SEP TecNM

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE ACAPULCO

TEMA:

ANÁLISIS Y DISEÑO DE UNA SOLUCIÓN UTILIZANDO TÉCNICAS DE TEST DE PENETRACIÓN PARA LA PREVENCIÓN DE RIESGOS DE VULNERABILIDADES DEL SISTEMA INTEGRAL DE INFORMACIÓN DEL ITA.

OPCIÓN I:
TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:
MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA:
ING. LEÓN GALEANA ABEL ISAAC

DIRECTOR DE TESIS:
DR. EDUARDO DE LA CRUZ GAMEZ

CO-DIRECTOR DE TESIS:
MTI. JORGE CARRANZA GÓMEZ

Acapulco, Gro, 4 de Diciembre, 2018

Descargo de Responsabilidad

El que suscribe declara que el presente documento de tesis titulado: “Análisis y diseño de una solución utilizando técnicas de test de penetración para la prevención de riesgos de vulnerabilidades del Sistema Integral de Información del **ITA**” es un trabajo propio y original, el cual no ha sido utilizado anterior mente en institución alguna para propósitos de evaluación, publicación y obtención de algún grado académico.

Además, se han reconocido todas las fuentes de información utilizadas, las cuales han sido citadas en la sección de referencias bibliográficas de este trabajo.

Abel Isaac Leon Galeana	4 de Diciembre del 2018
Nombre	Fecha y firma

Agradecimientos

A mis padres:

Abel León Aguilera y Elvia Galeana Benites por darme lo más importante en esta vida que es una educación y profesión, por su apoyo en los momentos más difíciles y no dejar que me rindiera, por enseñarme que lo más importante en la vida es superarse en pos de lograr las superaciones personales, forjando los cimientos de mi carácter y determinación para lograr todo lo que me proponga.

A mis instituciones:

El Instituto Tecnológico de Acapulco (**ITA**) siendo mi institución mater, la que me vio crecer brindándome los títulos profesionales que ahora poseo para culminar mi desarrollo como profesionista.

Al Consejo Nacional de Ciencia y Tecnología (**CONACYT**) por su apoyo económico durante estos dos años para el desarrollo de mi proyecto de tesis, dándome la oportunidad de escalar un peldaño más en mis objetivos de vida y profesional.

Al Departamento de Posgrado e Investigación (DEPI) del **ITA** por darme la oportunidad de superarme y de ser parte de su proyecto de maestría, disfrutando cada fase que conlleva el plan educativo en sus dos años de principio hasta su culminación.

A mis asesores:

MC. Francisco Javier Gutiérrez Mata por sus consejos y animo durante el tiempo de desarrollo de mi tesis, por nunca perder la fe en mí, así como toda la ayuda brindada más allá de las cuestiones académicas, por su amistad y guía que hicieron posible la culminación del proyecto.

Dr. Eduardo de la Cruz Gámez por su guía, ayuda, paciencia y asesoramiento durante el transcurso de la tesis, de igual manera el más cordial agradecimiento por darme la oportunidad de seguir con mi desarrollo profesional.

MTI. Jorge Carranza Gómez, por sus correcciones y asesoramiento permitiendo concluir de manera satisfactoria el presente escrito de fin de grado y por consiguiente lograr ascender más en mi desarrollo educativo y profesional

Resumen

El presente trabajo de tesis contiene los conocimientos fundamentales para la implementación de una auditoria informática, comúnmente conocida como test de intrusiones o *Pentesting;* con un enfoque relacionado a tecnologías Web; tomando como banco de pruebas una de las principales aplicaciones para la difusión y registro de datos del Instituto Tecnológico de Acapulco (**ITA**), nombrada Sistema Integral de Información (**SII**).

En el **SII** se manejan datos importantes para la organización, debido a esto se plantea la necesidad de auditar los principales niveles de seguridad y determinar los horizontes que se manejan relacionados al tema, del mismo modo tomar acciones en la corrección de las *vulnerabilidades* encontradas; siguiendo esta idea se llegan a acuerdos de delimitación y levantamientos de necesidades del departamento encargado en la gestión del **SII** para las principales flaquezas de toda aplicación de entorno Web, optando por la Metodología desarrollada por **OWASP**, debido que convenientemente se enfoca a la seguridad y desarrollo eficaz de este tipo de aplicación, esperando que con el apoyo del Estándar de Verificación en Aplicaciones en su versión 3.0.1 perteneciente a **OWASP** se pueda comprobar la veracidad de las pruebas

Las pruebas se centran en el cumplimiento de las necesidades del "Centro de cómputo (**CECOMP**)", departamento encargado de administrar el **SII**, y el análisis de las principales *vulnerabilidades* en aplicaciones de entorno Web, tomando como base los estudios realizados por Open Web Aplication Security Project (**OWASP**), en sus publicaciones periódicas con un periodo de diez años de seguimiento en sus estudios.

Abstract

This thesis work contains the fundamental knowledge for the implementation of a computer audit, commonly known as intrusion test or Pentesting; with a focus related to Web technologies; taking as a test bank one of the main applications for the dissemination and registration of data from Instituto Tecnológico de Acapulco (**ITA)**, named Sistema Integral de Informacion (**SII**).

The **SII** handles important data for the organization, due to this the need to audit the main levels of security and determine the horizons that are handled related to the subject, in the same way take actions in the correction of the vulnerabilities found; Following this idea, agreements are reached to delimit and raise the needs of the department in charge of the **SII** management for the main weaknesses of any Web application, opting for the Methodology developed by OWASP, due to the fact that it is conveniently focused on security and development. effective of this type of App., hoping that with the support of the Verification Standard in Applications in version 3.0.1 pertaining to OWASP to verify the veracity of the tests

The tests focus on meeting the needs of the "Computer Center (**CECOMP**)", the department in charge of administering the SII, and the analysis of the main vulnerabilities in the Web Environment **App,** based on the studies carried out by Open Web Aplication Security Project (**OWASP**), in its periodical publications with a period of ten years of follow-up in its studies.

Introducción

Actualmente en el siglo XXl toda organización en su infraestructura cuenta con algún tipo de equipo y sistema de cómputo, los cuales a su vez están enlazadas a una red global conocida como Internet; debido a la necesidad de establecer enlaces de difusión de la información; recaído a esto se crean bases de datos (**BD**), mismas que contienen la información que se requiera almacenar, para posteriormente usarla. Por esto el uso de **BD** enlazadas a una aplicación de entorno Web, hoy en día son consideradas por expertos como un riesgo latente, en consecuencia, de los diferentes tipos de *vulnerabilidades* que existen relacionados a estos sistemas.

Teniendo la preocupante de que usuarios ajenos a los sistemas de cualquier organización y con conocimientos en términos informáticos, pudiesen acceder a las **TIC**'s, con la finalidad de realizar actividades ilícitas, tales como, la suplantación de identidad, la sustracción de información y el daño a la integridad de la organización; con la finalidad de establecer medios de prevención a esta problemática las organizaciones empiezan a tomar conciencia en relación a la seguridad informática. Es aquí donde se centra la temática principal de este trabajo de tesis, presentando un lineamiento en el desarrollo de *Pentesting* a una aplicación de entorno Web, perteneciente a una organización educativa, siguiendo la metodología de **OWASP** para el análisis de Riesgos.

Índice General

Índice de Figuras

Índice de Tablas

Índice de ecuaciones

Capítulo 1.- Especificaciones del Proyecto

El proyecto que da origen a este trabajo de tesis, se encuentra centrado en la necesidad de aumentar la seguridad informática para mejorar el desempeño en las operaciones del Centro de cómputo (**CECOMP**) del Instituto Tecnológico de Acapulco (**ITA**), este departamento es el responsable de gestionar el funcionamiento de la base de datos (**BD)** y servidor Web responsable del funcionamiento del Sistema Integral de Información (**SII**), herramienta de entorno Web, con el propósito de difundir, registrar y dar de alta los datos de los alumnos inscritos en el plantel educativo.

Por esta razón se plantea que el **SII** debe de estar a la par del constante crecimiento tecnológico en términos de seguridad, prevención y recuperación de la información, en caso de posibles intrusiones al sistema. En el desarrollo del presente capítulo se presentan los aspectos generales a tomar en consideración para el desarrollo del proyecto, dando a conocer en cada sección los puntos a destacar desde el surgimiento del proyecto, hasta su elaboración

1.1.- Planteamiento del Problema

En la actualidad los principales problemas en el país van relacionados a la seguridad de la información de acuerdo con la **UIT**[1] , México es uno de los países con más retrasos en el tema, a partir de los últimos años esta problemática ha tomado relevancia debido a que con el paso del

[1] Es el organismo especializado de las Naciones Unidas para las Tecnologías de la Información y la Comunicación (UIT, 2018).

tiempo los avances en dicha primicia y las nuevas tecnologías facilitan los ataques, en términos de robo y suplantación de datos.

En el año 2015 según los estudios elaborados por el **INCIBE**[2], teniendo como tema principal las principales *Vulnerabilidades* en aplicaciones Web, dividiéndolos por nivel de riesgo y explotación; demostrando que las *Vulnerabilidades* de **XSS** (por sus siglas en ingles Cross Site Scripting), es una de las más explotadas, de la misma manera, los errores generados por una mala codificación de programadores con poca experiencia en seguridad, genera que el código de las aplicaciones contenga muchas fallas de diseño, usándolas para el alojamiento de *Malware*.

Otro de los métodos que se considera necesario investigar son las inyecciones de código **SQL**, si bien solo figura con un 5% del total de las *Vulnerabilidades* más comunes, este tipo de ataque tiene como principal objetivo a los servicios y **BD** alojados en un servidor, que se encuentra enlazado a un servidor Web, al ser combinados con inyección de código en errores de programación y configuración tanto de la **BD** como del servidor, se crea un daño considerable a los datos y a los recursos materiales.

Dentro de **CECOMP**, departamento perteneciente al **ITA** se presenta la necesidad de auditar una de las principales aplicaciones de entorno Web, caso en el que su función corresponde a la difusión y resguardo de la información; esta **App.** recibe el nombre de **SII**, y su auditoria tiene como finalidad obtener conocimiento sobre las fortalezas y debilidades de la **App.**; partiendo de

[2] Es una sociedad dependiente de la Secretaría de Estado para la Sociedad de la Información y la Agenda Digital (SESIAD) y consolidada como entidad de referencia para el desarrollo de la ciberseguridad y de la confianza digital de ciudadanos, red académica y de investigación, profesionales, empresas y especialmente para sectores estratégicos(INCIBE, 2018)

problemáticas pasadas con el **SII** (las cuales se resolvieron en tiempo y forma), se considera que no es conveniente la espera de nuevas incidencias; por esto el utilizar los medios oportunos para la medición de la seguridad del **SII**, de la misma forma que la documentación que plasme los riesgos encontrados, cuantificando su nivel de incidencia o gravedad, permitiendo que seaposible generar métodos de prevención y corrección.

En el caso de no hacer una auditoria de seguridad y cuantificar el nivel de riesgo, es de esperarse que en un futuro un usuario no autorizado pueda aprovechar alguna *Vulnerabilidad* existente, para manipular la información a su conveniencia o en un caso extremo inhabilitar la aplicación. El panorama contemplado para estos casos no muestra una visión favorable para la organización debido que esto provocaría un desprestigio, así como repercusiones económicas.

1.1.1.- Formulación del problema.

Para establecer los cimientos del tema a tratar de resolver dentro de la línea de investigación, es necesario plantearse una serie de preguntas, que, ayudarán a crear un plan de acción para el desarrollo de los ataques, dichas interrogantes son las siguientes:

- ¿Qué tan seguro es el sistema?
- ¿Es posible romper su seguridad?
- ¿Qué *Vulnerabilidades* tiene el sistema?
- ¿Cómo es posible prevenirlo?

Como se ha mencionado con anterioridad, el tema se centra en la auditoria y el procesos de explotación de *Vulnerabilidades* del **SII** que maneja una institución educativa en específico (**ITA**); es pertinente decir que los resultados y la investigación en general puede ser un marco de referencia para futuras investigaciones, a pesar de que se cubran problemas específicos para la organización en el banco de datos, toda prueba de seguridad basada en *Hacking ético* o *Pentesting* parte de un mismo lugar.

1.2.- Objetivos del Proyecto

En el siguiente apartado se presentan los objetivos a cumplir durante el desarrollo del proyecto, los cuales durante su desempeño se pretende brindar un resultado de evaluación a una aplicación de entorno Web perteneciente al **ITA,** por medio de pruebas de penetración y explotación de *Vulnerabilidades* (*Pentesting),* apoyadas en un estándar de tipo *Open Source* (código libre o abierto) propuesto por Open Web Aplication Security Proyect (**OWASP)**, estandarizando el proceso de evaluación del **SII** para ofrecer un nivel de certidumbre a las pruebas.

1.2.1.- Objetivo general.

Analizar mediante pruebas de penetración (*Pentesting*) y búsqueda de *Vulnerabilidades*, al Sistema Integral de Información (**SII**), con la finalidad de determinar si cumple con los niveles mínimos en el ámbito de resguardo de la información. contenidas en la base de datos (**BD)** del **SII** del Instituto Tecnológico de Acapulco (**ITA**), sustentadas por medio de la metodología contenida

en el Testing Guide (guía de pruebas) de **OWASP**[3] versión 4.0 y el Estándar de Verificación de Seguridad en Aplicaciones (**EVSA**) en su versión 3.0.1 de **OWASP**.

1.2.2.- Objetivos específicos.

- Auditar el Sistema Integral de Información (**SII)**.
- Analizar diversos planes de contingencia para la protección de la información del **SII**.
- Promover la aplicación de auditorías enfocadas a la seguridad para evaluar las prácticas de seguridad dentro del departamento de Centro de cómputo (**CECOMP**).

1.3.- Universo y Muestra

El campo de trabajo o universo objeto para la realización de la presente tesis, está formado por la evaluación del **SII**, el cual es una **App.** de entorno Web usada por el **ITA**, sin embargo, debido a la naturaleza de las pruebas se opta por usar un duplicado del **SII**, dicha copia recibe el nombre de **SIItest**, encontrándose alojada dentro de un servidor Web y una base de datos (**BD**), enlazada a un servidor de **BD**, a la cual se accederá y evaluará a través de un navegador y acceso a Internet.

[3] Organización sin fines de lucro la cual se encarga de combatir las causas que propician que el Software contenga vulnerabilidades haciéndolo inseguro para el uso de los usuarios, esto hace que los proyectos apoyados por esta organización sean *Open Source* se encuentra conformado por empresas, instituciones educativas y particulares en todo el mundo, surgiendo desde el año 2001 y concretando su formación a partir del 2004 hasta la fecha (OWASP, 2008)

1.4.- Hipótesis

Los resultados arrojados por la auditoria al **SII** (**SIItest** en el banco de pruebas), apoyadas con la metodología de **OWASP** para la detección de *Vulnerabilidades*, permitirá generar una guía de referencia con el fin de detectar fallas y mejorar el blindaje del **SII**, con la finalidad de disminuir los principales riesgos relacionados al tema, mejorando los niveles de confiabilidad en las aplicaciones Web de cualquier organización.

1.5.- Justificación

El proyecto de investigación tiene la finalidad de encontrar, documentar y proponer alternativas para la solucion de las *Vulnerabilidades* encontradas en una **App.** de entorno Web, por medio de técnicas de *Pentesting* mejor conocidas como auditoria informática, con la intención de contribuir en los estudios relacionados al tema, usando una estandarización *Open Source* propuesta por **OWASP** y su Testing Guide (guía de pruebas) en la versión 4.0.

Las técnicas se demuestran por medio de la utilización de herramientas contenidas en Kali Linux 2.0, sistema operativo (**SO)**. Especialmente diseñado para el *Pentesting*; de este modo la investigación se enfoca en realizar un reporte de fallas encontradas en cualquier **App.** Web y desplegando un conjunto de recomendaciones que pueda garantizar la seguridad de estas.

Tanto las herramientas contenidas en la distribución y la metodología a seguir dentro del **EVSA 3.0.1** son de tipo *Open Source*, ambas son muy utilizadas por expertos en el tema para la

elaboración de *Pentesting* y creación de **App.** de entorno Web normalizadas. La naturaleza que los conforma, permite con mayor facilidad, poder obtener resultados debido a la facilidad para conseguir actualizaciones en la base de datos de instrumentos dedicados a seguridad.

Las medidas de prevención de riesgos para las **App.** Web, es la base para los parámetros, la integridad de la información por medio de administración y control de *Vulnerabilidades*. Los estudios apropiados de las amenazas a la seguridad proporcionan las ventajas de implantación de procedimientos y controles con el fin de proteger y salvaguardar los datos; para de esta manera poder demostrar la eficiencia en el desarrollo de estas prácticas.

A pesar de ser un caso de estudio con parámetros específicos a evaluar, se considera que, toda organización interesada en instituir pruebas de seguridad por medio de *Pentesting,* para el conocimiento de *Vulnerabilidades* dentro de sus sistemas, puede tomar este trabajo como marco referencial para proyectos propios.

1.6.- Descripción del Trabajo

El presente trabajo de tesis, plantea la evaluación al **SII**, aplicación de entorno Web perteneciente al **ITA**, una de las principales universidades en el Municipio de Acapulco y del estado de Guerrero, por medio de pruebas de *Vulnerabilidades* o *Pentesting* usando un estándar para el desarrollo de **App.** Web propuesto por **OWASP**.

El proyecto sigue un método científico para el uso de un marco referencial que pueda guiar la investigación, el uso combinado de la teoría y conocimientos empíricos ayudará en el manejo de la investigación apoyada por herramientas y tecnologías que involucre el proyecto.

Por medio del razonamiento inductivo se llegará a una propuesta de solución con base a las observaciones de los efectos propiciados por los ataques siguiendo una metodología de acción, para mantener un nivel óptimo de la seguridad para cualquier App. Web.

1.6.1.- Contribuciones.

Las contribuciones reflejadas al realizar el proyecto que compete a este trabajo de tesis son divididas en tres tipos, que se considera de mayor impacto para la comunidad interesada en el tema y para la ejecución de futuros proyectos relacionados al tema, dichas contribuciones son:

1.6.1.1.- Social.

Toda institución debe contar con un manejo integral y estandarizado de la seguridad informática con el fin de poder proteger los intereses de esta, así como mejorar los protocolos en ámbito de protección de los datos, esto con el objetivo de tener un respaldo y/o alternativa de enfoque en caso de un mal manejo de operación, siniestro o intrusión.

1.6.1.2.- Económico.

Al realizar la búsqueda de *Vulnerabilidades*, se plantea los posibles riesgos de intrusión, de este modo se puede prevenir posibles daños o riesgos de pérdidas tanto en bienes físicos, como es el *Hardware* e intangibles como es el caso del *Software*. De este modo la institución puede ahorrar capital de posibles pérdidas o según sea el caso minimizarlas, logrando detectar los sistemas que han permitido la intrusión y de este modo reemplazarlos, así como llevar un registro de los bienes que se han dañado o quedan obsoletos para una sustitución de material con el fin de reducir costos.

1.6.1.3.- Tecnológico.

Al realizar las actividades de la búsqueda de *Vulnerabilidades* se deben verificar cuales son los rasgos tecnológicos con los que se cuentan para la prevención de pérdida de información o robo de datos, cuáles son los que ya están obsoletos, en otras palabras, cuales son los que cumplen su función en la protección de la integridad de los datos y de esta forma buscar que la institución este a la vanguardia de las tecnologías de la información y comunicaciones (**TIC'S**).

1.7.- Organización de la Tesis

Tomando una línea de investigación que ha ido creciendo con el paso de los años, se presenta un capitulado para demostrar la importancia de establecer pruebas de *Pentesting* a las aplicaciones de entorno Web que se manejen en una organización, debido a que no es factible el

pensar que las **App.,** no contienen datos significativos para los usuarios maliciosos mejor conocidos como *Hacker*.

Debido a esto los capítulos contenidos en el presente trabajo de tesis recaban toda la información referente al caso de estudio anteriormente descrito en el **apartado 1.1.- Planteamiento del Problema**, si bien fue pensado que el escrito fuera leído de una forma lineal y secuencial como se muestra en la **Figura1.1.**, es considerado que personas con conocimientos previos al tema puedan hacer una lectura discontinua, omitiendo ciertos apartados, detal manera que sólo puedan regresar a ellos con la finalidad de refrescar conocimientos, como se muestra en la **Figura1.2.**

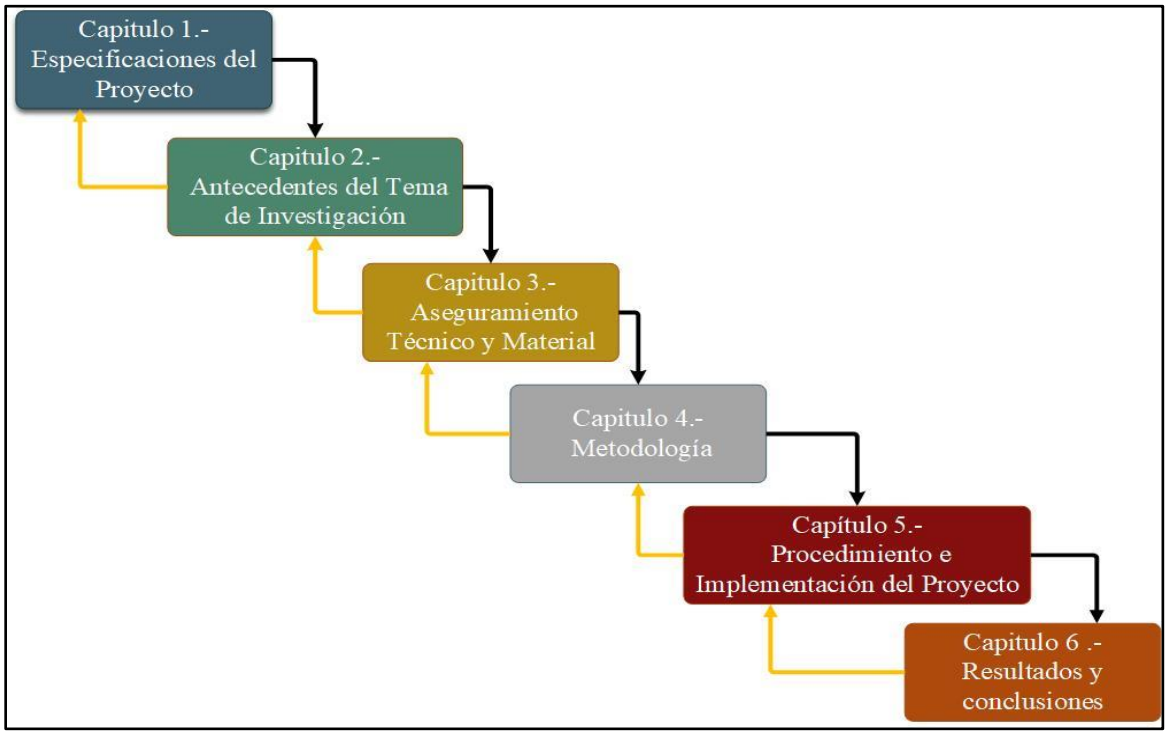


Figura 1.1.- Ejemplificación de una Lectura en Secuencia Lineal. (Autor).

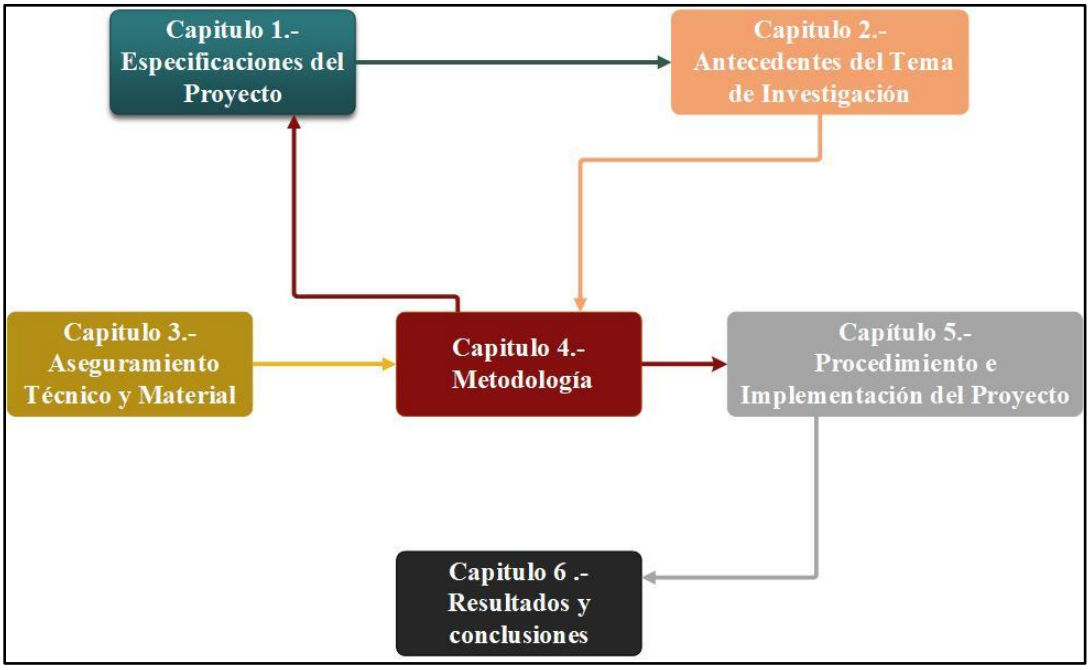


Figura 1.2.- Ejemplificación de una Lectura No Lineal. (Autor).

Capítulo1: se describe el ámbito general del proyecto, así como los aspectos que influenciaron la elección del tema que compete al campo de investigación las problemáticas a resolver, descripción general del capitulado y de los departamentos involucrados en el proyecto. Con la finalidad de que el lector obtenga un panorama general de los temas contenidos en el trabajo de investigación.

Capítulo 2: dentro de este apartado se plantea el panorama general del tema de investigación, los diferentes tipos de documentos de carácter científico relacionados al tema, los cuales servirán como discusión en la obtención de resultados, también se plantean las definiciones de temas con los que el lector debe estar familiarizado para la comprensión de la tesis; los apartados contenidos en el capítulo son denominados “Estado del arte y marco teórico”.

Capítulo 3: dentro de este capítulo se exponen los gastos de producción y adquisición de material para el proceso de investigación del proyecto, debido a la necesidad de usar equipo físico para duplicar los escenarios, con la finalidad de que las pruebas fueran los más fidedignas, se crea un gasto económico lo cual se desglosa a lo largo del capitulado, se piensa que el equipo usado y el presupuesto puede ser adaptado o mejorado en proyectos futuros según las necesidades a cubrir.

Capítulo 4: apartado que despliega el procedimiento utilizado para el desarrollo de la investigación, los métodos y estándares a seguir para obtener resultados fehacientes para la comunidad interesada, la descripción de los apartados que conforman al capítulo pretenden visualizar la planeación de una metodología imponiendo como método científico y normalizando el proceso de producción, estableciendo una jerarquía de actividades a realizar hasta la culminación del proyecto.

Capítulo 5: capitulo que contiene el proceso de elaboración del proyecto, los pasos, métodos y técnicas usadas para la implementación de las pruebas los cuales se estarán registrando para obtener los resultados que se plasmaran en el **Capítulo 6**.

Capítulo 6: dentro del capítulo se exponen los resultados, conclusiones y posibles trabajos a futuros derivados del proyecto, esto permite visualizar el comportamiento de las pruebas durante la explotación de las *Vulnerabilidades* y que trabajos pueden complementar el tema en un futuro.

Capítulo 2.- Antecedentes del Tema de Investigación

Dentro del presente capítulo se exponen parte de las investigaciones realizadas en el ramo de la seguridad informática y temas adyacentes, los cuales proporcionan una base para cimentar los conocimientos de la problemática que pertenece al tema de investigación.

El capítulo se encuentra dividido en dos partes, en la primera parte llamada "Estado del arte", se procederá a exponer los diferentes trabajos existentes en la comunidad científica, sus resultados, aportes y la relación con el problema a resolver; en la segunda sección nombrada "Marco teórico" se procede a enumerar, exponer y definir los principales temas y herramientas que se utilizaran para la realización del proyecto, teniendo como finalidad que el lector pueda adentrarse al tema de manera gradual, para posteriormente en capítulos avanzados como lo puede ser el **Capítulo 5.-** Procedimiento e Implementación del Proyecto, siempre pueda regresar en caso de no entender una referencia.

2.1.- Estado del Arte

En la actualidad las funciones burocráticas y la educación se apoyan en gran medida en las Tecnologías de la Información y Comunicación comúnmente abreviadas **TIC**, con esto es posible decir que en pocos años se podrá hacer una automatización completa de estas funciones, en materia del crecimiento tecnológico, son muchos los avances en ramas de investigación en un sentido informático, tantas que desgraciadamente con el aumento de estas tecnologías también lo ha hecho el mal manejo de las interfaces tecnológicas e informáticas como lo son las computadoras y las

tecnologías de redes, las malas prácticas que se le da, van desde robo de identidad, plagio de información personal, hasta robo y espionaje.

En base a esto se han creado líneas alternas de investigación, las cuales han tratado de erradicar de raíz la problemática, pero sin importar cuanto se cuide uno de no caer en estos problemas, los problemas siempre lo pueden encontrar a uno, y son lo que hoy se denominan con el nombre de *Hacker* los que tanto aterran a las organizaciones de cualquier rubro, el hecho de perder información es en términos de toda organización considerado como pérdida de dinero, sin embargo esto no quiere decir que la tecnología sea peligrosa, más bien, se considera de mayor riesgo a los usuario que la usan. Esto quiere decir que las personas que manejan la información no siempre están entrenadas en términos de resguardo de la información que manejan, en palabras de (Mitnick & Simon, 2007) "usted puede tener la mejor tecnología, *Firewall,* sistemas de detección, de ataques, dispositivos biométricos, etc. Lo único que se necesita es una llamada a un empleado desprevenido e ingresar sin más".

Esto dice que además del cuidado de los sistemas que se manejan a diario, también es pertinente tener cuidado de personas que dominan lo que hoy en día se le denomina ingeniería social, en palabras de (Borghello, 2009), está trata de manipular a las personas para que a base de engaños cometan una acción necesaria para vulnerar los sistemas de información, en el 2008 se suscitaron varios casos en que los usuarios cayeron en engaños abriendo archivos adjuntos que infectaron los sistemas que manejaban.

Relacionado a una de las actividades de la ingeniería social, se encuentra lo que se conoce como *Phishing* o mejor dicho suplantación de identidad, esta problemática básicamente se centra en la obtención de contraseñas de todo tipo (Florêncio, Herley, & Coskun, 2007) en su estudio elaborado para Microsoft en el cual se centraron en los hábitos de las personas para la creación de contraseñas, las cuales llegan a ser de poca robustez debido a que el usuario piensa que será más fácil recordarla, así como de la tendencia de los mismos por usar fechas o días importantes para su creación, esto sin mencionar que la gente tiende a olvidar las contraseñas, anotándolos inicialmente en lugares donde es fácil obtenerlas, este estudio muestra también que conforme más dígitos tiene la contraseña está se hace más difícil de descifrar o desencriptar siendo apoyado en estudios posteriores

(Harshavardhan, Vinay Reddy, Chintalapudi, & Viswanatham, 2018) en su artículo ejemplifican el uso del *criptoanálisis* a las contraseñas de usuarios usando técnicas de *Hash criptográficas* durante la creación de usuarios en el sistema, debido a la facilidad con la que los *Hacker* pueden piratear las contraseñas ayudados con el uso de *Hardware* especializado; en su caso de estudio, los ataques plantean el uso de texto plano para probar la eficiencia, contrastándola con el clásico ataque de fuerza bruta orientado en el uso de diccionarios.

Una de las medidas surgidas para la prevención de intrusiones ya sea por ingeniería social o el *Hacking,* es lo que hoy día conocemos como auditorías informáticas, este campo está siendo explotado en gran medida por muchas organizaciones; dentro del contexto de la seguridad informática, es necesario entender el control de riesgos, esta una actividad ayudada por las buenas practicas, en palabras de (Ramírez R. & Álvarez D., 2003) “la seguridad informática permite a la

organización buscar los medios para alcanzar los estándares internacionales en el uso adecuado de las tecnologías de la información con el objetivo de llegar a una certificación de calidad".

Las técnicas para la protección de datos a las organizaciones tiene un crecimiento exponencial, debido a los constantes riesgos que conlleva el tener que conectarse a la red de datos más grande del mundo que es la Internet, esto crea la necesidad de ramificar las ciencias en **TIC**'s en temas de estudio para la seguridad informática, algunos de estos estudios son los que cimientan los avances que la ciencia ha conseguido a lo largo de los años, es conveniente destacar que con el objetivo de que la información se considere veras, concisa y actual a las tecnologías que se manejan en estos tiempos, del mismo modo es necesario recabar información que ayude a las discusiones, la comparación y el seguimiento para el apoyo de la tesis, teniendo como finalidad implementar una auditoria de seguridad al Sistema Integral de Información (**SII**) del Instituto Tecnológico de Acapulco (**ITA**).

Muchos son los autores que señalan una serie de problemáticas que ha habido en los últimos años y los cuales han ido en aumento pretendiendo analizar las amenazas los ataques, el estudio de las consecuencias y como al estar en una sociedad que depende del correcto funcionamiento de las tecnologías de hoy en día.

(Gómez Vieites, 2012), En su estudio enfatiza que las principales motivaciones investigadas por el **FBI** son las motivaciones por dinero, compromiso y Ego; para que se realice un ataque informático además de las motivaciones ya mencionadas se recalca que esta persona debe contar con los conocimientos y herramientas adecuadas, así como la explotación de una

Vulnerabilidad al sistema que el atacante haya elegido o encontrado, los factores distribuidos en este artículo son representados de manera gráfica en la **Figura 2.1** en lo que el autor denomina como " Triángulo de la intrusión".

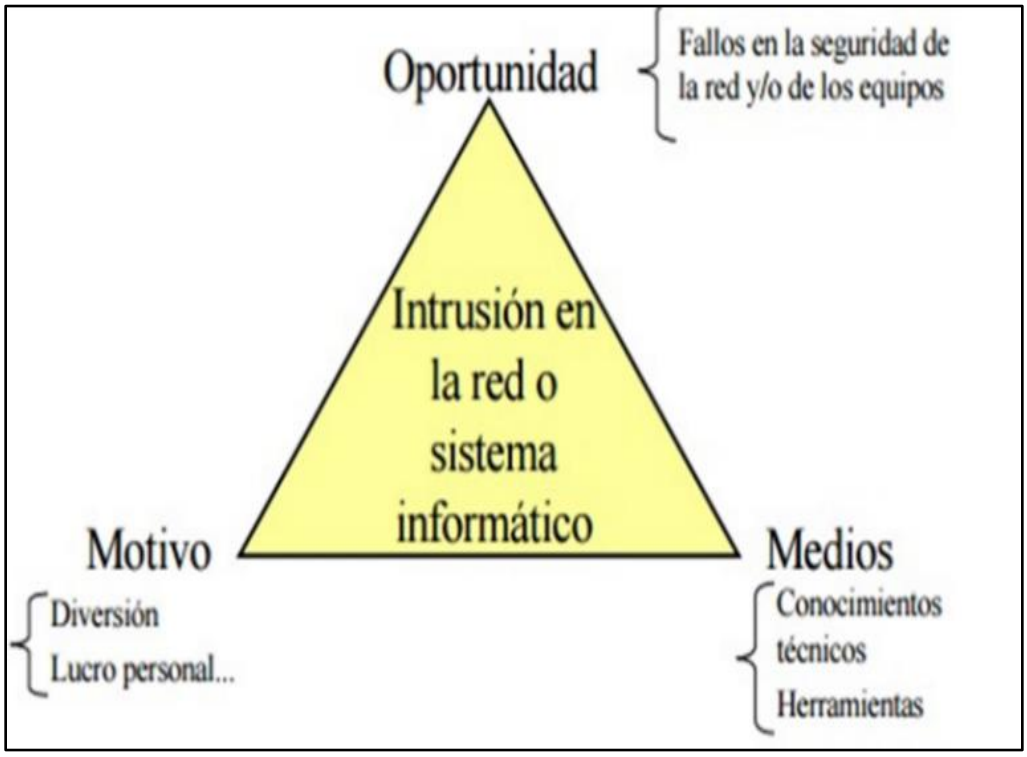


Figura 2.1.- Triángulo de la Intrusión.(Vieites, 2012, Pág. 5)

Dentro del artículo se enumeran una serie de recomendaciones las cuales al ser inspeccionadas se concluye que los conocimientos de estos pueden ser de interés al elaborar el proyecto, dichas recomendaciones son las siguientes.

- Constitución de un equipo de respuesta a incidentes.
- Definición de una guía de procedimientos.
- Detección de un incidente de seguridad.

- Análisis del incidente.
- Contención, erradicación y recuperación.
- Identificación del atacante y para la toma de acciones legales.
- Documentación del incidente de seguridad.
- Análisis y revisión "a posteriori" del incidente.

Dentro de la elaboración de auditorías informáticas existen diferentes tipos de riesgos los cuales pueden afectar la integridad de la información de la organización auditada, de la misma forma que los sistemas que esta maneje; para ello es recomendable que si no se tiene mucha experiencia en términos de seguridad, es recomendable ampliar los conocimientos mediante la elaboración de las pruebas de *Pentesting* en entornos virtuales, manejando las especificaciones del sistema lo más parecidas al de la organización, con el fin de obtener resultados lo más precisos que se puedan tener en un entorno real.

En su artículo titulado" Diseño de un Ambiente Simulado para Seguridad de la Información" (Fonseca Romero, 2015) muestra la arquitecturas ocupadas para la elaboración de pruebas de *Vulnerabilidades* a diferentes maquinas en un ambiente simulado por máquinas virtuales, si bien el autor especifica que las pruebas están hechas en máquinas virtuales, con sistema operativo XP, el cual ha dejado de obtener mantenimiento desde hace años y es considerado actualmente uno de los sistemas operativos (**SO**) más vulnerables; sin embargo el propósito es facilitar la alimentación de conocimiento a los usuarios que se estén adentrando al mundo de la seguridad informática en la **Figura 2.2** el autor propone una topología para la conexión de su simulador.

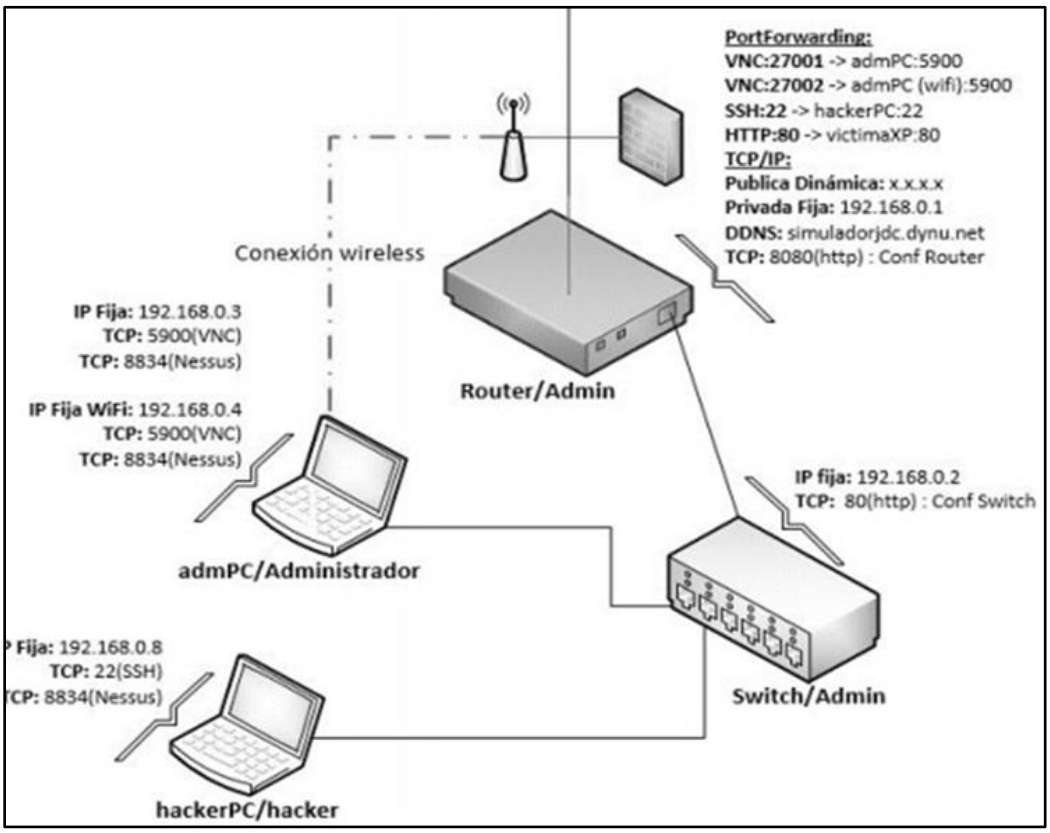


Figura 2.2.- Topología de Red Física del Simulador. (Fonseca Romero, 2015, pág 117)

Es recomendable que la obtención de experiencia sea gradual para ello el autor remarca que “hasta los pilotos y astronautas tienen que usar un ambiente simulado el cual les da la experiencia necesaria para reaccionar en el momento en que se presente la necesidad de estar en un ambiente real”, de la misma manera el auditor en un ambiente de simulación puede adquirir el manejo adecuado de las herramientas para la generación de pruebas de penetración sin los inconvenientes de tirar los servicios de la empresa o comprometer la información que esta maneje. Posteriormente en su estudio concerniente a su tesis de fin de grado de maestría (Fonseca Romero, 2017) describe los procedimientos para la implementación de un sistema con el principal propósito de capacitar y entrenar a usuarios interesados en los procesos de un test de intrusión, combinando

la virtualización y las tecnologías para las aplicaciones Web teniendo como principal motivo permitir la capacitación de los usuarios en términos de test de intrusión, en la **Figura 2.3** el autor ejemplifica por medio de un diagrama de casos de uso los elementos que conforman su sistema

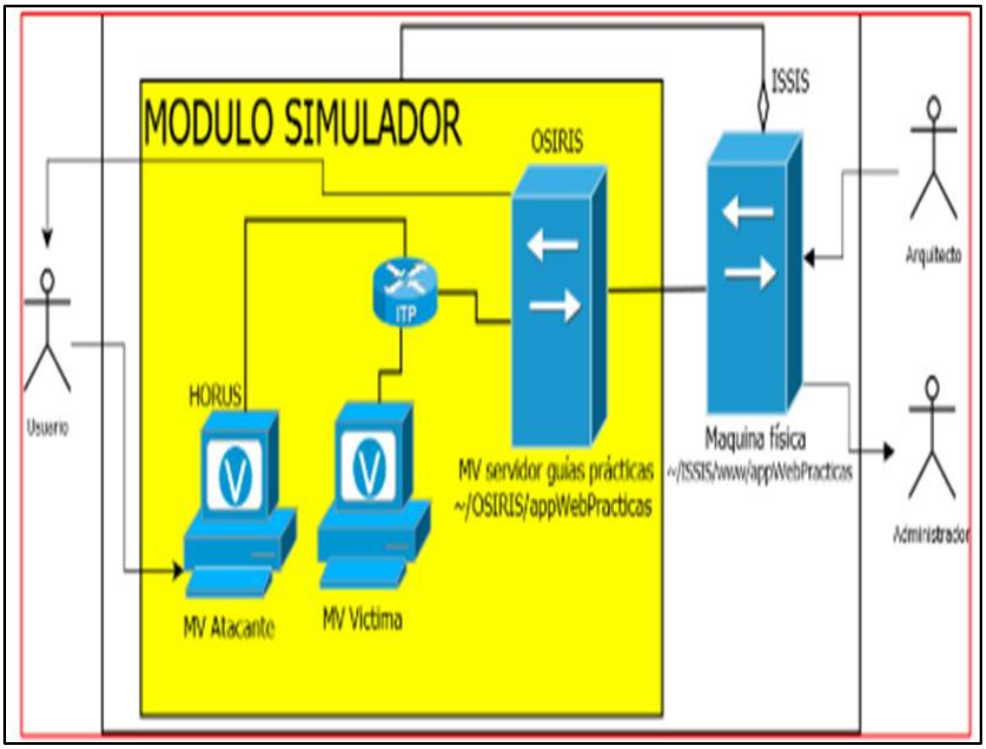


Figura 2.3.- Diagrama de Caso de Uso del Módulo Simulador. (Fonseca Romero, 2017, pág.22)

Los escritos que el autor plantea se considera de mucha importancia en el campo de estudio, debido a que el tema a tratar a lo largo del trabajo que compete a la presente tesis, es el planteamiento de un análisis de seguridad a una aplicación de entorno Web, para el centro de cómputo (**CECOMP**) del Instituto Tecnológico de Acapulco (**ITA**), este ha dejado como condicionante que la página, el cual es punto de consulta de varios usuarios, no debe de interrumpir sus servicios en ningún momento en que se puedan realizar las auditorias; este tema se abordara con más detalle en capítulos siguientes.

2.2.- Marco Teórico

Dentro de esta sección se desglosan los conocimientos necesarios y pertinentes dentro de la rama de la informática, a fin de expandir el panorama del lector que no esté familiarizado, incrementando su conocimiento del tema a resolver dentro del proyecto que da pie al presente escrito.

El desglose del marco teórico se sustentan las bases y la clasificación de los conocimientos que ayudan a la elaboración de proyectos relacionados al presente tema de investigación, el cual consiste en establecer pruebas de *Pentesting* o auditoria informática a una App. Web nombrada **SII**, esta **App**., es la encargada de gestionar y almacenar los datos personales de los alumnos, así como la difusión de sus estatus académicos; por ello es importante que la información que se maneja en esta aplicación siempre pueda ser irrebatible, para eso se determina que las pruebas de *Pentesting* son la mejor alternativa para detectar las *Vulnerabilidades* en una **App**. que ya se encuentran en funcionamiento y con el cual nunca se tuvo un análisis referente a este rubro en el **SDLC**.

2.2.1.- Vulnerabilidades en la Web.

En palabras de (Villalobos Murillo, 2012) “la protección de tecnologías Web de las entidades debe ser consideradas de alta prioridad, de la misma forma la protección a las **BD** enlazadas a las App. Web, que en su mayoría se encuentran montadas en gestores comerciales

como lo son MySQL, Access, PostgreSQL entre otras", esta ideología queda sustentada debido a que son amenazas con un tiempo prolongado entre los estudios elaborados por organizaciones especializadas en el tema como son ESET, Norton y CCN-CERT por mencionar algunas, sin embargó, a pesar de los esfuerzos por solucionar estas problemáticas, los estudios realizados por **OWASP** como se muestran en la **Tabla 2.1**, la evolución de las *Vulnerabilidades* mostrada en su informe de **Top ten** de las principales *Vulnerabilidades* a las **App.** Web ha estado en un cambio constante desde el 2007 hasta el 2017, en estos informes se enumeran las debilidades por orden de prioridad, tomando como estudio las organizaciones en un tiempo de por lo general tres años entre un informe y otro; como se observa en la **Tabla 2.1,** las *Vulnerabilidades* han ido cambiando y sustituyendo a las anteriores en el nivel de explotación.

Tabla 2.1.- Tabla Comparativa de las Principales Vulnerabilidades Publicadas Por OWASP 2007 – 2017. (Autor)

	2007	**2010**	**2013**	**2017**
A-1	Cross Site Scripting (**XSS**)	Inyección (**SQL**, LDAP, XPath)	Inyección (**SQL**, LDAP, XPath)	Inyección (**SQL**, LDAP, XPath)
A-2	Defectos de inyección	Cross-Site Scripting (**XSS**)	Autenticación rota y gestión de sesiones	Autenticación rota y gestión de sesiones
A-3	Ejecución de archivos maliciosos	Autenticación rota y gestión de sesiones	Cross-Site Scripting (**XSS**)	Cross-Site Scripting (**XSS**)

A-4	Referencia insegura de objeto directo	Referencias de objetos directos inseguros	Referencias de objetos directos inseguros	Control de acceso roto
A-5	Falsificación de solicitudes cruzadas (**CSRF**)	Cross-Site Request Forgery (**CSRF**)	Configuración incorrecta de seguridad	Configuración incorrecta de seguridad
A-6	Fuga de información y manejo incorrecto de errores	Configuración incorrecta de seguridad	Exposición de datos sensibles	Exposición de datos sensibles
A-7	Autenticación rota y gestión de sesiones	Almacenamiento criptográfico inseguro	Falta el control de acceso del nivel de función	Protección insuficiente de ataque
A-8	Almacenamiento criptográfico inseguro	Error al restringir el acceso a la URL	Falsificación de solicitudes entre sitios (**CSRF**)	Falsificación de solicitudes entre sitios (**CSRF**)
A-9	Comunicaciones inseguras	Insuficiente protección de la capa de transporte	Uso de componentes vulnerables conocidos	Uso de componentes con vulnerabilidades conocidas
A-10	Error al restringir el acceso a la URL	Redirecciones y reenvíos no validados	Redirecciones y reenvíos no validados	Aplicaciones desprotegidas

Otros sitios y organizaciones que apoyan los resultados divulgados por **OWASP** concluyen de manera similar un ejemplo de estas se mencionan a continuación.

El sitio *www.opensecurity.es* indica que los cinco principales tipos de *Vulnerabilidades* en aplicaciones Web son:

- Ejecución remota de código.
- SQL.
- *Vulnerabilidades* en formato de cadenas.
- XSS.
- Problemas atribuidos a los usuarios.

El sitio del Departamento de Seguridad en Cómputo de la **UNAM** en su página *http://www.seguridad.unam.mx/VulnerabilidadesDB* menciona en su lista de *Vulnerabilidades* más comunes a las siguientes:

- Cross Site Scripting (**XSS**).
- SQL Injection.
- Buffer Overflow

En la **Figura 2.4** propuesta por (Jiménez, 2016) se observa una representación de las principales *Vulnerabilidades* y los sistemas a los que atacan.

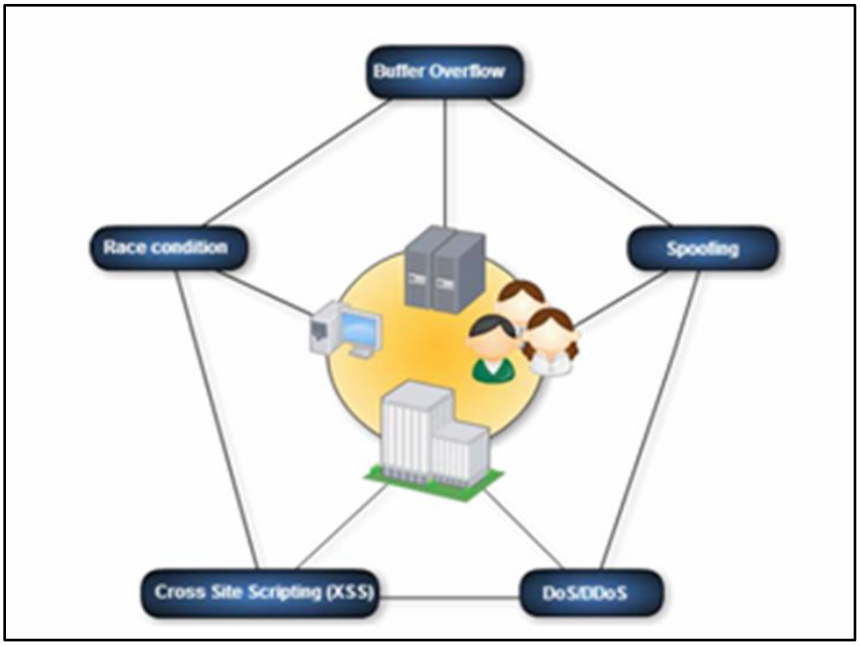


Figura 2.4.- Tipos de Vulnerabilidades Encontradas en App. Web. (Jiménez, 2016, pág 37)

2.2.1.1.- XSS.

La *Vulnerabilidad* conocida como Cross Site Scripting (**XSS**) o ejecución de comandos en sitios cruzados es una de las más habituales, por lo que centrarse en cómo funciona y cómo afecta a sus víctimas es uno de los temas más preocupantes al que programadores Web inexpertos, deben familiarizarse para evitar *Vulnerabilidades* relacionados al tema.

Es un tipo de *Vulnerabilidad* informática o agujero de seguridad típico de las **App.** Web, permite a una tercera persona inyectar código JavaScript o en otro lenguaje similar, evitando medidas de control y restricciones del mismo origen (Alonso Cebrián, Guzmán Sacristán, Laguna Durán, & Martín Bailón, 2014a), en la **Figura 2.5** se ejemplifica el proceso de explotación de una *Vulnerabilidad* por **XSS**.

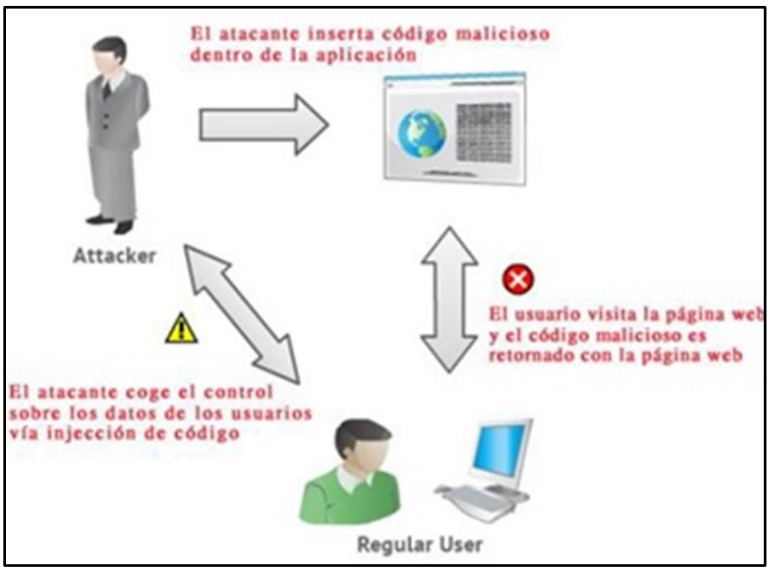


Figura 2.5.- Procesos de un Ataque de Tipo XSS. (UNAM-CERT, 2015)

En primer lugar, es importante tener en cuenta que, con esta *Vulnerabilidad* los atacantes explotan la confianza de un usuario en un sitio Web en particular, el impacto que puede tener este tipo de *Vulnerabilidad* al ser utilizada dependerá de su tipo los cuales se dividen en dos maneras:

- **No persistentes**: Los ataques no persistentes o reflejados no almacenan el código malicioso en el servidor, sino que lo pasan y presentan directamente a la víctima. Es el método más popular de ataques de tipo **XSS** y puede ser lanzado desde una fuente externa, por ejemplo, mediante correo electronico o un sitio de terceros.

- **Persistente:** El código malicioso ya ha superado la barrera del proceso de validación y está guardado en un almacén de datos. Puede ser un comentario, un Archivo log un mensaje de notificación, o cualquier otro tipo de sección del sitio Web que solicita algún *Input* al

usuario. Cuando esta información en particular se presenta en el sitio Web, el código malicioso se ejecuta.

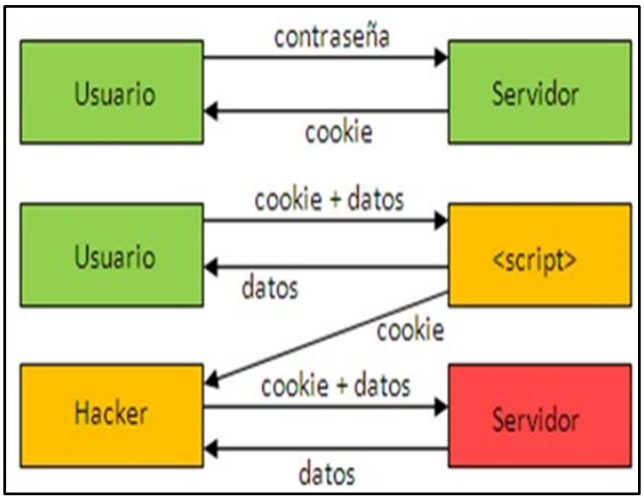


Figura 2.6.- Operación de un Ataque de Tipo XSS. (UNAM-CERT, 2015)

2.2.1.2.- SQL Injection

La *Vulnerabilidad* de inyección de código **SQL** o mejor conocida entre la comunidad informática como **SQL** Injection es una problemática con la que las **App.** Web no han podido mitigar desde el inicio del uso de bases de datos, diversas organizaciones relacionadas en el tema la clasifican entre las principales que todo programador y administrador de **BD** debe verificar en su codificación, sin embargo, pese a los esfuerzos por eliminarlo sigue estando en los lugares más altos de *Vulnerabilidades* explotadas por los *Hacker*; en la **Tabla 2.1** se observa el lugar en que se clasifica según el daño e índice de explotación en un estudio propuesto por **OWASP** y publicado en su Top10 de amenazas a **App.** Web desde el 2007 hasta el 2017

En palabras de (Sriphum, Chomsiri, Attanak, & Noitarong, 2011) **SQL** Injection consiste en un proceso que permite a los usuarios con conocimiento, el insertar comandos de código **SQL** en la base de datos conectada al servidor, para esto es necesario conocer el tipo de gestor de base de datos debido a que muchas de estas tienen pequeñas diferencias de codificación siendo, entre los gestores más populares se encentran MySQL. PostgreSQL, SQL Server etc. (Ramírez Castro, 2012)

Con la ayuda de esta *Vulnerabilidad* los usuarios no autorizados podrían realizar diversas acciones dentro de la base de datos entre las más importantes se encuentra la consulta de información con esta visualización el atacante modifica algunas consultas para acceder a los registros de la **BD** y la suplantación de usuarios consultando las credenciales de usuarios dados de alta en el sistema realizando acciones no autorizadas previamente con el usuario (Alonso Cebrián, Guzmán Sacristán, Laguna Durán, & Martín Bailón, 2014) en la **Figura 2.7** se ejemplifica el proceso de una petición por **SQL** Injection.

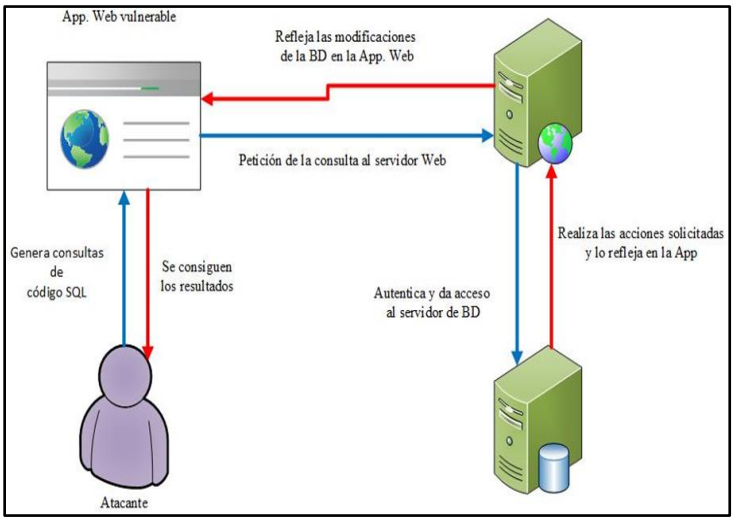


Figura 2.7.- Diagrama de una Explotación de Inyección de Código SQL. (Autor)

2.2.1.3.- CSRF.

Las vulnerabilidades de peticiones en sitios cruzados o **CSRF** (por sus siglas en ingles cross-site request forgery), es un tipo de *Exploit* de carácter malicioso encontrada en **App.** de entorno Web, donde usuarios pueden emplear comandos, que el servidor Web interpreta como verdaderos y forzándolo a realizar acciones a través de la víctima.

Si bien este tipo de vulnerabilidad es parecida a las de tipo **XSS**, es posible diferenciarlas en su modo de operación. Las vulnerabilidades de tipo **XSS** se basan en la explotación de la confianza del usuario y las vulnerabilidades de tipo **CSRF** están basadas en la explotación de confianza del servidor Web.

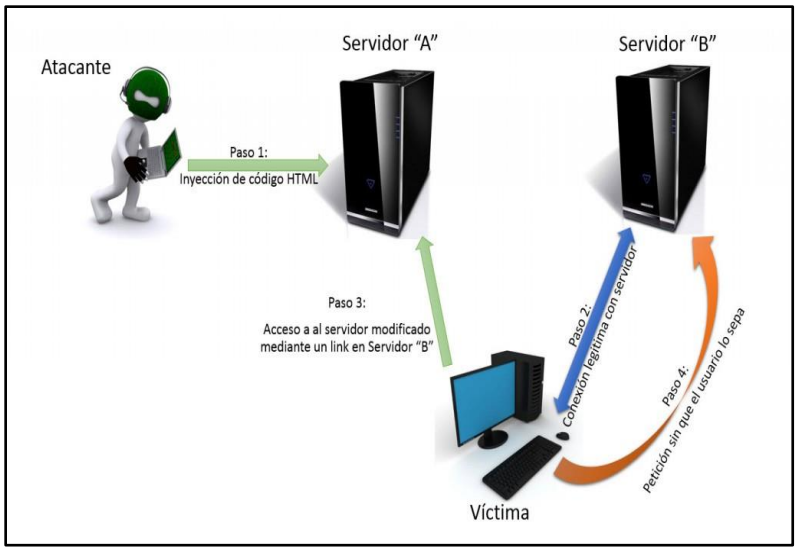


Figura 2.8.- Ejemplificación de un Ataque de Tipo CSRF. (Pérez, 2015)

En palabras de (Pérez, 2015) “Los riesgos que plantea la explotación de **CSRF** incluyen distintos tipos de actividades ilícitas e indeseadas. Desde el acceso a cuentas privadas de usuarios, pasando por acusar a alguien de acceder a sitios de contenido indebido e inclusive, hasta habilitar algún tipo de filtro o regla en el correo electrónico para que todos los correos se reenvíen a otra cuenta”. En la **Figura 2.8**, se muestra una ejemplificación de un ataque de tipo **CSRF**.

2.2.2.- Tecnologías orientadas a la seguridad Web.

Dentro de las tecnologías implementadas por los auditores informáticos o *Pentester* según las regulaciones que permiten dicha actividad se encuentran señalados ciertos algoritmos de apoyo, esto permite que las aplicaciones, den ciertas mediciones de decisión para que el auditor o diseñador Web pueda visualizar los parámetros vulnerables de su aplicación.

En el caso del diseñador, la tecnología de medición de *Vulnerabilidades* debe tomarse en cuenta desde el ciclo de vida de desarrollo del *Software*, por sus siglas en inglés **SDLC** (Systems Development Life Cycle), según (Meucci & Muller, 2014)**,** esto se debe a que es necesario obtener un enfoque equilibrado, este enfoque incluye el enfoque de revisiones periódicas de manuales, aprovechando todas las fases en la que se encuentre el desarrollo del **SDLC**.

En otros sentidos **OWASP,** recomienda el uso de prácticas de pruebas en busca de *Vulnerabilidades*, cuando ya se tiene una aplicación Web implementada y en funcionamiento, con la finalidad de llegar a los límites desafiando todos los escenarios posibles, tales como la restricción de acceso al código fuente y realizar una serie de pruebas complejas (Meucci & Muller, 2014).

Entre las soluciones a la problemática del diseño en **App**. Web y apoyo a la auditoria de seguridad informática, son consideradas las siguientes como las más usadas:

- Pruebas y uso de *Exploits* de tipo *Black box.*
- Pruebas y uso de *Exploits* de tipo *White box.*
- Escáner de seguridad de aplicaciones Web (escáner de *Vulnerabilidad*).
- *Firewall*s de aplicación Web (**WAF**).

Dentro de la tesis se abordarán a detalle diversas metodologías, estableciendo métricas de *Vulnerabilidades* proporcionadas por herramientas especializadas en su detección, las herramientas usadas en el banco de pruebas se encuentran orientadas al desarrollo de *Pentesting,* entre dichas herramientas se encuentran las siguientes.

2.2.2.1.- NMAP 7.70.

Actualmente, una de las principales herramientas para el mapeo de redes y escáner de puertos en busca de *Vulnerabilidades* a nivel de red, es **NMap**, creada originalmente por Gordon Lyon actualmente esta herramienta se encuentra en su versión 7.70 siendo una **App.** *Open Source,* según (Lyon, 2016) la herramienta usa paquetes de tipo **IP** sin procesar, con la finalidad de determinar que *Hosts* se encuentran disponibles dentro de la red, sus servicios, los sistemas operativos (**SO**) que usa, el filtrado en los *Paquetes*, entre otras características.

2.2.2.2.- Zed Attack Proxy Project (ZAP)2.7.0

De acuerdo a lo dicho por (OWASP, 2018b), **ZAP** es una de las herramientas multiplataforma más usadas en el escaneo para la búsqueda de *Vulnerabilidades* de **App**. Web, contando con actualizaciones comunitarias de manera internacional, esta herramienta se trata de un *Servidor proxy*, capaz de establecer peticiones en busca de *Vulnerabilidades* dentro de los componentes de una **App.** de entorno Web y capaz de establecer dicha búsqueda de manera manual o automática.

2.2.2.3.- Kali Linux 2017.3

(Allen, Heriyanto, & Shakeel, 2017) El **SO** Kali Linux actualmente en su versión 2017.3, es una distribución basada en debían orientado al *Pentesting* o auditoria informática contando con muchas herramientas en su repertorio orientadas a la seguridad informática y que facilitan el trabajo del auditor, en la búsqueda de *Exploits* para aprovechar *vulnerabilidades* en todo tipo de aplicaciones y redes informáticas, en palabras de (OSCP, 2018) otras utilidades para el **SO** van desde, Investigación de seguridad, Informática forense e Ingeniería inversa.

Capítulo 3.- Aseguramiento Técnico y Material.

En el siguiente apartado se describe el desglose y explicación de las diferentes herramientas que se consideran indispensable para la elaboración de una auditoria informática, estas herramientas consisten en el uso de *Software* y *Hardware* especializado, así como los costos de producción y personal involucrados en la metodología y elaboración del proyecto.

Los costos de producción de un *Pentesting* o auditoria informática varían según los requerimientos y nivel que requiera evaluar el **ITA**, para la producción del proyecto, el cual se ve reflejado en la presente tesis, consiste en auditar una App. de entorno Web con el nombre de **SII** perteneciente al **ITA**, basado en el cumplimiento del **EVSA** 3.0.1 de **OWASP**, considerando los costos de producción divididos de la siguiente forma:

- Costos de *Hardware*.
- Costos de *Software*.
- Otros gastos.

3.1.- Costo de Hardware.

Dentro del apartado se refleja el costo de los dispositivos y herramientas a nivel físico las cuales servirán para elaborar el proyecto de auditoria informática, aunque algunas de estas ya se encontraban dentro de **CECOMP** y otras son de pertenencia propia es necesario desglosar los

costos con la finalidad que el lector pueda establecer comparativas y buscar alternativas en caso de ajuste de presupuesto.

3.1.1.- Equipo de cómputo

El equipo de cómputo utilizado consta de dos computadoras las cuales servirán para establecer las pruebas de *Pentesting* en dos sistemas operativos (**SO**) diferentes (Kali Linux y Windows), el motivo de esto se debe a que es necesario que se establezcan diferentes tipos de escenarios de casos de intrusión utilizando diferentes herramientas de *Software* las cuales se describen en el Marco teórico, contenido en el **Capítulo 2.**

La descripción de estos equipos se muestra en la **Tabla 3.1**, mostrando los costos de todos los equipos utilizados dentro del proyecto.

Tabla 3.1.- Tabla Presupuestal del Proyecto. (Autor)

Modelo	cantidad	Características	Costos de mercado	Costos de producción
PC - ASUS X556U	1	• Procesador: Intel Core i7. • Pantalla: 15,6”. • Disco duro: 1TB HDD • RAM: 8GB • Salida HDMI. • Tarjeta de gráficos independiente (NVIDIA GEOFORCE 940M de 2GB). • S.O: WINDOWS 10 PRO 64bits. • Lector de CD	Los costos en el mercado de una PC portátil de estas características rondan entre $15,000.00 y $19,000.00 pesos (moneda nacional Mexicana).	Se adquirió una PC portátil del modelo especificado con un costo total de $15,900.00 pesos (moneda nacional Mexicana).

Modelo	cantidad	Características	Costos de mercado	Costos de producción
PC - Notebook Samsung 14 Np270	1	• Procesador: AMD A6 • Pantalla: 14” • Disco duro: 586 GB • RAM: 4 GB • S.O: Windows 7 Home premiun • Lector de CD	Los costos actuales en el mercado para una PC portátil con las características anterior mente mostradas ronda entre los $4,000.00 y los 6,000.00 pesos (moneda nacional Mexicana)	Equipo de uso personal adquirida a fechas previa al proyecto por lo tanto no fue contemplado dentro de los costos de producción. $0.00

Modelo	cantidad	Características	Costos de mercado	Costos de producción
Servidor - Dell PowerEdge R715 de 2U	1	• Procesadores AMD Opteron serie 6100 basados en la plataforma AMD Opteron serie 6000. • Memoria: 256 GB (16 ranuras DIMM). 1 GB/2 GB/4 GB/8 GB/16 GB. Hasta 1.333 MHz. • Opciones de disco duro conectables en caliente: SSD SATA de 2,5", SAS (10 000 rpm, 15 000 rpm), SAS nearline (7200 rpm) y SATA (7200 rpm). • Almacenamiento interno máximo: Hasta 6 TB. • Tarjeta gráfica: Matrox G200eW con 8 MB.	Los costos actuales del mercado para un servidor con las características ya descritas y en buen estado se encuentran con un estimado de $4 239.00 dólares, convirtiéndolo a moneda nacional Mexicana, queda un aproximado de $78,482.00 a fechas de cotización del dólar según la banca Mexicana para el año 2017	El servidor con las especificaciones mencionadas existe en el Centro de cómputo (**CECOMP**) con anterioridad por lo tanto no se consideró como gastos del proyecto. $0.00

Modelo	cantidad	Características	Costos de mercado	Costos de producción
Servidor HP Proliant ML110 Gen9	1	• procesador: Intel Xeon E5 v4 • Tipo de memoria interna DDR4-SDRAM. • 8 Ranuras de memoria. • Memoria interna 8 GB. • Memoria interna máxima 256 GB. • 2 Ethernet **LAN** (RJ-45) cantidad de puertos. • 8 puertos USB 2.0 • 1 puertos VGA (D-Sub). • Tecnología de cableado 10/100/1000Base-T(X). • Tipo de interfaz ethernet Gigabit Ethernet.	Los costos actuales del mercado para un servidor con las características ya descritas y en buen estado se encuentran con un estimado de $29,000.00 y $23,000.00 pesos (moneda nacional Mexicana)	El servidor con las especificaciones descritas fue adquirido por el Centro de cómputo por un precio aproximado de $29,000.00 pesos (moneda nacional Mexicana)

Modelo	cantidad	Características	Costos de mercado	Costos de producción
Servidor DELL PowerEdge T620	1	• Procesadores Procesador Intel® Xeon®, familia de productos de E5-2600 • 2 Sockets del procesador • Memoria Hasta 768 GB (24 ranuras DIMM): DDR3 hasta 1600 MHz • Una ranura x8 con ancho de banda x4, de longitud y altura completas • Capacidad máxima de almacenamiento interno Hasta 6TB • 1 Discos duros • Opciones de disco duro de conexión en marcha:	Los costos actuales del mercado para un servidor con las características ya descritas y en buen estado se encuentran con un estimado de $19,000.00 y $20,000.00 pesos (moneda nacional Mexicana)	El servidor con las especificaciones anterior mente mencionadas existe en el Centro de cómputo (**CECOMP**) con anterioridad por lo tanto no se consideró como gastos del proyecto. $0.00

		• 2.5” PCIe SSD, SAS SSD, SATA SSD, SAS (15 K, 10 K), nearline SAS (7.2 K), SATA (7.2 K) • 3.5” nearline SAS (7.2 K), SATA (7.2 K), SAS (15 K) • Dispositivos de auto cifrado disponibles • NIC incorporado LOM Intel de 1 GbE y dos puertos • Fuente de alimentación Efectividad de platino de 495 W, 750 W, ó 1100 W • Fuentes de alimentación de rango automático		
Total			44,900.00 pesos Mexicanos	

Los datos contenidos en la **Tabla 3.1** refleja los gastos del mercado en cuanto al material ocupado, como se expresa en dicha tabla la mayoría de los materiales que se ocupan para el desarrollo del proyecto, ya se encontraban en **CECOMP,** dicha situación genera un ahorro para la realización del proyecto.

En otras palabras, el total de gastos en adquisición de material para el proyecto se limita a la adquisición de una computadora portátil, modelo **ASUS X556U**, con un costo de $15,900.00 y un servidor con el modelo **HP Proliant ML110 Gen9** con un precio de $29,000.00. La justificación para el uso de este material se describe a continuación en la **Tabla 3.2.**

Tabla 3.2.- Justificación del Material Para el Desarrollo del Proyecto. (Autor)

Material	Justificación	Imagen
PC - ASUS X556U	Computadora portátil con la cual se plantea abordar escenarios de intrusión al Sistema integral de información (**SII**), utilizando Software en entornos de **SO** de *Windows 10*, así como la generación de reportes.	Imagen en anexo1
PC – Notebook Samsung 14 Np270.	Computadora portátil la cual cuenta con características óptimas para la explotación de vulnerabilidades utilizando herramientas de auditoría informática contenidas en el **SO** *Kali Linux,* por lo cual se reemplazará el **SO** de Windows 7 por la anteriormente mencionada.	Imagen en anexo 2
Servidor - Dell PowerEdge R715 de 2U	Hardware especializado en el almacenamiento, con este servidor, se pensaba duplicar la base de datos de la aplicación de entorno Web a evaluar, pero debido a fallos con el flujo eléctrico en el departamento del **ITA**, sufrió desperfectos en dispositivos que lo conforman, por esta	Imagen en anexo 3

	razón se llega a la conclusión que este ya no cumple con los requerimientos para la fiabilidad del proyecto, sumando que la base de datos actual del **SII** trabaja con un servidor de modelo y marca diferente a este, sin embargo es plasmado debido a que hay pruebas que se elaboraron inicialmente al desarrollo del proyecto..	
Servidor HP Proliant ML110 Gen9	Hardware especializado en el almacenamiento, con este servidor se duplicara el servidor Web a evaluar debido a que existen pruebas agresivas que podrían dañar la integridad de la App., así como, el funcionamiento de dicha aplicación, debido a esto se opta por el uso de un servidor de pruebas el cual cumple con las mismas características que el original.	Imagen en anexo 4
Servidor en torre PowerEdge T620	Hardware especializado en el almacenamiento, con este servidor se duplicara la base de datos de la aplicación de entorno Web a evaluar debido a que existen pruebas agresivas que podrían dañar la integridad de los datos contenidos en este, así como, el funcionamiento de dicha aplicación, debido a esto se opta por el uso de un servidor de pruebas el cual cumple con las mismas características que el original	Imagen en anexo 5

3.2.- Costos de Software

El *Software* utilizado para el proyecto puede variar en base a las necesidades del auditor y el tipo de evaluación a las **TIC'S** de la organización auditada, dicho esto se desglosan los costos de algunas aplicaciones que se utilizan para la ejecución del *Pentesting* al **SII**.

Tabla 3.3.- Tipos de Software y Costos. (Autor)

Software	Tipo de licencia	Costos del proyecto
Kali Linux	Open Source	$0.00
Nmap	Open Source	$0.00
Wireshark	Open Source	$0.00
OWASP Zed Attack Proxy	Open Source	$0.00
Total		$0.00

Como se muestra en la **Tabla 3.3** el *Software* utilizado para el proyecto es de tipo *Open Source*; por consiguiente, no se emite ningún costo por el uso de estas herramientas, aunque en algunos casos pueda verse restringido o limitado en algunas funciones, se considera que no es un factor relevante para la toma de resultados. En contraste, a los altos costos que implica el licenciamiento de algunos programas en su entorno comercial, se deduce que la versión gratuita cumple con las necesidades generales para llegar a emitir evaluaciones fiables y por consiguiente, se acuerda adquirir únicamente las versiones de código abierto.

3.3.- Otros Gastos

Dentro de esta sección se desglosan los gastos de transportación, alimentos e impresión de documentos, estos gastos son contemplados dentro del proyecto a forma de gastos del auditor.

Tabla 3.4.- Tabla de Gastos en un Lapso Semestral, Periodo Enero - Junio.2017 (Autor)

Periodo Enero – Junio 2017		
Tipo de gastos	**Gasto al día**	**Gastos semestral**
Transporte	$ 20.00	$3,640.00
Alimento	$50.00	$8,400.00
Impresiones	$15.00	$2,730.00
Total	$85.00	$15,470.00

Tabla 3.5.- Tabla de Gastos en un Lapso Semestral, Periodo Agosto – Diciembre 2017. (Autor).

Periodo Agosto – Diciembre 2017		
Tipo de gastos	**Gasto al día**	**Gastos semestral**
Transporte	$ 20.00	$3,640.00
Alimento	$50.00	$8,400.00
Impresiones	$15.00	$2,730.00
Total	$85.00	$15,470.00

Tabla 3.6.- Tabla de Gastos en un Lapso Semestral, Periodo Enero - Junio 2018. (Autor)

Periodo Enero – Junio 2018		
Tipo de gastos	**Gasto al día**	**Gastos semestral**
Transporte	$ 20.00	$3,640.00
Alimento	$50.00	$8,400.00
Impresiones	$15.00	$2,730.00
Total	$85.00	$15,470.00

En la **Tabla 3.4, Tabla 3.5** y **Tabla 3.6** es posible observar los gastos que se sopesan durante la producción del proyecto en términos de transporte alimentación e impresión de documentos los cuales han sido seccionados en periodos semestrales.

Destacando que el proceso de investigación de la tesis cubre un lapso de tiempo total a dos años al tiempo de ingreso; no se contempla el periodo inicial debido a que durante el transcurso de ese periodo no se encontraban definido los materiales ocupados para el desarrollo del proyecto.

Capítulo 4.- Metodología

Dentro de los trabajos de seguridad informática, son diversas las metodologías para las pruebas de *Pentesting,* en busca de fallos en la seguridad de la información, debido a esto el encargado de esta tarea, se ve en la necesidad de elaborar un plan de acción, debido a que muchas de estas actividades pueden afectar la integridad de las **TIC'S** de la organización.

Si bien es entendido que toda auditoria informática cuenta con una serie de pasos cuya finalidad es simular un ataque real a las **TIC´S,** para modificar y robar información valiosa para el atacante, justificado en el hecho de que estas pruebas proporcionan un control y planeación que permite determinar las fortalezas y debilidades.

Es vital que durante las pruebas de penetración se tomen en cuenta los diversos tipos de estándares existentes como puede ser el "Estándar de Verificación de Seguridad en Aplicaciones 3.0.1" **(EVSA)** auxiliándose con la metodología creada por **OWASP**, la cual se describe en su Testing Guide (Guía de pruebas) versión 4.0, ambas son recomendadas por expertos en el tema, debido a que se encuentran orientadas a App. Web.

4.1.- Metodología OWASP

"La metodología diseñada por **OWASP** propone un marco de trabajo el cual ayuda al proceso del proyecto, orientado al ciclo de vida del *Software* (**SDLC**)" (Meucci & Muller, 2014) otorgando soluciones con una alta flexibilidad para mejorar el proceso de desarrollo de las App.

estando orientado en gran porcentaje a entorno Web, teniendo la seguridad como uno de sus principales temas.

El método de riesgo estándar usado por **OWASP**, se presenta en la **Ecuación 4.1.**

$$\textbf{Riesgo} = \text{Probabilidad} * \text{impacto}$$

Ecuación 4.1.- Fórmula del Riesgo Estándar de OWASP

Como se puede ver en la **Figura 4.1, OWASP** demuestra que a medida que avance un proyecto, este obtendrá un costo mayor en el desglose de problemáticas y la seguridad con que toda App. debe de cumplir, esto quiere decir, que entre más avanzado se encuentre el proceso este tendrá un costo mayor para su solución

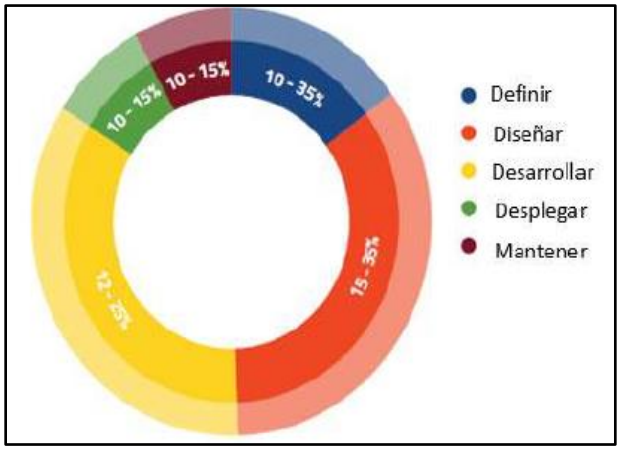


Figura 4.1.- Costo de Corrección de Errores en el SDLC. (Meucci & Muller, 2014, pág. 11)

La metodología está diseñada para la producción de App. seguras y con el menor número de fallos, para el caso de estudios, no se trabaja desde el punto de diseño e implementación, por

consiguiente, se presentan planes de trabajo que ayudan al diseño de un plan de acción para la evaluación del **SII,** usando otros medios de apoyo los cuales se expondrán en apartados siguientes.

En la evaluación de riesgos de **OWASP**, se indica una serie de pasos a seguir para la valoración de los niveles de riesgos que contenga una App. los cuales se plantean a continuación.

- Identificar el riesgo
- Factores para estimar la probabilidad
    - Factores de agentes de amenazas.
    - Factores de *Vulnerabilidad*.
- Factores para estimar el impacto
    - Factores de impacto técnico
    - Factores de impacto en el negocio
- Determinar la gravedad del riesgo
- Decidir que arreglar

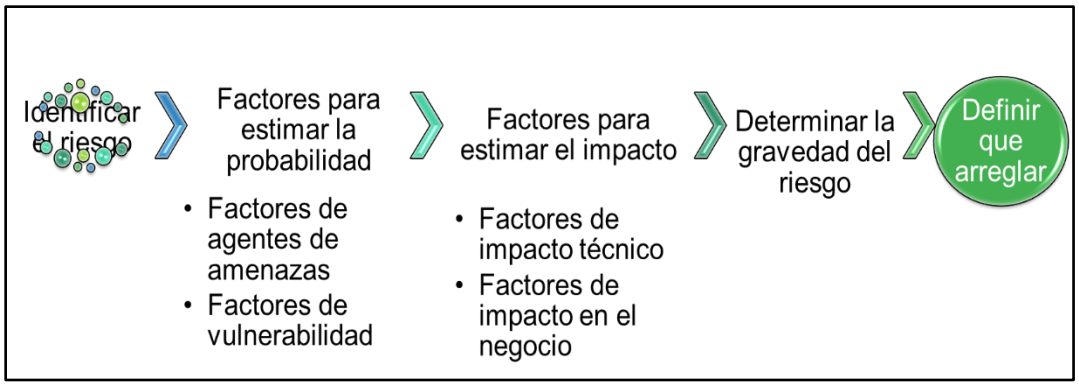


Figura 4.2.- Transición de la Metodología. (Autor)

4.1.1.- Factores para estimar la probabilidad.

Para identificar los riesgos es necesario identificar los peligros en seguridad que se necesitan tratar dentro del **SII**. para esto se hace una clasificación con la finalidad de recopilar toda la información sobre los agentes de amenazas que se encuentren involucrados, el ataque utilizado, la *Vulnerabilidad* y el impacto sobre la organización en caso de éxito de un *Exploit*; (OWASP, 2018a) recomienda usar el escenario del peor caso debido que al generar un riesgo alto este puede dar un amplio espectro en la toma de decisiones

En esta fase se cuantifica la probabilidad de ocurrencia de una *Vulnerabilidad* una vez se han identificado los riegos, las probabilidades de que una *Vulnerabilidad* pueda ser descubierta y explotada según (OWASP, 2017) esta clasificación puede dividirse en Alta, Media y Baja como se observa en la **Tabla 4.1,** esto se debe a que no es necesario ser muy preciso en esta estimación ya que solo se trata de dar un panorama de la probabilidad de que un atacante descubra y explote una *Vulnerabilidad*.

Los factores involucrados en este paso están asociados a un puntaje de probabilidad que va de **0 al 9**, en apartados siguientes se explica el simbolismo de este puntaje y su importancia para el cálculo de los escenarios.

Tabla 4.1.- Probabilidad de Ocurrencia de Vulnerabilidades de OWASP. (OWASP, 2018a)

Alta	**Media**	**Baja**
Vulnerabilidad que al ser explotada puede comprometer la seguridad de la información ocasionando un impacto negativo sobre la empresa	Vulnerabilidad que al ser explotada tendrá un impacto moderadamente significativo sobre las operaciones del negocio.	Vulnerabilidad que al ser explotada no ocasiona inconvenientes
Prioridad de solución		
Debe solucionarse inmediatamente	Puede ser solucionada en un tiempo prudente	La solución puede no ser inmediata

4.1.1.1.- Factores de agentes de amenazas.

Básicamente la finalidad de este conjunto se trata en la estimación de la probabilidad de que un ataque tenga éxito en el hallazgo y explotación a *Vulnerabilidades* en el **SII**, resaltando que **OWASP** siempre recomienda usar el peor agente apegado a las necesidades de la App. y la organización en este caso refiriéndose al **ITA**,

En la **Tabla 4.2** se pueden observar los diferentes agentes, el puntaje correspondiente a estos y su clasificación por habilidades, motivación, oportunidad y tamaño

Tabla 4.2.- Agentes de Amenazas y su Tabulación. (OWASP, 2018a)

Habilidades técnicas	**Motivación**
Habilidades de penetración de seguridad (1). Habilidades de red y programación (3). Usuarios avanzados en Computación (5). Algunas habilidades técnicas (6) Sin habilidades técnicas (9)	Baja o nula recompensa (1). Posible recompensa (4). Alta recompensa (9)
Oportunidad	**Tamaño**
Acceso completo (0) Acceso especial (4). Algunos accesos (7) Sin acceso (9)	Desarrolladores (2). Administradores de sistemas (2). Usuarios de la intranet (4). Socios (5). Usuarios autenticados (6). Usuarios de Internet anónimos (9)

4.1.1.2.- Factores de vulnerabilidad.

El principal objetivo es la estimación de las probabilidades de que una o más *Vulnerabilidades* sean descubiertas en el **SII** y posteriormente ser explotada, suponiendo que se tiene el escenario con alguno de los valores mencionados en la **Tabla 4.2**, se procede a identificar su factor, establecer la sumatoria entre los valores ya mencionados y los valores expuestos en la **Tabla 4.3,** para finalmente dividirlos entre el total de los factores**.**

Tabla 4.3.-Factores de Vulnerabilidad y Su Tabulación (OWASP, 2018a)

Facilidad de descubrimiento	**Facilidad de explotación**
Prácticamente imposible (1) Difícil (3) Fácil (7) Herramientas automatizadas disponibles (9)	Herramientas teóricas (1). Difíciles (3) Fáciles (5) Automatizadas (9)
Conocimiento	**Detección de intrusos**
Desconocido (1) Oculto (4) Obvio (6) Conocimiento público (9)	Detección activa en la aplicación (1) Registrada y revisada (3) Registrada sin revisión (8) No registrada (9)

4.1.1.3.- Cálculo del factor para estimar la probabilidad.

El cálculo de la Probabilidad e impacto se obtiene haciendo una sumatoria de los factores de agentes de amenazas y los de *Vulnerabilidad* para consecutivamente se hace una división con el total de los agentes de ambos factores, como se muestra en la **Ecuación 4.2**.

$$\frac{\sum Variables\ de\ amenazas + \sum variables\ de\ factores\ de\ vulnerabilidades}{8}$$

Ecuación 4.2.- Fórmula Para el Cálculo de la Probabilidad e Impacto

4.1.2.- Factores para estimar el impacto.

Al considerar las consecuencias de un ataque exitoso, es importante pensar en los dos tipos de impacto los cuales determinan el nivel de gravedad del ataque, estos son, los factores de impacto técnico y factores de impacto en el negocio. Siendo este último el más importante, sin embargo, es posible que no tenga acceso a toda la información necesaria para descubrir las consecuencias comerciales de un *Exploit* exitoso.

En este caso, proporcionar tantos detalles sobre el riesgo técnico permitirá al representante apropiado tomar una decisión sobre el riesgo comercial.

4.1.2.1.- Factores de impacto técnico.

El principal objetivo de estimar la magnitud y gravedad del impacto en el sistema en caso de explotarse una *Vulnerabilidad*; está fuertemente relacionado a términos clásicos de la seguridad informática, en la **Tabla 4.4** es posible observar las relaciones, así como la tabulación de los componentes.

Tabla 4.4.- Agentes de Impacto Técnico (OWASP, 2018a)

Pérdida de confidencialidad	**Pérdida de integridad**
Mínima de datos no sensibles divulgados (2) Datos críticos mínimamente divulgados (6) Divulgación de datos extensos no sensibles (6) Divulgación de datos críticos extensos (7) Divulgación de todos los datos (9)	Datos mínimos ligeramente corruptos (1) Datos mínimos corruptos (3) Datos extensos ligeramente corruptos (5) Datos muy corruptos (7) Todos los datos totalmente corruptos (9)
Pérdida de disponibilidad	**Pérdida de responsabilidad**
Interrupción mínima de servicios secundarios (1), Interrupción mínima de servicios primarios (5). Interrupción extensa de servicios secundarios (5), Interrupción extensa de servicios primarios (7), Pérdida total de los servicios (9)	Completamente rastreable (1). Posiblemente rastreable (7). Completamente anónimo (9).

4.1.2.1.1.- Cálculo para los factores de impacto técnico.

Una vez que se establezca el escenario y se visualicen los valores es posible obtener los cálculos haciendo una sumatoria de las variables de impacto técnico y después dividirlo ente el número de componentes involucrados como se muestra en la **Ecuación 4.3**.

$$\frac{\sum Variables\ de\ impacto\ tecnico}{4}$$

Ecuación 4.3.- Fórmula Para el Cálculo del Impacto Técnico

4.1.2.2.- Factores de impacto en el negocio.

El presente factor se encuentra estrechamente ligado a los factores de impacto técnico, con la diferencia de que se requiere conocer a profundidad y la importancia que tiene para **CECOMP** departamento que gestiona el **SII**., esto quiere decir que el objetivo general es el apoyo para la estimación de los riesgos con un impacto en el negocio, este riesgo le justifica a la organización educativa la inversión para la solución de los problemas de seguridad

Tabla 4.5.- Factores de Impacto en el Negocio. (OWASP, 2018a)

Daño financiero	**Daño de reputación**
Menor que el costo de corregir la vulnerabilidad (1) Efecto menor en el beneficio anual (3) Efecto significativo en el beneficio anual (7) Quiebra (9)	Daño mínimo (1) Pérdida de cuentas principales (4) Pérdida de credibilidad (5) Daño de la marca / imagen (9)
Incumplimiento	**Violación de la privacidad**
Infracción menor (2) Violación clara (5) Violación de alto perfil (7)	Un individuo (3) Cientos de personas (5) Miles de personas (7) Millones de personas (9)

Al mismo tiempo una estandarización ayuda en el enfoque de formalización para lo que realmente puede ser importante, en términos de seguridad; lo dicho anteriormente se considera obligatorio en caso de que el **SII**. no disponga de una estructura formalizada o estandarizada. En

la **Tabla 4.5** se exponen los factores que influyen sobre el impacto en el negocio, así como su tabulación.

4.1.2.2.1.- Cálculo para los factores de impacto en el negocio.

El cálculo del impacto que puede tener alguno de los escenarios sobre el negocio, es calculado siguiendo una fórmula, esta consiste en sumar los valores de impacto entre el número de factores, la visualización de lo anteriormente mencionado puede apreciarse en la **Ecuación 4.4**.

$$\frac{\sum Variables\ de\ impacto\ en\ el\ Negocio}{4}$$

Ecuación 4.4.- Fórmula Para el Cálculo del Impacto Sobre el Negocio. (OWASP, 2018a)

4.1.3.- Determinar la gravedad del riesgo.

Para determinar la gravedad en el riesgo de que un *Exploit* sea utilizado, es necesario combinar la estimación de probabilidad y de impacto, con la finalidad de calcular una gravedad general en determinado riesgo, tabulando los resultados en un intervalo de alta, media y baja, haciendo lo mismo para el impacto, en una escala que va de 0 a 9 como se muestra en al **Tabla 4.6.**

Tabla 4.6.- Valores de los Niveles de Probabilidad e Impacto. (OWASP, 2018a)

Niveles de probabilidad e impacto	
Entre 0 < 3	Bajo
Entre 3 < 6	Medio
Entre 6 < 9	Alto

4.2.- Requisitos para el Desarrollo.

Destacando que a pesar de que el proyecto en ámbitos generales y específicos no trata sobre la construcción de un sistema de *Software*, pero si en la evaluación de uno, se remarca un análisis de especificación de requerimientos de *Software* (**ERS**), el cual enumere los pasos para la producción y validación del proyecto, con el fin de que el cliente, en este caso de estudio "**CECOMP**", quede conforme y enterado de los pasos a cumplir durante el proyecto e informar sobre los puntos a cubrir hasta la culminación del proyecto.

4.2.1.- Estandarización.

Para lograr una normalización de las App. muchos de los desarrolladores optan por elegir un estándar que pueda ayudar en el diseño y programación de sus App., visto desde el punto de vista de la ingeniería del *Software*, el normalizar las aplicaciones. asegura que el programador pueda detectar fallos o errores con mayor facilidad.

Para el caso de estudio se decide el uso del **EVSA v3.0** de **OWASP** debido que al estar orientado en su totalidad a **App.** Web y apoyado en su **Testing Gide 4.0**, resulta útil para la

revisión de actividades que deben cumplir, del mismo modo se verifica el cumplimiento de análisis de los módulos que competen al proyecto dentro del **SII** relacionados a las principales *Vulnerabilidades* expuestas por **OWASP** en su Top10 el cual se describe a detalle en la **Tabla 2.1**.

4.2.2.- Requisitos de seguridad.

Los requerimientos de seguridad definen como funciona una **App.**, en un punto de vista enfocado a la seguridad de la información, por lo tanto, es fundamental ser lo más claro y concreto con la definición de éstos, además se deben probar y evaluar para corregir deficiencias con el objetivo de obtener mejoras.

Para un proyecto de *Pentesting*, el proteger los insumos de la organización, tiene que estar implícito en el desarrollo de las pruebas, en otras palabras, primeramente, es necesario entablar una conversación con el personal encargado del departamento que solicite la auditoria y establecer parámetros de seguridad, del mismo modo delimitar las pruebas a las necesidades que se deseen cubrir.

Llegando a un acuerdo con el Actual jefe de **CECOMP** departamento perteneciente al ITA a fechas de **3 de octubre del 2016**, se plantea la auditoria a una **App**. de entorno Web la cual es administrada y gestionada por el departamento ya mencionado, las evaluaciones que se plantean para las pruebas, según altercados pasados e investigaciones teóricas, se hace un desglose de las principales *Vulnerabilidades* en un lapso de tiempo de 10 años que va, desde el 2007 al 2017 elaborados y publicados por **OWASP** como se describe en la **Tabla 2.1.**

4.2.3.- Análisis de requerimientos.

Teniendo en cuenta lo anterior se crean los siguientes requerimientos que debe cubrir el proyecto, estos fueron proporcionados por los administradores del **SII** en base a sus necesidades de evaluación; Planteando ciertas limitaciones para la protección de la infraestructura, la adquisición de herramientas y la búsqueda de flaquezas, quedando de la siguiente manera:

4.2.3.1.- Requerimientos funcionales y no funcionales.

En la **Tabla 4.7**, se presentan los requerimientos funcionales que debe cumplir el proyecto para su elaboración, dichos requerimientos fueron proporcionados por el jefe del Centro de cómputo y serán realizados por el auditor,

Tabla 4.7.- Tabla de Requerimientos Funciónales. (Autor)

Identificador	Descripción
RF - 1	Las pruebas deben de cumplir un estándar para la confiabilidad de las pruebas y resultados que se contemplen en el proyecto.
RF - 2	Las pruebas deberán buscar vulnerabilidades de entornos específicos como lo son inyección **SQL**, **XSS**, etc.
RF - 3	Los escenarios deben de estar definidos de manera clara y específica para su entendimiento, es recomendable el uso se diagramas de casos de uso y diagramas de flujo para llevar un seguimiento de los procesos realizados.

RF – 4	Los resultados que se obtengan deberán ser cuantificados usando un estándar relacionado al tema, se sugiere el *Risk Rating Methodology* de **OWASP.**
RF- 5	Toda la evidencia obtenida deberá ser confidencial, respetando la integridad del departamento y la institución.

En la **Tabla 4.8,** se presentan los requerimientos no funcionales los cuales definen las restricciones del proyecto, del mismo modo estos requerimientos delimitaran el proyecto, enfocándose solamente en las necesidades a cubrir para el departamento evitando ambigüedades o pérdida de tiempo en la evaluación de puntos innecesarios.

Tabla 4.8.- Tabla de Requerimientos No Funcionales. (Autor)

Identificador.	Descripción.
RNF - 1	El uso de Hardware necesario para las pruebas deberá realizarse en infraestructura perteneciente al **ITA** y a **CECOMP**, en el caso de la auditoria el equipo debe de ser de uso personal del auditor.
RNF - 2	El uso y adquisición de Software deberá ser de preferencia de licencia open Source.
RNF - 3	Los servicios proporcionados por la App. deben estar funcionales en todo momento que se realicen las pruebas, no es admisible la baja de ninguno de los servicios en ningún lapso de tiempo o día.
RNF - 4	La integridad de la **BD** deberá estar integra en todo momento, no es admisible ningún tipo de cambio, manipulación o borrado de datos.

Capítulo 5.- Procedimiento e Implementación del Proyecto

Dentro del capítulo se presentan los trabajos realizados durante el tiempo que se realizó el proyecto, el seguimiento de la metodología para la realización de las pruebas, cálculos y recolección de datos y resultados; derivados de estos puntos, el capítulo justifica los resultados plasmados posteriormente en el **Capítulo 6**,

El procedimiento de la auditoria al **SII** está enfocada directamente a la búsqueda y explotación de las *Vulnerabilidades* publicadas por **OWASP**, con la finalidad de cuantificar las debilidades del **SII,** para determinar el nivel de seguridad, permitiendo visualizar las principales flaquezas dentro del **SII**. y del mismo modo distinguir los puntos prioritarios a resolver.

Otro punto a destacar es el escenario para el banco de pruebas, teniendo en cuenta las restricciones impuestas por la organización, según el cumplimiento de los **RNF - 3** y **RNF – 4** las cuales se realizan por medio del duplicado de los servidores Web y de **BD** del **SII**, cumpliendo con las mismas características que los servidores originales, dando origen al **SIItest**; se llegó a esta decisión debido que un ambiente simulado en un entorno virtual con cualquier herramienta existente en el mercado (VirtualBox, VMware) no podía igualar los niveles de procesamiento de los servidores en un ambiente real, por consiguiente esto restaría veracidad y confianza a los resultados obtenidos.

De este modo el desglose del proyecto compete a los apartados que se procede a exponer y detallar en los siguientes subtemas.

5.1.- Duplicando los Servicios Web y BD

Partiendo del modelo expuesto por (Fonseca Romero, 2017) se extraen las ideas de crear herramientas de entrenamiento con la diferencia que mientras en su estudio el autor presenta una simulación de servidores virtuales con la herramienta **VMware Workstation Player**[4], con la finalidad de entrenar a determinado personal en temas de seguridad informática. En el campo de estudio que compete a esta tesis se determina, que no es viable, debido a que al hacerlo se perdería poder de procesamiento, debido a que el tiempo de respuesta no es igual en un ambiente virtual comparado con un entorno real.

Una vez que se determinó este inconveniente, **CECOMP** proporcionó dos servidores tanto para la **BD**, el cual se puede observar en la **Figura 5.1** y otro para el servidor Web como se puede observar en la **Figura 5.2**, cabe destacar que estos servidores cumplen con las mismas características que los dispositivos donde se encuentra alojada el **SII** original, esto debe de brindar un índice de confianza en las pruebas, debido a que las *Vulnerabilidades* que se encuentren en el **SIItest**, deben de poder encontrarse y explotarse de la misma forma en la original (**SII**).

[4] "Herramienta ideal para ejecutar una única máquina virtual en una PC con Windows o Linux".(VMware, 2018)

Figura 5.1.- Servidor de BD. (Autor)

Figura 5.2.- Servidor Web. (Autor)

La arquitectura de los servicios determina como trabaja y sus funciones, por esto se procedió a configurar primeramente el servidor Web, el cual recibe todas las peticiones y enlaza la base de datos, teniendo terminada esta parte se procedió a duplicar la **BD** del servidor de base de datos original, y se enlaza por medio de un cableado Crossover (*T568A*). En la **Figura 5.3** se observa la arquitectura general del banco de pruebas, asimismo la transición de las peticiones y de respuesta entre el usuario y los servidores reflejándose en el navegador Web.

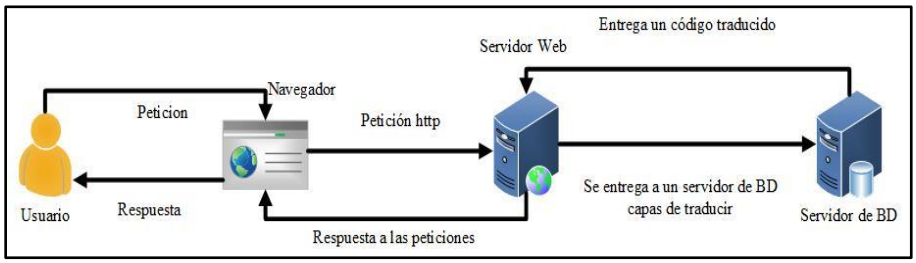


Figura 5.3.- Arquitectura de los Servicios de la App. (Autor)

5.1.1.- Características del servidor.

En la siguiente tabla se muestran las características con las que cuenta el servidor de base de datos.

Tabla 5.1.- Características del Servidor. (Autor)

Sistema operativo (SO)	Ubuntu 16.04.4 LTS (GNU - Linux 4.4.0-87-generic x86_64)
Procesador	6 Cores - Intel Xeon CPU E5-2603 v4 @ 1.70Ghz
Memoria RAM	8 GB
Nombre del Host	SIItest

5.1.2.- Configuración del servidor Web.

Como se mencionó anteriormente el **SO** gestor del Servidor Web es una distribución GNU - *Linux* (Ubuntu server en su versión 16.04.4), este sistema operativo será el encargado de contener las carpetas de diseño y visualización de la interface indexadas a una **IP** privada y posteriormente saldrá al exterior por una **IP** publica de un proveedor de *Hosting* que renombrará la **IP** con una **URL.**

Terminados los pasos de configuración y montaje del **SIItest**; se pone en ejecución y se duplica el sitio con la diferencia de acceso público por **URL** como se muestra en la **Tabla 5.2** es posible visualizar el contraste de **URL** e entre la App. original y la que se usa en el campo de estudio (**SIItest**).

Tabla 5.2.- Tabla de Datos del SII y SIItest. (Autor)

App. Original	App. Duplicada
SII	SIItest
www.siiacapulco.com	www.siitest.com

Puesta en ejecución el **SIItest** cumple con todas las funciones que el **SII** original, siguiendo el modelo de (IBM, 2014) la arquitectura del **SIItest** se encuentra dividida en tres capas, como puede visualizarse en la **Figura 5.4.**

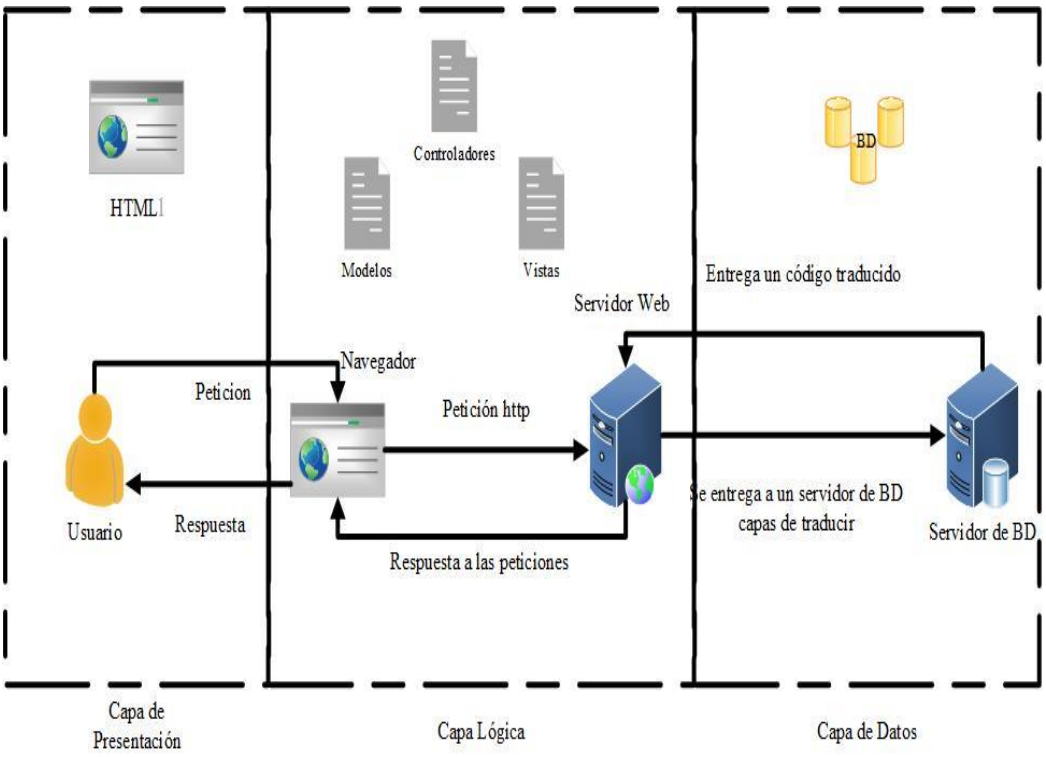


Figura 5.4.- Arquitectura de los Servicios en Capas. (Autor)

- **Capa de presentación:** se encuentra conformada por todo lo que el usuario puede visualizar desde su navegador (Opera, Firefox, Chrome, etc.) usando una traducción de lenguaje a **HTML**. En este nivel recae la responsabilidad de la presentación e interacción con el usuario, permitiendo al usuario interactuar con los componentes del segundo nivel por medio de peticiones, haciendo que el **SIItest** funcione de manera segura e intuitiva.

- **Capa lógica:** se encarga de gestionar el funcionamiento del **SIItest** utilizando un lenguaje especial del lado del servidor (PHP. ASP.net, Python, etc.). los procesos que gestionan esta capa, pueden acceder a los servicios de la capa de datos, en esta capa se produce la mayoría de los procesos debido a que esta debe ser capaz de gestionar sus propias transacciones.

- **Capa de Datos:** En esta capa se almacena la información de la organización usando un sistema gestor de base de datos (**SGBD**), el cual es el encargado de interpretar los lenguajes de **BD,** generalmente en **SQL**. Estos servicios deben estar protegidos en todo momento impidiendo el acceso directo a pesar de estar en una red segura y todos los procesos deben de hacerse a través de la capa lógica.

5.2.- Escaneo de Vulnerabilidades

Dentro del apartado se muestra el procedimiento de escaneo del **SIItest**, esta acción tiene el objetivo de buscar de manera general y remota las *Vulnerabilidades* que se puedan presentar en una App. de entorno Web, generando un conocimiento inicial del estado del **SIItest** y tener un punto de partida en la auditoria.

Dicho lo anterior se procede a mostrar los resultados obtenidos del escaneo al **SIItest**, usando diferentes herramientas con el objetivo de ampliar los resultados, entre las herramientas usadas para el escaneo se encuentran las mencionadas a continuación.

- Nmap 7.70, liberada en su reciente actualización el 20 de marzo de 2018 (Lyon, 2016).
- OWASP Zed Attack Proxy (ZAP) 2.7.0, liberada en su última versión el 28 de noviembre del 2017 (OWASP, 2018b)

5.2.1.- Escaneo con Nmap.

La herramienta Nmap 7.70, es usada en dos **SO** diferentes partiendo originalmente del instalado por defecto en la distribución Kali Linux como se muestra en la **Figura 5.5**, por medio de un escaneo sencillo a la **URL** del **SIItest** se verifican los puertos que se encuentran abiertos y sus servicios

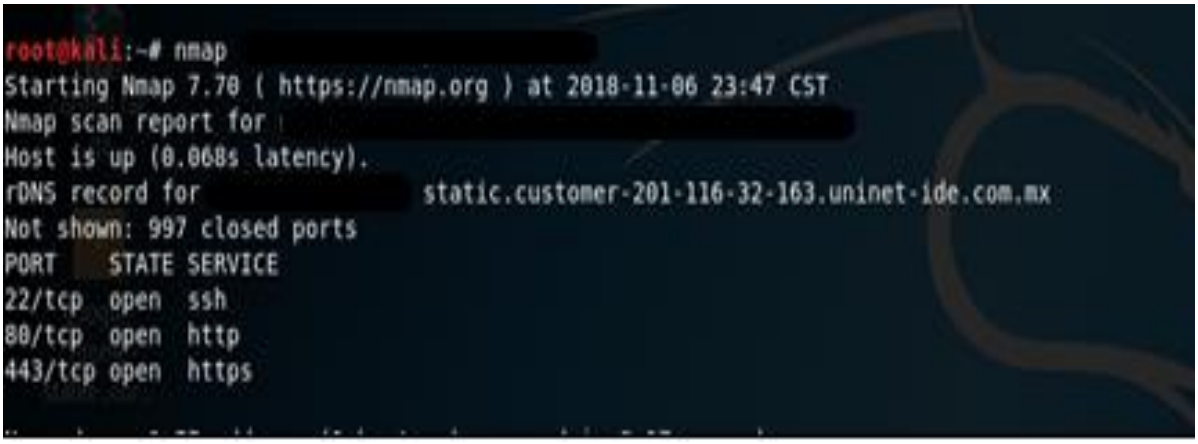


Figura 5.5.- Escaneo Básico al SIItest Con Nmap. (Autor)

En otra prueba usando el comando **–sV**[5] se analiza al conjunto de puertos abiertos detectados para tratar de descubrir servicios activos y versiones en puertos abiertos como se observa en la **Figura 5.6**, el comando tiene éxito al responder a las solicitudes de conexión con el protocolo usado por Ubuntu, la versión de *Apache* y que puertos responden.

[5] Interroga al conjunto de puertos detectados para tratar de descubrir servicios y versiones en puertos abiertos.(CSIRT-cv, 2018)

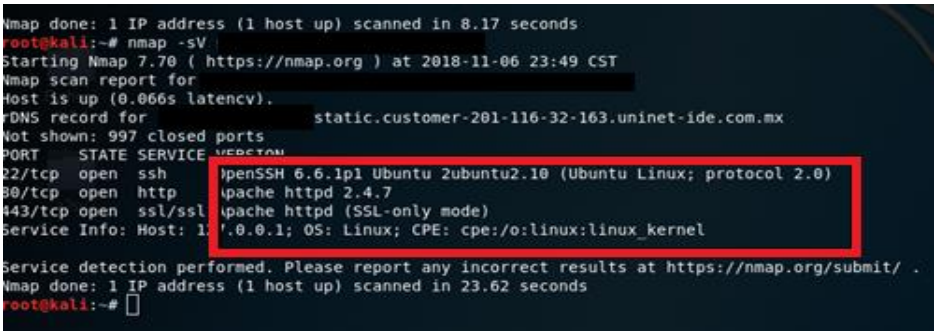
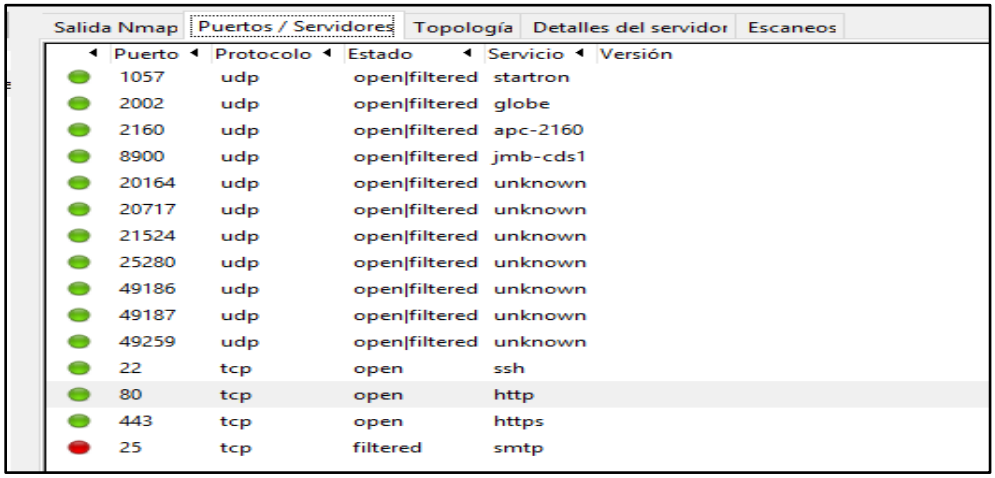


Figura 5.6.- Resultado del Uso del Comando -sV En Nmap. (Autor)

Pasando a un escaneo en un entorno Windows se usa el programa Nmap 7.70 existente para este SO, como se puede visualizar en la **Figura 5.7** la interfaz proporciona un entorno gráfico en contraste a su versión para Kali Linux y de igual manera hace el escaneo en un ambiente casi automático, permitiendo usar los comandos de Nmap 7.70 de manera manual, para este caso se usó uno de los perfiles por defecto (Slow comprehensive scan), en comandos sería al equivalente a usar los comandos < -sS -sU -T4 -A -v -PE -PP -PS80,443 -PA3389 -PU40125 -PY -g 53 --script "default or (discovery and safe)">, la nomenclatura de los comandos se muestra en el **Tabla 5.3**



Puerto	Protocolo	Estado	Servicio	Versión
1057	udp	open|filtered	startron	
2002	udp	open|filtered	globe	
2160	udp	open|filtered	apc-2160	
8900	udp	open|filtered	jmb-cds1	
20164	udp	open|filtered	unknown	
20717	udp	open|filtered	unknown	
21524	udp	open|filtered	unknown	
25280	udp	open|filtered	unknown	
49186	udp	open|filtered	unknown	
49187	udp	open|filtered	unknown	
49259	udp	open|filtered	unknown	
22	tcp	open	ssh	
80	tcp	open	http	
443	tcp	open	https	
25	tcp	filtered	smtp	

Figura 5.7.- Visualización de los Puertos y Servicios

Tabla 5.3.- Lista de Comandos Utilizados en Nmap. .(CSIRT-cv, 2018)

Comando	Nombre	Funcionamiento
-sS	Sondeo TCP SYN	Envía un *SYN*, relativamente sigilosa haciendo un análisis semi abierto debido a que no completa el enlace TCP.
-sU	Sondeo UDP	Envía UDP vacío combinándolo en paralelo a otras técnicas permite diferenciar entre puertos abiertos y filtrados.
-T<plant>	Plantilla de tiempo	Define una plantilla genérica de tiempo, en el banco de pruebas se usa –T4 donde 4 es el simbolismo que Nmap usa para agresivo
-A	Opción de sondeo agresivo	Esta opción activa algunas opciones avanzadas y agresivas, común mente es usada para la de sección de SO
-V	Versión	Muestra los números de versión
-PE	Ping ICMP echo	Envía un ICMP echo de tipo Request, la mayoría de las veces es filtrado por el Firewall
-PP	Ping ICMP timestamp	Envía un ICMP timestamp de tipo Request, la mayoría de las veces es filtrado por el Firewall
-PS<ports>	Ping TCP SYN	Envia un SYN especificando los puertos que desea, para las pruebas se sondeó el puerto 80 y 443
-PA<ports>	Ping TCP ACK	Envía un ACK vacío especificando los puertos a los que desea sondear, para las pruebas se sondea el puerto 3389

-PU<ports>	Ping UDP	Envía un UDP vacío especificando los puertos que se desean sondear, para las pruebas se sondea el puerto 40125
-PY<ports>	Ping SCTP	Envía paquetes SCTP de tipo INIT al puerto que se desea sondear, para las pruebas se dejó el puerto por defecto que es el 80
-g<port>	Falseo de dirección	Envía paquetes usando el puerto especificado para las pruebas se usó el puerto 53
--script <Valor>		Define los script a utilizar con algún valor, para las pruebas se usó –script default

Al contrastar ambos resultados es posible ver que los puertos que se encuentran abiertos proporcionan información al atacante del **SO** usado en el servidor, el tipo de protocolo y servicios. A partir de esta información es posible crear un método de ataque que involucre la información que se ha recabado.

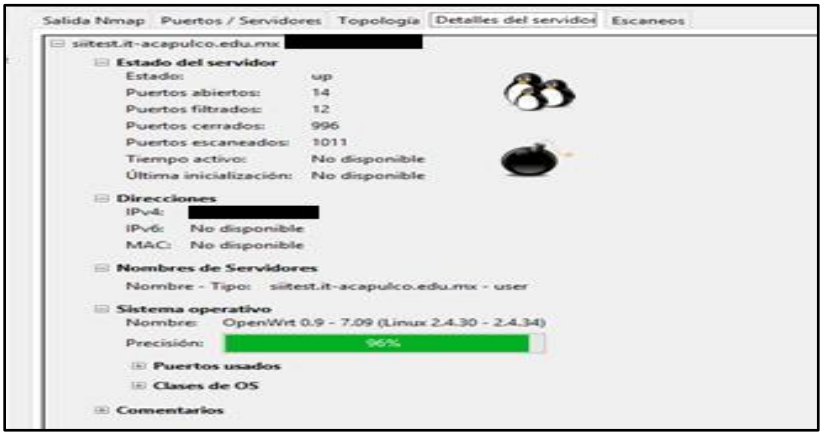


Figura 5.8.- Datos del Servidor Proporcionados Por NMap. (Autor)

5.2.2.- Escaneo con OWASP ZAP.

La herramienta **ZAP** perteneciente a **OWASP** es un *Servidor proxy* que permite capturar el tráfico de las peticiones en la navegación, sin embargo, otra de sus funciones es el escaneo de *Vulnerabilidades* en las **App**. Web, esta herramienta se encuentra orientada en la búsqueda de las *Vulnerabilidades* expuestas por **OWASP** en su top ten, la implementación de esta herramienta se justifica debido a la línea de investigación que compete a la tesis.

Como se muestra en la **Figura 5.9** la herramienta cuenta con un escáner automático a las *Vulnerabilidades* esta función genera varias peticiones de tipo **GET** al **SIItest**, y determina bajo que líneas es posible crear un ataque.

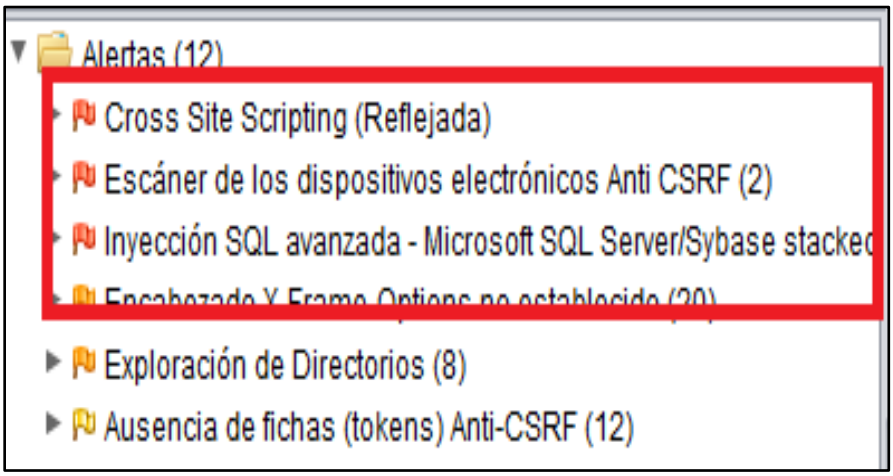


Figura 5.9.- Resultado del Escaneo al SIItest con la Herramienta ZAP. (Autor)

Como es posible apreciar en la F**igura 5.9**, los resultados obtenidos muestran la existencia de ciertas *Vulnerabilidades* dentro del **SIItest**, encontrando que entre las más destacables se encuentran las de tipo **XSS**, **CSRF** y las de inyección SQL, además de que también es posible ver

otras *Vulnerabilidades* de menor rango, como lo es la exposición de **URL**s que no deberían ser visibles debido al tipo de información que permiten extraer, así como la transición de las peticiones en la autenticación de usuarios; en los futuros apartados se procederá a explorar estos escenarios.

5.3.- Explotación de las vulnerabilidades

Dentro de este apartado se describen los escenarios fórmulados para la explotación de las *Vulnerabilidades*, partiendo de los resultados obtenidos en el escaneo al **SIItest**, dichos resultados generan un panorama inicial dentro de la línea de investigación, permitiendo compararlas con las expuestas en la Testing Guide v4 de **OWASP**, para posteriormente en base a la alimentación del banco de pruebas y los resultados obtenidos determinar el nivel de seguridad del **SIItest** y por consiguiente la del **SII.**

5.3.1.- Autenticación y gestión de sesiones

La autenticación y gestión de las sesiones está relacionada a un mal funcionamiento en el manejo de la información en los servidores Web, para el banco de pruebas se realizaron experimentos bajo dos posibles escenarios en los que la organización destaco mayor preocupación, entre los escenarios se encuentran:

- Explotación en la comunicación cifrada en el proceso de acceso a la aplicación.
- Inspección en contraseña de usuarios.

5.3.1.1.- Explotación en la comunicación cifrada en el proceso de acceso a la aplicación.

La explotación de esta *Vulnerabilidad* está basada en un ataque comúnmente denominado como hombre en el medio (man in the middle), dentro de este ataque se presenta el escenario donde una persona con entrenamiento en informática tiene acceso a la red de transferencia o comunicación de datos del **SII** (siendo en el banco de pruebas el **SIItest**), con la finalidad de poder capturar el tráfico de la red y visualizar la interacción que tienen los usuarios autenticados con el servidor Web.

En la explotación se usa la herramienta Wireshark 2.6.4, para la captura de paquetes dentro de la red, para esto primeramente es necesario verificar la **IP** del **SIItest**, con la finalidad de comprobar la seguridad en el protocolo de encriptación de las peticiones al servidor.

Dejando claro lo anterior se procede a generar el vector de ataque, primeramente, es necesario averiguar la **IP** de asociación a la **URL** para ello es necesario hacer un ping desde **CMD**, esto es posible hacerlo desde la plataforma de Windows como se muestra en la **Figura 5.10**, haciendo este procedimiento se visualiza la **IP** y se extraen para la intervención en la red.

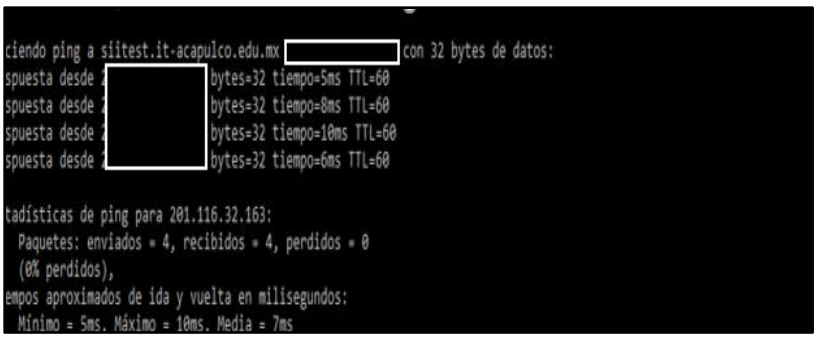


Figura 5.10.- Revisión de la IP Por CMD. (Autor)

Después se verifica el tráfico que existe para la **IP** que deseamos evaluar, en este caso se presenta en la **Figura5.11,** el escaneo de paquetes en el **SIItest** y usando uno de los filtros de Wireshark es posible observar de mejor manera el tráfico en la red únicamente en respuesta a las peticiones de inicio de sesión de usuarios del **SIItest**.

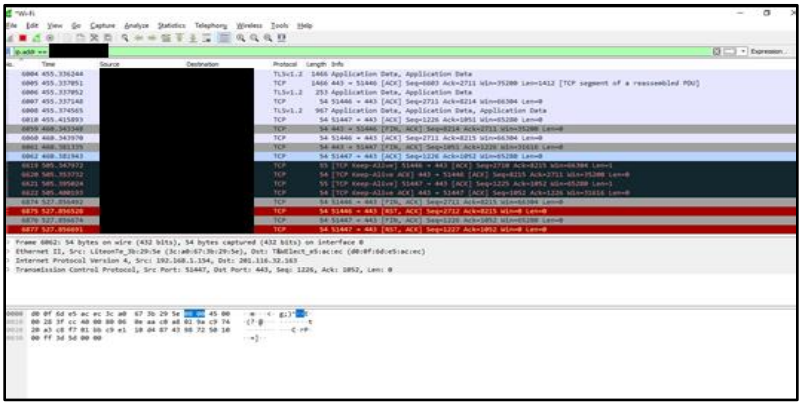


Figura 5.11.- Escaneo Con la Herramienta Wireshark. (Autor)

El escaneo de la red se desarrolla en dos escenarios, teniendo en cuenta que el atacante pueda tener acceso a la red de un usuario que se encuentre realizando peticiones al servidor, como se muestra en el **Figura 5.12** y el segundo contemplando la posibilidad de que el atacante obtenga acceso a la red de datos del **SII** (para el banco de pruebas **SIItest**), como se muestra en la **Fugura**

5.13, aclarando que el acceso a la red se encuentra restringido a personal no autorizado, sin embargo, es necesario la evaluación para la verificación de la encriptación de los protocolos en el certificado de seguridad.

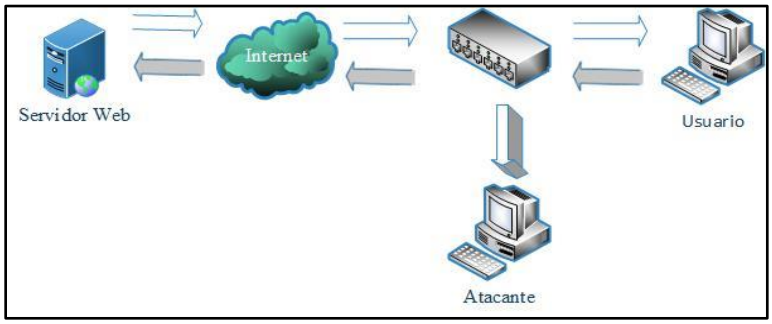


Figura 5.12.- Uso de Sniffer a Nivel de Usuario. (Autor)

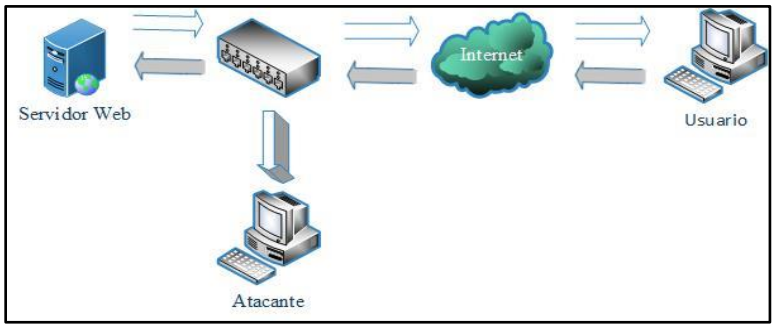


Figura 5.13.- Uso de Sniffer a Nivel de Servidor Web. (Autor)

Teniendo en cuenta lo anterior, se determina que la encriptación de las peticiones **GET** y **POST** no son visibles en el tráfico de paquetes **HTTP**, debido al certificado de seguridad que encripta la comunicación entre el usuario y servidor, por consiguiente, un usuario no autenticado no podrá hacerse con el usuario y contraseña de un usuario que si se encuentre autenticado en el sistema por lo que será necesario que el atacante pase a pruebas más agresivas.

5.3.1.2.- Ataque de diccionario en contraseña de usuarios.

La explotación de un ataque de fuerza bruta basadas en diccionario a cuentas de usuarios, es una de las *Vulnerabilidades* más explotadas en la actualidad según (Salazar, 2015), esta *Vulnerabilidad* se presenta cuando las cuentas de usuario no cuentan con una robustez significativa, en términos de caracteres que puedan ampliar el rango de combinaciones entre ellas.

Para el banco de pruebas se deja de lado las problemáticas en cuanto a ingeniería social, debido que muchos de los usuarios no tienen la cultura en relación a la protección de contraseñas (Florêncio et al., 2007), generando problemáticas sobretodo en usuarios de bajo perfil. Partiendo de esta restricción se procede a realizar pruebas a la clave de usuarios de bajo perfil (alumnos), para las pruebas se genera un diccionario que alimentara al programa que realiza las consultas de las contraseñas de manera automática (ZAP OWASP), como se puede observar en la **Figura 5.14**.

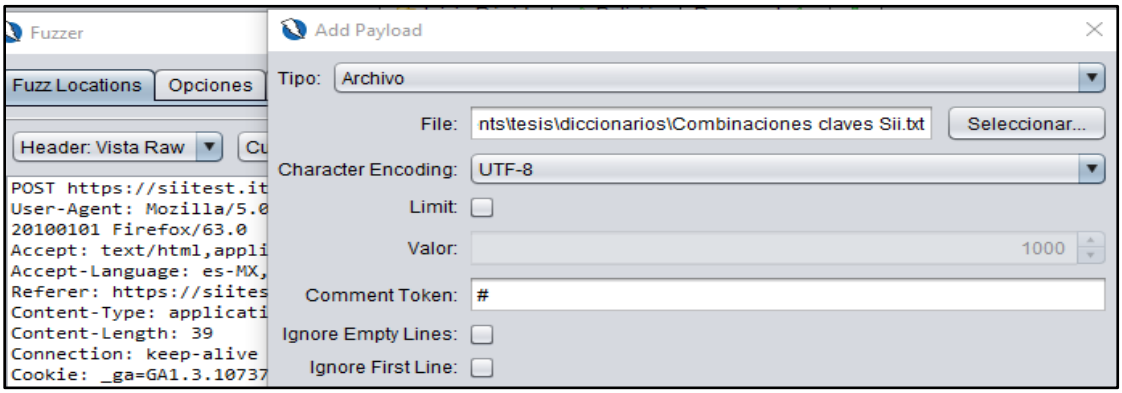


Figura 5.14.- Inserción del Diccionario Para la Alimentación de Contraseñas en ZAP. (Autor)

Una vez que se ha alimentado el programa con el diccionario, se procede a hacer la inspección de las posibles contraseñas para un usuario seleccionado con anterioridad como se

puede observar en la **Figura 5.15**, la clave es hallada y de esta manera es posible acceder a la cuenta del usuario en el **SIItest**, como se muestra en el **Figura 5.16,** en el banco de pruebas se realizó la explotación de la *Vulnerabilidad* en doce usuarios de bajo perfil teniendo éxito en todos ellos, considerando los resultados y el nivel de éxito en las pruebas, es determinante que el **SIItest** no cuenta con un sistema de seguridad para este tipo de ataque.

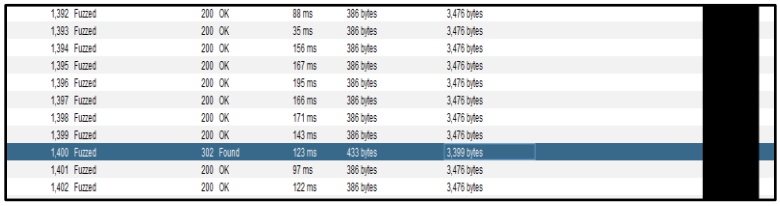


Figura 5.15.- Hallazgo de la Contraseña en el Diccionario. (Autor)

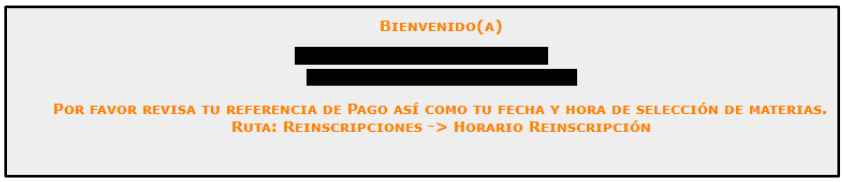


Figura 5.16.- Loguin del Usuario Atacado en el SIItest

5.3.1.3.- Identificación del riesgo para la vulnerabilidad de pérdida de autenticación y gestión de sesiones

Dentro de la línea de investigación se procede a clasificar el riesgo, siguiendo la **Tabla 2.1** de clasificación de *Vulnerabilidades* según **OWASP** contenida en el **Capítulo 2**, se determina que los resultados recabados en el banco de pruebas coinciden la *Vulnerabilidad* mostrada en la **Tabla 5.4.**

Tabla 5.4.- Ataque de Pérdida y Autenticación de Sesiones y Sus Característica. (Autor)

Ataque	Riesgo
Pérdida y autenticación de sesiones	Un ataque de este tipo es generado cuando un atacante obtiene la información de los usuarios autenticados quedando expuesta o si bien el atacante puede ingresar a la sesión de un usuario

Si bien dentro de los escenarios expuestos para la *Vulnerabilidad*, no fue posible hallar las peticiones del loguin de los usuarios, debido a la encriptación de los datos dentro del **SIItest** con el certificado de seguridad. Sin embargo, el **SIItest** no presento una resistencia significativa en ataques de fuerza bruta basada en diccionarios, este escenario representa un riesgo debido a que atacantes pueden acceder por este medio a cuentas de usuario, a pesar de ser usuarios de bajo perfil el riesgo de acceso no autorizado representa un riesgo en cuanto a la información personal de los usuarios, además de que, en base a esto se puede acceder a otras *Vulnerabilidades*, este tema se demuestra en otros apartados

5.3.1.4.- Establecer la probabilidad de ocurrencia para la Vulnerabilidad de pérdida y autenticación de sesiones

Como se puede observar, en las pruebas realizadas para la explotación de *Vulnerabilidades* de pérdida y autenticación de sesiones en el **SIItest**, se determinan los parámetros que involucran dicha explotación según la metodología expuesta en el **Capítulo 4**, se procede a clasificar los agentes de amenazas, esta clasificación es posible observarla en el **Tabla 5.5**

Tabla 5.5.- Clasificación de los Agentes de Amenazas Para la Vulnerabilidad de Pérdida y Autenticación de Sesiones. (Autor)

Amenazas				
Tipo de taque	**Habilidades técnicas**	**Motivaciones**	**Oportunidad**	**Tamaño**
Pérdida y autenticación de sesiones	Habilidades de res y programación (3)	Posible recompensa (4)	Algunos accesos (7)	Usuarios autenticados (6)

Habiendo tabulado los agentes de amenazas dentro del escenario se procede a establecer los agentes para los factores de *vulnerabilidad* y su tabulación como se observa en la **Tabla 5.6,** para que de esta manera se pueda proseguir con los cálculos de la probabilidad de ocurrencia, como lo dicta la metodología de clasificación de riesgos de **OWASP**

Tabla 5.6.- Asignación de Valores Para los Agentes de Vulnerabilidades en Pérdida y Autenticación de Sesiones. (Autor)

Vulnerabilidad				
Ataque.	**Facilidad de descubrimiento.**	**Facilidad de explotación.**	**Conciencia o conocimiento.**	**Detección de intrusos.**
Pérdida y autenticación de sesiones	Herramientas automatizadas disponibles (9)	Fáciles (5)	Oculto (4)	No registrada (9)

Teniendo los valores para los agentes de amenazas y *Vulnerabilidades* se procede a realizar el cálculo de la probabilidad de ocurrencia siguiendo la fórmula mostrada en **Ecuación 5.1**.

$$Probabilidad = \frac{(Ht + Mt + Op + Tm) + (Fd + Fe + Cc + Di)}{8}$$

Ecuación 5.1.- Fórmula Para la Probabilidad de Ocurrencia. (Autor)

Donde los valores para el cálculo se muestran en la **Tabla 5.7**, quedando tabuladas de la siguiente manera:

Tabla 5.7.- Tabla de Tabulación en la Probabilidad de Ocurrencia. (Autor)

Habilidades técnicas	Ht	3	Facilidad de descubrimiento	Fd	9
Motivaciones	Mt	4	Facilidad de explotación	Fe	5
Oportunidad	Op	7	Conciencia o conocimiento	Cc	4
Tamaño	Tm	6	Detección de intrusos	Di	9

$$Probabilidad = \frac{(3 + 4 + 7 + 6) + (9 + 5 + 4 + 9)}{8} = \frac{20 + 27}{8} = \frac{47}{8} = 5.87$$

Ecuación 5.2.- Cálculo de la Probabilidad de Ocurrencia Para Pérdida y Autenticación de Sesiones. (Autor)

Como se observa en la **Ecuación 5.2**, la probabilidad de ocurrencia para la *Vulnerabilidad* de autenticación y gestión de sesiones, encontrada en el **SIItest** da un grado Medio de explotación, según la tabulación de **OWASP**, con un rango total de 5.87.

5.3.1.5.- Estimación del impacto para la Vulnerabilidad de pérdida y autenticación de sesiones.

En esta fase se determina el impacto que genera en la organización la explotación de la *Vulnerabilidad* de pérdida y autenticación de sesiones, para esto es necesario calcular el impacto técnico y el impacto en el negocio, cuantificando los agentes que intervienen, para después dividirlo entre el total de los agentes, como se observa en la **Tabla 5.8** y **Tabla 5.9**, respectivamente; el resultado total contrastado con la probabilidad de ocurrencia determinara el nivel de seguridad que tiene en **SIItest** con esta *Vulnerabilidad*.

Tabla 5.8.- Asignación de Valores Para Calcular el Impacto Técnico en la Vulnerabilidad de Pérdida de Autenticación y Gestión de Sesiones. (Autor)

Impacto Técnico				
Ataque	**Pérdida de confidencialidad**	**Pérdida de integridad**	**Pérdida de disponibilidad**	**Pérdida de responsabilidad**
Pérdida y autenticación de sesiones	Mínima de datos no sensibles divulgados (2)	Datos mínimos corruptos (3)	Interrupción mínima de servicios secundarios (1)	Completamente anónimo (9)

Tabla 5.9.- Asignación de Valores Para Calcular el Impacto en el Negocio, Para la Vulnerabilidad de Pérdida de Autenticación y Gestión de Sesiones. (Autor)

Impacto en el negocio				
Ataque.	**Daño financiero.**	**Daño en la reputación.**	**Incumplimiento.**	**Violación de la privacidad.**
Pérdida y autenticación de sesiones	Efecto significativo en el beneficio anual (7)	Daño de la marca / imagen (9)	Violación de alto perfil (7)	Cientos de personas (5)

Teniendo los valores para los agentes de impacto técnico e impacto en el negocio se procede a realizar el cálculo de la probabilidad para la estimación del impacto general con la fórmula mostrada en **Ecuación 5.3**.

$$Impacto = \frac{(Pf + Pi + Pd + Pr) + (Df + Dr + In + Vp)}{8}$$

Ecuación 5.3.- Fórmula Para la Probabilidad de Impacto. (Autor)

Donde los valores para el cálculo se muestran en la **Tabla 5.10**

Tabla 5.10.- Tabla de Tabulación en la Probabilidad de Impacto. (Autor)

Pérdida de confiabilidad	Pf	2	Daño financiero	Df	7
Pérdida de integridad	Pi	3	Daño en la reputacion	Dr	9
Pérdida de disponibilidad	Pd	1	Incumplimiento	In	7
Pérdida de responsabilidad	Pr	9	Violación de privacidad	Vp	5

$$Impacto = \frac{(2+3+1+9)\ +\ (7+9+7+5)}{8} = \frac{15+28}{8} = \frac{43}{8} = 5.37$$

Ecuación 5.4.- Cálculo del Impacto Para la Vulnerabilidad de Perdías y Autenticación de Sesiones. (Autor)

Como se puede apreciar en la **Ecuación 5.4**, la estimación del impacto para la *Vulnerabilidad* de pérdida y autenticación de sesiones tiene un rango medio, esto según la tabulación de OWASP con un total de 5.37

5.3.2.- Explotando Vulnerabilidades de tipo XSS.

Siguiendo a línea de pruebas, se procede a explotar una de las *Vulnerabilidades* halladas durante la fase de escaneo por medio de la herramienta **ZAP**, como se muestra en la **Figura 5.9**, durante la fase de escaneo se encontró, que dos de las **URL** contenidas en el **SII** (para el banco de pruebas **SIItest**) eran vulnerables para ataques de tipo cross site scripting (**XSS**)**.**

Este tipo de *Vulnerabilidad* es común en **App.** de entorno Web, permitiendo a usuarios no autenticados la inyección de código JavaScript o similar, Según (OWASP, 2017) esta *Vulnerabilidad* se encuentra catalogada como el número tres entre las *Vulnerabilidades* más explotadas, otros autores como (Alonso Cebrián et al., 2014a) enfatizan la poca importancia que sectores orientados a la seguridad le han dado a este tema, debido que el ataque no compromete directamente al servidor, si no que el principal objetivo es el usuario, en el apartado **2.2.- Marco Teórico** se detalla este tipo de *Vulnerabilidad.*

En el banco de pruebas se presenta el escenario donde se secuestra una **URL** vulnerable, a la cual se le inserta código **PHP** como se ve en la **Tabla 5.11,** usando el navegador Firefox, caso en el que su función es crear una ventana de alerta como se puede observar en la **Figura 5.17**; dentro del código es posible escribir ciertas instrucciones con la finalidad de hacer creer a los usuarios que es una función normal del **SIItest** e introduzcan sus credenciales de acceso.

Tabla 5.11.- Código de Tipo PHP Para la Generación de Una Ventana de Alerta. (Autor)

Código de tipo PHP
""><script>alert(“teclea tu No de control y NIP”);</script>"

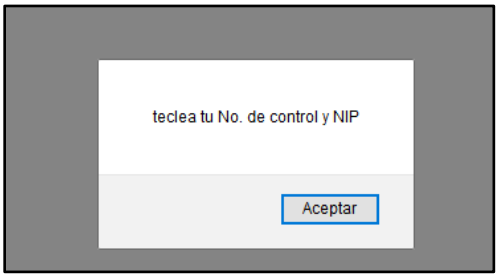


Figura 5.17.- Ventana Emergente Usando la URL Modificada. (Autor)

Como se mencionó con anterioridad, la explotación de esta *Vulnerabilidad* no representa un peligro directo al servidor, debido a que los cambios no permanecen dentro del **SIItest**, no obstante gracias a esta *Vulnerabilidad* es posible extraer información de los usuarios, debido que, al introducir las credenciales de acceso de un usuario legítimo, se visualizó que los datos de acceso se transmitían a la **URL** como se muestra en la **Figura 5.18** y se almacenaban en las *Cookie* del navegador como se puede observar en la **Figura 5.19**, esto representa un peligro debido a un atacante puede crear un histograma de usuarios con sus respectivas credenciales para posteriores ataques de mayor nivel

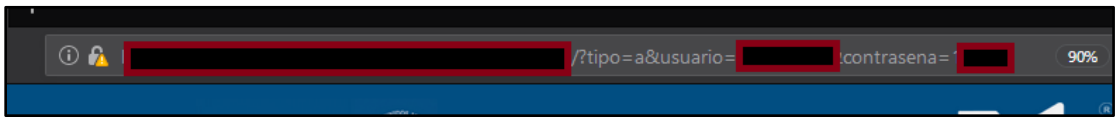


Figura 5.18.- Transición de la URL Con Información de las Credenciales de Acceso. (Autor)

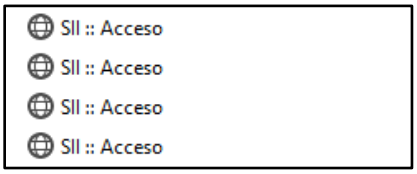


Figura 5.19.- Almacenamiento de las Credenciales de Acceso en las Cookies, en el Navegador Firefox. (Autor)

Realizándose algunas otras pruebas con la **URL** del **SIItest**, no fue posible obtener éxito para otros escenarios relacionados a la *Vulnerabilidad* de tipo **XSS**, sin embargo, debido a la naturaleza parecida entre los escenarios se halló una *Vulnerabilidad* de tipo **CSRF,** esta *Vulnerabilidad* se procederá a detallar en otro apartado.

5.3.2.1.- Identificación del riesgo para la Vulnerabilidad de XSS

Una vez que se visualizó la *Vulnerabilidad* y los efectos causados en el **SIItest,** se procede a la identificación del riesgo, siguiendo los pasos definidos en la metodología mostrada en el **Capítulo 4**, la cual dicta que es necesario identificar el tipo de riesgo. Concluyendo que la explotación de la *Vulnerabilidad* coincide para las de tipo **XSS**; afectando la capa de presentación, provocando que los usuarios que interactúan con el **SIItest** introduzcan sus datos para su recolección, en la **Tabla 5.12,** se presentan los aspectos generales de la *Vulneravilidad*.

Tabla 5.12.- Ataque de Tipo XSS y Sus Características. (Autor)

Ataque	Riesgo
XSS	El ataque permite a usuarios sin privilegios secuestrar sesiones de otros usuarios, alteración de la apariencia del sitio Web, inserción de código malicioso, la restricción de usuarios y la inserción de malware

5.3.2.2.- Establecer la probabilidad de ocurrencia para la Vulnerabilidad de tipo XSS.

Viendo el escenario y su método de explotación es posible determinar los agentes que amenazan al sistema por medio de los requisitos que los conforman, como se puede apreciar en la **Tabla 4.2**, la vulnerabilidad en el banco de pruebas afecta a usuarios desprevenidos o sin conocimiento de este tipo de temas. Estableciendo el análisis bajo esta línea, los factores de amenazas son las que se pueden apreciar en la **Tabla 5.13**, para posteriormente determinar sus valores según la tabulación de **OWASP** y establecer el cálculo.

Tabla 5.13.- Asignación de Valores Para los Agentes de Amenazas Para la Vulnerabilidad de Tipo XSS. (Autor)

Amenazas				
Tipo de taque	**Habilidades técnicas**	**Motivaciones**	**Oportunidad**	**Tamaño**
XSS	Usuarios avanzados en Computación (5)	Posible recompensa (4)	Sin acceso (9)	Socios (5)

Habiendo tabulado los agentes de amenazas dentro del escenario se procede a establecer los agentes para la Vulnerabilidad y su tabulación como se puede observar en la **Tabla 5.14.**

Tabla 5.14.- Asignación de Valores Para los Agentes de Vulnerabilidad para la Vulnerabilidad de Tipo XSS. (Autor)

Vulnerabilidad				
Ataque.	**Facilidad de descubrimiento.**	**Facilidad de explotación.**	**Conciencia o conocimiento.**	**Detección de intrusos.**
XSS	Difícil (3)	Fáciles (5)	Oculto (4)	No registrada (9)

Teniendo ambas tabulaciones se procede a establecer el cálculo de la seguridad del **SII** (en el banco de pruebas **SIItest**) bajo el escenario de **XSS**. Como se vio en el **Capítulo 4** se sigue la **Ecuación 5.5**, para el cálculo del nivel de seguridad en la probabilidad de impacto.

$$Probabilidad = \frac{(Ht + Mt + Op + Tm) + (Fd + Fe + Cc + Di)}{8}$$

Ecuación 5.5.- Fórmula Para el Cálculo de la Probabilidad de Ocurrencia en la Vulnerabilidad de Tipo XSS. (Autor)

Donde los valores para el cálculo se muestran en la **Tabla 5.15**,

Tabla 5.15.- Tabla de Tabulación en la Probabilidad de Ocurrencia en la Vulnerabilidad de Tipo XSS. (Autor)

Habilidades técnicas	Ht	5	Facilidad de descubrimiento	Fd	3
Motivaciones	Mt	4	Facilidad de explotación	Fe	5
Oportunidad	Op	9	Conciencia o conocimiento	Cc	4
Tamaño	Tm	5	Detección de intrusos	Di	9

$$Probabilidad = \frac{(5+4+9+5)+(3+5+4+9)}{8} = \frac{23+21}{8} = \frac{44}{8} = 5.5$$

Ecuación 5.6.- Cálculo de la Probabilidad de Ocurrencia Para la Vulnerabilidad de Tipo XSS. (Autor)

Como se observa en la **Ecuación 5.6**, la probabilidad de ocurrencia para la *Vulnerabilidad* de tipo **XSS**, encontrada en el **SIItest** da un grado Medio de explotación, según la tabulación de **OWASP**, con un rango total de 5.5.

5.3.2.3.- Estimación del impacto para la Vulnerabilidad de XSS.

En esta fase se determina el impacto que genera en la organización la explotación de la *Vulnerabilidad* de tipo **XSS**, para esto es necesario calcular el impacto técnico y el impacto en el negocio, cuantificando los agentes que intervienen, como se observa en la **Tabla 5.16** y **Tabla 5.17**, respectivamente; el resultado total contrastado con la probabilidad de ocurrencia determinara el nivel de seguridad que tiene en **SIItest** con esta *Vulnerabilidad*.

Tabla 5.16.- Asignación de Valores Para Calcular el Impacto Técnico en la Vulnerabilidad de Tipo XSS. (Autor)

Impacto Técnico				
Ataque	**Pérdida de confidencialidad**	**Pérdida de integridad**	**Pérdida de disponibilidad**	**Pérdida de responsabilidad**
XSS	Datos críticos mínimamente divulgados (6)	Datos mínimos, ligeramente corruptos (1)	Interrupción mínima de servicios secundarios (1)	Posiblemente rastreable (7)

Tabla 5.17.- Asignación de Valores Para Calcular el Impacto en el Negocio, en la Vulnerabilidad de Tipo XSS. (Autor)

Impacto en el negocio				
Ataque.	**Daño financiero.**	**Daño en la reputación.**	**Incumplimiento.**	**Violación de la privacidad.**
XSS	Menor que el costo de Corregir la Vulnerabilidad (1)	Daño mínimo (1)	Violación clara (5)	Cientos de personas (5)

Teniendo los valores para los agentes de impacto técnico e impacto en el negocio se procede a realizar el cálculo de la probabilidad para la estimación del impacto general para la *Vulnerabilidad* de **XSS** con la fórmula mostrada en **Ecuación 5.7**.

$$Impacto = \frac{(Pf + Pi + Pd + Pr) + (Df + Dr + In + Vp)}{8}$$

Ecuación 5.7.- Fórmula Para la Probabilidad de Impacto Para la Vulnerabilidad de Tipo XSS. (Autor)

Donde los valores para el cálculo se muestran en la **Tabla 5.18**, quedando tabuladas de la siguiente manera:

Tabla 5.18.- Tabla de Tabulación en la Probabilidad de Impacto Para la Vulnerabilidad de Tipo XSS. (Autor)

Pérdida de confiabilidad	Pf	6	Daño financiero	Df	1
Pérdida de integridad	Pi	1	Daño en la reputación	Dr	1
Pérdida de disponibilidad	Pd	1	Incumplimiento	In	5
Pérdida de responsabilidad	Pr	7	Violación de privacidad	Vp	5

$$Impacto = \frac{(6+1+1+7)\ +\ (1+1+5+5)}{8} = \frac{15+12}{8} = \frac{27}{8} = 3.37$$

Ecuación 5.8.- Cálculo del Impacto Para la Vulnerabilidad de Tipo XSS. (Autor)

Como se puede apreciar en la **Ecuación 5.8**, la estimación del impacto para la *Vulnerabilidad* de **XSS**, tiene un rango medio, esto según la tabulación de **OWASP** con un total de 3.37

5.3.3.- Explotando Vulnerabilidades de tipo CSRF.

Dentro del **SII** (en el banco de pruebas **SIItest**) se estableció una evaluación en seguimiento a la línea de investigación descubriendo que la *Vulnerabilidad* para ataques de falsificación de peticiones en sitios cruzados (**CSRF**), el cual pertenece a la lista de las principales *Vulnerabilidades* de **OWASP**, ocupando el octavo lugar en el Top ten de **OWASP** para el año 2017, no es explotable durante todo el tiempo que funciona el **SII** (en el banco de pruebas **SIItest**), debido que en periodos normales el **SII**. contiene la mayoría de sus módulos cerrados, dejándolos inaccesibles para usuarios de bajo perfil y para usuarios con alto perfil son autorizados en tiempos especiales.

Sin embargo, se planteó el escenario de una apertura de alta de materias en las cuales usuarios de bajo perfil hacen la selección para sus respectivos horarios, en otras palabras, se simuló un periodo de reinscripción, con la finalidad de ver el comportamiento del **SIItest** durante esta fase; descubriendo una anomalía.

Usando la herramienta **ZAP** de **OWASP** en su modo pasivo, se procede a hacer una inspección manual del paginado dentro del **SIItest**, debido a que la automatización solo cubre los parámetros básicos, en la **Figura 5.20,** se muestra la presentación inicial de la **App**. en donde se configura para la captura desde el navegador Firefox.

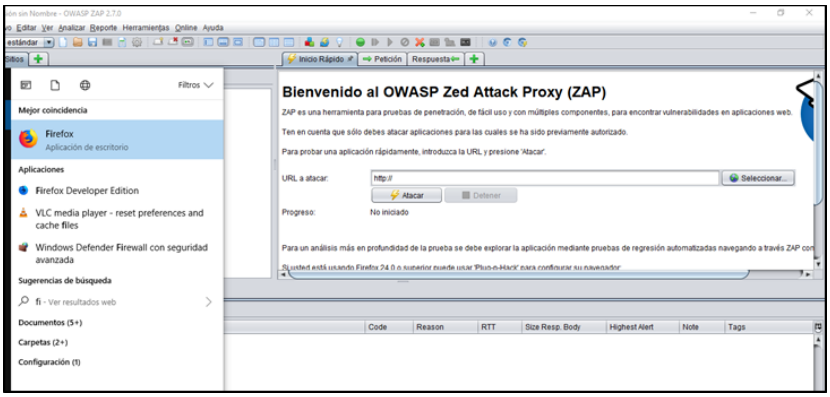


Figura 5.20.- Vista de la Interfaz de ZAP. (Autor)

A continuación, entrando a una cuenta de categoría baja, se simula el proceso de alta de materias, entrando al apartado de **selección de materias / periodo**, como se muestra en la **Figura 5.21**, hecho esto, aparecerá la retícula con las materias que pueden ser seleccionadas como se muestra en la **Figura 5.22.**

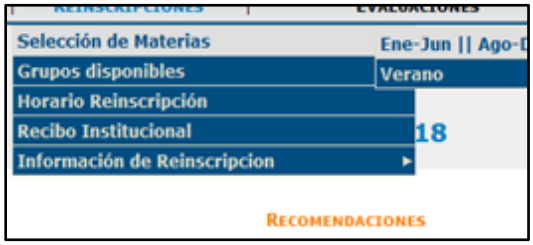


Figura 5.21.- Vista del Módulo Para Alta de Materias. (Autor)

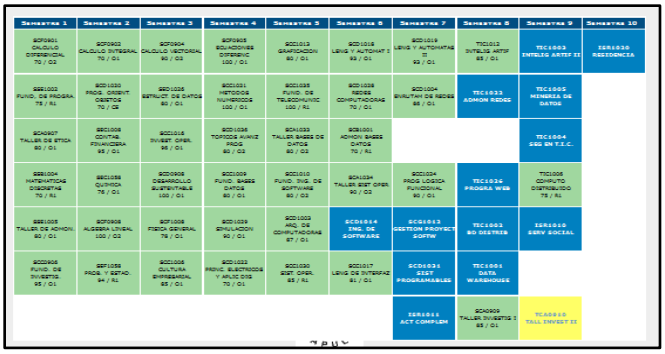


Figura 5.22.- Visualización de la Retícula Para Selección de Materias. (Autor)

Ya que se puede visualizar esta sección, se seleccionan las materias que se planean cursar dentro del siguiente periodo como se muestra en la **Figura 5.23.**

Figura 5.23.- Visualización de Materias a Seleccionar Por Paquete. (Autor)

Durante estas pruebas el **SIItest** respondió de manera normal, sin ningún tipo de problema, las consultas de alta de materias no es posible capturarlas o visualizarlas dentro de las *Cookies* del navegador o el *Servidor proxy*.

Caso contrario para la baja de materias, se descubrió que era posible obtener y capturar la consulta de la baja tanto en el *Servidor proxy* como en las *Cookies*, esto se puede observar en la **Figura 5.24** y en la **Figura 5.25** respectivamente.

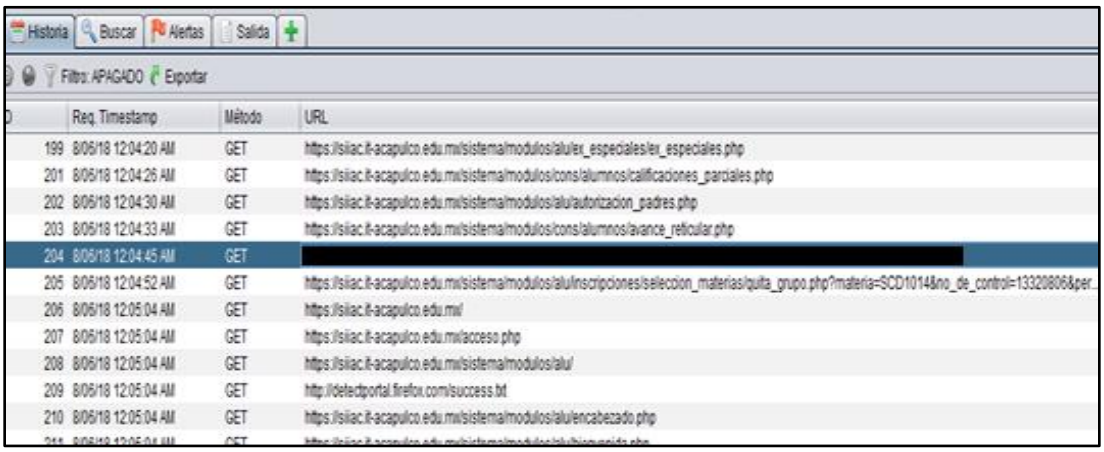


Figura 5.24.- Captura de la URL Para Baja de Materia en ZAP. (Autor)

Figura 5.25.- Captura de la URL Para Baja de Materia en las Cookies del Navegador Web. (Autor)

Al poder visualizar esta **URL** se crea una *Vulnerabilidad* de tipo **CSRF**, debido a que usuarios capacitados, pueden crear peticiones que el servidor toma como legítimas, permitiendo que usuarios de bajo perfil puedan borrar materias de su sesión, como de sesiones de otros alumnos sin la necesidad de ingresar a ellas, en la **Tabla 5.19,** se explica el funcionamiento de algunos parámetros que componen la **URL** y cómo afecta a la **BD**.

Tabla 5.19.- Estructura y Función de la URL Para Bajas. (Autor)

materia=xxx	Se selecciona el tipo de materia que se quiere borrar para esto es necesario visualizar la clave de la materia en la retícula
no_de_control=xxx	La consulta se asocia al número de control del alumno en esta parte es posible borrar materias de la sesión de otro alumno siempre y cuando cumpla con las características asociadas a la materia y grupo
periodo=xxx	Sección enlazada al periodo de alta de materias
grupo=xxx	Con esta sección es posible eliminar materias de un grupo en concreto, el cual es posible visualizarlos desde el tiempo de selección de materias o desde que se dan a conocer los paquetes para los horarios

Teniendo lo anteriormente descrito, se procede a explicar la transición en el proceso de la explotación de esta *Vulnerabilidad* por medio de un diagrama de flujo, ver la **Figura 5.26**.

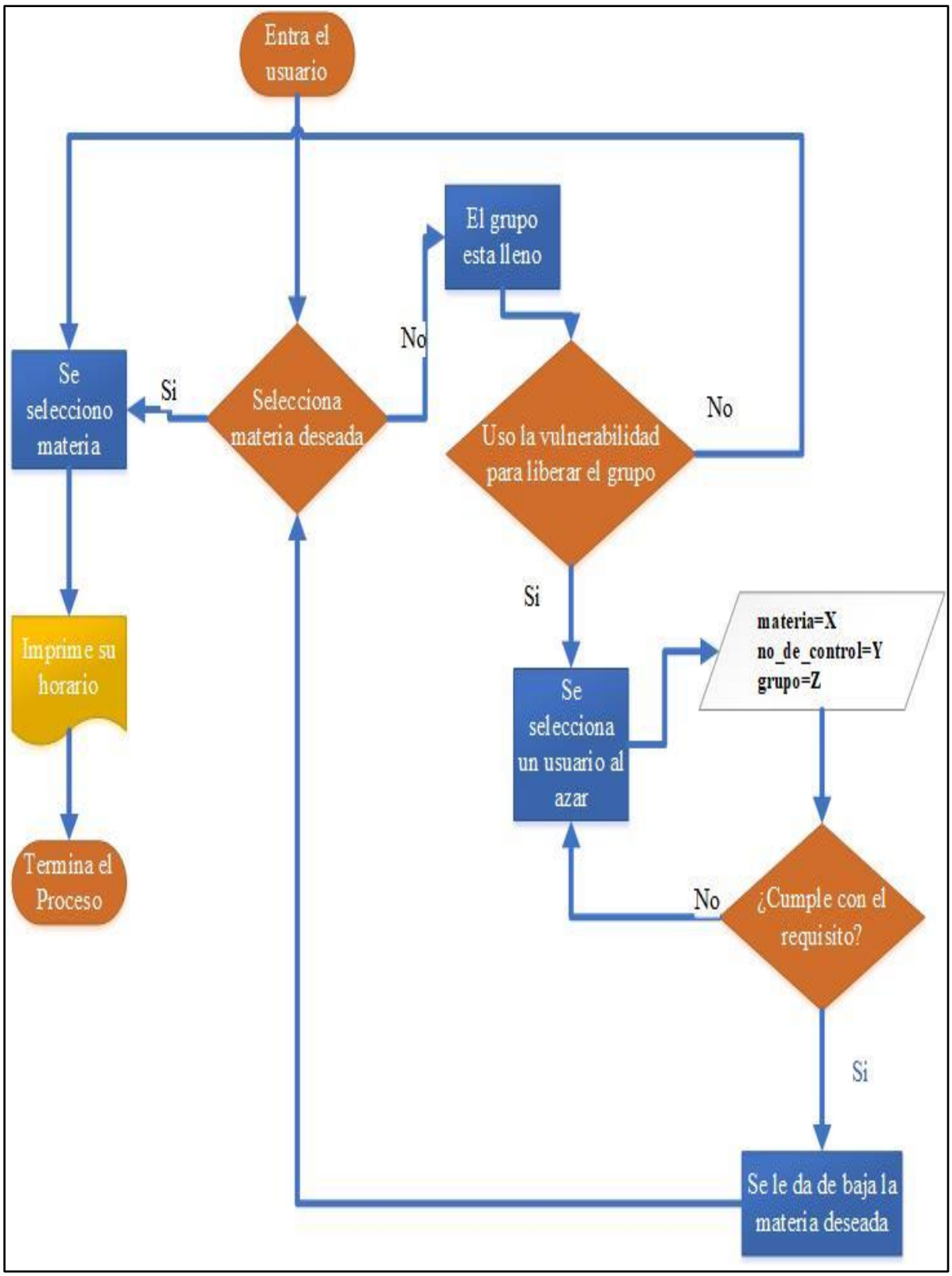


Figura 5.26.- Proceso de Explotación de la Vulnerabilidad. (Autor)

En el escenario contemplado se visualiza que usuarios no autorizados hacen cambios en la base de datos de los grupos mientras se hace el tiempo de alta de materias, esto quiere decir que, si el usuario no cumple con los requisitos o en dado caso el grupo al que desea inscribirse se encuentra saturado, este puede explotar la **URL** para borrar a un usuario que ya se encuentre inscrito siempre y cuando cumpla con los parámetros específicos expuestos en la **Tabla 5.19**, hecho esto aprovecha que el grupo se ha liberado para poder inscribirse en el grupo.

Para la organización esto supone un problema debido a que usuarios legítimos que estaban inscritos en el grupo pierden el lugar y su carga académica, generando molestias e inconvenientes con el personal administrativo debido al cambio no autorizado de materias.

5.3.3.1.- Identificación del riesgo para la Vulnerabilidad CSRF.

Una vez que se visualizó lo ya descrito se procede a la identificación del riesgo, siguiendo los pasos definidos en la metodología mostrada en el **Capítulo 4**, la cual dicta que es necesario identificar el tipo de riesgo.

Usando la clasificación de **OWASP** para las *Vulnerabilidades* de aplicaciones Web vistas en la **Tabla 2.1** se concluye que la explotación de la *Vulnerabilidad* coincide para las de tipo **CSRF**, en el **Capítulo 2** se detalla su clasificación y método de explotación, en la **Tabla 5.20** se clasifica el tipo de ataque y riesgo

Tabla 5.20.- Ataque CSRF y Sus Características. (Autor)

Ataque	Riesgo
CSRF	El ataque permite a usuarios sin privilegios secuestrar sesiones de otros usuarios, alteración de la apariencia del sitio Web, inserción de código malicioso, la restricción de usuarios y la inserción de malware

Las *Vulnerabilidades* se encuentran arraigadas dentro de la capa de presentación, en esta capa los usuarios interactúan con la **App.** para la recolección de información del usuario, mandarla a la capa de datos y recibir los resultados para generar una presentación final.

5.3.3.2.- Establecer la probabilidad de ocurrencia para la Vulnerabilidad de CSRF.

Observando el escenario y su método de explotación, es posible determinar los agentes que amenazan al sistema por medio de los requisitos que los conforman, como se puede apreciar en la **Tabla 4.2,** la *Vulnerabilidad* en el caso de estudio al cumplir con algunos requisitos específicos no lo hace explotable para todo tipo de personas.

Estableciendo el análisis bajo esta línea, los factores de amenazas son las que se pueden apreciar en la **Tabla 5.21 y Tabla 5.22**, para de esta manera determinar sus valores según la tabulación de **OWASP** y establecer el cálculo.

Tabla 5.21.- Asignación de Valores a los Agentes de Amenazas Para la Vulnerabilidad de Tipo CSRF. (Autor)

Amenazas				
Tipo de taque	**Habilidades técnicas**	**Motivaciones**	**Oportunidad**	**Tamaño**
CSRF	Usuarios avanzados en Computación (5)	Alta recompensa (9)	Acceso especial (4)	Usuarios autenticados (6)

Habiendo tabulado los agentes de amenazas dentro del escenario se procede a establecer los agentes para la *Vulnerabilidad* y su tabulación siguiendo la línea establecida por **OWASP**

Tabla 5.22.- Asignación de Valores a los Agentes de Vulnerabilidades de tipo SCRF. (Autor)

Vulnerabilidad				
Ataque.	**Facilidad de descubrimiento.**	**Facilidad de explotación.**	**Conciencia o conocimiento.**	**Detección de intrusos.**
CSRF	Fácil (7)	Fáciles (5)	Oculto (4)	Registrada sin revisión (8)

Teniendo ambas tabulaciones se procede a establecer el cálculo de la seguridad del **SII** en el escenario de **CSRF**. Para esto como se vio en el **Capítulo 4** se sigue la **Ecuación 5.**9,Ecuación 4.2 para el cálculo del nivel de seguridad en la probabilidad de impacto.

$$Probabilidad = \frac{(Ht + Mt + Op + Tm) + (Fd + Fe + Cc + Di)}{8}$$

Ecuación 5.9.- Fórmula Para el Cálculo de la Probabilidad de Ocurrencia, en la Vulnerabilidad de tipo CSRF. (Autor)

Donde los valores para el cálculo se muestran en la **Tabla 5.23**, quedando tabuladas de la siguiente manera:

Tabla 5.23.- Tabla de Tabulación en la Probabilidad de Ocurrencia en la Vulnerabilidad de Tipo CSRF.. (Autor)

Habilidades técnicas	Ht	5	Facilidad de descubrimiento	Fd	7
Motivaciones	Mt	9	Facilidad de explotación	Fe	5
Oportunidad	Op	4	Conciencia o conocimiento	Cc	4
Tamaño	Tm	6	Detección de intrusos	Di	8

$$Probabilidad = \frac{(5 + 9 + 4 + 6) + (7 + 5 + 4 + 8)}{8} = \frac{24 + 24}{8} = \frac{48}{8} = 6$$

Ecuación 5.10.- Cálculo de la Probabilidad de Ocurrencia Para la Vulnerabilidad de Tipo CSRF. (Autor)

Como se observa en la **Ecuación 5.10**, la probabilidad de ocurrencia para la *Vulnerabilidad* de tipo **CSRF** en el **SIItest** da un grado alto de explotación según la tabulación de **OWASP**, con un rango total de 6

5.3.3.3.- Estimación del impacto para la Vulnerabilidad CSRF.

En esta fase se determina el impacto que genera en la organización la explotación de la *Vulnerabilidad*, para esto es necesario calcular el impacto técnico como se muestra en la **Tabla**

5.24 y el impacto en el negocio como se muestra en la **Tabla 5.25**, cuantificando los agentes que intervienen según la tabulación de **OWASP** y dividirlo entre el total de los agentes que intervienen.

Tabla 5.24.- Asignación de Valores Para Calcular el Impacto Técnico en la Vulnerabilidad de Tipo CSRF. (Autor)

Impacto Técnico				
Ataque	**Pérdida de confidencialidad**	**Pérdida de integridad**	**Pérdida de disponibilidad**	**Pérdida de responsabilidad**
CSRF	Divulgación de datos críticos extensos (5)	Datos muy corruptos (7)	Interrupción extensa de servicios secundarios (5)	Posiblemente rastreable (7)

Tabla 5.25.- Asignación de Valores Para Calcular el Impacto en el Negocio Para la Vulnerabilidad de CSRF. (Autor)

Impacto en el negocio				
Ataque.	**Daño financiero.**	**Daño en la reputación.**	**Incumplimiento.**	**Violación de la privacidad.**
CSRF	Efecto significativo en el beneficio anual (7)	Daño de la marca / imagen (9)	Violación de alto perfil (7)	Miles de personas (5)

$$Impacto = \frac{(Pf + Pi + Pd + Pr) + (Df + Dr + In + Vp)}{8}$$

Ecuación 5.11.- Fórmula Para la Probabilidad de Impacto Para la Vulnerabilidad de CSRF. (Autor)

Donde los valores para el cálculo se muestran en la **Tabla 5.26**, quedando tabuladas de la siguiente manera:

Tabla 5.26.- Tabla de Tabulación en la Probabilidad de Impacto Para la Vulnerabilidad de Tipo CSRF. (Autor)

Pérdida de confiabilidad	Pf	5	Daño financiero	Df	7
Pérdida de integridad	Pi	7	Daño en la reputación	Dr	9
Pérdida de disponibilidad	Pd	5	Incumplimiento	In	7
Pérdida de responsabilidad	Pr	7	Violación de privacidad	Vp	5

$$Impacto = \frac{(5+7+5+7) \ + \ (7+9+7+5)}{8} = \frac{24+28}{8} = \frac{52}{8} = 6.5$$

Ecuación 5.12.- Cálculo del Impacto Para la Vulnerabilidad de Tipo CSRF. (Autor)

Como es posible observar en la **Ecuación 5.12,** el impacto total para la organización da un resultado alto según la tabulación de **OWASP**, dando un total de 6.5

Capítulo 6.- Resultados, Conclusiones y Recomendaciones

Dentro de este capítulo se plasman los resultados obtenidos durante la realización del proyecto, así como, las recomendaciones para **CECOMP**, las cuales, no solo reflejan las acciones para evitar problemas derivados de la explotación de *Vulnerabilidades* en el **SIItest** y sus herramientas de manejo de la información, refiriéndose a los servidores Web y de **BD**.

Dicho lo anterior, se aclara que los subtemas ya mencionados, son un reflejo derivado de la fase de desarrollo de la auditoria en busca de *Vulnerabilidades* realizada al **SIItest**, aplicación de entorno Web que se utilizó como banco de pruebas; teniendo en cuenta que esta aplicación es un clon del **SII**, toda vulnerabilidad hallada en el banco de pruebas debe hallarse por lógica en la original, todos los detalles que corresponde a la fase de desarrollo se encuentran plasmados en el **Capítulo 5.**

Por último, se resalta que las recomendaciones manejadas dentro de este capítulo están divididas en dos partes (recomendaciones del campo muestral y recomendaciones a futuro). Las recomendaciones del campo muestral, tienen como fin que el personal pertinente pueda tener registro y crear una bitácora del seguimiento en las mejoras a la aplicación; por consiguiente, las recomendaciones a futuro trata sobre puntos que no se tocaron en la investigación; pero se considera que representan un tema significativo para investigaciones y trabajos que puedan realizarse en un futuro.

6.1.- Resultados

Visualizando los escenarios, en el que los ataques al **SII** (para el banco de pruebas **SIItest**), generaron resultados exitosos en el hallazgo de vulnerabilidades y su explotación, permitiendo generar una cuantificación de los niveles de riesgo, se procede a realizar una tabla de concentración, conteniendo las tabulaciones manejadas en la metodología de calificación de riesgo de **OWASP.** La metodología mencionada se encuentra detallada en el **Capítulo 4**

En la **Tabla 6.1** se muestra la clasificación de los niveles de seguridad que tiene el **SII** (en el banco de pruebas **SIItest)** dentro de los diferentes escenarios en que se generó la explotación de las *Vulnerabilidades*.

La visualización de los niveles debe permitir al personal encargado determinar que *Vulnerabilidades* deben ser resueltas y en que prioridad según su nivel de riesgo. Como se puede observar desde la fase de escaneo y desarrollo del proyecto las vulnerabilidades encontradas aparecen en el **SII** (para el banco de pruebas **SIItest**), concuerdan con las líneas de investigación de **OWASP** como se menciona en el **Capítulo 2** y resaltando que muchas de estas *Vulnerabilidades* no fue posibles explotarlas dentro del banco de pruebas, mencionando en especial las de tipo inyección SQL que a pesar de aparecer en la fase de escaneo, no se pudo concretar un escenario óptimo para la explotación, teniendo esto en cuenta se concluye que el SII cumple con los medios para protegerse de esta *Vulnerabilidad*

Tabla 6.1.- Tabla General de Resultados de las Vulnerabilidades Halladas en el SIItest. (Autor)

<table>
<tr><th colspan="6">Tabla de resultados para la vulnerabilidad de autenticación y gestión de sesiones</th></tr>
<tr><td rowspan="13">Autenticación y gestión de sesiones</td><td colspan="5">Agentes de Amenazas.</td></tr>
<tr><td colspan="2">Habilidades técnicas (3)</td><td>Motivaciones (4)</td><td>Oportunidad (7)</td><td>Tamaño (6)</td></tr>
<tr><td colspan="5">Agentes de Vulnerabilidad</td></tr>
<tr><td colspan="2">Facilidad de descubrimiento (9)</td><td>Facilidad de explotación (5)</td><td>Conciencia o conocimiento (4)</td><td>Detección de intruso (9)</td></tr>
<tr><td colspan="5">Agentes de impacto Técnico.</td></tr>
<tr><td colspan="2">Pérdida de Confidencialidad (2)</td><td>Pérdida de integridad (3)</td><td>Pérdida de disponibilidad (1)</td><td>Pérdida de responsabilidad (9)</td></tr>
<tr><td colspan="5">Agentes de Impacto en el Negocio.</td></tr>
<tr><td colspan="2">Daño financiero (7)</td><td>Daño en la reputación (9)</td><td>Incumplimiento (7)</td><td>Violación de la privacidad (5)</td></tr>
<tr><td colspan="5">Gravedad del riesgo general = Probabilidad X Impacto</td></tr>
<tr><td rowspan="4">Impacto = 5.37</td><td>Alto.</td><td>Medio</td><td>Alto</td><td>Critico.</td></tr>
<tr><td>Medio.</td><td>Bajo.</td><td>Medio.</td><td>Alto.</td></tr>
<tr><td>Bajo.</td><td>Ninguno.</td><td>Bajo.</td><td>Medio.</td></tr>
<tr><td></td><td>Bajo.</td><td>Medio.</td><td>Alto.</td></tr>
<tr><td colspan="5">Probabilidad = 5.87</td></tr>
<tr><td colspan="5">Resultados:</td></tr>
<tr><td colspan="2">Impacto estimado = 5.37</td><td colspan="2">Probabilidad estimada = 5.87</td><td>Índice de riesgo medio</td></tr>
</table>

<table>
<tr><th colspan="6">Tabla de resultados para la vulnerabilidad de tipo XSS</th></tr>
<tr><td rowspan="14">XSS</td><td colspan="5">Agentes de amenazas:</td></tr>
<tr><td colspan="2">Habilidades técnicas (5)</td><td>Motivaciones (4)</td><td>Oportunidad (9)</td><td>Tamaño (5)</td></tr>
<tr><td colspan="5">Agentes de vulnerabilidad:</td></tr>
<tr><td colspan="2">Facilidad de descubrimiento (3)</td><td>Facilidad de explotación (5)</td><td>Conciencia o conocimiento (4)</td><td>Detección de intrusos (9)</td></tr>
<tr><td colspan="5">Agentes de impacto técnico:</td></tr>
<tr><td colspan="2">Pérdida de confidencialidad (6)</td><td>Pérdida de integridad (1)</td><td>Pérdida de disponibilidad (1)</td><td>Pérdida de responsabilidad (7)</td></tr>
<tr><td colspan="5">Agentes de impacto en el negocio:</td></tr>
<tr><td colspan="2">Daño financiero (1)</td><td>Daño en la reputación (1)</td><td>Incumplimiento (5)</td><td>Violación de la privacidad (5)</td></tr>
<tr><td colspan="5">Gravedad del riesgo general = Probabilidad X Impacto</td></tr>
<tr><td rowspan="4">Impacto = 3.37</td><td>Alto</td><td>Medio</td><td>Alto</td><td>Critico</td></tr>
<tr><td>Medio</td><td>Bajo</td><td>Medio</td><td>Alto</td></tr>
<tr><td>Bajo</td><td>Ninguno</td><td>Bajo</td><td>Medio</td></tr>
<tr><td></td><td>Bajo</td><td>Medio</td><td>Alto</td></tr>
<tr><td colspan="5">Probabilidad = 5.5</td></tr>
<tr><td colspan="5">Resultados:</td></tr>
<tr><td colspan="2">Impacto estimado = 3.37</td><td colspan="2">Probabilidad estimada = 5.5</td><td>Índice de riesgo medio</td></tr>
</table>

Tabla de Resultados para la vulnerabilidad de tipo CSRF					
	Agentes de amenazas.				
	Habilidades técnicas (5)		Motivaciones (9)	Oportunidad (4)	Tamaño (6)
	Agentes de Vulnerabilidad.				
	Facilidad de descubrimiento (7)		Facilidad de explotación (5)	Conciencia o Conocimiento (4)	Detección de intrusos (8)
	Agentes de impacto Técnico.				
	Pérdida de Confidencialidad (5)		Pérdida de integridad (7)	Pérdida de disponibilidad (5)	Pérdida de responsabilidad (7)
	Agentes de Impacto en el Negocio.				
	Daño financiero (7)		Daño en la reputación (9)	Incumplimiento (7)	Violación de la privacidad (5)
	Gravedad del riesgo general = Probabilidad X Impacto				
	Impacto = 6.5	**Alto.**	Medio	Alto.	Critico.
		Medio.	Bajo.	Medio.	Alto.
		Bajo.	Ninguno.	Bajo.	Medio.
			Bajo.	**Medio.**	**Alto.**
	Probabilidad. = 6				
	Resultados:				
	Impacto estimado = 6.5		Probabilidad estimada = 6		Índice de riesgo alto

6.1.1.- Definir qué arreglar.

Teniendo los resultados recabados de las pruebas, se genera un concentrado de los resultados como se puede observar en la **Tabla 6.1** con el índice de explotación de las *Vulnerabilidades* encontradas siendo catalogadas, partiendo de las que se consideren de menor impacto hasta la de mayor impacto, esto tiene la finalidad de presentar un panorama amplio para la toma de decisiones, considerando que las de mayor impacto requieren una atención inmediata en el banco de pruebas resultando la de mayor rango la *Vulnerabilidad* de tipo **CSRF**.

6.2.- Conclusiones

Como se ha demostrado a lo largo del presente escrito, la implementación de una auditoria por medio de *Pentesting* al **SII** (en el banco de pruebas **SIItest**), demostró que la aplicación es explotable en diferentes escenarios, para algunas de las *Vulnerabilidades* en la línea de investigación de **OWASP**.

Indicando en base a los resultados, el éxito en la promulgación de este tipo de actividades, generando un antecedente en el campo de la seguridad informática, en el centro de cómputo (**CECOMP**), en base a los resultados obtenidos, puede determinar bajo que medios es posible mitigar las Vulnerabilidades encontradas en el **SII**.

Teniendo en cuenta que dentro de la línea de investigación se logró demostrar la explotación de algunas de las vulnerabilidades preocupantes pare el centro de cómputo

(**CECOMP**), dando positivo en cuatro, según los resultados arrojados en la fase de escaneo, sin embargo, solo se logró192.168. demostrar tres de estas vulnerabilidades, como se puede observar en el **Capítulo 5**. De este modo se concluye que el **SII** no cuenta con una seguridad fiable en términos de vulnerabilidades para aplicaciones de entorno Web. Reiterando la importancia que representa al departamento ya mencionado la visualización de los resultados, con el fin de cumplir el objetivo de mejorar la seguridad en el **SII**.

6.3.- Recomendaciones.

En el presente apartado se presentan algunas recomendaciones a considerar y dirigidas al centro de cómputo (**CECOMP**), sin descartar que puedan ser usadas como marco referencial para personas orientadas en el desarrollo de aplicaciones Web seguras.

6.3.1.-Recomendaciones en el campo muestral.

Algunas recomendaciones relacionadas para mejorar la seguridad al **SII** es la implementación de algunos elementos divididos por el tipo de vulnerabilidad encontrada en el **SIItest** (**App.** que sirvió como banco de pruebas), las cuales se mencionarán a continuación:

- **Autenticación y gestión de sesiones**: La *Vulnerabilidad* como se pudo apreciar en el **Capítulo 5**, dentro del **SII** (en el banco de pruebas **SIItest**) muestra un índice aceptable de protección debido al certificado de seguridad, lo cual complica el uso de sniffeo a nivel de red, sin embargo, tiene un nivel alto de explotación en la *Vulnerabilidad* de ataque por

fuerza bruta, esto es debido a que el índice de combinaciones es muy bajo permitiendo solo $10^4 = 10000$, tipos de combinaciones de tipo numérico, complicando de mayor manera el índice de combinaciones posibles para una **App.** de su tipo esto representa un problema debido que con la tecnología actual en procesos computacionales no representa un problema.

Teniendo en cuenta esto se sugiere aumentar el índice de combinaciones posibles y pasar de un tipo numérico a uno alfanumérico, o bien implementar el uso de CAPTCHA, para impedir la automatización en las consultas de las claves a un usuario, mencionado que esta opción es una de las más usadas en la actualidad para este tipo de *Vulnerabilidad*.

- **Vulnerabilidad de tipo XSS:** Este tipo de *Vulnerabilidad* encontrada en una **URL** especifica dentro del banco de pruebas, representa un riesgo inminente a los usuarios al combinarla con ingeniería social, sin menospreciar que a pesar que la vulnerabilidad es muy evidente, no se puede descartar su solución debido a que nunca faltara un usuario desprevenido.

  Para este tipo de vulnerabilidad se recomienda ampliar el rango de la strip_tag, sistema que se usa en la mitigación de este tipo de vulnerabilidad intentando devolver un string con todos los bytes NULL y las etiquetas **HTML** y **PHP** retirados de un str dado, como se observa en el **SII** (en el banco de pruebas **SIItest**) cuenta con este tipo de seguridad, sin embargo, al subir el nivel de explotación se descubrió una flaqueza este proceso se detalla en su apartado dentro del **Capítulo 5**

- **Vulnerabilidad de tipo SCRF:** La Vulnerabilidad de este tipo represento un índice alto de riesgo, según lo visto en su apartado en el **Capítulo 5,** a pesar de que solo es explotable durante un tiempo específico, el nivel de problemáticas generadas por esta Vulnerabilidad repercute significativamente en la integridad económica y la reputación del negocio.

  Se sugiere el uso de Antiforgery Token (Tokens anti falsificacion), El cliente solicita una página HTML que contiene un formulario, después el servidor incluye dos *Token* en la respuesta (un *Token* se envía como una cookie, mientras que el otro se coloca en un campo de formulario oculto), cuando un usuario envíe el formulario, debe enviar ambos *Token* de vuelta al servidor para que el cliente envíe el *Token* de la *cookie* como una *cookie* enviando el *Token* del formulario con los datos, si una solicitud no incluye ambos *Token*, el servidor rechazará la solicitud.

6.3.2.- Recomendaciones a futuro.

Es recomendable que se consideren pruebas a nivel de red dentro del departamento, si bien no se abordó dentro de la tesis debido a que no competía dentro de la línea de investigación, si se concluyó que el servidor respondiera con credenciales del **SO** y sus versiones, no representa un nivel de seguridad favorable, para evitar estos problemas convendría promover el uso de *Firewall* dedicado exclusivo para los servidores.

Otra de las recomendaciones a considerar podría ser la estandarización del **SII** en base a una ISO o usando el estándar de ISECOM las cuales tienen abarcan temas que van desde el nivel de red hasta el de aplicativo.

Siglario.

B	
BD	Bases de datos
C	
CECOMP	Centro de cómputo
CSRF	Cross Site Request Forgery (falsificación de petición en sitios cruzados)
CVSS	Common Vulnerability Scoring System (sistema común de puntuación de vulnerabilidades)
D	
DNS	Domain Name System (sistema de nombres de dominio)
E	
EVSA	Estándar de Verificación de Seguridad en Aplicaciones
F	
FBI	Federal Bureau of Investigation (Oficina Federal de Investigación)
FIRST	Forum of Incident Response and Security Teams (Foro de Respuesta a Incidentes y Equipos de Seguridad)
FOSS	Free and Open Source Software (software libre y de código abierto)
H	
HTML	HyperText Markup Language (lenguaje de marcas de hipertexto)
HTTP	Hypertext Transfer Protocol (protocolo de transferencia de hipertexto)

HTTPS	Hypertext Transport Protocol Secure (protocolo seguro de transferencia de hipertexto)
I	
ICMP	Internet Control Message Protocol (protocolo de mensajes de control de internet)
INCIBE	Instituto Nacional de Ciberseguridad de España
IP	Internet protocol (Protocolo de Internet)
ITA	Instituto Tecnológico de Acapulco
ISECOM	Institute for Security and Open Methodologies (Instituto de Seguridad y Metodologías Abiertas.)
ISO	International Organization for Standardization (Organización Internacional de Normalización)
L	
LAN	Local Área Network (red de área local)
O	
OWASP	Open Web Aplication Security Proyect
P	
PHP	Hypertext Preprocessor (preprocesador de hipertexto)
S	
SDLC	Systems Development Life Cycle (siclo de vida del Software)
SGBD	Sistema Gestor de Bases de Datos
SII	Sistema Integral de Información
SO	Sistema Operativo
SQL	Structured Query Language (lenguaje de consulta estructurada)

T	
TCP	Transmission Control Protocol (Protocolo de control de transmisión)
TIC	Tecnologías de la Información y la Comunicación
U	
UIT	Unión Internacional de Telecomunicaciones
UNAM	Universidad Nacional Autónoma de México
URL	Uniform Resource Locator (Localizador Uniforme de Recursos)
USB	Universal Serial Bus
W	
WAF	Web Application Firewall (firewall de aplicación Web)
X	
XSS	Cross Site Scrpting (ejecución de comandos en sitios cruzados)

Glosario.

Allen, L., Heriyanto, T., & Shakeel, A. (2017). *Kali Linux Revealed - Mastering the Penetration Testing Distribution*. Cornelius, USA.

Alonso Cebrián, J. M., Guzmán Sacristán, A., Laguna Durán, P., & Martín Bailón, A. (2014a). Ataques a aplicaciones web. *Universitat Oberta de Catalunya*, 74.

Alonso Cebrián, J. M., Guzmán Sacristán, A., Laguna Durán, P., & Martín Bailón, A. (2014b). Ataques a BB. DD., SQL Injection. *Universitat Oberta de Ctalunya*, 68.

Borghello, C. (2009). El arma infalible : la Ingeniería Social. *Eset-La*, (619), 1–7. Recuperado a partir de http://www.eset-la.com/pdf/prensa/informe/arma_infalible_ingenieria_social.pdf

CSIRT-cv. (2018). *Nmap 6: Listado de comandos*. Valencia. Recuperado a partir de http://www.csirtcv.gva.es

Florêncio, D., Herley, C., & Coskun, B. (2007). Do strong web passwords accomplish anything? *Proceedings of the 2nd USENIX workshop on Hot topics in security (HOTSEC'07)*, 6. Recuperado a partir de http://portal.acm.org/citation.cfm?id=1361419.1361429

Fonseca Romero, J. C. (2015). Diseño de un Ambiente Simulado para Seguridad de la Información. *Revista Ciencia, Innovación y Tecnología (RCIYT), II*, 115–123.

Fonseca Romero, J. C. (2017). *Diseño e implementación de sistema informático para entrenamiento en test de intrusión*. Universidad Internacional de La Rioja.

Gómez Vieites, A. (2012). La lucha contra el ciberterrorismo y los ataques informáticos. *Memorias X Reunión Española sobre Criptología y Seguridad de la Información*. Recuperado a partir de http://www.edisa.com/wp-content/uploads/2014/08/La_lucha_contra_el_ciberterrorismo_y_los_ataques_informaticos.

pdf

Harshavardhan, M., Vinay Reddy, M., Chintalapudi, S., & Viswanatham, M. (2018). Cryptanalysis of Secure Hash Password Technique (CSHPT) in Linux. *IADS*, *01*(02), 1–4.

IBM. (2014). IBM Knowledge Center - Arquitecturas de tres niveles. Recuperado el 5 de noviembre de 2018, a partir de https://www.ibm.com/support/knowledgecenter/es/SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/covr_3-tier.html

INCIBE. (2018). Acerca de Nosotros. Recuperado el 18 de junio de 2018, a partir de https://www.incibe.es/

Jiménez, C. J. (2016). *Seguridad en redes y sistemas: Técnicas y conceptos sobre hacking y pentesting*. (P. S. Cristina, Ed.), *Seguridad Informática*. Catalunya, España. Recuperado a partir de http://openaccess.uoc.edu/webapps/o2/bitstream/10609/52944/9/cjmenezTFG0616memoria.pdf

Lyon, G. (2016). Nmap: the network mapper-free security scanner. Recuperado el 8 de noviembre de 2018, a partir de https://nmap.org/

Meucci, M., & Muller, A. (2014). *Testing Guide 4.0* (Vercion 4.). Open Web Application Security Project.

Mitnick, K. D., & Simon, W. L. (2007). *El arte de la intrusión: cómo ser hacker o evitarlos,. Historia* (Primera Ed). México, D.f.: RA-MA.

OSCP. (2018). What is Kali Linux ? – Kali Linux. Recuperado el 23 de noviembre de 2018, a partir de https://docs.kali.org/introduction/what-is-kali-linux

OWASP. (2008). Static Code Analysis --- The Open Web Application Security Project.

Recuperado a partir de https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project

OWASP. (2017). OWASP Top 10 - 2017 the Ten Most Crit ical Web Application Security Risks. *OWASP*, 20.

OWASP. (2018a). OWASP Risk Rating Methodology - OWASP. Recuperado el 26 de julio de 2018, a partir de https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

OWASP. (2018b). OWASP Zed Attack Proxy Project - OWASP. Recuperado el 9 de noviembre de 2018, a partir de https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project#tab=Main

Pérez, I. (2015). ¿En qué consiste la vulnerabilidad Cross Site Request Forgery (CSRF)? Recuperado el 4 de diciembre de 2018, a partir de https://www.welivesecurity.com/la-es/2015/04/21/vulnerabilidad-cross-site-request-forgery-csrf/

Ramírez Castro, A. (2012). Riesgo Tecnológico y su Impacto Para Las Organizaciones Parte l. ..*Seguridad Cultura de prevención para TI*, 13–16.

Ramírez R., G., & Álvarez D., E. (2003). Auditoría a La Gestión De Las Tecnologías Y Sistemas De Informacion. *Industrial Data*, *6*, 90–102. Recuperado a partir de http://www.redalyc.org/articulo.oa?id=81606114

Salazar, E. (2015). Pruebas de Seguridad en aplicaciones web segun OWASP, 9–128. Recuperado a partir de https://www.owasp.org/images/2/2f/OWASP_SUSCERTE.pdf

Sriphum, W., Chomsiri, T., Attanak, P., & Noitarong, P. (2011). SQL Injection Protector. *International Conference on Modeling, Simulation and Control*, *10*, 7–11.

UIT. (2018). Sobre la Unión Internacional de Telecomunicaciones (UIT). Recuperado el 18 de junio de 2018, a partir de https://www.itu.int/es/about/Pages/default.aspx

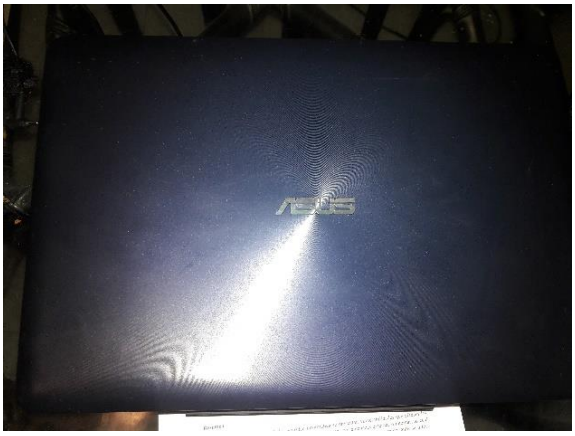
UNAM-CERT. (2015). ¿Qué es y cómo opera un ataque de Cross-Site Scripting (XSS). Recuperado el 28 de junio de 2018, a partir de https://www.seguridad.unam.mx/historico/documento/index.html-id=35

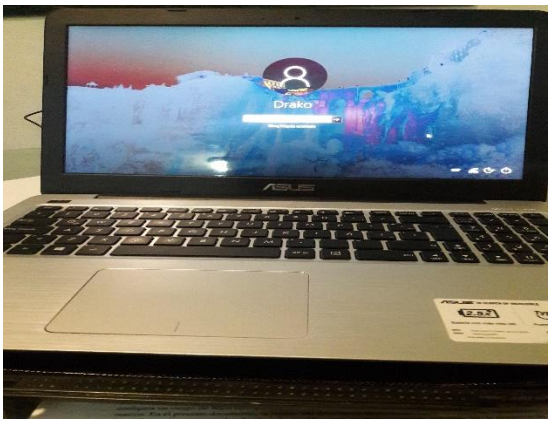
Villalobos Murillo, J. (2012). Principios Básicos de Seguridad en Bases de Datos. ..*Seguridad Cultura de prevención para TI, 12*, 1–5. Recuperado a partir de http://revista.seguridad.unam.mx/numero-12/principios-basicos-de-seguridad-en-bases-de-datos

VMware, I. (2018). Descargar VMware Workstation Player | VMware | MX. Recuperado el 5 de octubre de 2018, a partir de https://www.vmware.com/mx/products/workstation-player/workstation-player-evaluation.html

Anexos

Anexo 1: PC - ASUS X556U

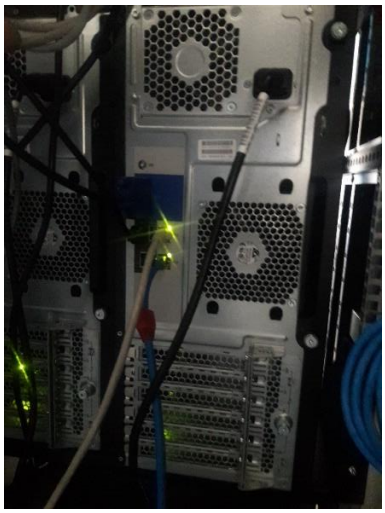




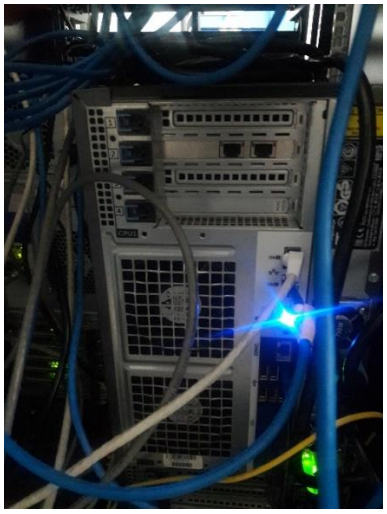
Anexo 2.- PC – Notebook Samsung 14 Np270.

Anexo 3.- Servidor - Dell PowerEdge R715 de 2U

Anexo 4.- Servidor HP Proliant ML110 Gen9



Anexo 5.- Servidor en Torre PowerEdge T620



Referencias.

Allen, L., Heriyanto, T., & Shakeel, A. (2017). *Kali Linux Revealed - Mastering the Penetration Testing Distribution*. Cornelius, USA.

Alonso Cebrián, J. M., Guzmán Sacristán, A., Laguna Durán, P., & Martín Bailón, A. (2014a). Ataques a aplicaciones web. *Universitat Oberta de Catalunya*, 74.

Alonso Cebrián, J. M., Guzmán Sacristán, A., Laguna Durán, P., & Martín Bailón, A. (2014b). Ataques a BB. DD., SQL Injection. *Universitat Oberta de Ctalunya*, 68.

Borghello, C. (2009). El arma infalible : la Ingeniería Social. *Eset-La*, (619), 1–7. Recuperado a partir de http://www.eset-la.com/pdf/prensa/informe/arma_infalible_ingenieria_social.pdf

CSIRT-cv. (2018). *Nmap 6: Listado de comandos*. Valencia. Recuperado a partir de http://www.csirtcv.gva.es

Florêncio, D., Herley, C., & Coskun, B. (2007). Do strong web passwords accomplish anything? *Proceedings of the 2nd USENIX workshop on Hot topics in security (HOTSEC'07)*, 6. Recuperado a partir de http://portal.acm.org/citation.cfm?id=1361419.1361429

Fonseca Romero, J. C. (2015). Diseño de un Ambiente Simulado para Seguridad de la Información. *Revista Ciencia, Innovación y Tecnología (RCIYT), II*, 115–123.

Fonseca Romero, J. C. (2017). *Diseño e implementación de sistema informático para entrenamiento en test de intrusión*. Universidad Internacional de La Rioja.

Gómez Vieites, A. (2012). La lucha contra el ciberterrorismo y los ataques informáticos. *Memorias X Reunión Española sobre Criptología y Seguridad de la Información*. Recuperado a partir de http://www.edisa.com/wp-content/uploads/2014/08/La_lucha_contra_el_ciberterrorismo_y_los_ataques_informaticos.

pdf

Harshavardhan, M., Vinay Reddy, M., Chintalapudi, S., & Viswanatham, M. (2018). Cryptanalysis of Secure Hash Password Technique (CSHPT) in Linux. *IADS*, *01*(02), 1–4.

IBM. (2014). IBM Knowledge Center - Arquitecturas de tres niveles. Recuperado el 5 de noviembre de 2018, a partir de https://www.ibm.com/support/knowledgecenter/es/SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/covr_3-tier.html

INCIBE. (2018). Acerca de Nosotros. Recuperado el 18 de junio de 2018, a partir de https://www.incibe.es/

Jiménez, C. J. (2016). *Seguridad en redes y sistemas: Técnicas y conceptos sobre hacking y pentesting*. (P. S. Cristina, Ed.), *Seguridad Informática*. Catalunya, España. Recuperado a partir de http://openaccess.uoc.edu/webapps/o2/bitstream/10609/52944/9/cjmenezTFG0616memoria.pdf

Lyon, G. (2016). Nmap: the network mapper-free security scanner. Recuperado el 8 de noviembre de 2018, a partir de https://nmap.org/

Meucci, M., & Muller, A. (2014). *Testing Guide 4.0* (Vercion 4.). Open Web Application Security Project.

Mitnick, K. D., & Simon, W. L. (2007). *El arte de la intrusión: cómo ser hacker o evitarlos,. Historia* (Primera Ed). México, D.f.: RA-MA.

OSCP. (2018). What is Kali Linux ? – Kali Linux. Recuperado el 23 de noviembre de 2018, a partir de https://docs.kali.org/introduction/what-is-kali-linux

OWASP. (2008). Static Code Analysis --- The Open Web Application Security Project.

Recuperado a partir de https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project

OWASP. (2017). OWASP Top 10 - 2017 the Ten Most Crit ical Web Application Security Risks. *OWASP*, 20.

OWASP. (2018a). OWASP Risk Rating Methodology - OWASP. Recuperado el 26 de julio de 2018, a partir de https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

OWASP. (2018b). OWASP Zed Attack Proxy Project - OWASP. Recuperado el 9 de noviembre de 2018, a partir de https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project#tab=Main

Pérez, I. (2015). ¿En qué consiste la vulnerabilidad Cross Site Request Forgery (CSRF)? Recuperado el 4 de diciembre de 2018, a partir de https://www.welivesecurity.com/la-es/2015/04/21/vulnerabilidad-cross-site-request-forgery-csrf/

Ramírez Castro, A. (2012). Riesgo Tecnológico y su Impacto Para Las Organizaciones Parte l. ..*Seguridad Cultura de prevención para TI*, 13–16.

Ramírez R., G., & Álvarez D., E. (2003). Auditoría a La Gestión De Las Tecnologías Y Sistemas De Informacion. *Industrial Data*, *6*, 90–102. Recuperado a partir de http://www.redalyc.org/articulo.oa?id=81606114

Salazar, E. (2015). Pruebas de Seguridad en aplicaciones web segun OWASP, 9–128. Recuperado a partir de https://www.owasp.org/images/2/2f/OWASP_SUSCERTE.pdf

Sriphum, W., Chomsiri, T., Attanak, P., & Noitarong, P. (2011). SQL Injection Protector. *International Conference on Modeling, Simulation and Control*, *10*, 7–11.

UIT. (2018). Sobre la Unión Internacional de Telecomunicaciones (UIT). Recuperado el 18 de junio de 2018, a partir de https://www.itu.int/es/about/Pages/default.aspx

UNAM-CERT. (2015). ¿Qué es y cómo opera un ataque de Cross-Site Scripting (XSS). Recuperado el 28 de junio de 2018, a partir de https://www.seguridad.unam.mx/historico/documento/index.html-id=35

Villalobos Murillo, J. (2012). Principios Básicos de Seguridad en Bases de Datos. ..*Seguridad Cultura de prevención para TI*, *12*, 1–5. Recuperado a partir de http://revista.seguridad.unam.mx/numero-12/principios-basicos-de-seguridad-en-bases-de-datos

VMware, I. (2018). Descargar VMware Workstation Player | VMware | MX. Recuperado el 5 de octubre de 2018, a partir de https://www.vmware.com/mx/products/workstation-player/workstation-player-evaluation.html

Formato de tesis.

El presente escrito contiene los siguientes formatos para su diseño de maquetación siguiendo el estilo de la American Psychological Association mejor conocidas como normas APA. Conteniendo los siguientes elementos:

- **Tipo de papel:** tamaño carta estándar.
- **Fuente:** Times New Román a 12 puntos. 14puntos en los subtítulos y 16 puntos en títulos principales
- **Encabezado:** sin encabezado.
- **Numeración:** romanos en las hojas preliminares, arábigos en las de contenido todas ubicadas en la esquina superior izquierda.
- **Sangrías:** 1.27 cm al inicio de todos los párrafos.
- **Alineación:** justificado, aunque la norma dice que la alineación debe de ser ala izquierda se considera que justificado da una mejor presentación.
- **Márgenes:** 2.54 cm en todos los márgenes de la página.

Para los niveles de titulación se establece los siguientes formatos los cuales ayudan en la organización y división del capitulado.

- **Nivel 1:** el encabezado debe de ir centrado en negrilla con mayúsculas y minúsculas.
- **Nivel 2:** debe de ir alineado a la izquierda en negrilla con mayúsculas y minúsculas.

- **Nivel 3:** el encabezado debe de llevar sangría a 1.27 cm en negrilla, minúsculas y punto final.
- **Nivel 4:** el encabezado debe de llevar sangría a 1.27 cm en negrilla, cursivas, minúsculas y punto final.
- **Nivel 5** en adelante: el encabezado debe de llevar sangría a 1.27 cm en cursiva, minúsculas y punto final.