cuál es la ventaja de utilizar una máquina virtual en Ubuntu?

Para este caso de estudio utilizare una distribución de seguridad informática llamada Parrot Security Os basada en Debian, por consiguiente expongo algunos beneficios que me ofrece el trabajar con Ubuntu como un sistema operativo general.

- Tener Ubuntu como sistema operativo base me beneficia porque tendré una estructura segura para trabajar con código malicioso sea este un Malware, Spyware, Ramsomware etc.
- Al ser Ubuntu un ambiente Linux basado en Debian me da la facilidad de minimizar el consumo de los recursos del sistema, permitiéndome tener un mayor rendimiento con otras aplicaciones en ejecución.
- Nuestra máquina virtual nos ayudara a realizar distintas pruebas sin preocuparnos, por la afectación a nuestro sistema operativo base.

Figura 29: Parrot Security Os sobre sistema operativo Ubuntu
Elaborado por: Rodríguez Zambrano Stalyn

4.2.1 Pasos a seguir

Para este de estudio se eligió la distribución de Linux Parrot Security Os, este sistema operativo de seguridad informática, tiene integrada una gran variedad de aplicaciones para realizar Hacking ético, en el cual se incluye la herramienta Owasp el mismo que es nuestra base de estudios en la presente Tesis.

Para la utilización del OWASP como primer paso se efectuara la instalación del sistema operativo Parrot Security Os, en una máquina virtual llamada Vmware Workstation 15 Player.

Paso 1: El proceso de instalación inicia con la configuración de la máquina virtual, asignándole los recursos necesarios, para que nuestro sistema operativo Parrot Security Os tenga un rendimiento adecuado de acuerdo a los procesos que va a ejecutar.

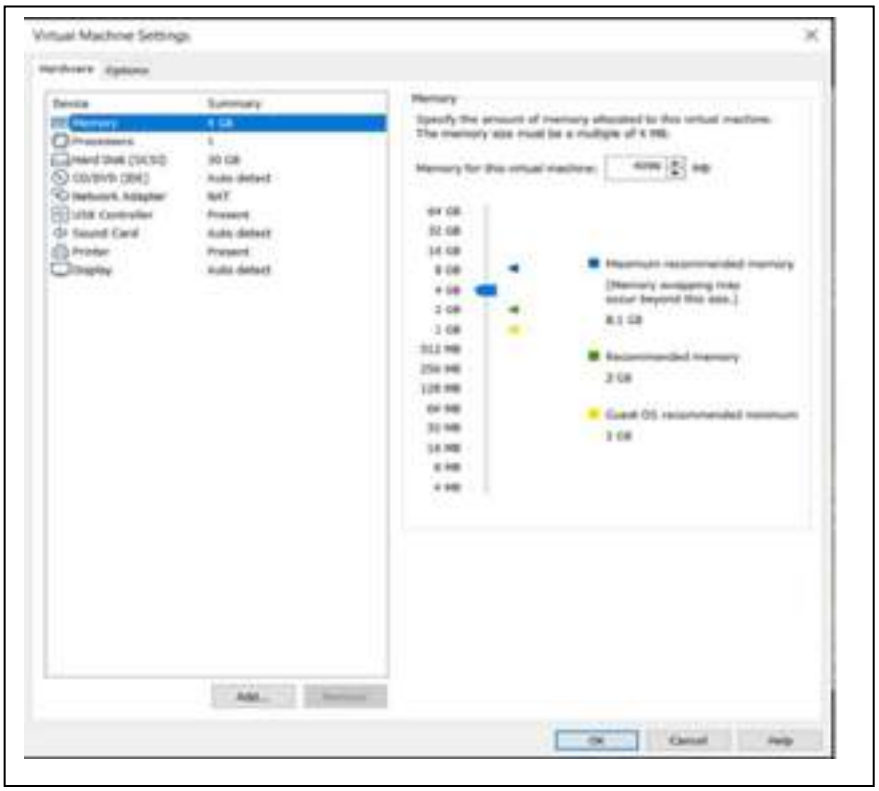

Figura 30: Configuración de máquina virtual Vmware Workstation 15 Player
Elaborado por: Rodríguez Zambrano Stalyn

Paso 2: Tenemos variedad de configuraciones y ejecuciones que nos permite realizar el Sistema Operativo Parrot Security Os. En este caso vamos a la opción Install para iniciar el proceso de instalación.

Figura 31: Ventana de Boteo del Sistema Operativo Parrot Security Os.
Elaborado por: Rodríguez Zambrano Stalyn

Paso 3: Debemos seleccionar el tipo de instalación que deseamos, en este caso instalamos con contenido grafico de bibliotecas.

Figura 32: Instalación con biblioteca de gráficos
Elaborado por: Rodríguez Zambrano Stalyn

Paso 4: Aplicamos el idioma del sistema operativo el cual va ser el español.

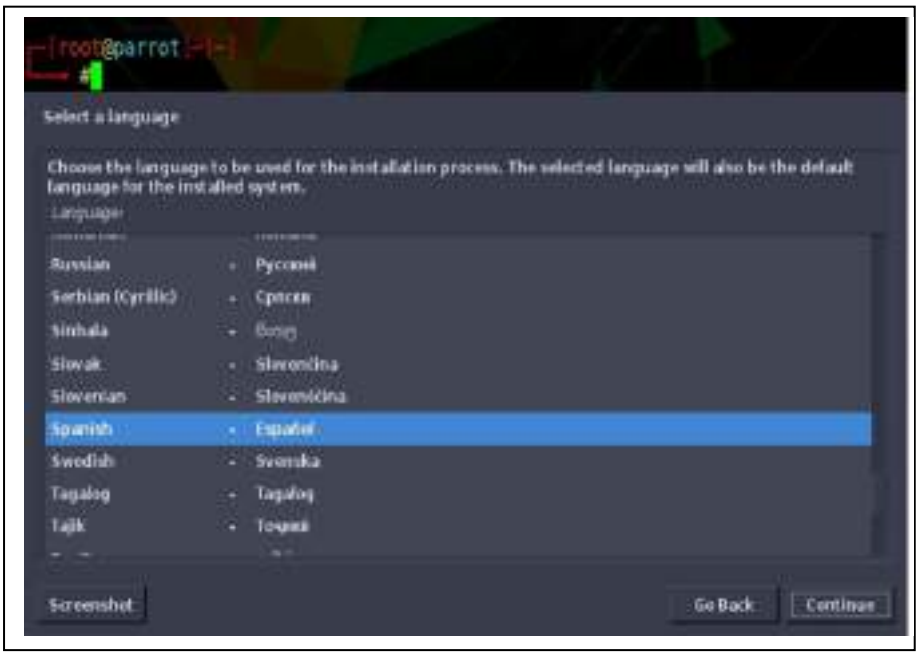

Figura 33: Selección del idioma del sistema operativo
Elaborado por: Rodríguez Zambrano Stalyn

Paso 5: Ubicación hacemos referencia al lugar donde nos situamos estos ayudara al sistema operativo recibir todas la actualizaciones en tiempo real.

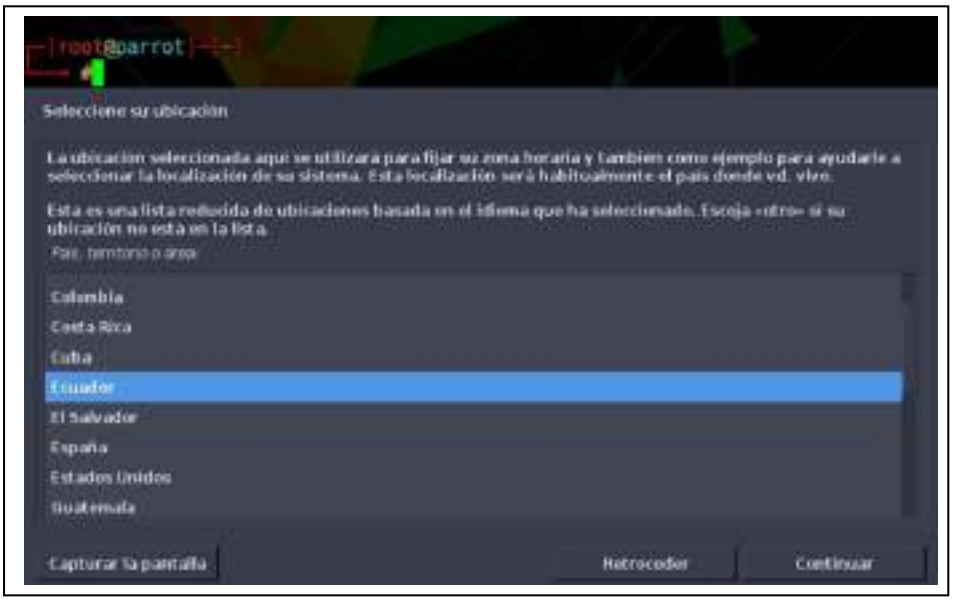

Figura 34: Ubicación de instalación del sistema operativo
Elaborado por: Rodríguez Zambrano Stalyn

Paso 6: Se configura la cuenta de usuario root para realizar varios procesos que requieren de un usuario administrador.

Figura 35: Creación de contraseñas como root
Elaborado por: Rodríguez Zambrano Stalyn

Paso 7: Configuramos el nombre de la cuenta que vamos a tener como superusuarios.

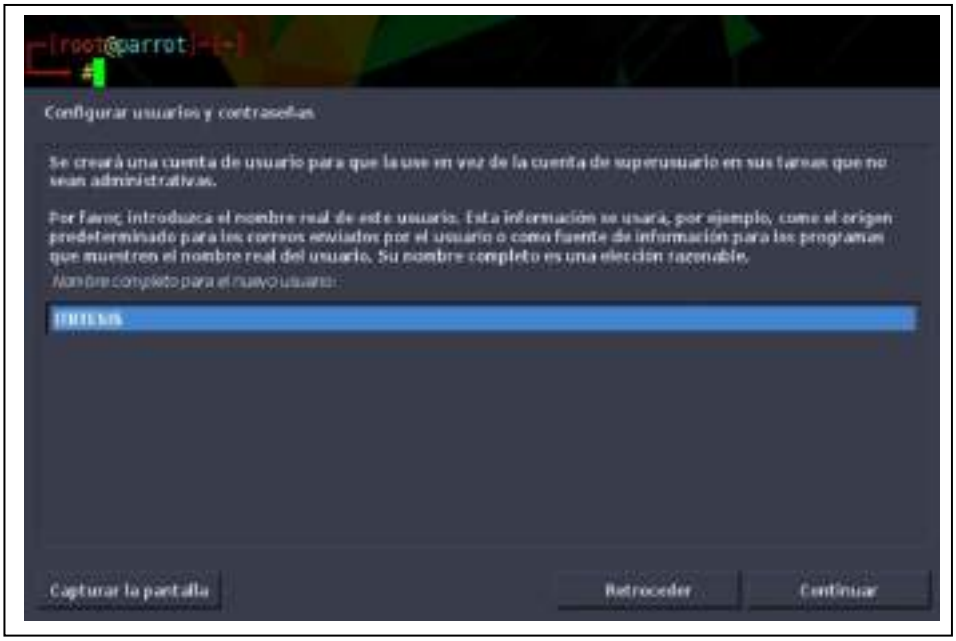

Figura 36: Configuración del nombre de la cuenta
Elaborado por: Rodríguez Zambrano Stalyn

Paso 8: Tener en cuenta la partición de disco que deseamos tener en nuestro sistema operativo.

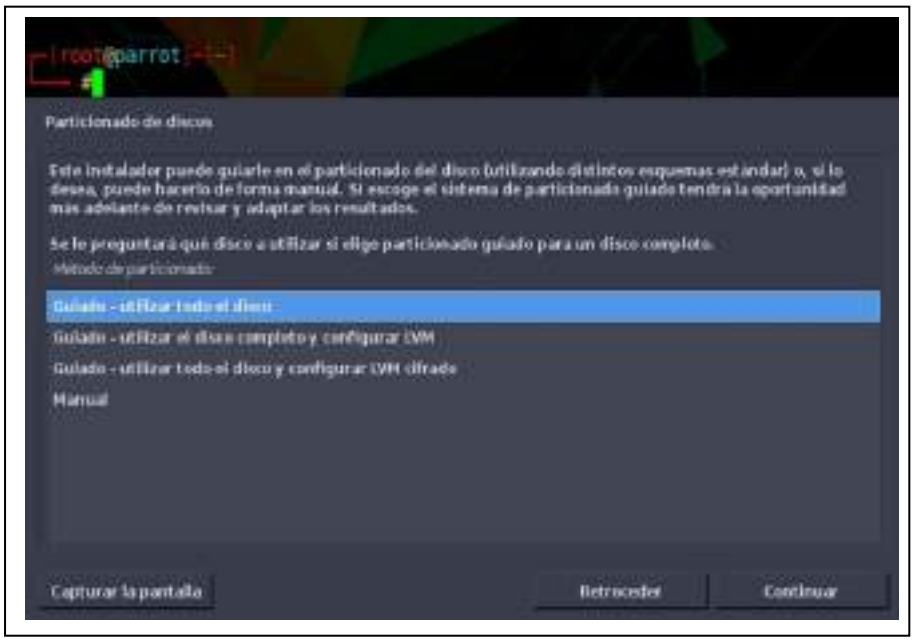

Figura 37: Partición de discos
Elaborado por: Rodríguez Zambrano Stalyn

Paso 9: Tenemos tres tipos de particiones, las cuales serán gestionada de acuerdo a nuestras necesidades.

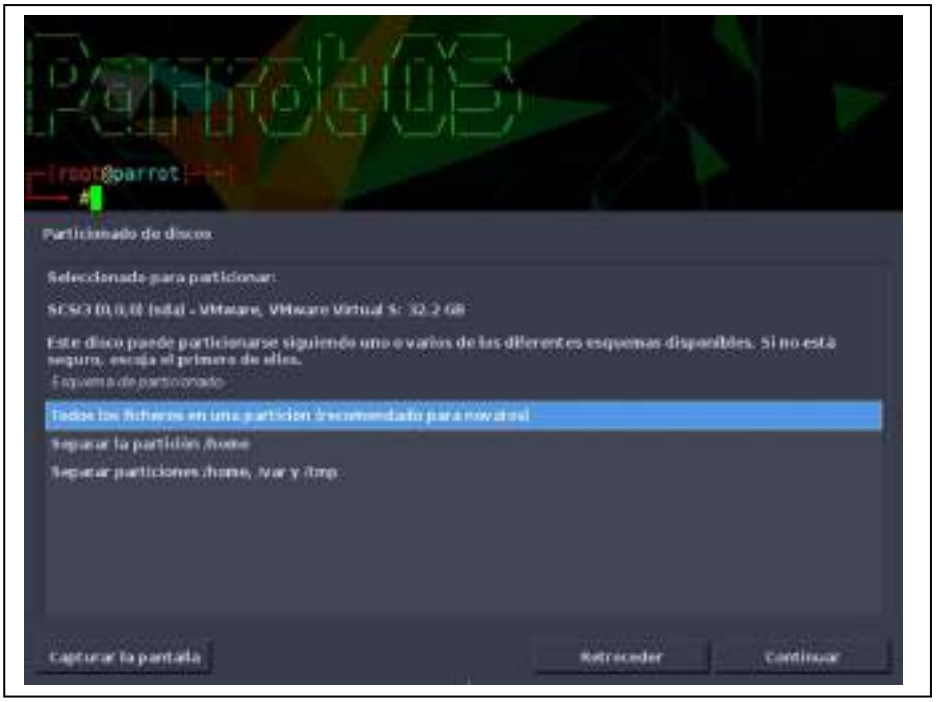

Figura 38: Partición de disco para novatos
Elaborado por: Rodríguez Zambrano Stalyn

Paso 10: Visualizamos el resumen de las particiones, como quedan distribuidas en nuestro disco virtual.

Figura 39: Configuración final de la partición de disco
Elaborado por: Rodríguez Zambrano Stalyn

Paso 11: Al final del proceso de instalación el sistemas nos consulta si deseamos cargar el arranque GRUB[1].

Figura 40: Arranque del GRUB en el disco duro
Elaborado por: Rodríguez Zambrano Stalyn

[1] GRUB" GRand Unified Bootloader": Se puede definir como un cargador de arranque unificado o gestor de arranque múltiple, este fue creado por el proyecto GNU que lidero el científico Richard Stallman, y su funcionalidad consiste en administrar el inicio de uno o más sistemas operativos que se encuentren instalados en un determinado equipo de cómputo.

Paso 12: Configuramos la dirección física desde donde va arrancar el GRUB en el disco duro.

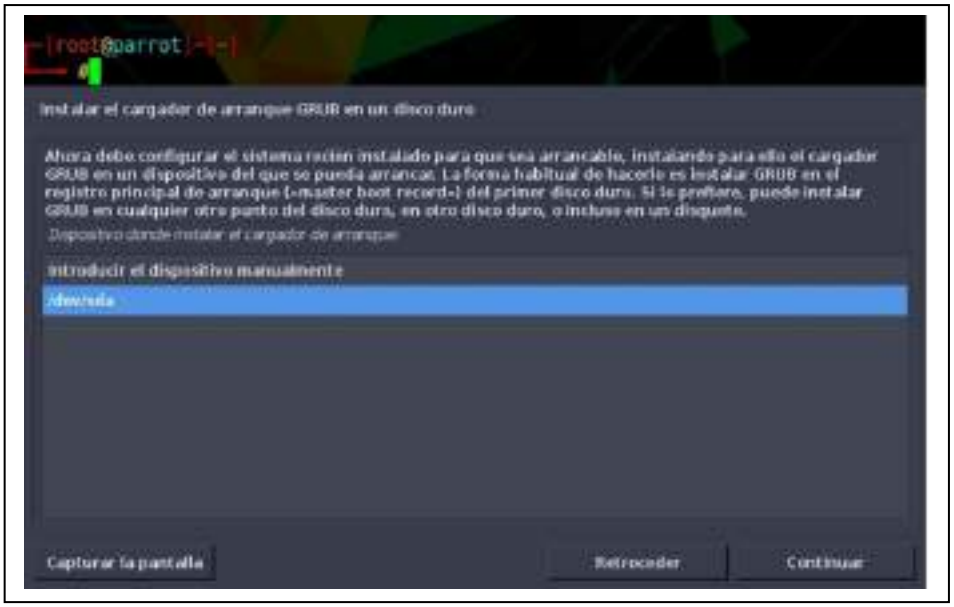

Figura 41: Dirección de arranque del GRUB
Elaborado por: Rodríguez Zambrano Stalyn

Paso 13: Finalizado el proceso de instalación, podemos acceder a trabajar en nuestro distribución de sistema operativo basado en seguridad informática.

Figura 42: Ingreso al sistema Parrot Security Os
Elaborado por: Rodríguez Zambrano Stalyn

4.2.2 Consideraciones

Las distribuciones de Linux son muy diversas, para este caso de estudio el sistema operativo Parrot Security Os es un sistema operativo basado en seguridad informática, se debe de tomar en cuenta las siguientes consideraciones para su uso y aplicación en el ámbito informático:

- Este sistema operativo Parrot Security Os cuenta con gran variedad de aplicaciones para realizar auditoria informática, hacking ético, informática forense, análisis de datos.
- Este sistema operativo fue concebido para encontrar vulnerabilidades informáticas de todo tipo.
- Parrot Security Os tiene un kit de aplicaciones importante cada una destinada para un fin específico:
    - Recopilación de Información.
    - Aplicaciones Web.
    - Ataques de contraseñas.
    - Ataques Wireless.

Figura 43: Kits de aplicaciones de Parrot Security OS
Elaborado por: Rodríguez Zambrano Stalyn

- Entre las aplicaciones destacadas para análisis de vulnerabilidades web tenemos la herramienta OWASP la cual es base de estudio para desarrollar esta tesis.

Figura 44: Aplicación de análisis Web OWASP
Elaborado por: Rodríguez Zambrano Stalyn

El sistema operativo Parrot Security Os ofrece un amplio marco de posibilidades con sus aplicaciones, que son muy útiles a la hora de utilizar la herramienta OWASP, lo cual nos garantiza procedimientos adecuados para la correcta practica de evaluación del sistema de gestión académico SGA.

4.3 Procedimiento

En este caso de estudio se estable como procedimientos los todas las configuraciones de acuerdo a la normativa de la guías de pruebas OWASP Foundation, con la finalidad de tener una herramienta que preste todas las garantías de funcionalidad antes de poner a prueba su ejecución.

Establecer un canal seguro de comunicación entre la aplicación OWASP y la aplicación web en este caso el navegador Firefox.

Se configura el localhost del navegador Firefox y el puerto de enlace 8080.

- Accedemos al menú de personalización de la página para ir a opciones generales.
- Ingresamos a la configuración de red y procedemos a realizar la configuración.

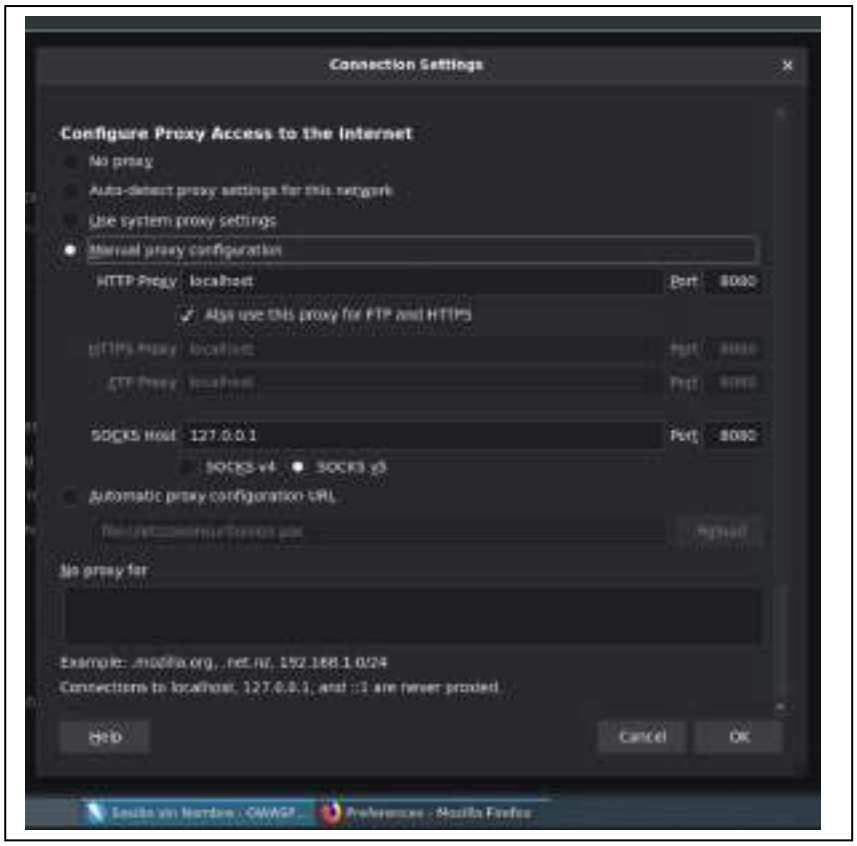

Figura 45: Configuración de Localhost y puerto de enlace.
Elaborado por: Rodríguez Zambrano Stalyn

Esta configuración se establece para tener una comunicación segura entre el cliente y servidor, con el fin de evitar otro tipo de entradas no autorizadas o desconocidas en nuestra prueba de la herramienta OWASP.

En nuestra herramienta OWASP procedemos a realizar la misma configuración que efectuamos en nuestro navegador Firefox, con el fin de

tener el mismo direccionamiento localhost y del puerto de enlace, con estos procedimientos efectuados tendremos una comunicación segura.

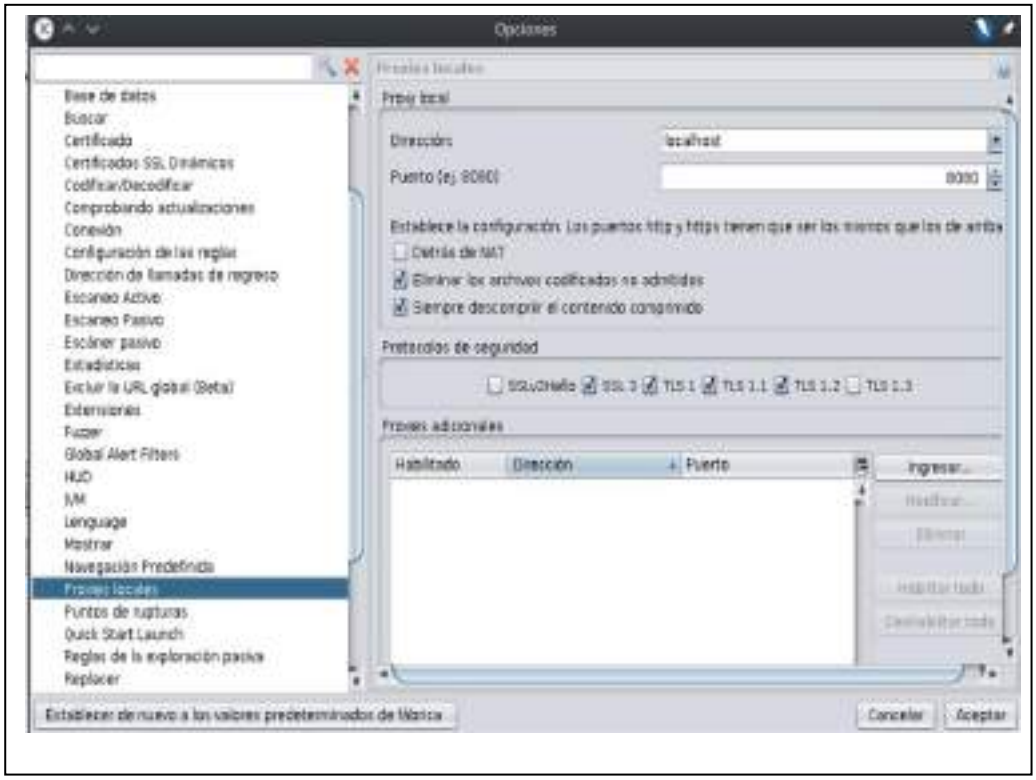

Figura 46: Configuración Herramienta OWASP
Elaborado por: Rodríguez Zambrano Stalyn

Generalmente cuando nos referimos a la seguridad que debe tener una aplicación web, se deben de establecer ciertos criterios con respecto a un canal seguro para la comunicación entre el cliente y servidor, por lo tanto debemos determinar varios métodos que nos faciliten el uso de una comunicación segura.

Utilizaremos el método de configuración de certificados digitales Secure Sockets Layer “SSL”, lo que nos permitirá cifrar todo el tráfico de datos que tendremos en nuestro navegador “Firefox” con algún sitio web especifico “SGA”.

En la herramienta OWASP procedemos a generar un tipo de certificado SSL dinámico para poder ser cargado en nuestro navegador Firefox.

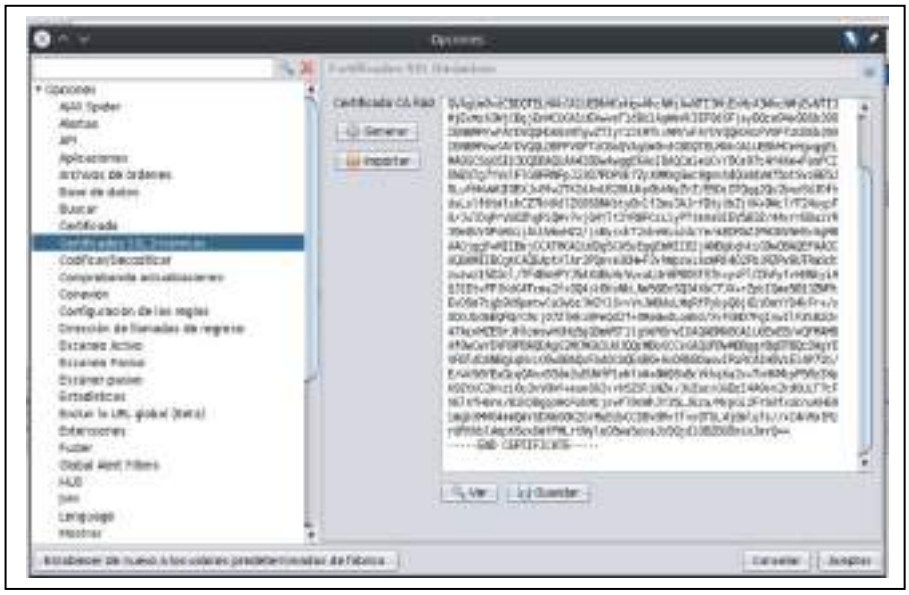

Figura 47: Certificado Dinámico SSL
Elaborado por: Rodríguez Zambrano Stalyn

Este tipo de cifrado impide la intercepción de la información que transmitimos desde nuestro servidor llamado punto A al punto B que en este caso sería un sitio Web.

Para este caso de estudio el sitio web destinado para el análisis en tiempo real será el sistema de gestión académico SGA.

Figura 48: Carga del Certificado SSL en navegador Firefox
Elaborado por: Rodríguez Zambrano Stalyn

Finalmente tenemos en línea la herramienta OWASP con nuestro navegador web Firefox lista para efectuar las pruebas.

Figura 49: OWASP configurado para pruebas de aplicación web SGA
Elaborado por: Rodríguez Zambrano Stalyn

Entre uno de los aspectos importantes del OWASP, se puede decir que esta aplicación dentro del modelo OSI de 7 capas, su estándar de gestión se encuentra definido en la capa 5, 6 y 7, debido a que en estas capas realiza todas las gestiones de control, iniciando con el módulo de aplicación que es la encargada de direccionar las acciones desde el sistema operativo, utilizando cada uno de los protocolos como FTP, HTTP, y otros. En el módulo de presentación la aplicación OWASP, estandariza los métodos de comunicación que va establecer con el usuario, es decir que será un intérprete de datos.

Finalmente en la capa de sesión gestionara la conexiones existentes entre usuarios estableciendo la comunicación entre varios hots, con el fin de tener la comunicación de la sesión abierta de manera continua, los protocolos más conocidos que gestiona esta capa están ZIP, ZAP, NetBios, SMTP, SQL y otros.

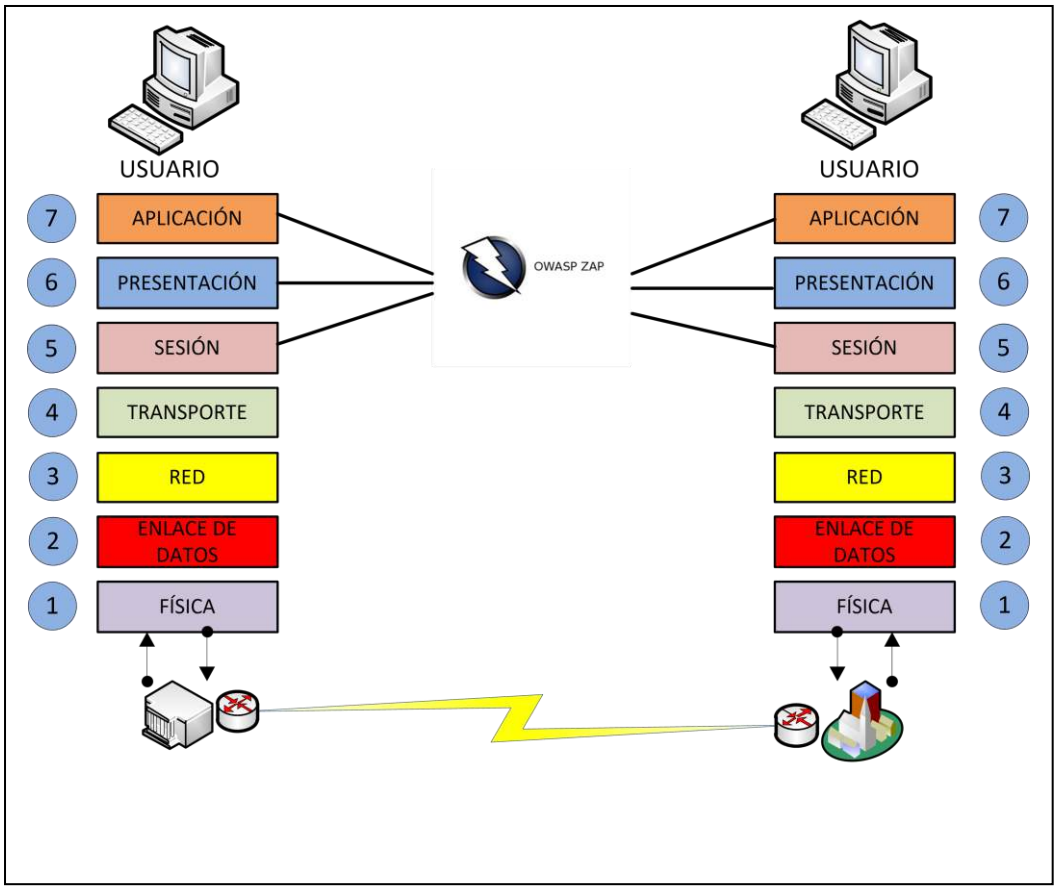

Figura 50: OWASP en el modelo OSI de 7 capas.
Elaborado por: Rodríguez Zambrano Stalyn

En el gráfico 50 podemos ver la interacción de la aplicación OWASP con el modelo OSI de 7 capas, procesando el intercambio de información entre usuarios, que a su vez se ejecutan entre los diferentes host de origen y destino.

4.3.1 Pruebas de laboratorio

En esta prueba se realizaran algunos procesos de conectividad a la red utilizando anonimato de direccionamiento IP con la herramienta Annosurf

y utilizando el enmascaramiento de red con un Vpn gratuito, con estas herramientas se aplicaran métodos seguros para efectuar una prueba de análisis con la herramienta OWASP.

Anonsurf

Esta herramienta nos brinda un anonimato en la conexión de la IP pública de nuestro proveedor de servicios, se ejecuta de forma automática al iniciar la exploración con el navegador, realiza cambios en nuestra DNS y dirección IP publica, lo que permitirá que nuestra conexión sea anónima. Esta aplicación requiere la utilización de navegadores como TOR, pero para este caso de estudio se utiliza el navegador Firefox.

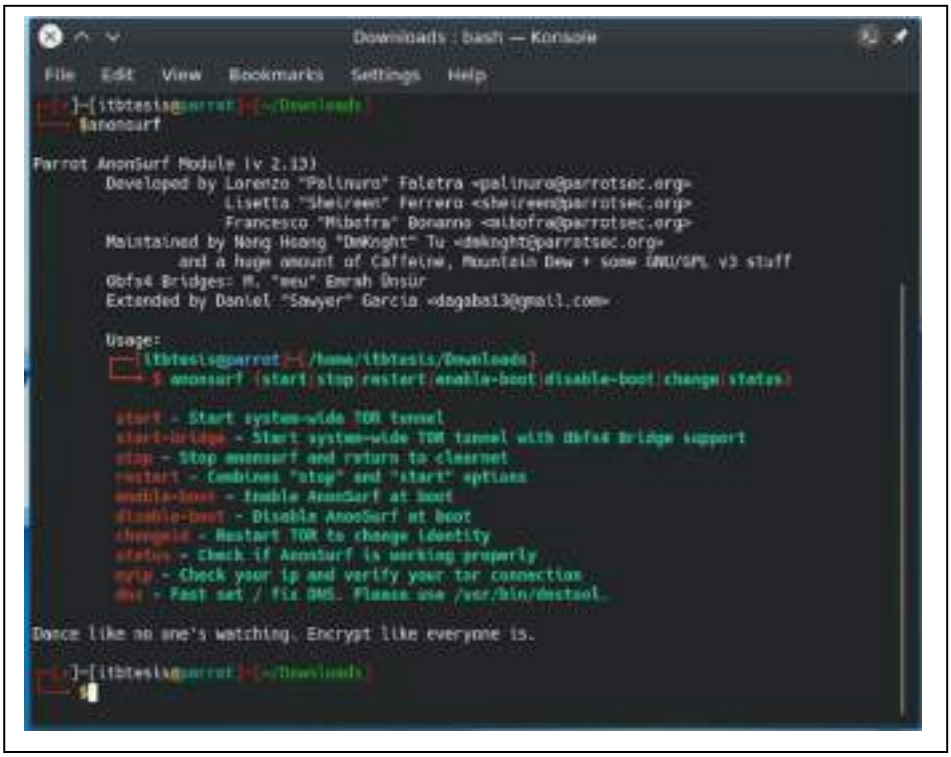

Figura 51: Terminal de Parrot Security invocando comando Anonsurf. Elaborado por: Rodríguez Zambrano Stalyn

Con el anonimato de nuestra ip tenemos un método efectivo para evitar rastreos durante nuestra utilización de la herramienta OWASP.

Para invocar a la aplicación Anonsurf su activación, debemos inicializar el modo de ejecución de superusuarios ejecutando el comando Sudo Su en la terminal de sistema operativo Parrot Security Os:

Figura 52: Comando anonsurf para invocar a la aplicación.
Elaborado por: Rodríguez Zambrano Stalyn

Con Anonsurf podemos visualizar el estado de nuestra conexión y poder determinar por cada uno de los eventos, cual es el ancho de banda que consumimos y el tiempo de acceso a cada sitio web.

Utilizamos el comando **Anonsurf status** para acceder a este servicio de la aplicación.

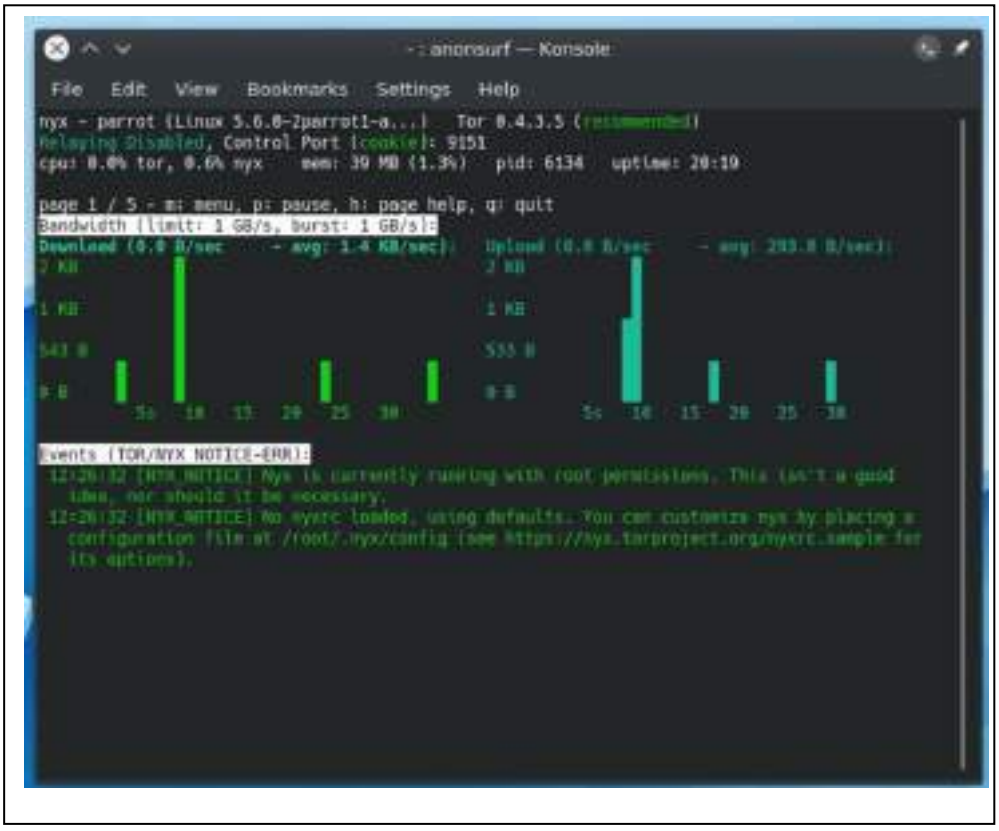

Figura 53: Estado del servicio de anonimato de anonsurf.
Elaborado por: Rodríguez Zambrano Stalyn

Tener anonimato con la aplicación Anonsurf en nuestra conexión hace que tengamos seguridad al momento de iniciar un escaneo de vulnerabilidades con la herramienta OWASP, adicional necesitamos cifrar nuestros datos por lo cual se creara una sesión de VPN “Virtual Private Network”, para este caso de estudio seleccione la aplicación gratuita VpnBook.

Virtual private Network “VPN”

Una VPN es un software de aplicación que se ejecuta sobre un servidor físico el mismo que puede estar ubicado en nuestro país o en cualquier otra parte del mundo, mediante su configuración ejecuta un túnel enmascarando nuestra dirección física IP y cifrando nuestros datos, este cifrado se denomina de extremo a extremo.

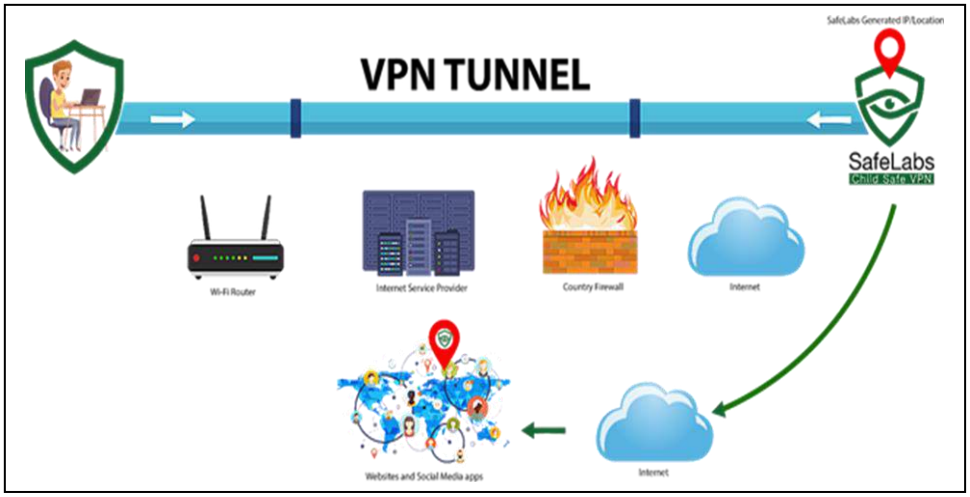

Figura 54: Conexión de un VPN.
Fuente: https://www.redeszone.net/tutoriales/seguridad/ocultar-direccion-ip-internet/

VPNBook

Es una aplicación gratuita la cual nos brinda cifrado de información manteniendo nuestra IP oculta, utilizando el método AES de 256 bits con este tipo de cifrado es invulnerable ante cualquier tipo de amenaza. Esta VPN cuenta con un proxy propio el cual permite la utilización de todo el ancho de banda sin limitaciones.

Figura 55: Página de inicio de VPNBook.
Elaborado por: Rodríguez Zambrano Stalyn

Para establecer la configuración de cifrado de datos en nuestra red doméstica, accedemos a la página web de VNPBook, procedemos a descargar las certificaciones de cliente con su respectiva autenticación de usuario y contraseña.

Para iniciar la correcta conexión con el servidor de VPNBook y realizar la tunelización y cifrado de datos debemos crear una nueva sección de conexión en nuestro equipo y cargar los parámetros ya obtenidos.

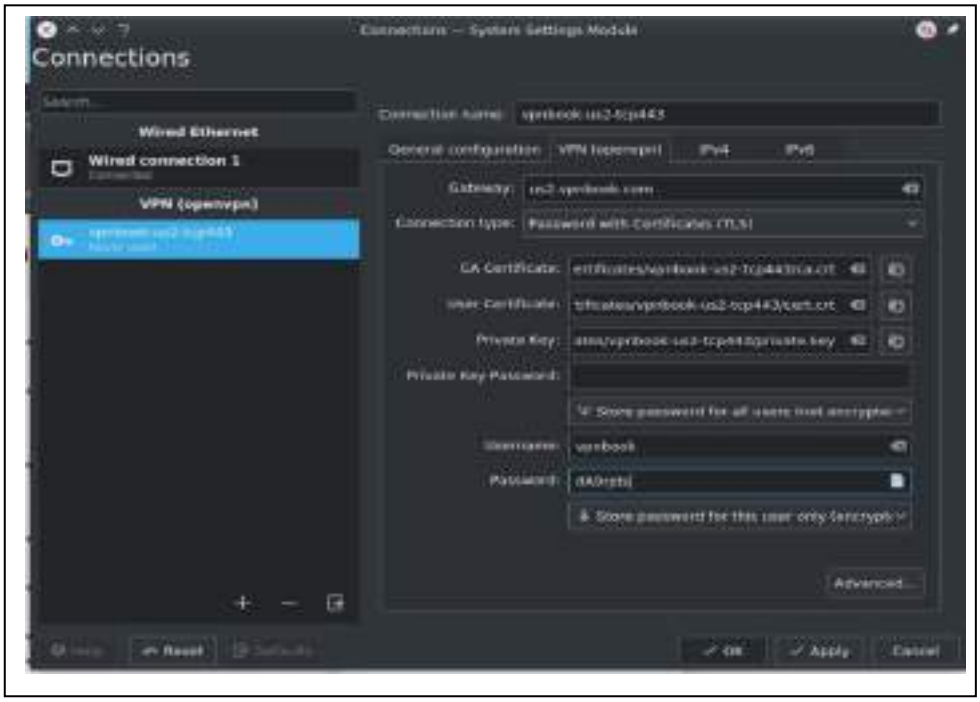

Figura 56: Configuración de VPNBook de nuestro equipo.
Elaborado por: Rodríguez Zambrano Stalyn

Una vez configurado nuestro equipo de cómputo a nivel de su conexión de red, para nosotros poder certificar que tenemos privacidad y anonimato, realizamos la comprobación de las configuraciones ya establecidas, accediendo a nuestro navegador Firefox digitamos la **palabra clave IP** y nos debe demostrar, que nuestro equipo esta automáticamente trabajando con un servido ubicado en otro país.

Esto quiere decir que no expondrá ni nuestro proveedor de servicios habitual ni tampoco expondrá ubicación, ni nuestros datos.

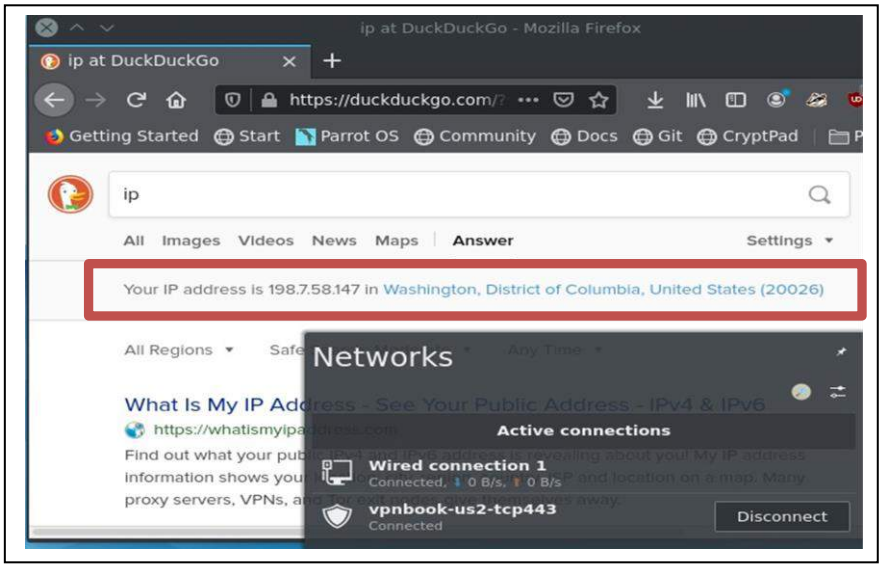

Figura 57: Conexión activada con VPNBook.
Elaborado por: Rodríguez Zambrano Stalyn

En el gráfico 55 podemos visualizar que nuestra IP está conectada con un servidor VPNBook en otro país siendo este Estados Unidos y en estado de Washington distrito de Columbia con una dirección IP 198.7.58.147.

4.3.2 Pruebas de ejecución

OWASP es una herramienta utilizada para realizar diversos tipos de auditorías basadas en desarrollo de aplicaciones web o búsqueda de vulnerabilidades web.

En esta prueba vamos a utilizar diversos métodos que nos permite realizar la herramienta OWASP para verificar, analizar y detallar cada una

de las novedades que se encuentre en el sistema de gestión académico “SGA”.

Es importante conocer cada uno de los aspectos de la herramienta OWASP, para poder interpretar cada escenario que nos presente de acuerdo a cada una de las vulnerabilidades y riesgos que pueda presentar la aplicación web “SGA”, en este caso de estudio.

El usuario de la herramienta OWASP debe tener conocimientos en distintos campos de aplicación como manejo de base de datos, redes, administración de centro de cómputo, lenguajes de programación etc.

Al iniciar la herramienta OWASP el primer mensaje que nos muestra es, si nosotros como usuarios deseamos permanecer en la sesión que ejecutamos. OWASP nos permite almacenar cada sesión iniciada facilitándonos el poder almacenar en una parte especifica de nuestro disco la información y luego continuar con el proyecto iniciado anteriormente.

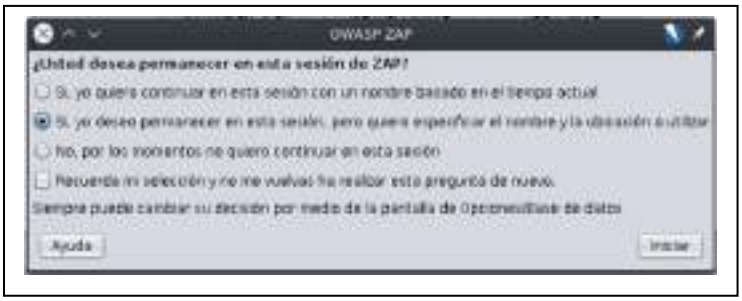

Figura 58: Inicio de sesión de OWASP.
Elaborado por: Rodríguez Zambrano Stalyn

Para iniciar el proceso de escaneo pasivo de la herramienta OWASP a la aplicación web del “SGA”, procedemos con ingresar la dirección https://sga.itb.edu.ec/ en el campo URL to attack de la herramienta OWASP y procedemos a dar inicio con el proceso de análisis de vulnerabilidades de sistema de gestión académico “SGA”.

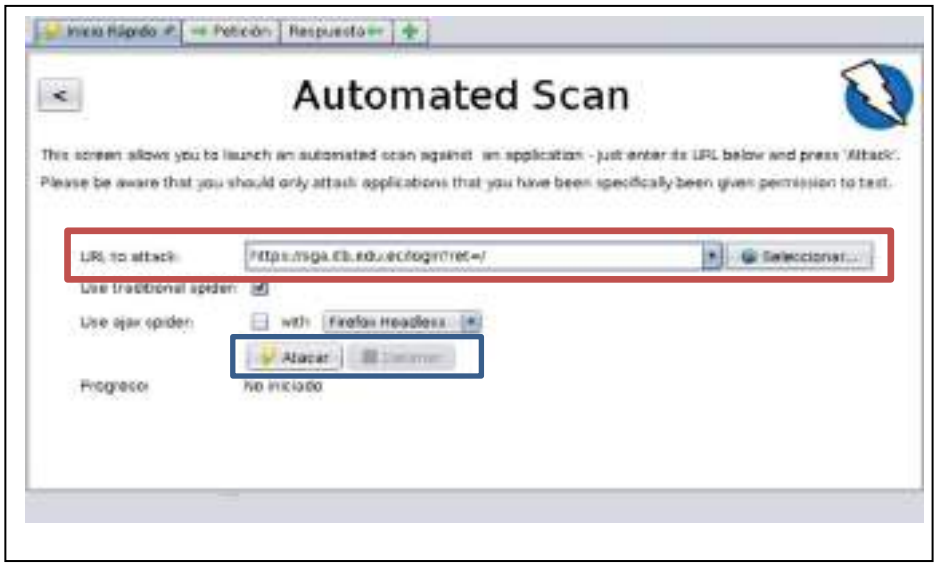

Figura 59: Escaneo de dirección https.
Elaborado por: Rodríguez Zambrano Stalyn

Una vez efectuado el escáner Spider nos muestra un árbol de resultados que nos indica el tipo de peligros que tiene la aplicación web, para este caso de estudio la aplicación web analizada es el sistema de gestión académico “SGA”.

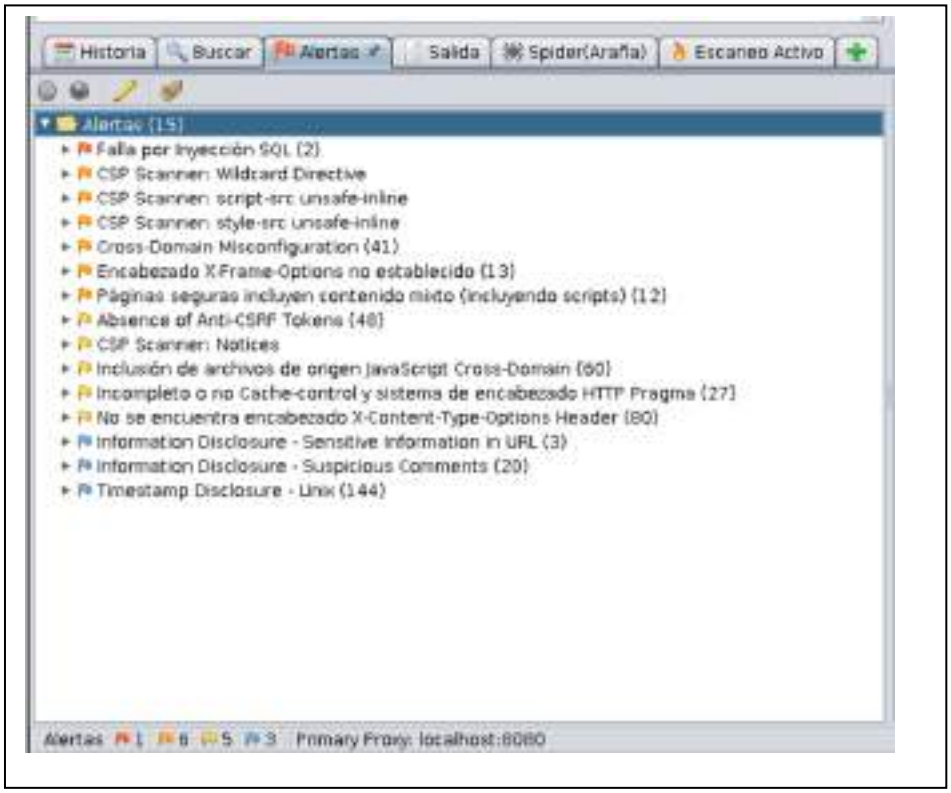

Figura 60: Alertas de vulnerabilidades OWASP.
Elaborado por: Rodríguez Zambrano Stalyn

Cada color de bandera significa el tipo de prioridad o importancia que debemos de dar a los resultados obtenidos.

Tabla 19: Estados de alerta

	Alerta con alta prioridad
	Alerta con prioridad media
	Alerta con baja prioridad
	Alertas informativas

Tabla 18: Elaborado por: Rodríguez Zambrano Stalyn

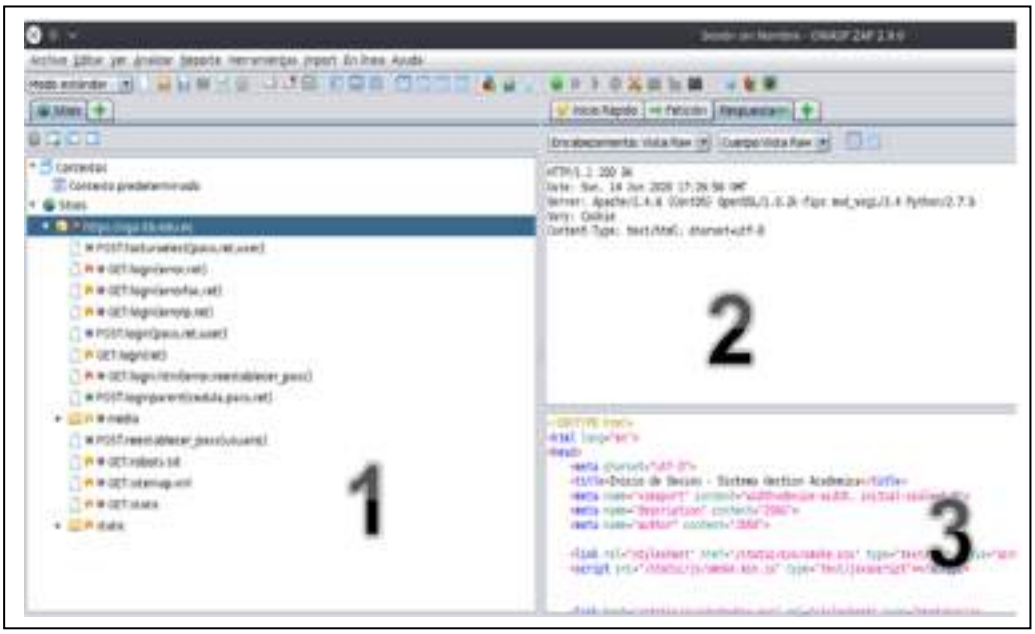

Figura 61: Pantalla de resultados del análisis de vulnerabilidades.
Elaborado por: Rodríguez Zambrano Stalyn

En la figura 59 tenemos un gráfico con tres puntos importantes los mismos que explicare a continuación:

1. **Árbol de directorio:** El árbol de directorios contiene la navegación del dominio resultante del escaneo. Nos mostrara el detalle de las novedades encontradas por cada directorio escaneado.
2. **Encabezamiento de vistas Raw:** Esta vista nos muestra los datos sensibles que nosotros enviamos y recibimos del servidor al que estamos escaneando. En este caso se visualiza que la aplicación

web del sistema de gestión académico “SGA” funciona con la versión de apache 2.4.6, Centos, Python 2.7.5.

3. **Cuerpo de vistas Raw:** Muestra la estructura del lenguaje de programación sobre el cual está diseñada la aplicación web.

4.3.3 Interpretación de resultados

La prueba de escaneo de vulnerabilidades se efectuada el día domingo 14 de junio del 2020 en la que tenemos los siguientes resultados.

Alertas de Alta Prioridad

- Falla por inyección SQL.
- Inyección remota de comandos OS.

Alertas de prioridad media

- CSP Scanner: Wildcard Directive.
- CSP Scanner: Script-src unsafe-inline.
- CSP Scanner: Style-src unsafe-inline.
- Configuración incorrecta entre dominios, “Cross-Domain Mis configuration”.
- Encabezado X-Frame-options no establecidos.

Alertas de prioridad baja

- Ausencia de tokens Anti-CSRF “Absence of Anti-CSRF Tokens”.
- Inclusión de archivos de origen JavaScript Cross-Domain.
- No se encuentra encabezado de opciones de tipo de contenido X, “X-Content-Type-Options Header”.

Alertas informativas

- Divulgación de información confidencial en URL,”Information Disclosure - Sensitive information in URL”.
- Divulgación de información de comentarios sospechosos, “Information Disclosure - Suspicious Comments”.
- Divulgación de marca de tiempo Unix, “Timestamp Disclosure-Unix”.

4.3.3.1 Vulnerabilidad o alerta de alta prioridad

Fallas por Inyección Sql

Es un fallo muy común encontrar en aplicaciones web que tienen un acceso directo al servidor, debido al uso de métodos de ingreso al sistema por medio de usuarios y contraseñas.

El método de ataque por inyección de código Sql puede ser de diferentes tipos, de acuerdo a la condición lógica que se estructure una consulta dirigida al servidor.

SELECT* FROM User = WHERE usuario = $usuario AND clave = '$clave';

En este ejemplo vemos como realizamos una consulta normal a nuestro servidor para saber si los datos ingresados son los correctos.

Pero qué pasaría si decidimos cambiar los parámetros $usuario y $password por operadores lógicos como OR en la estructura de la línea de comandos de Sql.

SELECT* FROM User = WHERE usuario = 'OR'1' = '1' AND clave = 'OR'1' = '1';

En este caso tendremos que el valor 1=1 nos devolverá un valor por verdadero y por ende nos dará un registro de la tabla usuarios. Con este sencillo método prácticamente el atacante puede tener acceso a nuestra web server, y aplicar otros métodos de inyección de código Sql, para poder tomar control total de todas las acciones que realicemos en nuestro servidor.

Inyección remota de comandos OS

Esta vulnerabilidad se refiere a la inyección de código malicioso directamente sobre el sistema operativo. Esta vulnerabilidad puede ser utilizada mediante el método Get de la aplicación web.

4.3.3.2 Vulnerabilidad o alerta de prioridad media

CSP Scanner: wildcard directive

El CSP "Content Security Policy" es una política de seguridad para mitigar ciertos ataques a las aplicaciones web.

Wildcard directive: Es una directiva de comodín, un comodín es una lista de caracteres especial utilizado como método de asignación en una determinada línea de código, un comodín puede ser *, -, +, [],{},%, etc.

Ejemplo:

Realizamos una selección de los clientes que tienen como patrón “es”.

SELECT * FROM Clientes WHERE Ciudad LIKE **'%GYE%';**

Normalmente esta búsqueda considerando de 1 a 150000 registros demora 1 segundo, pero si la misma búsqueda la realizamos con caracteres especiales, el tiempo de repuesta de esta petición al servidor puede ser mayor de lo usual.

Esta vulnerabilidad puede generar lo que se conoce una denegación de servicios sobre la capa de aplicación.

SELECT * FROM Clientes WHERE Ciudad LIKE **'%_[^!_%/%a?F%_D)_(F%)_%([](){}%){()}£$&N%_)$*£()$*R"_)][%](%[x])%a][$*"£$-9]_%';**

Script-src unsafe-inline

Es un recurso utilizado por la web para cargar y validar líneas de códigos JavaScript, además de cargar URL, controles, comandos y otras secuencias de eventos. Esta política permite que no se pueda reescribir líneas de código JavaScript en tiempo en tiempo real para evitar inyección de código malicioso.

Ejemplo del código fuente tomada de la página web del sistema de gestión académico “SGA”.

CSP Scanner: Style-src unsafe-inline

Es un recurso utilizado para validar Fuentes de textos para hojas de estilos CSS. En si controla los estilos de texto en las líneas de código pero no realiza el controla los atributos.

Cross-Domain Misconfiguration

Se utiliza para el realizar el intercambio de información entre un sub dominios o entre muchos dominios. Al existir una configuración no adecuada se puede comprometer la información confidencial de la

aplicación web, como Identificación de usuarios, identificación de correos electrónicos conocer, cual es el origen de la transmisión de sus datos.

Encabezado X-Frame-options no establecidos.

X-Frame-options es un encabezado HTTP que proporciona un parámetro de solicitud de permiso sobre una determinada página web, pueda ser renderizada en un frame, iframe, objetc. Técnicamente no permite asociar un conjunto de elementos de una página secundaria html, sobre una página html principal, esta opción evita que atacantes puedan utilizar la técnica de Clickjacking.

4.3.3.3 Alertas de baja prioridad

Absence of Anti-CSRF Tokens

Cross-Site Request Forgery(CSRF) La función de este tokens es evitar la falsificación de cualquier tipo de solicitud de datos entre sitios web. Generalmente los fallos de seguridad son ocasionados por parte del cliente, el cual puede usar peticiones para redirigir a sus usuarios a un sitio web en específico, extremadamente peligroso para hurtar, falsificar información valiosa.

Inclusión de archivos de origen JavaScript Cross-Domain

Esta solicitud de dominios cruzados de JavaScritp hace referencia a los datos de alojamiento de información que se encuentran en dominios y hosting de terceros y que a su vez hacen peticiones de información y de recursos a otros servidores de alojamiento.

X-Content-Type-Options Header

Esta cabecera de opciones de tipo de contenido X, evita que el browser cargue cualquier tipo de contenido que sea distinto al parámetro ya configurado en el Content-Type de las cabeceras HTTP, evita que código malicioso diseñado en el lenguaje de programación JavaScript cree sentencias autoejecutables de nuevos códigos algorítmicos, que afecten el funcionamiento correcto de los MIME “Multipurpose Internet Mail Extensions”.

4.3.3.4 Alertas informativas

Information Disclosure-Sensitive information in URL

La divulgación de la información sensible a URL es un mensaje informativo que se basa en políticas de cumplimiento con respecto a la filtración de información a través de URL. Para casos de estudio se basa en la etiqueta de restablecimiento de contraseñas.

Information Disclosure- Suspicious Comments

La divulgación de información – comentarios sospechosos, este mensaje informativo nos indica que en nuestra estructura de código Script tenemos información que puede ser sensible para exponerla en la red, porque puede suponer una vulnerabilidad debido a que ciertos comentarios pueden ser de ayuda para que un atacante se aproveche de dicha información.

Timestamp Disclosure- Unix

Divulgación de marca tiempo de Unix este mensaje informativo nos indica que nuestro servidor tiene una filtración de información con respecto al tiempo en línea. Este tipo de filtración de información puede exponer patrones de datos que pueden ser aprovechados por un atacante.

4.3.4 Seguimiento

Durante el desarrollo de esta tesis, se han efectuado seguimientos de procesos con la herramienta OWASP a las distintas anomalías que presenta el sistema de gestión académico “SGA”.

4.4 Resultados

De acuerdo al análisis con la herramienta OWASP, se realiza un muestreo de resultados por evaluaciones mensuales al sistema de gestión académico “SGA”.

Parte de este estudio se toma como referencia principal la guía top ten de OWASP, con la misma obtendremos una métrica, para saber el grado de riesgo que está expuesto sistema de gestión académico “SGA”.

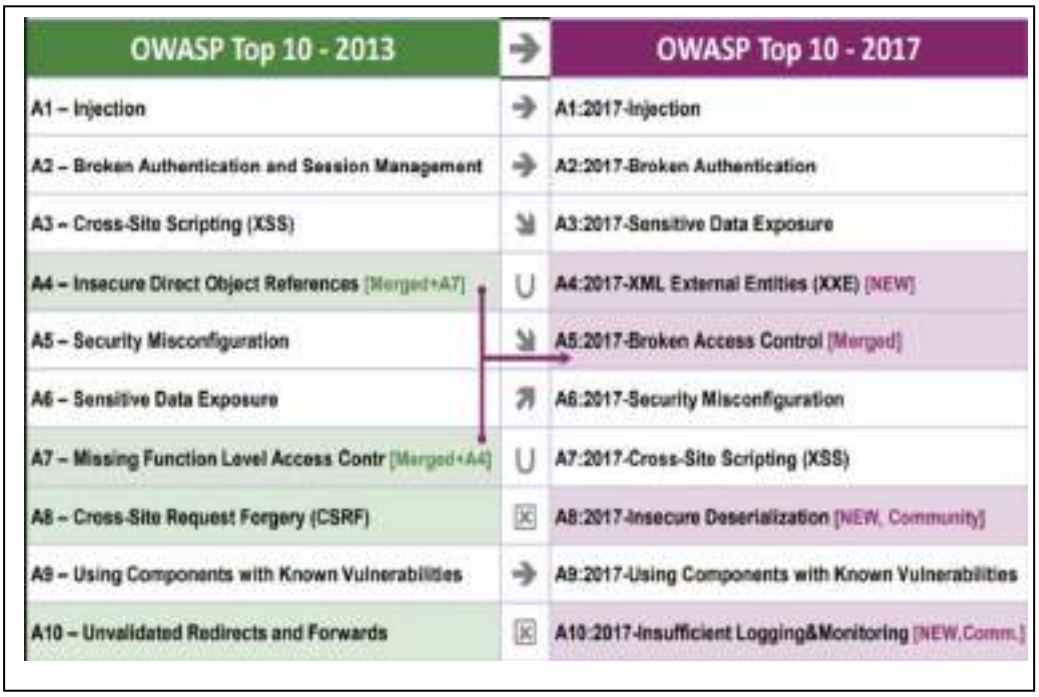

Figura 62: Guía OWASP de vulnerabilidades 2013 antigua y 2017 actual. Fuente: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/

Resultado de Prueba 1: OWASP 02 de septiembre del 2019

En este análisis efectuado el domingo 2 de septiembre del 2019, se evidenciaron: 2 alertas de Prioridad media y 5 alertas de prioridad baja teniendo un total de 7 alertas, estos parámetros nos muestran posibles fallos o vulnerabilidades. De acuerdo a la con el top ten de OWASP estas vulnerabilidades se encuentran en el parámetro A5: Control de Acceso roto “Broken Access Control”.

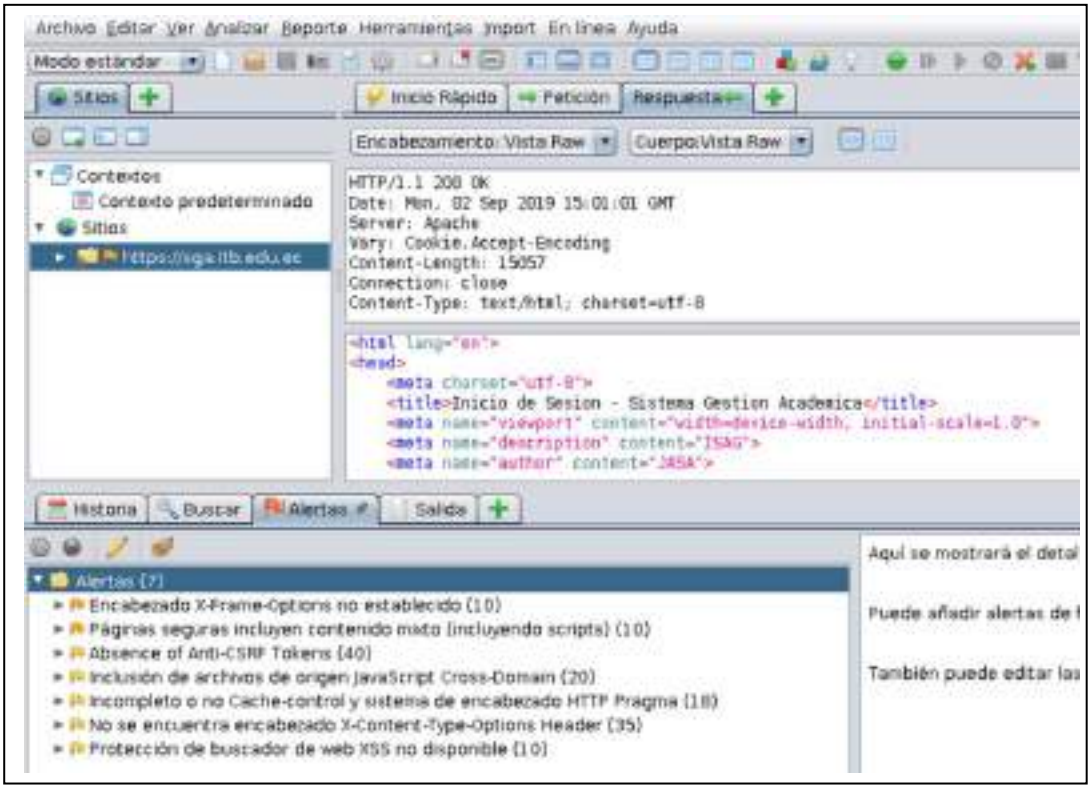

Figura 63: Prueba número 1 de la herramienta OWAS en el “SGA”.
Elaborado por: Rodríguez Zambrano Stalyn

Un control de acceso roto puede ser cualquier tipo de dato que se encuentre alojado en el servidor como un archivo o una data con registros que no tienen el respectivo control de acceso, por consiguiente pueden será objetos de una manipulación por personas no autorizadas en el sistema.

Resultado de Prueba 2: OWASP 14 de octubre del 2019

En este análisis efectuado el día domingo 14 de octubre del 2019, se evidenciaron: 1 alerta de alta prioridad, 3 alertas de prioridad media y 5 alertas de prioridad baja, teniendo un total de 8 alertas, estos parámetros nos muestran posibles fallos o vulnerabilidades. De acuerdo a la con el top ten de OWASP estas vulnerabilidades se encuentran en el parámetro A1: Inyección “Injection”.

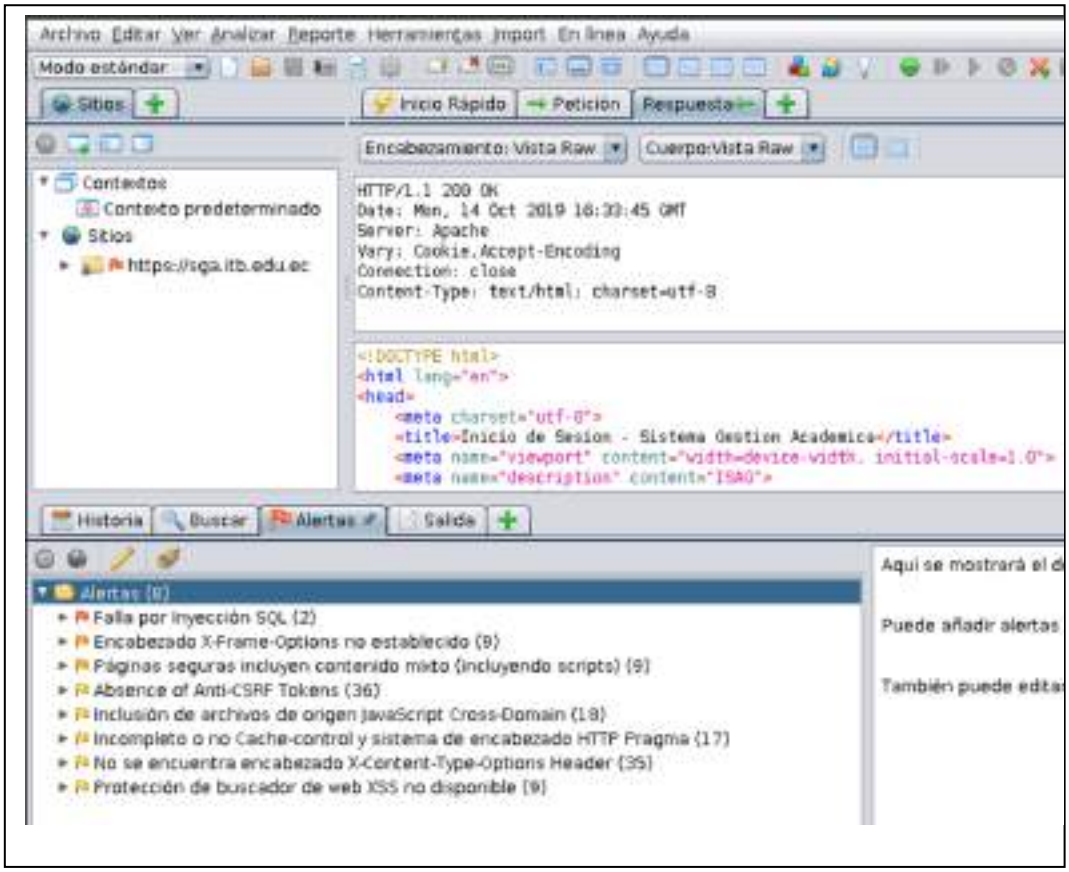

Figura 64: Prueba número 2 de la herramienta OWAS en el “SGA”.
Elaborado por: Rodríguez Zambrano Stalyn

Resultado de Prueba 3: OWASP 11 de noviembre del 2019

En este análisis efectuado el día domingo 11 de noviembre del 2019, se evidenciaron: 2 alertas de prioridad media y 7 alertas de prioridad baja, teniendo un total de 9 alertas, estos parámetros nos muestran posibles fallos o vulnerabilidades. De acuerdo a la con el top ten de OWASP estas vulnerabilidades se encuentran en el parámetro A5: Control de Acceso roto “Broken Access Control”.

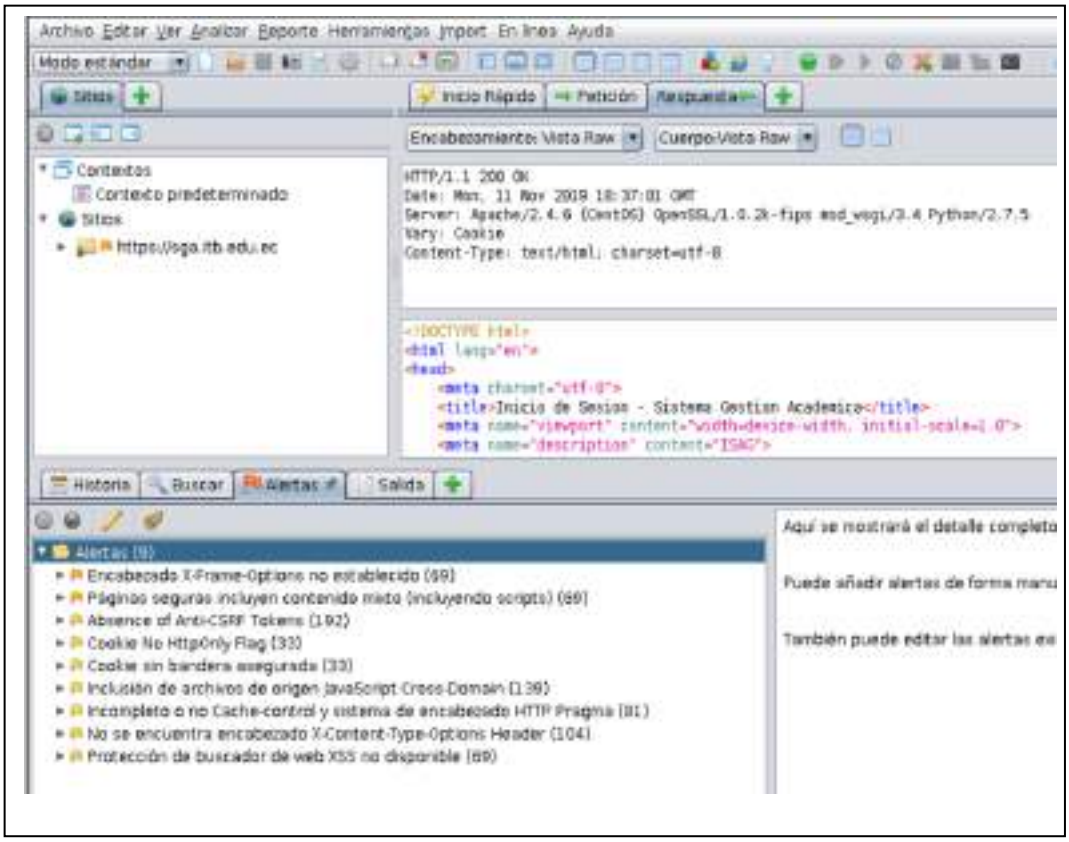

Figura 65: Prueba número 3 de la herramienta OWAS en el “SGA”. Elaborado por: Rodríguez Zambrano Stalyn

Se realizaron pruebas en los meses posteriores dando los mismos resultados de vulnerabilidades. Sin embargo se efectúa un a última prueba a fecha de junio del 2020.

Resultado de Prueba 4: OWASP 16 de junio del 2020

En este análisis efectuado el día martes 16 de junio del 2020, se evidenciaron: 1 alerta de alta prioridad, 2 alertas de prioridad media, 4 alertas de prioridad baja y 3 alertas informativas teniendo un total de 9 alertas, estos parámetros nos muestran posibles fallos o vulnerabilidades. De acuerdo a la con el top ten de OWASP estas vulnerabilidades se encuentran en el parámetro A1: Inyección “Injection”, A5: Control de Acceso roto “ Broken Access Control”.

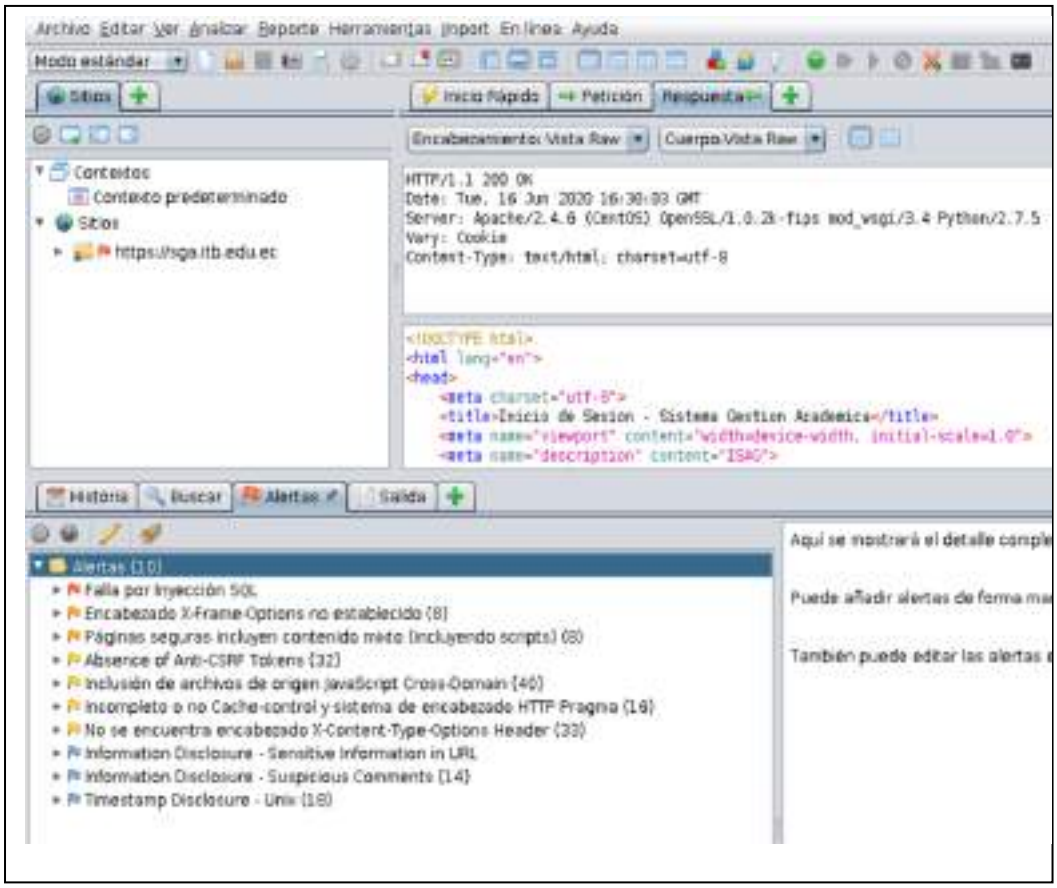

Figura 66: Prueba número 4 de la herramienta OWAS en el “SGA”.
Elaborado por: Rodríguez Zambrano Stalyn

4.4.1 Análisis de los resultados obtenidos

Después de realizar varios análisis de vulnerabilidades, con la herramienta OWASP sobre la aplicación web “SGA”, se identificaron alertas o anomalías persistentes.

Cada uno de los resultados obtenidos en las muestras de análisis representa una prioridad de acuerdo al grado de vulnerabilidad, que representan para la aplicación web “SGA”, por lo tanto cada uno de estos valores presentados serán tomados en consideración para este caso de estudio.

Los datos estadísticos serán importantes para la toma de decisiones futuras con respecto a la seguridad que debe tener el sistema de gestión académico “SGA”.

Se realiza un detalle estadístico de anomalías, de las pruebas efectuada los cuales detallare a continuación.

Tabla 19: Elaborado por: Rodríguez Zambrano Stalyn

Día	Meses	Año	Vulnerabilidades Encontradas
2	Septiembre	2019	7
14	Octubre	2019	8
11	Noviembre	2019	9
16	Junio	2020	10

Tabla 20: OWASP Escaneo de vulnerabilidades

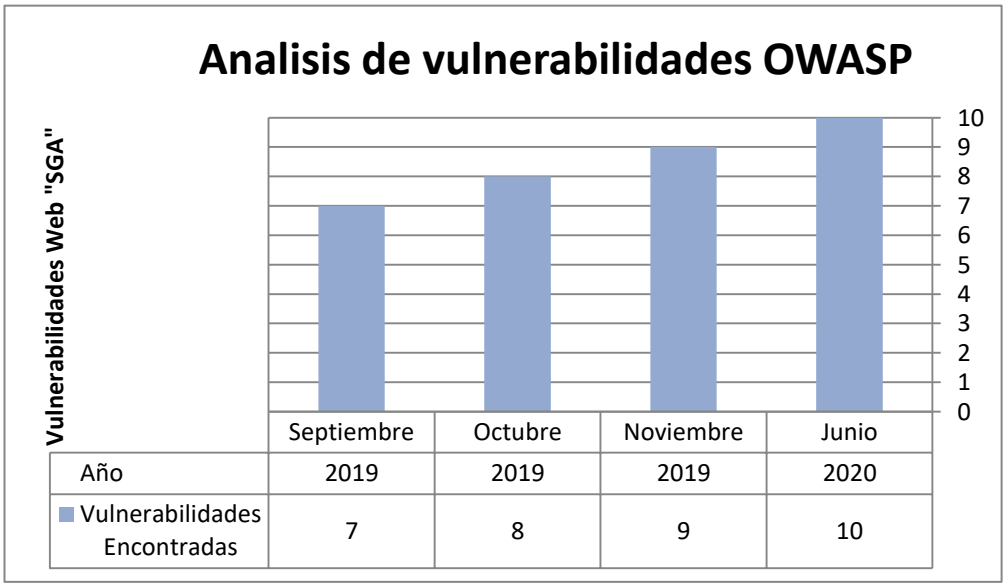

Figura *67: Cuadro estadístico escaneo de vulnerabilidades con OWASP. Elaborado por: Rodríguez Zambrano Stalyn*

Tabla 20: OWASP cuadro de porcentajes de vulnerabilidades

Detalle de Vulnerabilidades	2019 Septiembre	2019 Octubre	2019 Noviembre	2020 Junio	Porcentajes %
Falla por Inyección de código SQL		1		1	50%
Encabezado X-Frame-Option	1	1	1	1	100%
Paginas seguras contenidos mixtos (Scripts)	1	1	1	1	100%
Absence of Anti-CSRF Token	1	1	1	1	100%
Cookie No HttpOnly Flag			1		25%
Cookie sin bandera segura			1		25%
Inclusión de archivos de origen JavaScript Cross-Domain	1	1	1	1	100%
No Cache-Control y encabezado HTTP Pragma	1	1	1	1	100%
X-Content-Type-Options Header	1	1	1	1	100%
Sensitive information in URL				1	25%
Protección Buscador Web XSS	1	1	1		75%
Suspicious Comments				1	25%
TimeStamp Disclosure Unix				1	25%

Tabla 21: Cuadro estadístico de vulnerabilidades que afectan al sistema de gestión académico "SGA".
Elaborado por: Rodríguez Zambrano Stalyn

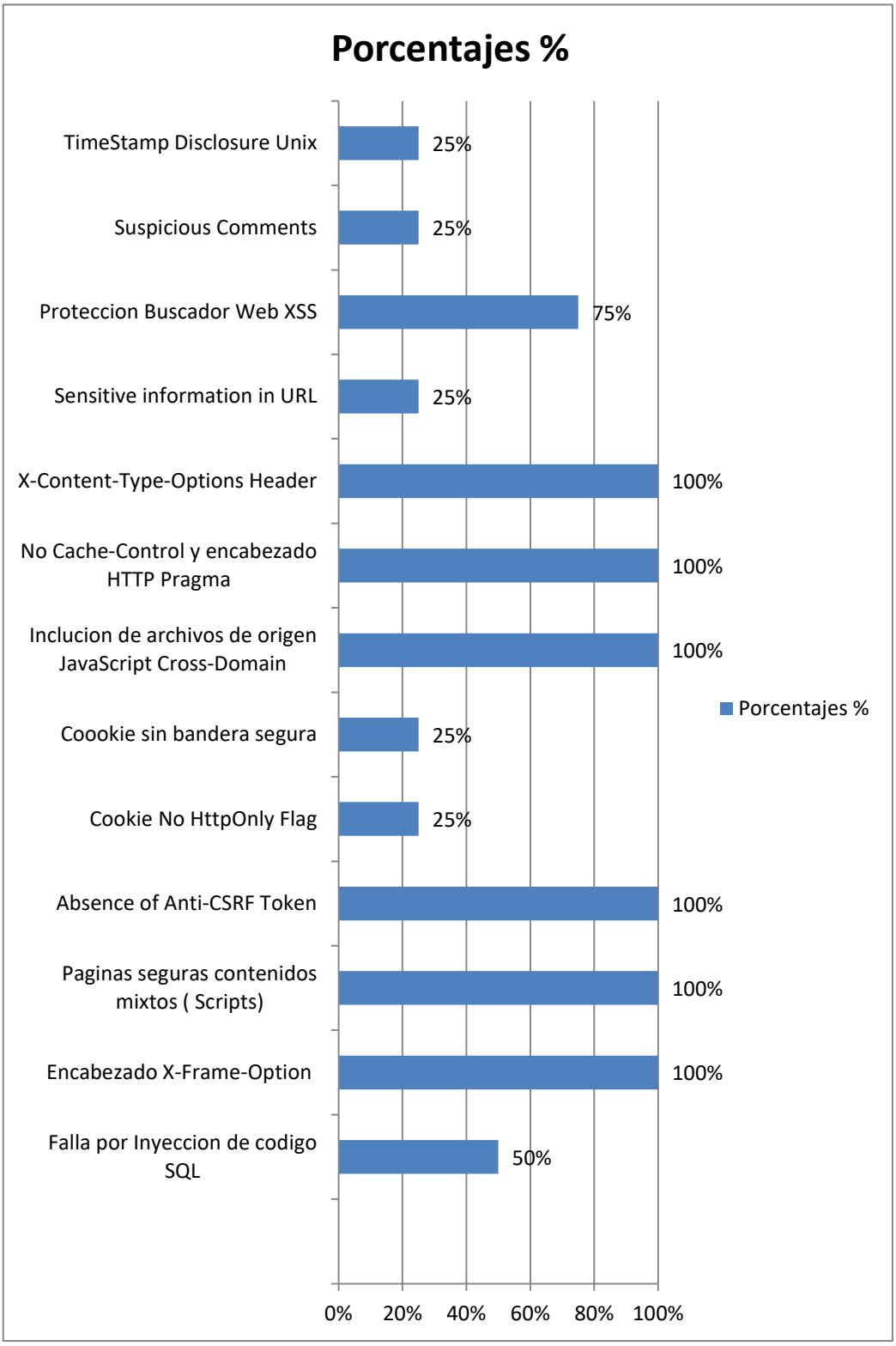

Figura 68: Porcentajes de vulnerabilidades sistema "SGA".
Elaborado por: Rodríguez Zambrano Stalyn

En este presente estudio se analizó la plataforma web del sistema de gestión académico “SGA”, con la herramienta de auditoria de aplicaciones web OWASP y se determina la existencia de factores de riesgo que pueden incidir en la seguridad del sistema de información del “SGA”.

Una vez efectuado el escaneo de vulnerabilidades con la herramienta OWASP, tenemos las más importantes incidencia, en nuestra gráfico mostramos la Inyección de código SQL, como una incidencia de alta prioridad, la cual está presente en un porcentaje del 50% lo que significa que es un dato variable, y que puede este fallo representar en cualquier momento un problema a futuro.

Tenemos otro tipos de vulnerabilidades importantes sin embargo la tasa frecuente de esta anomalía es del 100%, por lo tanto en este presente estudio con la recopilación de información efectuada, se iniciara un proceso de hacer conocer este fallo al departamento de TIC’s con la finalidad de corregir estas anomalías en la plataforma de gestión académica “SGA”.

4.4.2 Ventajas y desventajas de la instalación del software

Ventajas

- La herramienta OWASP es una aplicación fácil de instalar, en los sistemas operativos Windows, Linux y MacOS.
- En el Marketplace de OWASP se pueden encontrar otros paquetes adicionales para un escaneo más predictivo de aplicaciones web.
- En el sistema operativo Linux puedes crear instancias de instalación, para generar un instalador multiplataforma, para otras distribuciones de Linux.
- En la versión de MacOs no se necesita de la instalación de Java, porque viene incluida por defecto en el sistema operativo.

Desventajas

- Para instalar la herramienta OWASP, como requisito principal se debe tener instalada, las últimas versiones de Java para tener una mejor experiencia de usuario.

- En el sistema Operativo Ubuntu, por lo general presenta problemas de dependencias de paquetes incumplidos”APT - GET” en la mayoría de instalaciones de aplicaciones a terceros, por lo cual se debe aplicar comandos vía terminal para restablecer las dependencia de paquetes y poder instalar OWASP.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

En este trabajo de titulación, luego de identificar el funcionamiento de la herramienta OWASP y de las vulnerabilidades existentes en la plataforma “SGA”, se desglosan las siguientes conclusiones:

1. OWASP demuestra ser una herramienta inteligente a la hora de profundizar en la búsqueda o escaneo de fallos informáticos o brechas de seguridad, permitiendo conocer cada una de estas anomalías informáticas y a su vez mostrarnos el nivel de riesgo de vulnerabilidad, al que está expuesto el sistema de gestión académico “SGA”. Este análisis permite tomar acciones y medidas preventivas para mitigar estas vulnerabilidades identificadas.
2. Conocer las la estructura compuesta del sistema de gestión académica “SGA”, fue de suma importancia porque identificamos las debilidades que son muy frecuentes encontrarlas en muchas de las aplicaciones web que tenemos en el internet, estas debilidades están también presentes en el sistema de gestión académica “SGA”, la cuales son la principal causa de grandes factores de riesgos y explotación de datos, los cuales pueden provocar pérdidas financieras y fuga de información para el Instituto Tecnológico Bolivariano.
3. El sistema de gestión académica “SGA” de acuerdo al análisis efectuado trabaja con software de desarrollo que requieren actualización de sus versiones, como son en el caso concreto Python 2.7, PostgreSql 9.4. Actualmente el departamento de sistemas del ITB está trabajando en la migración de datos con respecto a estos dos software en mención.
4. En la entrevista efectuada al personal de desarrollo de la plataforma de gestión académica “SGA”, se evidencio una falta de conocimiento en el uso de herramientas, que permitan identificar fallos y errores en el proceso del desarrollo de una aplicación web.

En este caso se expuso el tema del grado de conocimiento que tenían con respecto a la herramienta OWASP.

5. Con respecto a las vulnerabilidades detectadas en el sistema de gestión académica “SGA” se detectó y a su vez se identificaron dos riesgos importantes, uno de ellos es el fallo por inyección de código SQL y el encabezado de X-Frame, ambos riesgos en conjunto permiten al atacante o hacker utilizar métodos para la validación de contraseñas y enmascarar o embeber, un código normal escrito en lenguaje JavaScript por otro código malicioso escrito en el mismo lenguaje de programación, en una determinada página web, con el fin de validarse como usuario administrador del sistema, y efectuar un escalamiento de privilegios para conseguir un acceso total al servidor. Es importante tener en cuenta esta vulnerabilidad para evitar problemas futuros.

5.2 RECOMENDACIONES

Las siguientes recomendaciones que se detallan a continuación serán objeto de análisis, para considerar nuevas metodologías seguras en el desarrollo e implementación de la aplicación web “SGA”, y mejorar en el paradigma de la seguridad informática, que en este estudio no fueron consideradas en un mayor contexto.

1. Para tener un enfoque más dinámico y versátil en la búsqueda de vulnerabilidades o fallos graves, en el sistema de gestión académico “SGA”, se debe implementar políticas de desarrollo seguro utilizando las normativas OWASP.
2. Es muy importante tener un conocimiento amplio en los nuevos estándares de prevención contra riesgos informáticos, ya que todos los días se presentan nuevas brechas de seguridad sobre las plataformas con la que interactuamos a diario.
3. Se debería implementar un esquema o banco de recolección de datos, que registre todas las anomalías que el sistema reporta, con respecto a fallos de acuerdo al grado de vulnerabilidad que presente el sistema de gestión académica “SGA”, el mismo que ayude a conocer como mitigar un determinado fallo a futuro.
4. Como recomendación final toda la información suministrada en este trabajo de titulación, debe ser considerado como un precedente a futuras investigaciones, por otros académicos en el área de seguridad de la información y sobretodo, ser objeto de un amplio análisis para evitar afectaciones en la integridad de la información.

BIBLIOGRAFÍA

Andrew S. Tanenbaum, D. J. (2012). *Redes de computadoras* (Quinta ed.). Mexico: PEARSON EDUCACIÓN.

Anonimo. (1996). *Postgresql.org*. Obtenido de Postgresql.org: https://www.postgresql.org/docs/current/history.html

Anonimo. (1996). *postgresql.org/*. Obtenido de postgresql.org/: https://www.postgresql.org/docs/9.5/mvcc-intro.html

Anonimo. (1996). *todopostgresql.com*. Obtenido de todopostgresql.com: https://todopostgresql.com/ventajas-y-desventajas-de-postgresql/

Anonimo. (2005-2020). *docs.djangoproject*. Obtenido de docs.djangoproject: https://docs.djangoproject.com/en/3.0/misc/design-philosophies/

Anonimo. (01 de 08 de 2017). *https://www.telecomunicaciones.gob.ec/*. Obtenido de https://www.telecomunicaciones.gob.ec/: https://www.telecomunicaciones.gob.ec/ecuador-ocupa-sexto-lugar-en-la-region-segun-indice-de-ciberseguridad/

Anonimo. (17 de 09 de 2019). *EL UNIVERSO*. Recuperado el 17 de 09 de 2019, de EL UNIVERSO: https://www.eluniverso.com/noticias/2019/09/16/nota/7521358/masiva-filtracion-online-informacion-casi-cada-ciudadano

Anonimo. (11 de 11 de 2019). *precisesecurity*. Obtenido de precisesecurity: https://www.precisesecurity.com/antivirus/windows

Anonimo. (s.f.). *http://viamatica.com/nosotros/*. Obtenido de http://viamatica.com/nosotros/: http://viamatica.com/nosotros/

Anonimo. (s.f.). *http://www.eclipsoft.com/*. Obtenido de http://www.eclipsoft.com/: http://www.eclipsoft.com/

Anonimo. (s.f.). *https://agrosoftcomec.wordpress.com/nosotros/*. Obtenido de https://agrosoftcomec.wordpress.com/nosotros/: https://agrosoftcomec.wordpress.com/nosotros/

Arias, F. (2012). *El Proyecto de Investigacion - Introduccion a la Metodologia Cientifica.* Caracas: Episteme C.A.

Ariganello Ernesto. (2014). Redes Cisco: Guia de estudio para la certificacion CCNA SECURITY. En Ariganello Ernesto, *Redes*

Cisco: Guia de estudio para la certificacion CCNA SECURITY. Madrid: RA-MA S.A.

Ary, D., Jacobs, L. C., & Razavieh, A. (1989). *Introducción a la investigación pedagogica.* Mexico D.F.: México [D.T México] : McGraw-Hill Interamericana.

Baltrusaitis, J. (19 de 12 de 2019). *https://www.precisesecurity.com/*. Obtenido de https://www.precisesecurity.com/: https://www.precisesecurity.com/articles/ms-office-represents-73-of-the-most-commonly-exploited-applications-worldwide/

Berners-Lee, T. (mayo de 1990). *w3org*. Recuperado el 20 de 09 de 2019, de w3org: http://www.w3.org/History/1989/proposal.html

Carballar A. Jose. (2014). WI-FI Instalacion, Seguridad y Aplicaciones. En Carballar A. Jose, *WI-FI Instalacion, Seguridad y Aplicaciones* (pág. 212). Madrid: RA-MA S.A.

Carlos Muñoz Rocha. (2015). *Metodologia de la Investigacion.* Mexico: Editorial Progreso S.A de C.V.

Database, N. V. (06 de 05 de 2020). *https://nvd.nist.gov/vuln/detail/CVE-2020-11884#vulnCurrentDescriptionTitle*. Recuperado el 18 de 5 de 2020, de https://nvd.nist.gov/vuln/detail/CVE-2020-11884#vulnCurrentDescriptionTitle

Eoin, K., Muller, A., & Meucci, M. (Noviembre de 2014). *OWASP Testing Guide*, V4. Recuperado el 19 de 09 de 2019, de OWASP Testing Guide: https://www.owasp.org/images/1/19/OTGv4.pdf

Ernesto Rodriguez Moguel. (2005). *Metodologia de la Investigacion .* Mexico: Universidad Juárez Autónoma de Tabasco.

Gomez, A. (2012). *Enciclopedia de la Segridad Informatica.* (Vol. Segunda Edicion). Mexico: Alfaomega.

Gómez-Peresmitré, G., & Martínez, L. R. (s.f.). *http://blogs.fad.unam.mx/*. Obtenido de http://blogs.fad.unam.mx/: http://blogs.fad.unam.mx/asignatura/carlos_salgado/wp-content/uploads/2012/10/Metodolog%C3%ADa-de-la-Invetigaci%C3%B3n-en-ciencias-sociales.pdf

Gonzalez Pablo, A. A. (2016). *Hacking Web Technologies.* Mostoles, Madrid, España: 0XWORD.

Guerra, M. (2016). *Interconexion de Redes Privadas y Publicas.* Madrid: RA-MA.

Ilic, J. (23 de 12 de 2019). *https://www.precisesecurity.com/*. Obtenido de https://www.precisesecurity.com/: https://www.precisesecurity.com/articles/cross-site-scripting-xss-makes-nearly-40-of-all-cyber-attacks-in-2019/

Kendall, K. E. (2005). *Analisis y Diseño de Sistemas.* Mexico D.F.: Pearson Educación de México, S.A. de C.V.

Lopez, A. (2010). *Seguridad Informatica.* Madrid, España: Editex.

López, P. L. (2004). *Scielo*. Obtenido de http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1815-02762004000100012

Mathew, M. (26 de 12 de 2019). *precisesecurity*. Obtenido de precisesecurity: https://www.precisesecurity.com/about-us/

Meak, L. (26 de 02 de 2018). *https://newsroom.cisco.com/*. Obtenido de https://newsroom.cisco.com/: https://newsroom.cisco.com/feature-content?type=webcontent&articleId=1912213

OWASP. (2008). *https://www.owasp.org*. Obtenido de https://www.owasp.org: https://www.owasp.org/images/8/80/Gu%C3%ADa_de_pruebas_de_OWASP_ver_3.0.pdf

OWASP. (21 de 09 de 2019). *OWASP ORG*, 3.0. Recuperado el 21 de 09 de 2019, de OWASP ORG: https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project

Pablo Gonzalez, A. A. (2016). Hacking Web Tecnhologies. En P. Gonzalez, A. Aparicio, E. Rando, R. Martin, & C. Alonso, *Hacking Web Tecnhologies* (pág. 29). Mostoles (Madrid): 0xWORD Computing S.L.

Rodriguez, L. d. (2011). *Metodologia de la Investigacion en Ciencias Sociales.* Mexico D.F.: Universidad Juarez Autonoma de Tabasco .

Rodriguez, M. L. (19 de 8 de 2013). Obtenido de https://guiadetesis.wordpress.com/2013/08/19/acerca-de-la-investigacion-bibliografica-y-documental/

Sabino, C. (1992). Obtenido de http://paginas.ufm.edu/sabino/ingles/book/proceso_investigacion.pdf

Samperi, H. (2014). *Metodología de la Investigación.* Mexico DF: McGraw-Hill.

Sampieri, R. H. (2014). *Metodologia de la Investigacion.* Mexico D.F.: McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V.

Santos Gonzalez Manuel. (2014). Sistemas Telemáticos. En S. G. Manuel, *Sistemas Telemáticos.* Madrid: RA-MA S.A.

Stalling, W. (2004). *Fundamentos de Seguridad en Redes Aplicacion y Estandares* (Vol. Segunda Edicion). Madrid: PEARSON EDUCACION S.A.

Urbina Baca Gabriel. (2016). *Introduccion a la Seguridad Informatica.* (J. E. Callejas, Ed.) Mexico D.F.: Grupo Editorial Patria.

Wikipedians. (s.f.). *https://books.google.com.ec/*. Obtenido de https://books.google.com.ec/: https://books.google.com.ec/books?id=u5pZX6xt2PoC&printsec=frontcover#v=onepage&q&f=false