UNIVERSIDAD NACIONAL DE LOJA

Área de la Energía, las Industrias y los Recursos Naturales no Renovables

Carrera de Ingeniería en Sistemas

” Revisión Sistemática de Literatura: Seguridad en Ambientes Web Utilizando Framework.”

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS

Autor:

ECINF8635, Jesennia Maribel Iñiguez Banegas

Director:

Ing. Edwin René Guamán Quinche, Mg. Sc.

LOJA – ECUADOR
2016

CERTIFICACIÓN DEL DIRECTOR

Ing. Edwin René Guamán Quinche, en calidad de director del trabajo de titulación designado por disposición de la coordinación de la carrera de Ingeniería en Sistemas, certifico que la Egresada Jesennia Maribel Iñiguez Banegas, ha culminado el trabajo de titulación, con el tema **"REVISIÓN SISTEMÁTICA DE LITERATURA: SEGURIDAD EN AMBIENTES WEB UTILIZANDO FRAMEWORK"**, quien ha cumplido con todos los requisitos legales exigidos por lo que se aprueba la misma. Es todo cuanto puedo decir en honor a la verdad, facultando al interesado hacer uso de la presente, así como también se autoriza la presentación para la evaluación por parte del jurado respectivo.

Loja, 28 de octubre del 2016

Atentamente,

Ing. Edwin René Guamán Quinche, Mg. Sc.

DIRECTOR DE TESIS

AUTORÍA

Yo **JESENNIA MARIBEL IÑIGUEZ BANEGAS**, declaro ser autora del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja, y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente, acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional-Biblioteca Virtual.

Firma: 

Cédula: 1104718166

Fecha: 12 de diciembre del 2016.

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LA AUTORA, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo **JESENNIA MARIBEL IÑIGUEZ BANEGAS**, declaro ser autora de la tesis titulada: **"REVISIÓN SISTEMÁTICA DE LITERATURA: SEGURIDAD EN AMBIENTES WEB UTILIZANDO FRAMEWORK"**, como requisito para optar el grado de: **INGENIERO EN SISTEMAS;** autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copias de las tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los doce días del mes de diciembre del dos mil dieciséis.

Firma: 

Autora: Jesennia Maribel Iñiguez Banegas

Cedula: 1104718166

Dirección: Loja (Piura y Cali).

Correo Electrónico: jesennniaib@gmail.com

Teléfono: 072613166 **Celular:** 0983818115

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Edwin René Guamán Quinche, Mg. Sc.

Tribunal de Grado: Ing. Hernán Leonel Torres Carrión, Mg. Sc.,

Ing. Pablo Fernando Ordoñez Ordoñez Mg. Sc.,

Ing. Alex Vinicio Padilla Encalada, Mgs.

DEDICATORIA

Con el más profundo agradecimiento, amor y cariño dedico mi tesis.

En primer lugar, a ti DIOS que me diste la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi espíritu y mi corazón e iluminar mi mente, por una familia maravillosa y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía.

Con mucho cariño principalmente a mis padres Miller Iñiguez y Rosa Banegas que me dieron la vida y han estado conmigo en todo momento. Gracias por todo papá y mamá por darme una carrera para mi futuro y por creer en mí, aunque hemos pasado momentos difíciles siempre han estado apoyándome y brindándome todo su amor, tiempo y paciencia. A mi hija Analía que es el motor de mi vida y mi razón de ser, a mi esposo, mis hermanos y a todas las personas que con su cariño y aprecio han hecho posible cumplir una meta más en mi vida profesional.

Jesennia I.

AGRADECIMIENTO

En primer lugar, a Dios y a mis padres por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en momentos de debilidad y por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad.

Mi más sincero agradecimiento a la Universidad Nacional de Loja, que me abrió sus puertas y me concedió el privilegio de estudiar en esta noble institución de gran prestigio y trayectoria.

A los docentes que conforman la carrera de Ingeniería en Sistemas por compartir sus experiencias y conocimientos en cada uno de los módulos hasta alcanzar mi meta.

Al Ingeniero René Guamán director de tesis, quien dedicó parte de su valioso tiempo a pesar de sus múltiples ocupaciones, para guiarme durante el desarrollo del proyecto de fin de carrera.

Así mismo agradezco a todos mis amigos y compañeros con quienes compartimos la vida universitaria y fueron un gran apoyo en momentos difíciles, familiares y demás personas que me supieron dar aliento en toda la etapa de estudiante.

Jesennia I.

a. Título

"REVISIÓN SISTEMÁTICA DE LITERATURA: SEGURIDAD EN AMBIENTES WEB UTILIZANDO FRAMEWORK"

b. Resumen

Las aplicaciones web se han convirtiendo en una parte esencial de nuestra vida cotidiana, ya que muchas de nuestras actividades dependen de la funcionalidad y seguridad de las mismas. Su principal activo es la información, esto implica que estén expuestas a un sinnúmero de vulnerabilidades, como: inyección SQL, perdida de autenticación y gestión de sesiones, Cross-site scripting XSS, entre otros, que se ha convertido en los principales desafíos de seguridad. De acuerdo a OWASP el 80% de fallos de la seguridad, se debe principalmente al desconocimiento de la estructura de frameworks de desarrollo en la construcción de aplicaciones, porque, los programadores no configuran los mecanismos de seguridad. El presente trabajo, detalla el proceso de una Revisión sistemática de Literatura, el mismo que servirá para encontrar estudios primarios acerca de propuestas de solución, para prevenir problemas de seguridad. Para partir con la revisión se debe en primera instancia iniciar con la pregunta de revisión, que evidencia los estudios primarios existen sobre mecanismos de seguridad para problemas de seguridad en frameworks de desarrollo. Resultados basados en estudios de artículos relevantes, que luego de cumplir con los criterios de selección se redujeron en 13 estudios primarios relevantes. De estos estudios se hace las respectivas conclusiones.

c. Abstract

Web applications have become an essential part of our daily life since many of the activities depend on the functionality and security of these. It's main function is the information, this means that they are exposed to an infinite number of vulnerabilities, like: SQL injection, loss of authenticity and session management, Cross site scripting XSS, among others, that have become the main challenges of security. According to OWASP, 80% of security fails, is due mainly to the lack of knowledge of the structure of the frameworks of building development of the applications because programmers don´t setup security mechanisms. The following written work, details the systematic process of Literature, the same that will be useful to find primary studies to offer solutions, to prevent security problems. To be able to start with the revision you must start with the revision question that highlights existent primary studies about security mechanism for security problems of development frameworks. Results based on studies of relevant articles that after fulfilling with the selection criteria were reduced into 13 relevant primary studies. From these studies the following conclusions have been made.

ÍNDICE

ÍNDICE DE GRÁFICOS

ÍNDICE DE TABLAS

d. Introducción

La seguridad del software es una inquietud cada vez más significativa para las instituciones del sector público o privado. Sin embargo, pocos programadores abordan este carácter de calidad de forma estratégica [1]. Los arquitectos y desarrolladores continuamente ponen un énfasis mayor en satisfacer los requerimientos prácticos y funcionales, y la seguridad usualmente es aplicada como un “adicional” para arreglar una vulnerabilidad durante o después de que la aplicación ha sido desarrollada [2].

Desarrollar código encaminado a la seguridad es una tarea que pocos la realizan por ser compleja [3] y, por ello, asiduamente se recurre a la adopción y uso de frameworks que se orientan en satisfacer distintas áreas de la seguridad como por ejemplo el control de acceso a los distintos sistemas, el cifrado de la información y la validación de entradas, entre las más importantes [4].

Se pueden considerar tres enfoques referentes al acogimiento de frameworks para mejorar los aspectos de seguridad como parte del diseño de la arquitectura: ninguna adopción, cuando la seguridad no se considera para el diseño, sino soluciones adicionales para cubrir aspectos puntuales; en adopción a medias se usan frameworks de seguridad luego del diseño inicial; y adopción total considera la seguridad desde el inicio dentro del diseño de la arquitectura e influye en todo el proyecto [5] [6].

Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado, por lo tanto se depende de lo sólido y flexible del mismo a la vez lo cual es un problema para el programador si no lo sabe utilizar [7][8]. Siendo así que el 80% de problemas de seguridad de sistemas web se debe a que los programadores no configuran los mecanismos de seguridad de frameworks [9].

Las vulnerabilidades de aplicaciones Web se han convertido, en los últimos años, en una gran amenaza para la seguridad de sistemas informáticos [10]. Esta situación se explica por el aumento de la complejidad de tecnologías de la Web [11], por la evolución frecuente de estas tecnologías, por los ciclos cortos de desarrollo de aplicaciones Web durante el cual las actividades de prueba y validación son limitados, y también, en algunos casos, por la falta de seguridad, habilidades y cultura de los desarrolladores [12].

En base a todo esto, en el presente trabajo de titulación se desarrolló una Revisión Sistemática de seguridad en ambientes web utilizando framework, basados en la metodología de Barbara Kitchenham [13] , siendo un medio para evaluar e interpretar todas las investigaciones disponibles acerca de una pregunta en particular de investigación, área temática, o fenómeno de interés. Las revisiones sistemáticas tienen como objetivo presentar una evaluación razonable de un tema de investigación mediante el uso de una metodología fiable, rigurosa y auditable. Los estudios individuales que contribuyen a una revisión sistemática se denominan estudios primarios [14].

El presente documento consta de un contenido esquematizado de acuerdo a los lineamientos establecidos por la Universidad Nacional de Loja, en donde se refleja todas las etapas de ejecución del Proyecto. Inicia con el Resumen, que relata de forma general aspectos relevantes del proyecto; la Introducción describe de manera global el ámbito del trabajo, y su estructura; en la Revisión de literatura, se encuentra los conceptos fundamentales que permiten entender y abordar el desarrollo del proyecto, cuyo contenido se encuentra distribuido en tres capítulos en los que se trata: Seguridad web, Gestores bibliográficos y Revisiones bibliográficas; el siguiente apartado corresponde a Materiales y Métodos, en donde se detalla todas las herramientas de hardware, software y recursos de oficina, luego se desarrolló el Método para la selección de estudios primarios; posteriormente se muestra los Resultados, que detalla cómo fue planteado el desarrollo del trabajo de titulación basado en el método de Revisiones Sistemáticas de Bárbara Kitchenham; luego se especifica la Discusión, en donde se hace la valoración de los objetivos determinando su cumplimiento y argumentando las actividades realizadas para conseguirlos; a continuación las Conclusiones, que se relacionan directamente con los objetivos específicos; Recomendaciones, se plantea aspectos a considerar para el desarrollo de futuros trabajos. Se finaliza con la bibliografía de las fuentes de donde se extrajo la información necesaria para el desarrollo del proyecto, y los anexos.

e. Revisión de literatura

1. Seguridad web

En la actualidad el crecimiento de internet ha impactado directamente en la seguridad de la información manejada cotidianamente. Sitios de comercio electrónico, servicios, bancos e incluso redes sociales contienen información sensible que en la mayoría de los casos resulta ser muy importante.

Las aplicaciones Web por lo general están en línea y eso las expone a la visita de usuarios que llegan con objetivos distintos, uno de ellos el hacerse de fama violando la seguridad de sitios Web valiéndose de diferentes estrategias y herramientas [15].

La seguridad en la Web es un conjunto de procedimientos, prácticas y tecnologías para proteger a los servidores y usuarios de la Web, y las organizaciones que los rodean. La seguridad es una protección contra el comportamiento inesperado [16].

Internet es una red de dos sentidos. Así como hace posible que los servidores Web divulguen información a millones de usuarios, permite a los hackers, crackers, criminales irrumpir en las mismas computadoras donde se ejecutan los servidores Web.

Las empresas, instituciones y los gobiernos utilizan cada vez más el Word Wide Web para distribuir información importante y realizar transacciones comerciales. Al violar servidores Web se pueden dañar reputaciones y perder dinero.

Los principios básicos de la seguridad web son:

- **Integridad:** los objetos sólo pueden ser modificados por elementos autorizados, y de una manera controlada. Hace referencia al hecho de que la información no pueda ser manipulada en el proceso de envío.

- **Disponibilidad:** los objetos del sistema tienen que permanecer accesibles a elementos autorizados. La pérdida de disponibilidad puede implicar, la pérdida de productividad o de credibilidad de la entidad.

- **Confidencialidad:** los objetos de un sistema han de ser accedidos únicamente por elementos autorizados a ello. Asegura el secreto de las comunicaciones

contenidas en los mensajes. La pérdida de confidencialidad puede resultar en problemas legales, pérdida del negocio o de credibilidad [9].

Las aplicaciones Web, por definición, permiten el acceso de usuarios a recursos centrales tal como al servidor Web y el cual permite el acceso a otros servidores de base de datos. Con los conocimientos y la implementación correcta de medidas de seguridad, se pueden proteger los recursos, así como proporcionar un entorno seguro en el cual los usuarios trabajen cómodos con su aplicación.

Una aplicación Web, especialmente que se ejecuta en Internet, es muy vulnerable a ataques de los hacker que una aplicación autónoma o cliente-servidor típica. Hay varias razones para esto:

- **Disponibilidad y accesibilidad:** Muchas aplicaciones Web están disponibles para los usuarios públicos en cualquier momento del día o de la noche. Como los servidores Web tienen que permitir el acceso a usuarios públicos y no tienen la protección completa de los cortafuegos típicos de una empresa.

- **Familiaridad:** La mayoría de los atacantes, incluso los menos sofisticados, conocen las interfaces Web. Un navegador Web es fácil de obtener y es uno de los programas de aplicación más comunes. El protocolo HTTP está bien definido, y existen muchas herramientas de hacking creados específicamente para ayudar a los atacantes a penetrar y comprometer las aplicaciones Web.

- **Facilidad:** La configuración de un servidor Web, contenedor Web y aplicación Web para uso público es extremadamente compleja. Los atacantes, frecuentemente, pueden aprovechar esta complejidad y explotar deficiencias en la configuración de la aplicación o del sistema. Publicidad: El ego de algunos hackers experimentados es la publicidad, la fama, o un simple deseo de probar que pueden hacer algo que no todas las personas pueden hacer [17].

1.1 Problemas principales en la programación de sistemas web

Gran parte de los problemas de seguridad en las aplicaciones web son causados por la falta de seguimiento en dos rubros muy importantes de los que depende cualquier aplicación, las entradas y salidas del sistema se muestra en el Gráfico 1.

Gráfico 1. Entrada y salida del sistema.

1.2 Prácticas básicas de seguridad web

- **Balancear riesgo y usabilidad:** Si bien la usabilidad y la seguridad en una aplicación web no son excluyentes una de la otra, alguna medida tomada para incrementar la seguridad con frecuencia afecta la usabilidad. Normalmente siempre se debe pensar en las maneras en que usuarios ilegítimos nos pueden atacar y la facilidad de uso para los usuarios legítimos por lo tanto es conveniente emplear medidas de seguridad que sean transparentes a los usuarios y que no resulten engorrosas en su empleo.

- **Rastrear el paso de los datos:** Es muy importante mantener conocimiento de los pasos que ha recorrido la información en todo momento. Conocer de dónde vienen los datos y hacia dónde van. En muchas ocasiones lograr esto puede ser complicado, especialmente sin un conocimiento profundo de cómo funcionan las aplicaciones web.

- **Filtrar entradas**: El filtrado es una de las piedras angulares de la seguridad en aplicaciones web. Es el proceso por el cual se prueba la validez de los datos. Si nos aseguramos que los datos son filtrados apropiadamente al entrar, podemos eliminar el riesgo de que datos contaminados sean usados para provocar funcionamientos no deseados en la aplicación. Si llegamos a utilizar algún framework se debe tener especial cuidado, ya que estos brindan tantas comodidades que muchos desarrolladores inexpertos los utilizan sin preocuparse en entender el código que están observando y por lo tanto implementan medidas

de validación en entradas, variables, entre otros, sin entender exactamente el funcionamiento de la solución empleada.

- **Escapado de salidas:** otro punto importante de la seguridad es el proceso de escapado y su contraparte para codificar o decodificar caracteres especiales de tal forma que su significado original sea preservado. Si llegamos a utilizar una codificación en particular es necesario conocer los caracteres reservados los cuales serán necesarios escapar [18].

1.3 ¿Qué son los riesgos de seguridad en aplicaciones?

Los atacantes pueden potencialmente usar rutas diferentes a través de la aplicación para hacer daño a su negocio u organización como se indica en el Gráfico 2. Cada una de estas rutas representa un riesgo que puede, o no, ser lo suficientemente grave como para justificar la atención.

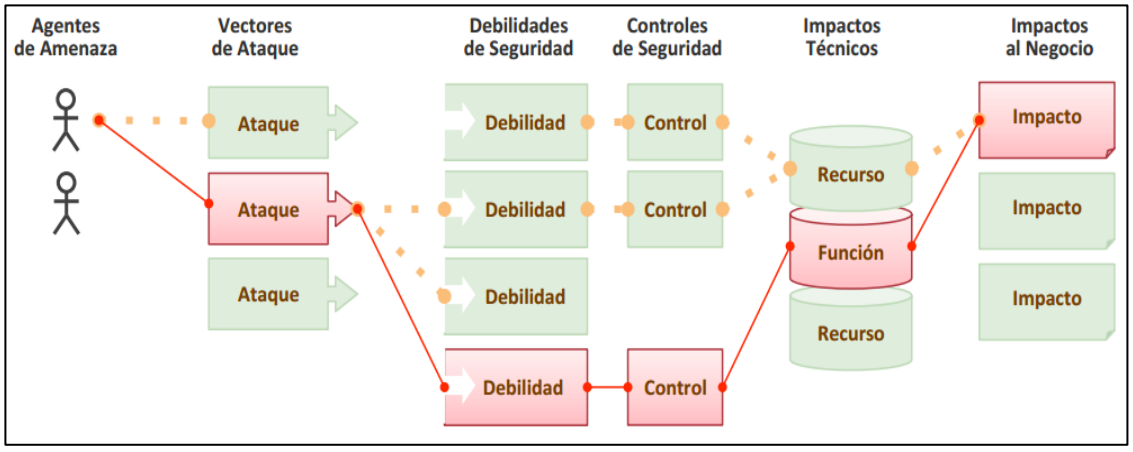

Gráfico 2. Rutas de riesgos

A veces, estas rutas son triviales de encontrar y explotar, y a veces son muy difíciles. Del mismo modo, el daño que se causa puede ir de ninguna consecuencia, o ponerlo fuera del negocio. Para determinar el riesgo en su organización, puede evaluar la probabilidad asociada a cada agente de amenaza, vector de ataque, y la debilidad en

la seguridad, y combinarla con una estimación del impacto técnico y de negocios para su organización. En conjunto, estos factores determinan el riesgo global.

Según OWASP los principales problemas de seguridad en aplicaciones web son [19]:

- **A1 Inyección:** Las fallas de inyección, tales como SQL, OS, y LDAP, ocurren cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta. Los datos hostiles del atacante pueden engañar al intérprete en ejecutar comandos no intencionados o acceder a datos no autorizados. Siendo las inyecciones SQL una de las más comunes.

- **A2 Pérdida de Autenticación y Gestión de Sesiones:** Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son frecuentemente implementadas incorrectamente, permitiendo a los atacantes comprometer contraseñas, claves, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios.

- **A3 Secuencia de Comandos en Sitios Cruzados (XSS):** Las fallas XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencia de comandos en el navegador de la víctima los cuales pueden secuestrar las sesiones de usuario, destruir sitios web, o dirigir al usuario hacia un sitio malicioso.

- **A4 Referencia Directa Insegura a Objetos:** Una referencia directa a objetos ocurre cuando un desarrollador expone una referencia a un objeto de implementación interno, tal como un fichero, directorio, o base de datos. Sin un chequeo de control de acceso u otra protección, los atacantes pueden manipular estas referencias para acceder a datos no autorizados.

- **A5 Configuración de Seguridad Incorrecta:** Una buena seguridad requiere tener definida e implementada una configuración segura para la aplicación, marcos de trabajo, servidor de aplicación, servidor web, base de datos, y plataforma. Todas estas configuraciones deben ser definidas, implementadas, y mantenidas ya que por lo general no son seguras por defecto. Esto incluye mantener todo el software actualizado, incluidas las librerías de código utilizadas por la aplicación.

- **A6 Exposición de datos sensibles:** Muchas aplicaciones web no protegen adecuadamente datos sensibles tales como números de tarjetas de crédito o credenciales de autenticación. Los atacantes pueden robar o modificar tales datos para llevar a cabo fraudes, robos de identidad u otros delitos. Los datos sensibles requieren de métodos de protección adicionales tales como el cifrado de datos, así como también de precauciones especiales en un intercambio de datos con el navegador.

- **A7 Ausencia de Control de Acceso a Funciones**: La mayoría de aplicaciones web verifican los derechos de acceso a nivel de función antes de hacer visible en la misma interfaz de usuario. A pesar de esto, las aplicaciones necesitan verificar el control de acceso en el servidor cuando se accede a cada función. Si las solicitudes de acceso no se verifican, los atacantes podrán realizar peticiones sin la autorización apropiada.

- **A8 Falsificación de Peticiones en Sitios Cruzados (CSRF):** Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una petición HTTP falsificado, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable. Esto permite al atacante forzar al navegador de la víctima para generar pedidos que la aplicación vulnerable piensa son peticiones legítimas provenientes de la víctima.

- **A9 Utilización de componentes con vulnerabilidades conocidas:** Algunos componentes tales como las librerías, los frameworks y otros módulos de software casi siempre funcionan con todos los privilegios. Si se ataca un componente vulnerable esto podría facilitar la intrusión en el servidor o una perdida seria de datos. Las aplicaciones que utilicen componentes con vulnerabilidades conocidas debilitan las defensas de la aplicación y permiten ampliar el rango de posibles ataques e impactos.

- **A10 Redirecciones y reenvíos no validados:** Las aplicaciones web frecuentemente redirigen y reenvían a los usuarios hacia otras páginas o sitios web, y utilizan datos no confiables para determinar la página de destino. Sin una validación apropiada, los atacantes pueden redirigir a las víctimas hacia sitios de phishing o malware, o utilizar reenvíos para acceder páginas no autorizadas.

1.4 Ataques a aplicaciones web:

- Un ataque de **inyección SQL** consiste en la inserción o “inyección” de una consulta SQL a través de los datos de entrada del cliente de la aplicación. Una inyección SQL éxito exploit puede leer los datos sensibles de la base de datos, modificar datos de bases de datos (Insertar / Actualizar / Eliminar), ejecutar operaciones de administración en la base de datos (tales como apagar el DBMS), recuperar el contenido de un archivo determinado presente en el sistema de archivos DBMS y en algunos casos emitir comandos al sistema operativo. Los ataques de inyección SQL son un tipo de ataque de inyección, en el que comandos SQL se inyectan en la entrada de datos de plano a fin de efectuar la ejecución de comandos predefinidos SQL [20].

Los ataques por inyección SQL permiten a los atacantes suplantar identidad, alterar datos existentes, causar problemas de repudio, permite la revelación de todos los datos en el sistema, destruir los datos o si no volverlos inasequibles, y convertirse en administradores del servidor de base de datos.

La inyección SQL es muy común con aplicaciones PHP y ASP. Debido a la naturaleza de las interfaces programáticas, las aplicaciones J2EE y ASP.NET tienen menor probabilidad de ser fácilmente atacadas por una inyección SQL.

La gravedad de una inyección SQL está limitada por la habilidad e imaginación del atacante, y en menor medida a las contramedidas, como por ejemplo las conexiones con bajo privilegio al servidor de bases de datos, entre otras. En general, se considera a la inyección SQL de alto impacto.

La inyección SQL se ha convertido en un problema común con sitios web que cuentan con base de datos. La falla es fácilmente detectada y fácilmente explotada, y como tal, cualquier sitio o paquete de software con incluso una mínima base de usuario es propenso a ser objeto de un intento de ataque de este tipo. Esencialmente, el ataque es llevado a cabo mediante la colocación de una meta carácter en los datos de entrada para colocar comandos SQL en el plano de control, el cual antes no existía. Este error depende del hecho de que SQL no hace real distinción entre los planos de datos y los de control.

Los ataques realmente varían de unos a otros, se pueden clasificar en dos categorías:

1. **Datos Exfiltración:** Exfiltración de datos a través de la inyección de SQL es lo que ha contribuido a algunas de las violaciones de datos más grandes hasta la fecha. Los atacantes encuentran una vulnerabilidad que les permite a la lista de todas las tablas y volcar todas las cuentas de usuario, correos electrónicos y contraseñas.
2. **Código Inyección:** No vemos esto muy a menudo, que a menudo se basan en algunas de las vulnerabilidades iniciales pre-pruebas que nos bloqueamos automáticamente a través de nuestro Sitio Web Firewall por lo que es mucho más difícil de grabar y ejecutar.

- **Ataques XSS** aprovecha la falta de mecanismos de filtrado y validación en campos de entrada, permitiendo así el envío de scripts completos (como Visual Basic Scripts o Java Scripts) con secuencias de comandos maliciosos que podrían impactar directamente en el sitio web o en el equipo de un usuario. Esta limitación se debe a que el código HTML se interpreta en el navegador de un usuario y no en el servidor. Así que si alguien inyecta código HTML en alguna aplicación web no podría hacer daño alguno al servidor, ya que éste nunca interpreta el código HTML, sólo los navegadores. Por eso este ataque se denomina: ataque del lado del cliente.

Los recursos para explotar la vulnerabilidad son: formularios de contactos de sitios web, mensajes en Foros, Firma de libro de visitas, Buscadores, Variables, Correo Web. Los tipos de ataques XSS son:

Ataques Directos, El ataque de forma directa de XSS (también llamado XSS persistente), se presenta cuando el atacante consigue embeber código HTML malicioso, directamente en los sitios Web que así lo permiten. Funciona localizando puntos débiles en la programación de los filtros de HTML si es que existen, para publicar contenido. Este tipo de ataques suele ser el más común, y el código del atacante, se basa en etiquetas HTML (del tipo <frame> o <script>), entre las cuales incluye el código malicioso.

Ataques Indirectos: Este tipo de ataques se presenta, cuando el código maligno se inyecta a través de formularios, a través de los parámetros de una URL, programas en Flash, un enlace malicioso e incluso vídeos. Un ejemplo de un XSS indirecto, puede darse a través de un enlace malicioso, del tipo: http://www.App Victima.com/search?p=<script>alert('Hola-mundo')</script>. El resultado muestra una

ventana con el texto "hola-mundo". Esta vulnerabilidad suele ser usada para efectuar robo de sesiones y phishing [21].

- **Ataques DoS:** Este tipo de ataque tiene como objetivo dejar sin servicio a la propia aplicación en lugar de la máquina. Una de las ventajas que tiene el atacante es que no necesita tantos recursos para llevar a cabo el ataque como un ataque DoS normal. En muchas ocasiones este tipo de ataques son más difíciles de detectar debido a que se camuflan en peticiones aparentemente comunes.

Los ataques DoS contra aplicaciones web se pueden llevar a cabo de diferentes maneras:

Cuelgue de la Aplicación: Buffer Overflows, Varios

Buffer Overflows: Los fallos de desbordamiento de Buffer pueden estar presentes si escribimos aplicaciones Web en C o lenguajes similares. Este tipo de ataques se producen cuando se escriben datos en un buffer que sobre escriben otros adyacentes, se suelen producir al copiar cadenas de caracteres de un buffer a otro. Estos desbordamientos de buffer pueden provocar muchas veces ataques de denegación de servicio contra la aplicación. Hay que prestar especial atención a funciones como strcpy(), strcat(), sprintf(), gets().

Varios: Los fallos de Format String se pueden utilizar en ocasiones para realizar ataques de denegación de servicio. Muchas veces un atacante puede aprovechar un ataque de SQL injection, Remote File Inclusion, etc. si los permisos del servidor no están debidamente configurados para llamar a comandos del sistema que paren las aplicaciones o servicios.

Modificación y destrucción de datos: Consumo de recursos, CPU, Ancho de Banda, Memoria, Espacio en Disco.

Este tipo de ataques no provocan que el servidor deje de dar un servicio, no obstante estos ataques afectan al funcionamiento normal del servicio provocando un mal funcionamiento del mismo [22].

1.5 Frameworks de desarrollo Web

Se define un framework como un conjunto de librerías y componentes junto con una documentación y metodología de uso, que permite diseñar, construir e implementar aplicaciones con mayor calidad y agilidad de programación [23].

A continuación, se detalla las características más importantes o descripción básica de los frameworks más populares

- **Spring**

Spring es un framework de código abierto, creado con el objetivo de facilitar el desarrollo de aplicaciones empresariales. Fue desarrollado gracias a la colaboración de grandes programadores, quienes lograron combinar las siguientes herramientas javas J2EE, EJB, Servlets y JSP, en un solo framework de aplicación al cual llamaron Spring. Se lo considera un framework liviano de gran funcionalidad; ya que, no requiere grandes recursos para su ejecución y aporta una gran funcionalidad al desarrollador.

Spring ofrece un modelo MVC para el desarrollo de aplicaciones web, con lo que da solución a varias de las capas de la arquitectura Web como: presentación, lógica del negocio e integración de datos. De igual manera, ofrece soporte AOP (Programación orientada a Aspectos) y abstracciones para manejo de datos JDBC.

En la parte central de Spring se incorporan los conceptos de IoC (Inversión de control) y DI (Inyección de dependencia). Estos se centran en el principio de que en lugar que el código de la aplicación llame a una clase de una librería, el framework llama al código y de esta acción nace el nombre de la tecnología ya que se invierte la acción de la llamada.

Sus características son: Proporciona el patrón MVC para la construcción de aplicaciones. Utiliza inyección de dependencias para independizar el código del negocio con el del framework. Ofrece AOP para centrar el código en las preocupaciones y aspectos importantes del negocio. Es un framework liviano y no invasivo. Usa objetos POJO en la implementación de la lógica del negocio. Permite la integración con otros frameworks para el mapeo de objetos de la base de datos.

Las Ventajas son: La inyección de dependencia aporta soporte para otros frameworks. Soporte JDBC. Mucha documentación disponible. Herramientas adicionales como Spring IDE. Es un framework completo para todas las capas.

Las Desventajas son: Su configuración es costosa, se requiere mucho código XML. Curva de aprendizaje media-alta. Los fallos solo pueden detectarse en tiempo de ejecución debido a la inversión de control. Es demasiado flexible, tanto que no existe un controlador padre.

Seguridad: Spring trae por defecto la protección contra ataques CSRF (Cross Site Request Forgery), esto hace que sea un poco más complejo hacer llamadas tipo POST, pero es altamente recomendable mantenerlo habilitado. Al principio puede parecer engorroso, pero al final es un hábito que cuando lo tienes no añade trabajo extra.

Spring nos inyecta en cada vista dos variables, el parámetro (Normalmente X-CSRF-TOKEN) que espera recibir en la cabecera o como parámetro en la petición y el valor de dicho parámetro.

Si usamos vistas JSPs lo implementaríamos de la siguiente manera:

```
1 <form action="/mi-accion" method="post">
2 <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>
3 <input type="Enviar" value="Log out" />
4 </form>
```

- **Richfaces**

Richfaces es un framework web open-source que extiende el framework JSF, añadiéndole capacidades Ajax sin la necesidad de usar JavaScript. Se caracteriza por una gran facilidad de integración de Ajax en la vista y que sus componentes están listos para su uso, con lo cual el desarrollador puede ahorrar tiempo de inmediato aprovechando las características de estos componentes. Además, ya que es una extensión de JSF implementa también el patrón de diseño MVC.

Richfaces se crea a partir del framework Ajax4jsf y en la actualidad es mantenido por JBoss y Red Hat. En un comienzo Richfaces era una librería de componentes comerciales, luego JBoss y Red Hat convinieron que sea un proyecto open source.

El framework funciona mediante sus propias etiquetas, las cuales generan eventos que envían peticiones al contenedor Ajax. Estos eventos son generados según acciones del usuario, disparando la funcionalidad antes descrita. De acuerdo a la forma de funcionamiento se concluye que el desarrollador no tiene que preocuparse de crear el código JavaScript.

Las Características son: Extiende la funcionalidad de JSF. Incorpora Ajax en sus componentes. Se centra en la creación de páginas web ricas. Orientado a la interfaz de usuario.

Las ventajas son: Se integra perfectamente en el ciclo de vida de JSF. Incluye funcionalidades Ajax. Contiene elementos visuales para el desarrollo de una aplicación web rica. Soporte de CSS themes o skins. Tiene una comunidad activa. Mejor aspecto y más componentes que JSF.

Las desventajas son: Alto consumo de memoria del navegador. Requiere de conocimientos JSF.

- **Seam**

Seam es un framework de integración de tecnologías que tiene como objetivo facilitar el desarrollo de aplicaciones JEE. Es un proyecto open-source creado por JBOSS que lo provee de una comunidad de respaldo. El aspecto más relevante de Seam está en su arquitectura; ya que, integra el uso de varias tecnologías existentes para la creación de aplicaciones web. Por lo que facilita la implementación del modelo MVC de forma intuitiva y rápida para la programación, pero sin perder la potencia y características de JEE.

Las características son: Define un modelo de componentes uniforme para la implementación de la lógica del negocio. Integración de JSF con EJB3, Seam unifica el framework de presentación JSF (Java Server Faces) y de lógica del negocio con EJB3 (Enterprice Java Bean en su versión 3). Implementación AJAX, permite la utilización de tecnologías de dos frameworks JSF basados en AJAX como lo son Richfaces e

IceFaces. Facilita la opción de manejo de procesos de negocio transparentes con jBPM. Poca dependencia de XML con la implementación EJB3 para los archivos de configuraciones exceptuando JSF que sigue siendo muy dependiente de XML.

Las ventajas son: Integra varias tecnologías estándar. Permite al desarrollador usar una arquitectura uniforme en sus aplicaciones con el modelo MVC. Integra Ajax a sus aplicaciones sin la necesidad de escribir código JavaScript. Poca dependencia de XML para la configuración. Aplicaciones más robustas pues se centra en el desarrollo y no simplemente en servir paginas HTML.

Las desventajas son: Curva de aprendizaje media-alta. Demasiado contenido lo que puede hacer que el desarrollador no aproveche todas sus funcionalidades.

Seguridad: El Seam Security API es una parte de Seam que proporciona funcionalidades de autenticación y autorización basado en JAAS (Java Authentication and Authorization Service). Puede usarse el modo sencillo, por defecto, que se basa en roles o usar el framework JBoss Rules, que ofrece un sistema más poderoso basado en reglas. El sistema se encarga del manejo de errores de autorización o autenticación, permitiendo redirigir al usuario a una página determinada. Las restricciones pueden establecerse mediante el uso de anotaciones. Permite restringir tanto acciones como entidades o páginas o componentes de la página.

La seguridad en la configuración como: instalar el componente de identidad, crear una clase de autenticación, crear un formulario de login.

- **Struts**

Struts fue el primer frameworks J2EE que salió al mercado. Es un framework que implementa el patrón de diseño MVC y básicamente está construido sobre las tecnologías de Servlet, JSP, JavaBeans y XML. Puede operar bajo cualquier contenedor de Servlets. Aunque ofrece un patrón de diseño MVC, no proporciona características propias para el desarrollo de la capa del modelo; es decir, a la hora de afrontar un proyecto será flexible a utilizar distintos objetos de negocio como lo son JavaBeans, Java Data Objects, u objetos de acceso de datos.

Las características son: Struts proporciona un controlador el cual es el corazón del framework. Proporciona acceso rápido a las clases java desde la vista. Simplifica mucho el proceso de desarrollo de la capa de presentación, permitiendo dedicar más

tiempo a codificar la lógica del negocio. Las interrelaciones entre Acciones y páginas u otras acciones se especifican por tablas XML. Librerías de entidades para facilitar las operaciones que realizan las páginas JSP. Contiene herramientas para validación de campos de plantillas bajo varios esquemas que van desde validaciones locales en la página, hasta las validaciones de fondo hechas a nivel de acciones. Soporte para Ajax.

Las ventajas son: El objetivo del proceso de desarrollo es la calidad del producto. Muy popular, con mucho material disponible. Curva de aprendizaje media. Apropiado tanto para desarrolladores independientes como equipos grandes de desarrollo. Permite crear aplicaciones Web de manera rápida y efectiva. Admite plug-ins para poder incrementar su funcionalidad. Se adapta a la incorporación de diferentes bibliotecas de tags.

Las desventajas son: No está diseñado para facilitar la creación de componentes propios. Las vistas quedan atadas al dispositivo en el que se renderizan, No define de manera adecuada el nivel de lógica de negocio. Muy Orientado a la capa de presentación.

Seguridad: La versión 1.1 de Struts introdujo la seguridad basada en acciones. Esto quiere decir que podemos combinar los mecanismos estándar de seguridad declarativa J2EE con el funcionamiento de nuestra aplicación Struts. Por tanto, primero necesitamos configurar el web.xml para restringir el acceso a los recursos protegidos a los usuarios que tengan determinado rol.

Para cada acción especificaremos qué rol o roles pueden ejecutarla, mediante el atributo roles de la etiqueta <action> en el struts-config.xml. Por ejemplo:

```
<action roles="admin, manager"
        path="/admin/borrarUsuario"
        ...
</action>
```

Si se intenta ejecutar una acción para la que no se tiene permiso, el controlador lanzará una excepción de tipo org.apache.struts.chain.commands.UnauthorizedAction Exception. Ya hemos visto cómo capturar una excepción y hacer que el navegador se redirija a una página de error.

2. Gestores bibliográficos y bases de datos científicas

2.1 Gestores bibliográficos

Son programas que permiten crear, mantener, organizar y dar forma a referencias bibliográficas de artículos de revista o libros; obtenidas automáticamente de una o de varias fuentes de información (bases de datos, revistas, páginas web, etc.), y que añaden a esta función básica su versatilidad para generar cientos de formatos de entrada y salida, utilizados para citar referencias bibliográficas en los trabajos de investigación [24].

Esto es fundamentalmente lo que les diferencia de cualquier otro programa de bases de datos, por una parte, la capacidad para compilar información desde la mayoría de las fuentes de información, y por otra la integración en los programas de proceso de textos (Word, OpenOffice) para facilitar a los investigadores la inclusión de citas en los trabajos de investigación.

Organización

Los Gestores de Referencias (GRs) facilitan la organización de sus registros, tanto desde el uso de carpetas (como si estuviéramos en el explorador de archivos de Windows), así como también la posibilidad de colocar etiquetas a cada referencia. Esto facilita la búsqueda al recuperar los registros luego.

Búsqueda y ordenamiento

Esta función permite realizar búsquedas por los metadatos (información “acerca de” los documentos: título, autor, fuente, palabras clave, resumen, etc.) De cada registro. Algunos GRs como Mendeley, permiten inclusive una búsqueda a texto completo en caso de que los PDFs se incluyen en el manejador. Adicionalmente los resultados de búsqueda pueden ser ordenados por más de un criterio: relevancia, fecha, autor, otros.

Anotaciones

Se pueden añadir notas, comentarios y resaltar parte del texto si fuera el caso. Además, si es que el GR tiene capacidades sociales, es posible compartir estas notas con otros usuarios.

Importación/Exportación

Introducir conjuntos de referencias desde bases de datos es posible mediante el uso de archivos de metadata standard (Ris, BibTeX) que permiten la comunicación inclusive entre software de diversas compañías. Se pueden hacer cargas masivas hacia o desde un determinado gestor.

Almacenamiento

Estos programas facilitan espacio para almacenar los documentos (pdfs, docs, etc.) referenciados que van desde los 100 megas hasta 1GB.

Compartir/Redes sociales

Es posible compartir tanto los documentos, como la metadata (incluyendo comentarios y anotaciones) de los registros con individuos o grupos de usuarios con intereses comunes. Inclusive es factible seguir la actividad de estos grupos o investigadores, consultar los documentos que comparten, así como sus actualizaciones de estado.

Desde el punto de vista profesional, los gestores de referencias son de gran interés para bibliotecarios y documentalistas por la facilidad que ofrecen para compilar datos de diferentes fuentes de información y distintos formatos, organizar la colección, personalizar formatos de salida y proporcionar servicios de información, tales como [25]: Información bibliográfica, Referencia en línea, Difusión Selectiva de Información (DSI), Monográficos, Biblioteca Digital.

También por ser una herramienta válida para: La evaluación de recursos, Apoyo a las actividades de formación en información (alfabetización digital), Elaboración de estudios bibliométricos.

Funciones básicas de los gestores de referencias se muestra en el Gráfico 3.

Gráfico 3. Funciones de los Gestores bibliográficos

Entrada de datos

La entrada de datos puede realizarse de manera manual, introduciendo los datos correspondientes a los diferentes tipos de documentos: artículos de revista, libros, discos, mapas, partituras, etc. Todos los formatos comparten una serie de campos básicos, comunes a todos ellos, como son: el autor, el título, las materias y unos campos específicos para cada tipo de documento.

Pero la función principal de un gestor de referencia no es esta, sino la de compilar información de manera automática sin necesidad de teclear los datos de forma manual. Los gestores permiten recoger los metadatos que nos proporcionan las fuentes de información preexistentes (bases de datos, catálogos, revistas electrónicas, fuentes comerciales). Para ello los gestores trabajan con un formato en texto plano (.txt) denominado RIS1, desarrollado por Research Information Systems de quien toma el nombre que permite que los programas de citas puedan intercambiar datos entre sí y exportarlos con sorprendente facilidad desde cualquier fuente de información. El formato RIS consta del nombre estándar del campo, identificado con dos letras mayúsculas, espacio guion espacio, y la información correspondiente a ese campo. Si el campo fuera repetible utilizaría la primera letra en mayúscula y el número ordinal correspondiente.

Los gestores también trabajan con otros formatos de importación como MARC, o los correspondientes a cada editor de bases de datos como SILVER PLATER, EBSCO,

Emerald, MLA, CSIC, etc. La facilidad para compilar información desde diferentes formatos es una de las características esenciales de los gestores de referencias. Y la importancia es tal, que prácticamente todas las fuentes de información especializadas disponen de formatos de salida en RIS o en otros formatos de exportación, ya sea directa o indirecta.

Cuando hablamos de exportación directa nos referimos a que la fuente proporciona un enlace que permite enviar la información en formato RIS directamente al gestor de referencias para que se integre en el mismo, sin más. La importación indirecta exige la utilización de filtros de información desde el gestor de referencias para integrar la información a través de un fichero de texto plano en RIS, u otro formato.

Organización y consulta

La mayoría de los gestores permiten organizar la información que recopilan en carpetas que el usuario puede crear, organizar y compartir con otros usuarios en función de sus intereses. Los datos se organizan fundamentalmente en torno a tres índices básicos: Autores, Títulos y Materias. Podemos disponer de valores relativos de la frecuencia de aparición de los términos en la base de datos. Esto garantiza el poder llevar a cabo un buen sistema de control de autoridades. Esto es muy importante ya que vamos a manejar datos procedentes de diferentes fuentes, en diferentes lenguas, que cuentan con sistemas de indización distintos, por lo que es necesario dar integridad a la base de datos, de manera que los términos que estén indexados bajo una forma sólo lo estén bajo esa, y no otra.

Salida de datos

Como antes pusimos de manifiesto, uno de los valores fundamentales de los gestores de referencias es la capacidad y versatilidad de los formatos de salida. Tanto de formatos relativos al tipo de soporte de los documentos: salidas en texto plano (.txt), en texto enriquecido compatible con los procesadores de texto (.rtf y .doc), en formato Adobe Reader (.pdf), o directamente en formato hipervínculo (HTML). Como en formatos de cita normalizados (ASCI, Vancouver, Chicago, UMI…) o estilos de publicación de las revistas más importantes de cada campo del conocimiento. (MLA, AIA, Biochemistry, AIAA Journal, etc.). Algunos de ellos ya ofrecen la posibilidad de crear formatos de salida personalizados.

Actualmente, existe una gran variedad de estos programas: Mendeley, Procite, EndNote, Reference Manager, Bibus, BixTex, RefWorks, Zotero, y muchos más [26].

Tabla 1. Gestores bibliográficos tabla comparativa.

	Mendeley	Endnote	Zotero	RefWorks
Disponibilidad	Libre	Suscripción Pucp	Libre	anual US$ 100.00
Sistemas Operativos	Windows, Mac OsX, Ios	Windows, Mac, Ios	Windows, Mac, Ios*, Android*	Windows, Mac
Formatos de salida	BibTeX, Ris, XML	BibTeX, Ris, XML	BibTeX, Ris, XML	BibTeX, Ris, XML
Formatos de entrada	BibTeX, Ris	Ris	BibTeX, Ris, XML	BibTeX, Ris
Estilos para Citas Bibliográficas	Apa, Chicago/Turabiam, Harvard, MLA, otros 12 instalados. 6,400 pueden ser instalados.	Apa, Chicago/Turabia, Harvard, MLA, Otros 5,000 incluídos por default.	Apa, Chicago/Turabiam, Harvard, MLA, Otros 16 instalados. 6,400 pueden ser instalados.	Apa, Chicago/Turabiam, Harvard, MLA, Otros 1,600 disponibles.
Bibliografías	HTML, RTF, texto plano, otros	HTML, RTF, texto plano, otros	HTML, RTF, otros	RTF, texto plano, otros
Plugin para Word	Sí	Sí	Sí	Sí

	Mendeley	Endnote	Zotero	RefWorks
Captura web de metadatos desde:	ACM, ACS, Amazon, APA Psycnet, arXiv.org, Ebsco, Google Books, Google Scholar, IEEE, IOP, Web of Science, Jstor, Sage, Science Direct, Scopus, Springer, SSRN, Wikipedia, Wiley, WorldCat.	ACS, Amazon.com, arXiv.org, CrossRef, EBSCOHost, ERIC, Google Scholar, IEEE Xplore, JSTOR, Lexis/Nexis, Oxford Journals, ProQuest, Web of Science, Wiley, WorldCat, YouTube	EBSCO, IEEEXplore, JSTOR, Google Scholar, ProQuest, Cambridge University Press, Oxford University Press, Project MUSE, ScienceDirect, SpringerLink, Taylor and Francis, Primo, WorldCat.	Ebsco, Web of Science, WorldCat, Google Scholar, Web of Science, otros.
Ordenamiento de las referencias	Autor, título, año, Fuente, fecha de ingreso	Autor, título, año, Fuente, fecha de ingreso	Autor, título, año, Fuente, fecha de ingreso	Autor, título, Fuente, fecha de ingreso
Capacidades de Búsqueda	Búsqueda rápida, búsqueda por campos, conectores booleanos, búsqueda dentro de los PDFs	Búsqueda rápida, búsqueda por campos, conectores booleanos, búsqueda dentro de los PDFs	Búsqueda rápida, búsqueda por campos, conectores booleanos, Búsqueda por tags, búsqueda dentro de los PDFs	Búsqueda rápida, búsqueda por campos, conectores booleanos, búsqueda dentro de los PDFs
Completar metadata	Sí, incluso desde los PDFs.	Sí	Sí, incluso desde los PDFs.	Sí
Interconexión con otras versiones **	No	No	Sí	No
Links a documentos	Sí	Sí	Sí	Sí

	Mendeley	Endnote	Zotero	RefWorks
Anotaciones y comentarios permitidos	Sí	Sí	No	No
Chequea duplicados	Sí	Sí	Sí	Sí
Creación de carpetas	Sí, multinivel	Sí, multinivel	Sí, multinivel	Sí, multinivel
Edición simultánea de referencias	Sí	Sí	Sí	Sí, pero solo en licencias a nivel de campus
Trabajo a nivel de redes sociales	Comunicación dentro de grupos de interés. Compartir publicaciones. Mantener perfil de investigación. Buscar y seguir individuos.	No	Comunicación dentro de grupos de interés. Mantener perfil de investigación. Buscar y seguir individuos	No
Licencias a nivel de Campus	Sí	Sí	No	No
Links	http://www.mendeley.com	http://biblioteca.pucp.edu.pe	http://www.zotero.org	http://www.refworks.com/

Después de analizar la tabla comparativa se optó por utilizar Mendeley ya que es un gestor de referencias bibliográficas con características avanzadas de red social que está integrado en las bases de datos comerciales, el repositorio institucional y el catálogo del CRAI de la Universidad de Barcelona. Con Mendeley podrás organizar tu investigación, colaborar con otros usuarios en línea y conocer los últimos documentos publicados en tu ámbito temático. Nos ofrece [27]:

- Crear una biblioteca personal.
- Importar documentos a la biblioteca personal y organizarlos en carpetas.
- Compartir la investigación: crear grupos públicos y privados, compartir documentos y trabajar colaborativamente en la revisión de artículos científicos.
- Añadir citas en procesadores de texto y generar bibliografías.
- Crear un perfil personal con el currículum, publicaciones y filiación.

2.1.1 Edición Institucional Mendeley (MIE)

Actualmente la Universidad de Barcelona dispone de un acceso institucional, juntamente con el resto de universidades del CSUC, para los miembros de la comunidad universitaria, que ofrece:

- Biblioteca personal de hasta 5 GB.
- Almacenamiento para grupos de investigación hasta 20 GB.
- Número ilimitado de grupos públicos y privados de hasta 25 personas.
- Recomendaciones de artículos basados en tu biblioteca personal con la opción Mendeley suggest en la versión Desktop

Mendeley trabaja con una versión de escritorio: Mendeley Desktop, y con una versión web: Mendeley Web.

Descarga Mendeley Desktop desde el botón Download Mendeley Desktop. La versión Desktop permite crear y organizar tu colección de referencias y documentos. También es posible arrastrar archivos en PDF para introducir de manera inmediata la referencia bibliográfica, así como subrayar y añadir notas a los documentos mientras los lees. También puedes compartirlos con colegas de manera privada.
Para tener siempre actualizadas las dos versiones de bibliografías y datos, es necesario sincronizarlas periódicamente con el botón Sync del Desktop.

Desde Mendeley Desktop ir al menú Tools y descargar:

- Web importer (o Save to Mendeley). Botón que es necesario instalar en los marcadores o favoritos del navegador para poder importar referencias y documentos de forma automática desde bases de datos o páginas web.

- Los plug-in MS Word Plug In, Libre Office u Open Office Plug in permiten añadir citas a documentos de texto.

2.2 Bases de datos científicos

Son recopilaciones de publicaciones de contenido científico-técnico, como artículos de revistas, libros, tesis, congresos, etc., de contenido temático, que tienen como objetivo reunir toda la producción bibliográfica posible sobre un área de conocimiento. Contienen información relevante, actualizada, precisa, contrastada y de calidad. Para todas las áreas científicas existe alguna base de datos específica o al menos alguna multidisciplinar [28].

Bases de datos de referencia mundial cubren las mejores revistas en todas las áreas temáticas, indiza más de 10000 revistas y 100000 actas de congreso [29] .

A su vez, este grupo puede dividirse [28]:

Bases de datos **multidisciplinares**: abarcan varias disciplinas científicas o técnicas.

Bases de datos **especializadas**: recopilan y analizan documentos pertinentes para una disciplina o subdisciplina concreta: investigación biomédica, farmacéutica, química, agroalimentaria, social, humanística, etc.

En las Bases de Datos la **información** está estructurada, es decir, ordenada en registros y campos.

Registros: cada registro representa un único documento (una referencia a un artículo de revista, libro, tesis, etc.).

Campos: a su vez los registros se dividen en campos. Cada campo representa un tipo de información sobre un documento, por ejemplo, el título, el autor, etc. Se identifican con una etiqueta: **Au**=autor, **TI**=título, **DE**=descriptores.

Software o interfaz de búsqueda. Es el programa informático que permite hacer búsquedas. Varía según la empresa con la que se contrate la suscripción y suele disponer de cajones de texto y menús desplegables para filtrar las búsquedas.

Ejemplo de registro:

Título	Towards an integrated crowdsourcing definition
Autor	Estelles-Arolas, Enrique [1] ; Gonzalez-Ladron-de-Guevara, Fernando
	[1] Technical University of Valencia, Spain
Autor de la correspondencia	Estelles-Arolas, Enrique
Título de publicación	Journal of Information Science
Tomo	38
Número	2
Páginas	189-200

Gráfico 4. Ejemplo de registro

Software de Búsqueda Bases de datos de libre acceso y virtuales se muestran en la Tabla 2. [30]**:**

Tabla 2. Base de datos de libre acceso

Noticier Oficial	Es un sitio WEB creado y desarrollado por la compañía BTC LTDA; con escritura pública No 2041 (Dos Mil Cuarenta y Uno) y con Número de Identificación Tributaria de la Cámara de Comercio de Bogotá. Así mismo la Dirección de Impuestos y Aduanas Nacionales DIAN**.**
SCIELO	Scientific Electronic Library Online Base de datos de acceso libre a revistas con texto completo que cubren disciplinas tales como: Arquitectura, Ciencias Agrícolas, Ciencias Biológicas, Ciencias de la Salud, Ciencias de la Tierra, Ciencias Jurídicas, Ciencias Sociales, Humanidades, Ingeniería, Matemática, Oceanógrafa y Química. De temas de América Latina y el Caribe, principalmente revistas editadas en las universidades de la región.
Redalyc	Redalyc Red de Revistas Científicas de América Latina y el Caribe, España y Portugal. Base de datos multidisciplinaria.
Dialnet	Dialnet Portal de difusión de la producción científica hispana.
DOAJ	DOAJ - Directory of Open Access Journals Cubre servicio gratis y

	completo de revistas científicas. El objetivo es cubrir todos los temas e idiomas. En la actualidad hay 2934 revistas en el directorio.
BioMed Central	Biomed Central Publicaciones en ciencia, tecnología y medicina. Pionera del movimiento de acceso abierto.
Biblioteca Virtual	Biblioteca Virtual de CervantesExilio, Venezuela, Literatura Gauchesca, Infantil y Juvenil, Historia y crítica del Cine español, etc.

Tabla 3. Bases de datos virtuales

IEEEXplore	Publica información de la más alta calidad técnica desde 1988. Comprende las siguientes áreas: ingeniería eléctrica, ciencias de la computación, telecomunicaciones y electrónica. Permite acceso a los registros y abstract plus, texto completo publicados desde 1988. Además, ofrece documentos como: revistas y magazines, memorias de conferencias (proceedings), normas y libros (sólo resúmenes).
ACM	Es la colección más completa de artículos de texto completo y los registros bibliográficos en existencia hoy abarca los campos de la informática y la tecnología de la información. La base de datos de texto completo incluye la colección completa de las publicaciones de ACM, incluyendo revistas, actas de congresos, revistas, boletines y títulos multimedia
ScienceDirect	Base de datos multidisciplinaria que cubre las áreas de Química, Ingeniería Informática, Matemáticas, Física, Astronomía, Agricultura, Biología, Biología Molecular, Inmunobiología, Microbiología, Neurociencia, Medicina, Enfermería, Farmacología, Veterinaria, Humanidades, Economía, Administración de Empresas, Sicología, Ciencias Sociales y Artes, también ofrece artículos de periódicos y capítulos de libros por más de 3500

	artículos y más de 34.000 libros.
E-libro	Son aproximadamente 24.000 títulos en inglés y 17.000 en español. Cubre las siguientes áreas: negocios, mercadeo, economía, computadores y tecnología informática, educación, ingeniería y tecnología, salud y ciencias clínicas y biomédicas, humanidades, ciencias físicas y de la vida. Para acceder a estos libros electrónicos el usuario debe ser docente, estudiante o empleado con vínculo vigente con la Institución Universitaria Pascual Bravo. **Usuario:** pascual **Clave:** bravo
EBSCO	Este paquete contiene una amplia selección de aproximadamente 180.000 títulos de libros electrónicos, artículos de revistas en todas las áreas del conocimiento que representan una gran variedad de temas académicos, asegurando a los usuarios acceso a la información relevante para sus necesidades de investigación. Plataforma que integra 14 bases de datos. Ofrece información referencial y en texto completo en diferentes áreas de conocimiento como economía, historia, derecho, literatura, filosofía, psicología, administración, religión, sociología, entre otras. **Usuario:** pascual **Clave:** bravo
Gale	Gale Virtual Reference Library es una base de datos de enciclopedias y fuentes especializadas de referencia para investigaciones multidisciplinarias disponible en línea las 24 horas del día desde cualquier lugar. La Biblioteca de La Institución Universitaria Pascual Bravo ofrece libros digitales en español de las editoriales McGraw Hill y Cengage, las cuales contienen títulos de diferentes áreas del conocimiento: administración, ciencias básicas, ingeniería y psicología.

	Estos libros son de consulta en línea, no permiten descarga. **Características del servicio:** Para acceder a estos libros electrónicos el usuario debe ser docente, estudiante o empleado con vínculo vigente en la Institución Universitaria Pascual Bravo. El acceso a este material bibliográfico está disponible dentro y fuera del campus universitario. La plataforma donde están alojados los libros:

Para la realización del trabajo de titulación se utilizó las siguientes bases de datos científicas SCOPUS, IEEE, ScienceDirect.

3. Revisiones Bibliográficas

3.1 Definición

La revisión bibliográfica se ha definido como "la operación documental de recuperar un conjunto de documentos o referencias bibliográficas que se publican en el mundo sobre un tema, un autor, una publicación o un trabajo específico [31]. Para la revisión bibliográfica como "la selección de los documentos disponibles sobre el tema, que contienen información, ideas, datos y evidencias por escrito sobre un punto de vista en particular para cumplir ciertos objetivos o expresar determinadas opiniones sobre la naturaleza del tema y la forma en que se va a investigar, así como la evaluación eficaz de estos documentos en relación con la investigación que se propone" [32] .

Las revisiones no se llevan a cabo únicamente con la finalidad de hacer investigación. La importancia de la revisión de la literatura se considera una herramienta básica para avanzar en la práctica. Puede ayudar a inspirar y generar nuevas ideas, poniendo de relieve las incoherencias en los conocimientos actuales [33]. Tienen una función importante en la evaluación de las prácticas actuales y formular recomendaciones para la elaboración de políticas y el cambio de la práctica asistencial. Por otra parte, también son útiles para el estudio de los marcos teóricos o conceptuales existentes

sobre un tema determinado o para facilitar el desarrollo de marcos teóricos o conceptuales a través de la exploración y la evaluación crítica de los conocimientos existentes [34].

3.2 Clasificación de las Revisiones Bibliográficas

Considerando la definición que dan los distintitos autores para los diferentes tipos de revisión, se pueden observar las diferentes clasificaciones y la denominación que reciben las revisiones a continuación:

- Clasificación 1.- Pertenecen a Cronin [34]

Revisión tradicional o Narrativa, Revisión Sistemática, Meta-Análisis, Meta-síntesis.

- Clasificación 2.- Pertenecen a Grant [35]

Revisión Crítica, Revisión de Literatura, Revisión Sistemática, Meta-Análisis, Revisión Sistemática Cualitativa, Revisión Panorámica, Revisión Paraguas, Revisión de Estudios Mixtos, Revisión de Mapeo Sistemático, Revisión Rápida, Revisión Sistematizada.

- Clasificación 3.- Pertenecen a Whittemore

Revisión de Integradora, Revisión Sistemática, Meta-Análisis, Síntesis Cualitativa, Revisión Panorámica, Revisión Paraguas, Revisión de Estudios Mixtos, Revisión RE-AIM.

- Clasificación 4.- Pertenecen a Goris [31].

Revisión de Narrativa, Revisión de Integradora, Revisión Panorámica, Análisis Conceptual, Revisión Sistemática, Revisión Paraguas, Revisión Realista.

3.3 Propósito de la Revisiones Bibliográficas

Primeramente, se realizó una descripción de cada uno de los tipos de revisión bibliográfica existentes, con el fin de identificar sus propósitos y elegir el tipo de revisión adecuada para desarrollar el Trabajo de Titulación. Los tipos de revisiones bibliográficas son:

Narrativa: Conocida como tradicional o crítica. Su objetivo es identificar, analizar, valorar e interpretar el cuerpo de conocimientos sobre un tema específico. El enfoque y profundidad de la revisión está en función del contexto para el que se realice. Sin embargo, la característica común en la revisión narrativa es que se revisa la literatura publicada, y ello implica que los materiales incluidos poseen cierto grado de permanencia. Ocasionalmente este proceso puede haber sido realizado por pares (dos autores de forma independiente realizan la revisión, y después ponen en común sus conclusiones, realizando un proceso de contrastación), aunque no siempre es una condición imprescindible [34].

Integradora: Este tipo de revisión fundamentalmente se centra en sintetizar el conocimiento sobre metodología, conocimientos teóricos o sobre la investigación realizada esbozando una conclusión sobre un tema específico esta propuesta puede ser confusa porque en cierto modo es parecida a la revisión narrativa, sin embargo, el propósito de la revisión integradora puede ser aportar una comprensión más profunda o incluso crear una nueva conceptualización del tema. Tiene como objetivo demostrar que el autor ha investigado ampliamente la literatura y evaluado críticamente su calidad [35].

Sistemática: Una revisión sistemática es definida como un resumen de evidencias, La diferencia más importante entre las revisión sistemática y otro tipo de revisiones es que fundamentalmente la metodología utilizada es explícita y precisa, y además se sigue un protocolo claramente delineado, estandarizado y replicable que asegura la calidad, consistencia y transparencia del proceso de revisión [35].

Meta-Análisis: El meta-análisis es la combinación cuantitativa, mediante las técnicas estadísticas adecuadas, de los resultados de investigaciones anteriores (por lo general publicadas como artículos originales). Es un tipo de diseño metodológico en sí mismo, por lo que podría considerarse investigación original, en el que las unidades de análisis son estudios originales publicados previamente sobre el tema de interés. Últimamente el meta-análisis por la magnitud que está adquiriendo es considerada en sí como un tipo de revisión [36].

Panorámica: pretende identificar los conceptos clave que sustentan un área de investigación, las principales fuentes y tipos de evidencias disponibles sobre todo cuando un área es compleja o no ha sido revisado exhaustivamente antes. Una de sus características es que no son una revisión sistemática que sea llevada a cabo con un

protocolo preestablecido por la amplia variedad de estudios que se incluyen. Otro elemento que caracteriza a este tipo de revisión es que se utiliza la técnica de mapeo conceptual, de mapeo de la bibliografía y la opinión de los usuarios (2, 16). Esta metodología contribuye a identificar vacíos y carencias en el conocimiento sobre un tema [35]

Análisis conceptual: Un tipo de revisión emergente en ciencias de la salud. El análisis conceptual es un método por el cual los conceptos que son de interés para una disciplina se examinan con el fin de aclarar sus características y conseguir una mejor comprensión del significado de ese concepto. Muchos de los conceptos que se usan en las ciencias de la salud son conceptos conductuales que se encuentran imbuidos de la forma en que se comprende el proceso de salud y enfermedad. Cuando se inicia un análisis conceptual se debe tener en cuenta que el concepto que se analiza debe ser relevante para la práctica y la disciplina [37].

Sistematizada: Las revisiones sistematizadas o revisiones estructuradas intentan incluir uno o más elementos del proceso de revisión sistemática, sin llegar a afirmar que el producto resultante es una revisión sistemática. Podría catalogarse como una "revisión sistemática" pero en muchas ocasiones carecen de algún elemento que no permite etiquetarlas como tales. Un ejemplo puede ser los trabajos de revisión que pueden realizar los estudiantes de postgrado que aun siguiendo toda la metodología de una revisión sistemática, carecen de recursos suficientes para realizarla (por ejemplo disponer de dos revisores que evalúen los documentos sometidos a análisis) [38].

Revisión de revisiones o paraguas: Las revisiones paraguas o revisión de revisiones son un nuevo tipo de revisiones que están ganando un creciente interés, de manera que el número de reseñas publicadas es cada vez mayor. Estas revisiones se centran fundamentalmente en resumir la evidencia disponible. Pueden ser utilizadas para evaluar las similitudes y diferencias en las revisiones publicadas, para resumir lo que se sabe sobre un tema y normalmente implican un amplio número de diferentes tipos de revisiones. Este tipo de revisiones tienen limitaciones por la calidad metodológica utilizada al realizar este tipo de revisiones y que deberían establecerse normas metodológicas y directrices para mejorar la calidad de este nuevo tipo de publicación.

Realista: Este tipo de revisiones han surgido como respuesta a la complejidad que tiene el diseño de políticas de intervención en salud. Los métodos tradicionales de

análisis de las políticas de salud se centran en la medición y Muy frecuentemente estos informes encuentran que la evidencia es confusa, y ofrecen poca o ninguna idea de porqué la intervención funcionó o no funcionó cuando se aplica en diferentes contextos o circunstancias. La revisión realista está diseñada para trabajar con las intervenciones o programas sociales complejos, de modo que en el enfoque emerge una evaluación "realista". Este tipo de revisiones proporciona un análisis explicativo dirigido a discernir lo que funciona, en quién, en qué circunstancias, y en qué aspectos y cómo funciona [31].

3.4 Selección de la Revisión del Estado de arte

Una vez identificado el propósito de las Revisiones Bibliográficas se elaboró una tabla comparativa que permitió reconocer el método adecuado a utilizar, en la **Tabla 4** podemos visualizar los aspectos que se calificó en donde se resalta si la revisión es cualitativa (si=1, no=0), cuantitativa (si=1, no=0), la profundidad de esta propuesta de acuerdo al nivel de estudio presentado en la revisión el mismo que puede ser básico (1), medio (2), avanzado (3), y si aplica a la ingeniera (si=1, no=0).

Tabla 4. Comparación de las revisiones bibliográficas.

Revisión	Cualitativa	Cuantitativa	Profundidad	Para ingeniería	Total
Revisión Narrativa:	1	0	1 0	0	2
Revisión Integradora	1	0	2	0	3
Revisión Sistemática	1	1	2	1	5
Meta-Análisis	0	1	3	1	5
Revisión Panorámica	1	0	1	0	2
Análisis conceptual	1	0	1	0	2
Revisión Sistematizada	1	1	1	1	4
Revisión Realista	1	0	1	0	2
Revisión paraguas	1	0	1	0	2

Se consideró los siguientes aspectos para la selección de la Revisión Bibliográfica:

- Las Revisiones Sistemáticas son de las pocas que se aplican a la Ingeniería.

- Cuentan con un protocolo de revisión claramente delineado y estandarizado que asegura la transparencia del proceso.
- Identifica las lagunas en la investigación actual con el fin de sugerir nuevas áreas de estudio.
- Permite la evaluación constante durante el transcurso de la revisión.
- Por su rigurosidad y eficiencia esta entre las revisiones bibliográficas aceptadas como para trabajos de titulación.
- Se puede realizar un análisis estadístico de todo el proceso realizado.

En vista de que para el Trabajo de Titulación se tenía previsto desarrollar una investigación en la que se pueda analizar la calidad y la cantidad de información, además se necesitaría una revisión que este en un punto medio de calidad de investigación y que se pueda aplicar a la ingeniería, se tomó como la mejor elección a la Revisión Sistemática de Kitchenham, la misma que cumple con los aspectos propuestos anteriormente.

3.5 Revisión Sistematice de Barbara Kitchenham

Definición: Una revisión sistemática de la literatura es un medio para identificar, evaluar e interpretar todas las investigaciones disponibles acerca de una pregunta en particular la investigación, o área temática, o fenómeno de interés. Los estudios individuales que contribuyen a una revisión sistemática se denominan estudios primarios.

Importancia: La mayoría de la investigación se inicia con una revisión de la literatura de algún tipo. Sin embargo, a menos que una revisión de la literatura sea exhaustiva y justa, es de poco valor científico. Esta es la principal razón para la realización de revisiones sistemáticas. Una revisión sistemática sintetiza el trabajo existente de una manera que sea justa y visto para ser justos. Por ejemplo, las revisiones sistemáticas deben llevarse a cabo de acuerdo con una estrategia de búsqueda predefinida. La estrategia de búsqueda debe permitir que la exhaustividad de la búsqueda para ser evaluado. En particular, los investigadores que realizan una revisión sistemática deben hacer todos los esfuerzos para identificar y reportar investigación que no apoyan su hipótesis de investigación preferente, así como la identificación y presentación de informes de investigación que lo soporta.

Proceso de Revisión

Una revisión sistemática de la literatura implica varias actividades. Las directrices existentes para las revisiones sistemáticas tienen ligeramente diferentes sugerencias sobre el número y el orden de las actividades. Sin embargo, Barbara Kitchenham las plantea como se muestra en la Tabla 5.

Tabla 5. Protocolo de Barbara Kitchenham

REVISIÓN SISTEMÁTICA	
ETAPA 1	**PLANIFICACIÓN DE LA REVISIÓN.**
	• Identificación de la necesidad de una revisión. • Especificación de la pregunta(s) de investigación. • El desarrollo de un protocolo de revisión.
ETAPA 2	**EJECUCIÓN DE LA REVISIÓN.**
	• Identificación de la investigación. • Selección de los estudios primarios. • La extracción de datos y el seguimiento. • Síntesis de los datos.
ETAPA 3	**INFORME DE LOS RESULTADOS.**

1) Planificación

Antes de llevar a cabo una revisión sistemática es necesario confirmar la necesidad de un examen. Las actividades más importantes antes de la revisión son la definición de la pregunta de investigación, se define los procedimientos básicos de revisión. El protocolo de examen debe estar sujeta a un proceso de evaluación independiente. Esto es particularmente importante para un estudio encargado.

1.1) Identificación de la necesidad de una revisión.

La necesidad de una revisión sistemática surge de la exigencia de los investigadores para resumir toda la información existente sobre algún fenómeno de manera exhaustiva e imparcial. Esto puede ser con el fin de sacar conclusiones más generales sobre algún fenómeno que es posible a partir de los estudios individuales, o puede llevarse a cabo como preludio a otras actividades de investigación.

1.2) Especificación de la pregunta(s) de investigación.

Las preguntas de la investigación es la parte más importante de cualquier revisión sistemática. Las preguntas ponen en marcha toda la metodología de revisión sistemática por lo tanto la elaboración debe considerar que:

• El proceso de búsqueda debe identificar los estudios primarios que abordan las cuestiones de investigación.

• El proceso de extracción de datos debe extraer los elementos de datos necesarios para responder a las preguntas.

• El proceso de análisis de datos debe sintetizar los datos de tal manera que las preguntas pueden ser contestadas.

1.3) El desarrollo de un protocolo de revisión.

Un protocolo de revisión especifica los métodos que se utilizarán para llevar a cabo una revisión sistemática específica. Un protocolo predefinido es necesario reducir la posibilidad de sesgo investigador. Por ejemplo, sin un protocolo, es posible que la selección de los estudios individuales o el análisis puedan ser impulsados por las expectativas del investigador.

2) Ejecución

Una vez que el protocolo ha sido acordado, el examen adecuado puede comenzar. Sin embargo, como se señaló anteriormente, los investigadores deben esperar a probar

cada uno de los pasos que se describen en esta sección cuando construyen su protocolo de investigación.

2.1) Identificación de la investigación

El objetivo de una revisión sistemática es encontrar el mayor número de estudios primarios relacionados con la pregunta de investigación como sea posible utilizando una estrategia de búsqueda imparcial. El rigor del proceso de búsqueda es un factor que distingue a las revisiones sistemáticas de las revisiones tradicionales.

Es necesario determinar y seguir una estrategia de búsqueda. A continuación, elaborar una lista de palabras clave que van a servir para construir sofisticadas cadenas de búsqueda utilizando los operadores lógicos AND y OR para concatenar. Paso seguido se especifica las fuentes de búsqueda y se procede a realizar las consultas.

Como punto final a este paso se identifica los criterios de inclusión y exclusión que se va aplicar a los estudios encontrados.

2.2) Selección de los estudios primarios

Una vez que se han obtenido los estudios primarios potencialmente relevantes, que deben ser evaluados por su importancia real. Para esto se describe los criterios de selección que se utilizan para identificar los estudios primarios que proporcionan evidencia directa acerca de la pregunta de investigación. Con el fin de reducir la probabilidad de sesgo, los criterios de selección se decidirán durante la definición del protocolo, aunque pueden ser refinados durante el proceso de búsqueda.

2.3) La extracción de datos y el seguimiento.

El objetivo de esta etapa es el diseño de formularios de extracción de datos para registrar con precisión la información de los investigadores que obtienen a partir de los estudios primarios. Para reducir la posibilidad de sesgo, los formularios de extracción de datos deben ser definidos y puestos a prueba cuando se define el protocolo de revisión.

2.4) Síntesis de los datos.

La síntesis de los datos consiste en recopilar y resumir los resultados de los estudios primarios incluidos. Síntesis puede ser descriptivo (no cuantitativa). Sin embargo, a veces es posible complementar una síntesis descriptiva con un resumen cuantitativo utilizando técnicas estadísticas.

3) Publicación de resultados.

La fase final de una revisión sistemática consiste en la redacción de los resultados de la revisión y difusión de los resultados a las partes potencialmente interesadas. Es importante comunicar los resultados de una revisión sistemática de manera efectiva. Por esta razón la mayoría de las guías recomiendan la planificación de la estrategia de difusión durante la etapa de puesta en marcha (si existe) o la hora de preparar el protocolo de revisión sistemática.

Por lo general, las revisiones sistemáticas serán notificadas en al menos dos formatos:

• En un informe técnico o en una sección de una tesis doctoral.

• En un artículo de revista o conferencia.

f. Materiales y métodos

1. Materiales

Detallamos algunos materiales en la Tabla 6, que fueron necesarios para el desarrollo del trabajo de titulación.

Tabla 6. Materiales

Hardware	
Material	**Detalle**
Computadora	Para realizar la digitación de todo el proceso
Impresora	Se utilizó en la impresión del trabajo cada que fue necesario.
Dispositivo de almacenamiento	Se utilizó una memoria USB y distintas herramientas de la nube como Dropbox, Mendeley.
Software	
Material	**Detalle**
Licencia Windows	Contribuyó en el funcionamiento del sistema operativo da la computadora.
Mendeley Desktop	Permitió la gestión bibliográfica para citar o referenciar desde el procesador de texto.
Procesador texto	Facilito la edición del texto y revisión online.
Materiales Varios	
Material	**Detalle**
Papel	Para la impresión de avances y presentación del trabajo de titilación.
Internet	La principal herramienta pues permitió el acceso a diferentes herramientas utilizadas en la elaboración de este trabajo.
Transporte	Ayudó a trasladarme y acudir al sitio necesario para culminar el proyecto.
Accesorios de oficina	Necesario para el desarrollo y culminación del trabajo de titulación.
Fuentes de búsqueda	
Material	**Detalle**
IEEE	Base de datos científica que se utilizó para obtener documentos para el desarrollo de la investigación.
Scopus	Base de datos científica que se utilizó para obtener documentos para el desarrollo de la investigación.
ScienceDirect	Base de datos científica que se utilizó para obtener documentos para el desarrollo de la investigación.

2. Métodos

- **Método de Revisiones Sistemáticas de Barbara Kitchenham:** siendo un medio para evaluar e interpretar todas las investigaciones disponibles acerca de una pregunta en particular de investigación, área temática, o fenómeno de interés, constituyéndose en el más conveniente para el desarrollo de este trabajo de titulación. Al realizar una revisión sistemática debemos cumplir tres etapas: la planificación de la revisión, desarrollo de la revisión e informe de resultados.

El resultado de la revisión sistemática y extracción de información que se presenta en este trabajo, se basa en [14][39]. Se compone en tres fases:

1. Planificación de la revisión.

Se desarrollan tres actividades que son:

1.1 Identificación de la necesidad de una revisión

En la actualidad las vulnerabilidades de las aplicaciones web, se han convertido, en una gran amenaza para la seguridad de sistemas informáticos, por que dependemos de estas aplicaciones para llevar acabo un sin número de actividades como transacciones bancarias, correos electrónicos, redes sociales, etc. y se debe conocer cuáles han sido los avances en canto a seguridad de las aplicaciones web, siendo un tema de mucho interés en la actualidad se decide Realizar una revisión sistemática de literatura: seguridad en ambientes web utilizando framework, con la finalidad de contribuir con algunas de las características que puedan servir y aportar con este tema.

1.2 Especificación de la pregunta de investigación

Se dirigió el alcance de este trabajo y planteo como objetivo “Realizar una revisión sistemática de cómo afecta las vulnerabilidades en ambientes web” sobre artículos relacionados a mecanismos de seguridad aplicando frameworks de desarrollo. En base a esto la pregunta de investigación planteada es:

¿Qué investigaciones primarias existen sobre mecanismos de seguridad en los Frameworks?

1.3 Desarrollo de un protocolo de revisión

El protocolo a seguir para la Revisión Sistemática es el propuesto por Barbara Kitchenham mostrado en la Tabla 5, que se compone de tres fases y cada fase tiene sus respectivas actividades.

2. Desarrollo de la revisión.

2.1 Identificación de la investigación.

La búsqueda para la selección de artículos de calidad, se llevó acabo en bases de datos científicas en línea accesibles, que permitan realizar búsquedas avanzadas sobre el área de investigación. Las bases de datos contienen trabajos publicados de mucho interés en la revisión y son utilizadas por muchos investigadores de Ingeniería en software. Varios artículos fueron descargados de las diferentes bases de datos en función de su relevancia.

Fuentes y estrategias de búsqueda se detallan a continuación en la Tabla 7.

Tabla 7. Bases de datos científicas

Bases de datos	URL
SCOPUS Library	https://www.scopus.com.
SCIENCEDIRECT Library	http://www.sciencedirect.com.
IEEEXPLORE Library	http://ieeexplore.ieee.org/

Se consideró para la elección de palabras claves lo siguiente: una revisión de literatura previa, que consistió en analizar algunos documentos relacionados (títulos, resúmenes e introducción) que se muestran en la Tabla 8, y la pregunta de investigación.

Tabla 8. Revisión preliminar y términos

Título	Palabras clave
Towards SQL Injection Attacks Detection Mechanism Using Parse Tree	SQL injection attacks, parse tree, detection, web environments [40]
Securing Web Applications from Injection and Logic Vulnerabilities: Approaches and Challenges	SQL injection, Cross-site scripting, Business logic vulnerabilities, Application logic vulnerabilities, Web application security, Injection flaws [12]
Effective detection of vulnerable and malicious browser extensions	Browser extensions, Web security, Malware, Hidden Markov Model, JavaScript [41]
Mitigating SQL Injection Attacks Via Hybrid Threat Modelling	SQL Injection Attacks, Software Security, SDLC, SSDL, Hybrid Threat Modeling, Attack Trees, Misuse Cases, State Machines [42].

Una vez obtenidos los resultados en la Tabla 8, se definieron las siguientes palabras clave:

Security in web environments, framework, security mechanisms, vulnerabilities, vulnerabilities in web environments.

Para generar la cadena de búsqueda se utilizaron los operadores lógicos “OR” y “AND”. Tomando en cuenta los estudios como: artículos de revistas y conferencias. Las cadenas de búsquedas(C) utilizadas se detallan en la Tabla 9.

Tabla 9. Cadenas de búsqueda.

Biblioteca digital de SCOPUS
C01: TITLE-ABS-KEY (security in web environments AND framework)
C02:TITLE-ABS-KEY (security in web environments OR security mechanisms

AND framework OR vulnerabilities AND vulnerabilities in web environments)
C03:TITLE-ABS-KEY (security mechanisms AND vulnerabilities in web environments AND framework.)
C04: TITLE-ABS-KEY (security mechanisms OR vulnerabilities in web environments AND framework)
C05: TITLE-ABS-KEY (vulnerabilities in web environments AND framework.)
Biblioteca digital de SCIENCEDIRECT
C06: Security in web environments and framework
C07: Security in web environments or security mechanisms and framework or vulnerabilities and vulnerabilities in web environments
C08: security mechanisms and vulnerabilities in web environments and framework.
C09: security mechanisms or vulnerabilities in web environments and framework.
C10: vulnerabilities in web environments and framework.
Biblioteca digital de IEEE
C11: (Security in web environments and framework)

2.2 Selección de estudios primarios.

Obtenidos los resultados de las búsquedas es conveniente describir el criterio a seguir para la selección de estudios primarios basados en la pregunta de investigación y según el protocolo de Kitchenham deben ser orientados a garantizar que los estudios puedan ser interpretados de forma fiable y que se clasifiquen correctamente, considerando los siguientes:

Criterios de inclusión.

- Considerar sólo publicaciones desde el año 2014 en adelante.
- Los resultados de la búsqueda solo sean en el área de Ciencias y Computación.

- La búsqueda por su relevancia científica será en el idioma inglés.
- El resumen del estudio debe contener, información actual sobre mecanismos de seguridad en los Frameworks.

Criterios de exclusión.

Los estudios que no han sido relevantes en este estudio se han excluido mediante los siguientes criterios:

- Los estudios que no contengan información relevante a la pregunta de investigación.
- Que no contengan información relevante para la revisión sobre mecanismos de seguridad en aplicaciones web en la introducción o conclusión.
- Artículos informales y que no contengan información relevante.
- Estudios que no cumplan con los criterios de inclusión.

Los resultados generales de las búsquedas se presentan en la Tabla 10.

Tabla 10. Resultados de la fase de selección de artículos incluidos y excluidos.

Biblioteca	Cadena de búsqueda	Total, de estudios	Estudios incluidos	Estudios excluidos
SCOPUS	C01	768	83	685
	C02	52	8	44
	C03	18	4	14
	C04	203	23	180
	C05	81	13	68
ScienceDirect	C06	25187	92	25095
	C07	5146	65	5081
	C08	3419	46	3373
	C09	17494	85	17409
	C10	8423	61	8362
IEEE	C11	392	40	352
Total		61183	520	60663

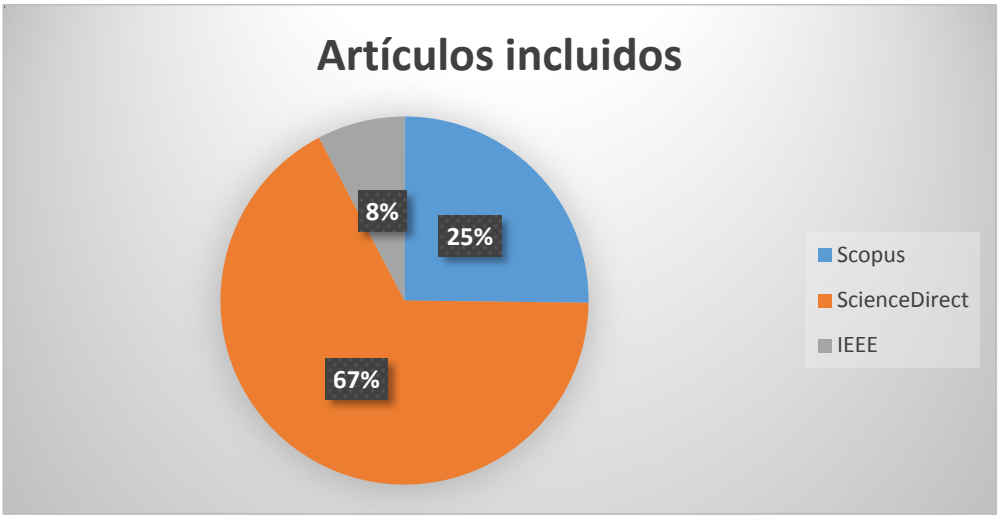

Gráfico 5. Artículos por base de datos.

Una vez aplicados los criterios de inclusión y exclusión se generaron como resultado 520 documentos siendo: el 67% de la librería ScienceDirect, el 25% de Scopus y el 8% de IEEE (Gráfico 5). En el Gráfico 6 se muestra, que el año 2015 en la base de datos ScienceDirect hay el mayor número de publicaciones. Del conjunto de resultados se registró 294 coincidencias y el número de artículos revisados fueron 226 que se detallan en la Tabla 25 del Anexo 1.

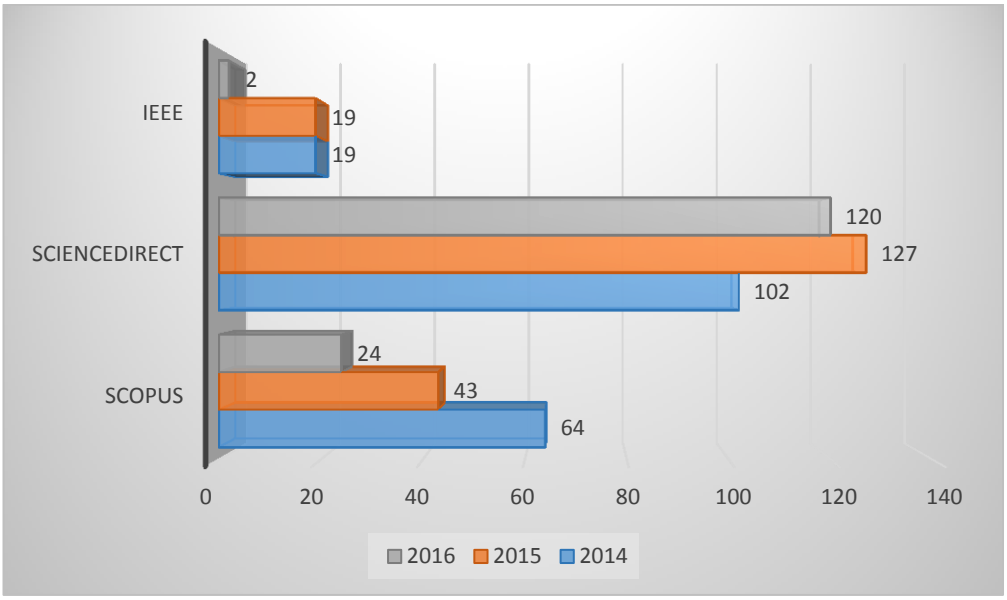

Gráfico 6. Estudios incluidos por año.

2.3 Extracción de datos y seguimiento.

Para la selección de estudios basados en la Tabla 25 del Anexo 1, se aplicaron los criterios de selección de estudios que establecen la pauta de extracción de información relevante para este trabajo. Por cada artículo seleccionado, se sintetizaron al menos uno de los siguientes elementos:

- Propuestas o técnicas de solución.
- Problemas de seguridad
- Resultados de propuesta
- Conclusiones relevantes.

2.4 Síntesis de datos.

El criterio utilizado para la selección de artículos fue que aportaran sobre la existencia de mecanismos de seguridad en los framework. Dando como resultado un total de 13 artículos.
La información detallada de los puntos **2.3** y **2.4** se muestra en la sección de Resultados.

Artículos Seleccionados en La Revisión Sistemática.

- **SA01:** H. Asghar, Z. Anwar, and K. Latif, “A Deliberately Insecure RDF-based Semantic Web Application Framework for Teaching SPARQL / SPARUL Injection Attacks and Defense Mechanisms,” Comput. Secur. 2015.

- **SA02:** T. N. Aung and S. S. Khaing, “Towards SQL Injection Attacks Detection Mechanism Using Parse Tree,” Adv. Intell. Syst. Comput., vol. 388, pp. 405–411, 2016.

- **SA03:** A. Costin, “Automated Dynamic Firmware Analysis at Scale: A Case Study on Embedded Web Interfaces ∗,” pp. 437–448.

- **SA04:** E. Ferry, J. O. Raw, K. Curran, E. Ferry, and J. O. Raw, “Security evaluation of the OAuth 2. 0 framework,” 2015.

- **SA05:** S. Gupta and B. B. Gupta, “CSSXC: Context-Sensitive Sanitization Framework for Web Applications against XSS Vulnerabilities in Cloud

Environments," Procedia - Procedia Comput. Sci., vol. 85, no. Cms, pp. 198–205, 2016.

- **SA06:** D. Kar, S. Panigrahi, and S. Sundararajan, "SQLiGoT: Detecting SQL Injection Attacks using Graph of Tokens and SVM," Comput. Secur., 2016.
- **SA07:** V. R. Mouli and K. P. Jevitha, "Web Services Attacks and Security- A Systematic Literature Review," Procedia - Procedia Comput. Sci., vol. 93, no. September, pp. 870–877, 2016.
- **SA08:** R. A. Oliveira, N. Laranjeiro, and M. Vieira, "Assessing the security of web service frameworks against Denial of Service attacks," J. Syst. Softw., vol. 109, pp. 18–31, 2015.
- **SA09:** A. M. Ortiz, E. Rios, W. Mallouli, E. Iturbe, E. M. De Oca, R. Montimage, and D. Paris, "Self-protecting multi-cloud applications," no. SPC, pp. 643–647, 2015.
- **SA10:** O. Z. De Paiva, W. V. Ruggiero, A. Prof, and L. Gualberto, "A Survey on Information Flow Control Mechanisms in Web Applications," pp. 211–220, 2015.
- **SA11:** A. Patel, S. Al-janabi, and I. Alshourbaji, "ScienceDirect A novel methodology towards a trusted environment in mashup web applications," Comput. Secur., vol. 49, pp. 107–122, 2014.
- **SA12:** M. S. Tajbakhsh, "A Sound Framework for Dynamic Prevention of Local File Inclusion," 2015.
- **SA13:** D. A. Vishwavidyalaya and D. A. Vishwavidyalaya, "Secure Integrated Framework for Business Processes," 2015.

3. Informe de resultados.

El presente trabajo sirvió como base para generar un artículo secundario que fue presentado el 15 de septiembre en la plataforma EasyChair, y ha sido aprobado para su presentación y publicación en la revista Latin American Journal of Computing-LAJC.

g. Resultados

1. Extracción de la información

Una vez aplicados los criterios de selección se obtuvo 13 artículos, de los cuales se extrae la información más relevante. Las tablas de la 11 a la 23 muestran la información extraída de cada documento seleccionado. Información como: título del artículo seleccionado, propuesta o técnica de solución, problemas de seguridad abordados, resultados de la propuesta y las conclusiones relevantes presentadas en cada documento analizado.

Tabla 11. Resultados del artículo SA01.

Artículo	A Deliberately Insecure RDF-based Semantic Web Application Framework for Teaching SPARQL/SPARUL Injection Attacks and Defense Mechanisms
Propuesta o técnica de solución.	Resource Description Framework (RDF), framework WebGoat donde los desarrolladores de aplicaciones pueden experimentar con seguridad y explorar vulnerabilidades en aplicaciones Web Semántica. También proporcionamos un conjunto personalizado de reglas ModSecurity WAF que SemWebGoat los usuarios pueden practicar y ser capaz de detectar los ataques de inyección SPARQL / sparul.
Problemas de seguridad	Ataques de inyección para la Web Semántica: SPAROL / SPARUL, inyección SQL, XML, Dos, XSS, CSRF, Cross-Site.
Resultado de la propuesta	Los desarrolladores deberían estar conscientes de las vulnerabilidades de aplicaciones Web Semántica. La tasa de detección del conjunto de reglas personalizadas ModSecurity ilustra como una medida de defensa viable para bloquear los ataques de inyección SPARQL / sparul. La evaluación también estableció que existe una necesidad de mejorar los escáneres de aplicaciones web para la detección automática de vulnerabilidades / sparul SPARQL.
Conclusiones relevantes	Los desarrolladores web no están familiarizados con las vulnerabilidades de inyección demostradas. Las aplicaciones de Web Semántica son propensas a ataques de inyección que pueden permitir a un atacante acceder o modificar datos no autorizados.

Tabla 12. Resultados del artículo SA02.

Artículo	Towards SQL Injection Attacks Detection Mechanism Using Parse Tree
Propuesta o técnica de solución.	Framework DSD (Dynamic Detección SQLIAs) para contrarrestar SQLIAs en entornos web, mediante el uso de árbol de análisis sintáctico. No requiere acceder al código fuente de las aplicaciones. El DSD puede ser directa y fácilmente incorporado para entornos web existentes.
Problemas de seguridad	Inyección SQL
Resultado de la propuesta	La exactitud del mecanismo es del 99.9%. El mecanismo tiene una mayor precisión, menor tasa de falsos positivos y bajo consumo de tiempo.
Conclusiones relevantes	El mecanismo de detección de SQLAs, para entornos web es eficiente por su alta tasa de precisión, baja tasa de falsos positivos, y bajo consumo de tiempo.

Tabla 13. Resultados del artículo SA03.

Artículo	Automated Dynamic Firmware Analysis at Scale: A Case Study on Embedded Web Interfaces
Propuesta o técnica de solución.	Presenta una nueva metodología, para ejecutar análisis a gran escala de la seguridad de interfaces web dentro de los dispositivos embebidos. Para ello, diseñan el primer framework que logre hacer análisis dinámico escalable y automatizada del firmware, y que fue desarrollado precisamente para descubrir vulnerabilidades en dispositivos embebidos, utilizando el enfoque de software solamente. El framework aprovecha off-the-shelf herramientas de análisis estático y dinámico.
Problemas de seguridad	Inyección de comandos, XSS, CSRF.
Resultado de la propuesta	Se encontraron vulnerabilidades en al menos el 24% de las interfaces web, incluyendo 225 de alto impacto, vulnerabilidades encontradas y verificadas por análisis dinámico. El análisis estático reportó 145 imágenes únicas firmware para exponer 9046 vulnerabilidades posibles. Agregados los informes de análisis estático y dinámico de activación periódica, un total de 185 imágenes de firmware son responsables de vulnerabilidades de

	9271, afectando casi una cuarta parte de los proveedores de su base de datos.
Conclusiones relevantes	Expone que en algunos sistemas embebidos los fabricantes tienen que empezar a considerar la seguridad en ciclo de vida del software, por ejemplo, mediante el uso off-the-shelf escáneres de seguridad como parte de su garantía de calidad del producto.

Tabla 14. Resultados del artículo SA04.

Artículo	Security evaluation of the OAuth 2.0 framework
Propuesta o técnica de solución.	Framework OAuth se compone de tres partes: una aplicación cliente necesaria para consumir servicios OAuth habilitados, un servidor de autorización para gestionar el acceso al servidor de recursos, y un servidor de recursos que exponga los datos de la base de datos en base a la autorización del usuario, se dan desde el servidor de autorización.
Problemas de seguridad	Cross Site Request Falsificación (CSRF), los ataques de inyección SQL, XSS (Cross Site Scripting)
Resultado de la propuesta	La solución resultó no ser vulnerable a la mayoría de las amenazas. Muchos de los problemas identificados en el modelo de amenaza fueron mitigados mediante la implementación de cifrado de extremo a extremo. El artefacto, sin embargo, no se protegió de la amenaza de suplantación de recursos y ataques de denegación de servicio en el servidor de recursos. Aquí es donde una aplicación cliente malicioso mediante programación, completa todo el flujo del lado del servidor a través de OAuth peticiones POST sin el conocimiento del usuario. El usuario debe tener una sesión válida con el proveedor OAuth para llevar a cabo este ataque.
Conclusiones relevantes	Se logró un alto nivel de seguridad. Seguridad a través de sus mensajes cifrados y firmados, pero su complejidad hace que sea difícil de integrar. Para crear un amplio conjunto de aplicaciones web universalmente integrables, los vendedores tienen que alinear su OAuth.

Tabla 15. Resultados del artículo SA05.

Artículo	CSSXC: Context-Sensitive Sanitization Framework for Web Applications against XSS Vulnerabilities in Cloud Environments
Propuesta o técnica de solución.	Un Framework de defensa XSS robusta para aplicaciones web basadas en HTML 5 desplegados en el entorno de nube, que detecta y mitiga las vulnerabilidades XSS. Descubre todos los puntos de inyección ocultos en las aplicaciones web en la nube y lleva a cabo la desinfección contextual en las cargas útiles de ataque XSS inyectado en esos puntos para mitigar el efecto de las vulnerabilidades XSS de estas aplicaciones web.
Problemas de la seguridad	Ataques XSS
Resultado de la propuesta	El análisis realizado revela que el marco propuesto exhibe un alto rendimiento como el valor observado de F-Medidas en todas las plataformas de aplicaciones web en 0,9. El Framework tiene una alta tasa de precisión y de impacto casi insignificante en el rendimiento de las aplicaciones Web desplegadas en las plataformas en la nube.
Conclusiones relevantes	Se integra fácilmente en las plataformas de nube existentes y se evaluó en la suite de prueba de la aplicación web en el mundo real.

Tabla 16. Resultados del artículo SA06.

Artículo	SQLiGoT: Detecting SQL Injection Attacks using Graph of Tokens and SVM
Propuesta o técnica de solución.	SQLiGoT (Detección de inyección SQL mediante Gráfica de tokens). Una nueva técnica o método para detectar SQLIA mediante: la conversión de una consulta SQL en una secuencia de tokens conservando su composición estructural, generar un gráfico que consiste en tokens como los nodos y la interacción entre ellos como bordes ponderados, la formación de una SVM (máquina de vectores soporte) clasificador utilizando la medida de centralidad de los nodos, y utilizando el clasificador para identificar las consultas maliciosos en tiempo de ejecución.
Problemas de seguridad	Inyección SQL

Resultado de la propuesta	Los resultados experimentales demuestran que esta técnica puede identificar con eficacia las consultas SQL maliciosas con sobrecarga de rendimiento insignificante. El 99,47% de precisión y 0,31% tasa de falsos positivos.
Conclusiones relevantes	El sistema no requiere la construcción de un modelo de uso normal de las consultas, ni requiere el acceso al código fuente. Se puede proteger múltiples aplicaciones web alojadas en un servidor compartido, lo que le da una ventaja sobre los métodos existentes

Tabla 17. Resultados del artículo SA07

Artículo	Web Services Attacks and Security- A Systematic Literature Review
Propuesta o técnica de solución.	Análisis dinámico, análisis estático analiza una aplicación web, su código fuente para construir modelos de consultas legítimas, modelo basado en razonamiento de caso, programación segura,
Problemas de seguridad	Inyección SQL, XML ataque de inyección, XPath ataque de inyección, ataque de Denegación de servicio.
Resultado de la propuesta	La mayoría de los ataques abordados son ataques de denegación de servicio, seguido de ataques de inyección XML. Las técnicas para hacer frente a los ataques predominantemente se centran en medidas de detección de ataques.
Conclusiones relevantes	Los ataques del servicio web no se pueden eliminar por completo, la penetración y la automatización de pruebas se deben realizar como parte de cada desarrollo. Esto garantizará mayor protección a los servicios web

Tabla 18. Resultados del artículo SA08.

Artículo	Assessing the security of web service frameworks against Denial of Service attacks
Propuesta o técnica de solución.	Un enfoque experimental para evaluar la seguridad de los framework de servicios web, consistió en el despliegue de una WSFAggressor cliente en funcionamiento que se basa en un conjunto de fases que incluyen la ejecución de un gran número de ataques de DoS contra un framework objetivo y la clasificación del comportamiento observado. Los frameworks seleccionados son: Metro de Oracle, Apache CXF, Apache Axis 2, Apache Axis 1,

	Primavera JAX-WS, Spring-WS, y XINS.
Problemas de seguridad	Denegación de servicio (DoS)
Resultado de la propuesta	Muestran que cuatro de los seis marcos probados son vulnerables a por lo menos un tipo de ataque DoS, e indican que incluso plataformas muy populares requieren mejoras urgentes de seguridad.
Conclusiones relevantes	Define un enfoque de pruebas de múltiples fases, con el objetivo de observar el comportamiento de las plataformas de servicio durante los ataques y detectar cualquier posible efecto de ataque durante el servicio normal u operación de inactividad del sistema. En la práctica, los desarrolladores y proveedores pueden utilizar el enfoque propuesto para evaluar la seguridad de las plataformas de servicio.

Tabla 19. Resultados del artículo SA09.

Artículo	Self-protecting multi-cloud applications
Propuesta o técnica de solución.	Framework MUSA una solución global para apoyar la seguridad de todo el ciclo de vida de aplicaciones multi-nube, proporcionando de vigilancia y seguridad de mecanismos de aseguramiento avanzados en entornos multi-nube.
Problemas de seguridad	Problemas de seguridad en aplicaciones multi-nube.
Resultado de la propuesta	No se presentan resultados.
Conclusiones relevantes	La aplicación innovadora de MUSA proporcionará controles de seguridad y mecanismos de aplicación que abarca múltiples áreas de seguridad relacionados con la protección de datos (confidencialidad de los datos, integridad de los datos, la localización de datos, acceso a datos).

Tabla 20. Resultados del artículo SA10.

Artículo	A Survey on Information Flow Control Mechanisms in Web Applications
Propuesta o técnica de solución.	Técnicas IFC (Información de control de flujo) utilizado para proteger la confidencialidad e integridad de la información. Funciona mediante el seguimiento de flujos de información entre los elementos vistos como fuentes y /o destinos de datos. AIM-ing para eliminar las vulnerabilidades mencionadas o para mitigar sus efectos.
Problemas de seguridad	Inyección SQL, Cross-Site Scripting
Resultado de propuesta	Con el fin de establecer un escenario comparativo entre los trabajos presentados, los clasificamos en 3 dimensiones "no ortogonales": (1) garantías de seguridad en el servidor: la mayor parte de los trabajos presentados (bestos AS, SafeWeb, SIF y Swift) tienen como objetivo proteger BE- aplicaciones web benignas contra los insectos y los usuarios maliciosos., (2) la protección del código del lado del cliente: SIF y Swift es naturalmente capaz de regular los flujos del lado del cliente debido a su modelo de programación de nivel superior que abarca código de cliente y servidor, que permite una verificación de tipos de códigos de ambos en un solo paso. Y (3) los impactos de programabilidad: SIF y Swift imponen los dos más grandes impactos de programación, ya que se basan en una versión muy modificada de Java, en la que los programadores se encargan de la tarea de asignar las anotaciones de la etiqueta a las variables, clases, métodos y argumentos.
Conclusiones relevantes	Dadas las garantías y los inconvenientes de las técnicas actuales de la CFI, se acerca a los impactos de programabilidad baja intensidad, como SafeWeb o algún OS- CFI, se recomiendan en un escenario de aplicación benigna.

Tabla 21. Resultados del artículo SA11.

Artículo	A novel methodology towards a trusted environment in mashup web applications
Propuesta o técnica de solución.	Framework de seguridad Mashup examina, analiza y evalúa los datos de transición entre el servidor y el lado del cliente (en la arquitectura cliente-servidor), para asegurar que las fuentes de

	datos son seguros contra amenazas de seguridad y actividades maliciosas. El framework implementa un mecanismo de protección de varios niveles, que clasifica la fuente de datos sobre los orígenes de fuentes confiables y no confiables. La clasificación se basa en un proceso de análisis de riesgos en línea y un monitoreo en línea de los cambios de los datos de intercambios entre los proveedores de la API (interfaz de programación de aplicaciones) y el navegador del cliente mediante la medición de los riesgos residuales y la sensibilidad de las áreas y los activos que se requieren para ser visitada.
Problemas de seguridad	Cross-Site Scripting (XSS)
Resultado de propuesta	El modelo de seguridad actual no satisface las necesidades del usuario y los requisitos de seguridad de protección que hacen que los sistemas de información que ejecuta la aplicación mashup sean vulnerables. Esto significa que un nuevo marco de seguridad debe ser construido desde cero con el fin de tener una forma única y eficaz de protección.
Conclusiones relevantes	El marco de seguridad propuesto se compone de tres módulos principales: 1) El módulo de monitoreo, Fuente encargada de validar y autenticar las citas de la fuente de origen del proveedor de servicios de terceros mediante el control de los enlaces URL de fuentes incrustadas en la escritura de HTML de la página antes de que se integró y se combina con el resto del documento solicitante. 2) El segundo módulo fue el flujo de datos del monitor, que era responsable de controlar el flujo e intercambio de datos entre el navegador y el servidor remoto. También fue responsable de controlar el acceso a áreas restringidas predefinidas en las máquinas de los clientes y para prevenir la exposición de los datos confidenciales, privados y protegidos. 3) El seguimiento de la ejecución, que era el responsable de los datos agregados de compuerta y ejecutar el código JavaScript solicitado

Tabla 22. Resultados del artículo SA12

Artículo	A Sound Framework for Dynamic Prevention of Local File Inclusion
Propuesta o técnica de solución.	Framework para la prevención de forma dinámica maliciosa. El método propuesto garantiza que ningún malintencionado código PHP se puede incluir en aplicación web sin permiso de actualización para cambiar el código fuente principal de la aplicación.
Problemas de seguridad	LFI (Inclusión de archivos locales)
Resultado de propuesta	El marco propuesto indica que solo es local, PHP scripts que se encuentran en la carpeta raíz del sitio web (o subcarpetas), deben ser incluidos. Estos archivos están etiquetados como archivos de confianza. Cualquier otro archivo se excluye y se marca como archivo que no es de confianza. De esta manera, la función incluye, funciona sin ningún problema, pero con una restricción: solo los archivos desarrollados por desarrolladores deben ser incluidos.
Conclusiones relevantes	El marco propuesto se beneficia de un enfoque dinámico para la prevención de vulnerabilidades. El marco se basa en el concepto de seguros y no seguros PHP scripts en los que se permiten guiones fiables para ser incluidos los que no se confía y no pueden incluir.

Tabla 23. Resultados del artículo SA13.

Artículo	Secure Integrated Framework for Business Processes
Propuesta o técnica de solución.	Framework para procesos de negocio que protege a los usuarios finales de los ataques maliciosos. Tiene varios componentes, tales como, un consumidor de servicios, la fase II de autenticación, sistema basado en múltiples hilos, matrices clave, la capa de agente, proveedor de servicios, módulos ENC y DEC, procesos de negocio, servicios web y bases de datos. Cada componente tiene sus propias responsabilidades y comportamientos.
Problemas de seguridad	Ataques maliciosos.
Resultado de propuesta	Los usuarios finales pueden utilizar el marco integrado para cargar y descargar sus datos importantes en y desde el servidor de manera segura con la ayuda del sistema multihilo. Cada archivo cargado o

	descargado desde y hacia el servidor está cifrada o descifrada con pares de claves relacionadas con matrices clave.
Conclusiones relevantes	El marco de seguridad integrado para los procesos de negocio ha sido desarrollado y probado con éxito.

2. Síntesis de datos.

En la Tabla 24, se presenta el resumen del proceso de selección de estudios de cada etapa. Las búsquedas realizadas generaron un total de 61183, obteniendo 520 artículos al aplicar los criterios de inclusión y exclusión, de los cuales se registraron 294 coincidencias, es decir el número de artículos revisados fueron 226, de los mismos se seleccionaron 13 artículos de acuerdo al criterio de selección. En la Tabla 24, se muestran los resultados de la selección de artículos: fuente de búsqueda, total general de las búsquedas, estudios incluidos, coincidencias o duplicados, artículos revisados y estudios primarios seleccionados, luego de aplicar los criterios de selección, los cuales nos servirán para la problemática propuesta.

Tabla 24. Resultados de selección de estudios primarios.

	Artículos				
Base de Datos	Total	Encontrados	Coincidencias	Revisados	Seleccionados
Scopus	1122	131	41	90	9
ScienceDirect	59669	349	235	114	3
IEEE	392	40	18	22	1
Total	61183	520	294	226	13

En el Grafico 7, se muestra que la mayor cantidad de artículos seleccionados son los publicados en el año 2015, pero a pesar que estamos a mediados del año 2016 la cantidad de documentos encontrados es considerable, en base a esto se podría decir que a partir del año 2014 ha crecido notablemente las investigaciones referentes a el tema de estudio.

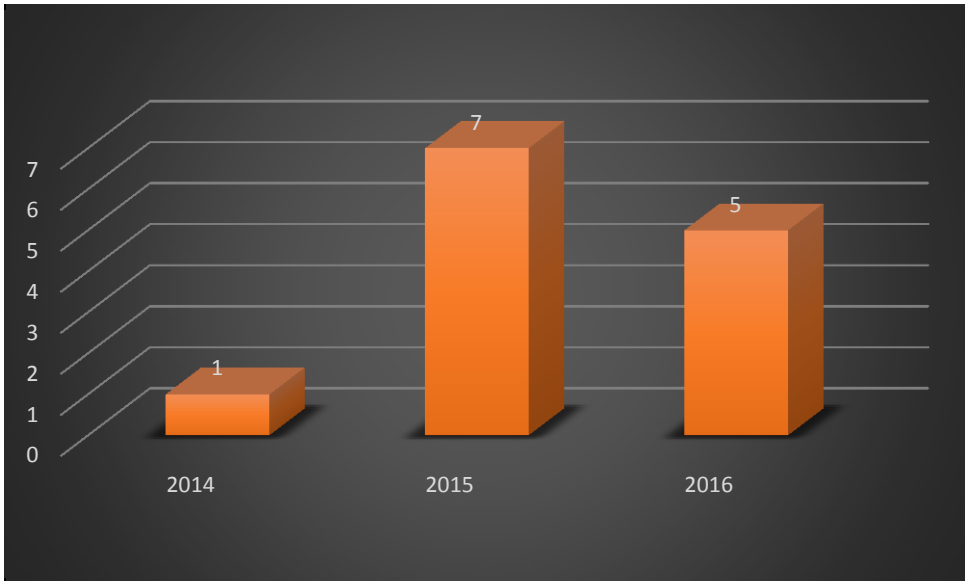

Gráfico 7. Artículos seleccionados por año.

Luego de extraer la información de cada estudio seleccionado se presenta en el Gráfico 8, el porcentaje de los problemas abordados en los mismos, dando como resultado que la mayoría de documentos abordan el problema de Inyección SQL con el 32%, ataques XSS con el 22% y atacas DoS el 13%.

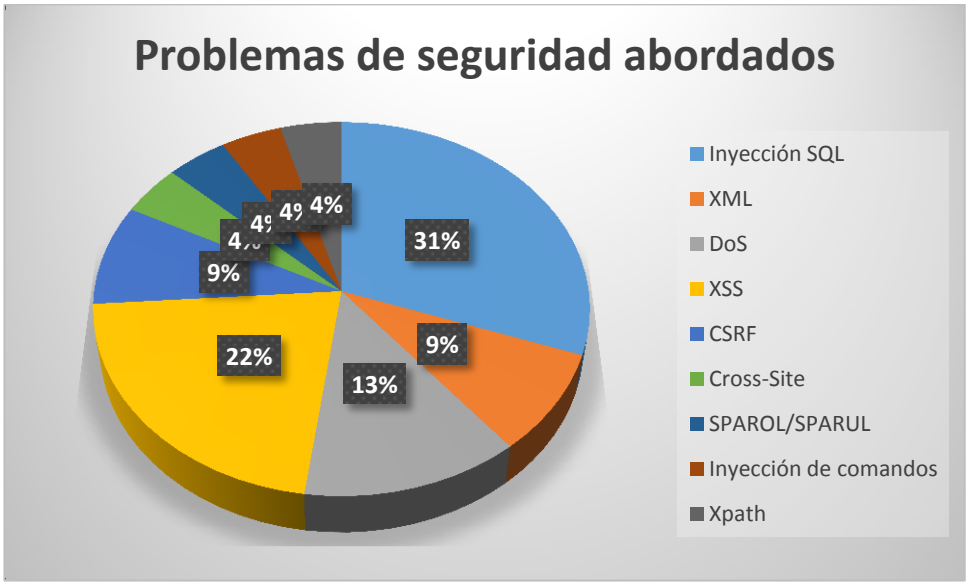

Gráfico 8. Problemas de seguridad abordados por los artículos estudiados.

h. Discusión

1. Discusión de los resultados obtenidos

- No hay una solución única hasta el momento que pueda eliminar las vulnerabilidades y prevenir ataques maliciosos. Sin embargo, la solución más ideal es eliminar vulnerabilidades de la raíz, es decir, el código fuente. Pero, en las aplicaciones web en el mundo real la obtención del código fuente o parches puede ser difícil. Por lo tanto, una serie de técnicas de mitigación deben ser empleadas para frenar la propagación de los ataques y eliminar las vulnerabilidades.

- Es así que el estudio SA01 plantea la Web Semántica basada en RDF (Resource Description Framework), para hacer pruebas de seguridad y explorar vulnerabilidades, llegando a la conclusión de que, los desarrolladores no están familiarizados con las diferentes vulnerabilidades y ataques inyección (SPAROL/SPARUL entre otras) a las que es propensa la Web Semántica.

- SA02 y SA06 utilizan técnicas de detección de SQLIA como: Framework DSD (Dynamic Deteccion SQLIAs) y SQLiGoT (Detección de inyección SQL mediante graficas de tokens), fáciles de incorporar para detectar las consultas maliciosas en tiempo de ejecución, es decir sin necesidad de acceder al código fuente, lo que es una ventaja sobre los métodos existentes. Al igual SA05 expone un framework de defensa XSS robusta, de fácil integración, que detecta y mitiga las vulnerabilidades XSS en aplicaciones web HTML5 desplegados en torno a la nube.

- SA03 presenta una nueva metodología, para el análisis a gran escala de seguridad de interfaces web dentro de dispositivos embebidos, diseñando el primer framework que hace un análisis dinámico escalable y automatizado de firmware, para descubrir vulnerabilidades. Utiliza herramientas Off-the-shelf de análisis estático y dinámico. Concluyen que los fabricantes deben considerar la seguridad en el ciclo de vida del software.

- SA04 expone Framework OAuth que hace frente a diferentes ataques maliciosos, a través de mensajes cifrados y firmados, dando como resultado un alto nivel de seguridad, pero su complejidad hace que sea difícil de integrar.

- SA07 muestra una Revisión de Literatura de seguridad y ataques en Servicios web. De la cual concluye, que la penetración y la automatización de pruebas se deben realizar como parte de cada desarrollo, porque, los ataques del servicio web no se pueden eliminar por completo.

- SA08 da a conocer un enfoque experimental a desarrolladores y proveedores para evaluar la seguridad de los framework de servicio web, contra ataques DoS, cuyo resultado indican que incluso plataformas muy populares requieren mejoras urgentes.

- SA09 enseña el Framework MUSA como una solución global y de apoyo a la seguridad (relacionada con la protección de datos), de todo el ciclo de vida de aplicaciones milti-nube. SA10 también protege la confidencialidad e integridad de la información mediante técnicas IFC (Información de control de flujo).

- SA11 exhibe un Framework de seguridad Mashup que examina, analiza y evalúa los datos de transición entre el servidor y el lado del cliente, cuyo resultado ha sido fallido ya que no satisface las necesidades de los usuarios.

- SA12 presenta un Framework sólido para la prevención de inclusión de archivos locales (LFI) y SA13 un Framework frente a procesos de negocios, que protegen a usuarios finales de los ataques maliciosos, permitiendo cargar y descargar datos en y desde el servidor de manera segura con la ayuda de sistemas multihilo. Este método fue desarrollado y probado con éxito.

- El trabajo se centra en 13 artículos relacionados con la investigación de mecanismos de seguridad. Se identifica las soluciones, métodos, técnicas propuestas en los estudios. Las soluciones propuestas son muchas y diversas, en su mayoría se enfocan en la prevención de ataques maliciosos y detección de vulnerabilidades. Solo el estudio SA02, discute la eliminación de vulnerabilidades a partir del código fuente, lo que es importante para prevenir ataques y ahorrar

recursos en post-implementación. Los artículos SA03, SA04, SA05, SA06, SA09, SA12, SA13 hacen un análisis sintáctico a través de off-the-shelf herramientas de análisis estático y dinámico, Framework OAuth, framework basado en HTML 5, SQLiGot (Detección de inyección SQL mediante grafica de tokens), MUSA, se basa en el concepto de seguros y no seguros PHP scripts, framework para procesos de negocio. Los documentos SA01, SA02, SA08, SA11, realizan análisis semánticos de diferentes mecanismos de seguridad como RDF (Resource Description Framework), DSD (Detección SQLIAs Dinámico), WSFAggresor herramienta de pruebas de seguridad, HMM (Modelo oculto de Markov) y Mashup framework de seguridad. Y el artículo SA07 y SA10 proporciona una revisión sistemática de los estudios de seguridad de servicios web, y mecanismos de control de flujo en aplicaciones web. Todos los estudios hacen frente a los problemas de seguridad para eliminarlos. Pero los ataques son cada vez más frecuentes, así que la seguridad debe ser tratada en todas las fases de desarrollo considerando la seguridad desde el principio y en todo el ciclo de vida de la aplicación y usar framework para soportarla puede dar excelentes resultados. El análisis de estudios primarios indica la facilidad con la que puede ser vulnerada una aplicación cuando no se les asigna una prioridad adecuada a los controles de seguridad en las distintas etapas de desarrollo.

2. Desarrollo de la propuesta

El tema planteado para el presente trabajo de titulación es “Revisión Sistemática de Literatura: Seguridad en Ambientes Web Utilizando Framework”, se lo propuso considerando algunos aspectos, en primer lugar esta área de investigación en la actualidad es muy importante ya que las vulnerabilidades de las aplicaciones web, se han convertido en una gran amenaza para la seguridad de sistemas informáticos, que son utilizados para llevar acabo un sinnúmero de actividades, y es necesario conocer los avances de los mecanismos de seguridad para la protección de las mismas. En segundo lugar, se cuenta con una gran variedad de métodos aplicables al área de investigación para su exitoso desarrollo, en este caso se decidió utilizar una revisión sistemática siguiendo el protocolo de Barbara Kitchenham considerando que es una de las pocas que se direcciona a la ingeniería de software, y además cuenta con un protocolo claramente establecido y estandarizado que asegura la claridad y

transparencia en los procesos de revisión. Para poder realizar con normalidad este Trabajo de Titulación se cumplió con los siguientes objetivos:

2.1 Identificar los problemas de seguridad más comunes en las aplicaciones web.

Para el cumplimiento de este objetivo, se desarrolló la revisión sistemática en donde se identificaron los problemas de seguridad más comunes en la extracción de datos, los mismos que se encuentran planteados en la sección de Resultados en las tablas de la 11 a 23, dando como resultado que los problemas de seguridad más abordados o comunes en las aplicaciones web son: Inyección SQL con el 31 %, XSS con el 22% y DoS con el 13%, que se encuentran plasmados en el gráfico 8.

2.2 Realizar un proceso de revisión sistemática para estudios primarios.

El proceso de la revisión comenzó con la elección del protocolo a seguir que es el de Barbara Kitchenham, siendo el más adecuado para realizar la investigación y uno de los pocos aplicados a ingenierías. Este protocolo consta de tres etapas, cada una con sus respectivas actividades. Comenzando con el paso más importante que es la especificación de la pregunta de investigación, la misma que guio en como recolectar los estudios, como controlar si los estudios son elegibles y como llevar a cabo el análisis. Luego se procedió con la elección de las fuentes de información considerando la accesibilidad a la web así como la inclusión de motores de búsqueda que permitan realizar consultas avanzadas, a continuación se seleccionó las palabras claves y una vez definidas se realizó concatenaciones utilizando los operadores lógicos AND y OR, formando las cadenas de búsqueda que sirvieron para recopilar los estudios, a los cuales se aplicaron los criterios de inclusión y exclusión quedando 520 documentos, encontrando 294 coincidencias y siendo revisados 226 estudios primarios, de los mismos que se redujeron en 13 artículos de acuerdo al criterio de selección, de estos artículos seleccionados se analizó y se sintetizo los aspectos principales tomando en consideración las propuestas o técnicas de solución, problemas de seguridad, resultados de propuesta y conclusiones relevantes. Y así conocer cuáles han sido los avances en cuanto a seguridad de las aplicaciones web y que vulnerabilidades o ataques han sido abordados.

2.3 Sintetizar la información recopilada en la revisión sistemática

Luego de cumplir rigurosamente el proceso del objetivo 2 se procedió a sintetizar la información a través de una discusión entre los aspectos relevantes de los 13 estudios utilizados para esta Revisión Sistemática, tomando en cuenta las propuestas o técnicas de solución, problemas de seguridad, resultados de propuesta y conclusiones relevantes.

Adicionalmente se presentó un artículo científico en la séptima edición de las Jornadas de Ingeniería en Sistemas Informáticos y de Computación JISIC-2016, el primer encuentro sobre Seguridad Informática, que tiene como finalidad reunir a investigadores y profesionales de la academia y la industria, para enfocarse en los avances de la seguridad Informática y fomentar la difusión de la investigación, aplicación e innovación que se realiza dentro de este tema. En el Anexo 2 y 3 se justifica la participación en este evento y la aceptación del artículo para su publicación.

Se pudo evidenciar en el transcurso de la Revisión que no existe gran cantidad de información publicada con respecto a mecanismos de seguridad para aplicaciones web, pese a esto se logró concluir con éxito la investigación rescatando valiosos resultados que ayudaron a emitir las respectivas conclusiones y a finalizar el presente Trabajo de Titulación.

3. Valoración Social, Técnica Económica y Científica.

Se expresa la valoración del Trabajo de Titulación describiendo los beneficios presentados en cuatro aspectos:

3.1 Valoración Social

- Conocer el procedimiento adecuado para la realización de una Revisión Sistemática.
- Conocer los mecanismos de seguridad propuestos para las aplicaciones web.
- Conocer el tipo de problemas de seguridad más abordados.
- Conocer los resultados obtenidos de cada mecanismo de seguridad propuesto.

3.2 Valoración Técnica

- A través del gestor bibliográfico Mendeley se ahorró tiempo, ya que este permite organizar las referencias de manera sencilla desde las fuentes y de varias maneras.

- Con la utilización Google Drive y Google Docs. se facilitó la revisión del Trabajo de Titulación, puesto que está diseñado para permitir la fácil y rápida colaboración de varios usuarios a la vez en un mismo proyecto en tiempo real.

- El uso del correo electrónico y las redes sociales permitió la constate comunicación entre el investigador y el director del Trabajo de Titulación.

3.3 Valoración Económica

- Unos de los principales beneficios son el aporte de la Universidad Nacional de Loja con el control y seguimiento del Trabajo, que cubre los gastos del Tutor o Director de Tesis.

- El uso de herramientas tecnológicas colaboró al ahorro de tiempo y dinero, evitando realizar impresiones innecesarias, así como asistencias personales a la UNL.

3.4 Valoración Científica

El beneficio en el aspecto científico radica en el aporte de trabajos futuros que resultaron del proceso de la investigación.

i. Conclusiones

- Los estudios analizados en la Revisión Sistemática revelan que no existe un mecanismo de seguridad que elimine por completo las vulnerabilidades de Inyección SQL, XSS, DoS de las aplicaciones web, ya que cada una tiene diferente arquitectura y funcionalidad, atacando a la base de datos, al URL, congelan el funcionamiento de un sitio web, acceden a cookies o tokens entre otros.

- En el análisis de los artículos seleccionados el 75% de los ataques y vulnerabilidades abordados son: ataques de inyección SQL, ataques Cross Site Scripting (XSS), ataques de denegación de servicio (DoS), seguido de ataques de inyección XML, debido al impacto que causan en la seguridad de las aplicaciones poniendo en riesgo la información de cada uno de sus usuarios.

- La Revisión sistemática empleada en este trabajo de titulación nos permitió seguir un proceso metodológico, con el objetivo de ir estructurando, planificando y extrayendo la información de estudios primarios, basándonos en cada uno de los parámetros del protocolo de Kitchenham los cuales son la planificación de la Revisión, Ejecución de la Planificación e informe de resultados, para extraer la información más relevante de cada estudio.

j. Recomendaciones

- Usar diferentes técnicas y combinación de mecanismos de seguridad para mitigar ataques maliciosos y vulnerabilidades de las aplicaciones web como: análisis estático que son útiles antes del despliegue y más aún durante el desarrollo de la aplicación; análisis dinámico como técnica de pruebas de penetración pueden ser utilizadas para explotar las aplicaciones web durante el tiempo de ejecución con el fin de determinar si todavía son vulnerables a ataques maliciosos. Ya que todas son diferentes, es decir están estructuradas según su finalidad.

- Estructurar bien las cadenas de búsqueda, concatenando las palabras claves con los operadores lógicos AND y OR en base a la pregunta de investigación ya que de ello depende la obtención de estudios primarios relevantes y de calidad.

Trabajos futuros

- En el futuro, realizar una nueva Revisión Sistemática en cuanto a los mecanismos de seguridad para mitigar la inyección SQL en aplicaciones web, compararlo con nuestro estudio y saber cuáles han sido los avances en la seguridad a partir de la publicación de mismo.

- Realizar estudios secundarios para obtener información, acerca de que estudios existen de los diferentes tipos de escáneres de vulnerabilidades existentes, para mitigar la Inyección SQL en las aplicaciones web.

k. Bibliografía

[1] R. A. Oliveira, N. Laranjeiro, and M. Vieira, “Assessing the security of web service frameworks against Denial of Service attacks,” *J. Syst. Softw.*, vol. 109, pp. 18–31, 2015.

[2] M. Castro-león, F. Boixader, M. Taboada, D. Rexachs, E. Universitaria, and T. Cerdá, “Servicios y Seguridad, un enfoque basado en estrategias de ataque y defensa,” pp. 39–48, 2015.

[3] D. CAMACHO, G. MARTINEZ, and D. BIANCHA, “Diseño De Framework Web Para El Desarrollo Dinámico De Aplicaciones,” no. 44, pp. 178–183, 2010.

[4] M. D. P. Salas-Zárate, G. Alor-Hernández, R. Valencia-García, L. Rodríguez-Mazahua, A. Rodríguez-González, and J. L. López Cuadrado, “Analyzing best practices on Web development frameworks: The lift approach,” *Sci. Comput. Program.*, vol. 102, pp. 1–19, 2015.

[5] H. Cervantes, R. Kazman, and J. Ryoo, “Seguridad y uso de Frameworks _ SG.” p. SG # 47, 2015.

[6] A. R. Sartorio, G. L. Rodríguez, and M. Vaquero, “Investigación en el diseño y desarrollo para el enriquecimiento de un framework colaborativo web sensible al contexto,” *XIII Work. Investig. en Ciencias la Comput.*, pp. 1–5, 2011.

[7] C. García, R. Hervás, and P. D. A.-/9 L. B.-G. Gervás, “Una Arquitectura Software para el Desarrollo de Aplicaciones de Generación de Lenguaje Natural,” *Soc. Española para el Proces. del Leng. Nat. Proces. Leng. Nat.*, vol. 33, pp. 111–118 ST – Una Arquitectura Software para el De, 2004.

[8] G. Martínez Villalobos, G. D. Camacho Sánchez, and D. A. Biancha Gutiérrez, “Diseño de Framework web para el desarrollo dinámico de aplicaciones,” *Sci. Tech.*, vol. XVI, no. 44, pp. 178–183, 2010.

[9] H. T. Quinche, René Guamán, “Seguridad en Entornos Web para Sistemas de Gestión Académica,” pp. 1–47.

[10] R. Akrout, E. Alata, M. Kaaniche, and V. Nicomette, “An automated black box approach for web vulnerability identification and attack scenario generation,” *J. Brazilian Comput. Soc.*, vol. 20, no. 1, p. 4, 2014.

[11] A. María Reina Quintero, “Separación avanzada de conceptos en entornos WEB.,” pp. 3–16.

[12] G. Deepa and P. S. Thilagam, “Securing web applications from injection and logic vulnerabilities: Approaches and challenges,” *Inf. Softw. Technol.*, vol. 74,

pp. 160–180, 2016.

[13] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3,” *Engineering*, vol. 45, no. 4ve, p. 1051, 2007.

[14] S. E. Group and R. Unido, “Directrices para la realización sistemática de la literatura críticas en Ingeniería de Software Sección de Control de Documentos,” 2007.

[15] Oscar Capuñay Uceda, “Desarrollo Web con PHP: Aprende PHP paso a paso - Oscar Capuñay Uceda - Google Libros,” 2013. [Online]. Available: https://books.google.com.ec/books?id=1GQUAgAAQBAJ&printsec=frontcover&hl=es#v=onepage&q&f=false.

[16] A. Gómez Vieites, *Seguridad informática: básico*. Starbook, 2010.

[17] J. Manuel Saura Martín DIRECTOR and P. Sánchez Palma, “Implantación de seguridad en entornos Web,” 2006.

[18] G. GALLARDO AVILS, *SEGURIDAD EN BASES DE DATOS Y APLICACIONES WEB*. EISENBRAUNS, 2015.

[19] D. de Autor Licencia, “O Acerca de OWASP.”

[20] G. G. Velásquez Gaby, Chicaiza Giovanny, “INYECCIÓN DE SQL, CASO DE ESTUDIO OWASP | Gaby Velásquez - Academia.edu,” 2016. [Online]. Available: http://www.academia.edu/11491488/INYECCI%C3%93N_DE_SQL_CASO_DE_ESTUDIO_OWASP.

[21] A. Rodríguez, R. Consultor, and S. Grupo, “Ataques XSS en Aplicaciones Web.”

[22] J. B. Bermejo, “Ataques DoS en aplicaciones Web.”

[23] M. Gerardo Orellana Cordero Director and I. Pablo Esquivel Cuenca, “" Evaluación de frameworks realizados en java para aplicaciones on-line " Tesis previa a la obtención del Título de: Ingeniero de sistemas,” 2013.

[24] A. Alonso, “GESTORES BIBLIOGRÁFICOS,” *vol. 20*, 2010. [Online]. Available: https://sites.google.com/a/pucp.pe/gestores-bibliograficos/home.

[25] J. Alonso, A. José, A. Cordón, G. Helena, and M. Rodero, “LA GESTIÓN DE REFERENCIAS EN EL DESARROLLO DE SERVICIOS BIBLIOTECARIOS.”

[26] I. Armenteros Vera and I. Alfonso Sánchez, “Los gestores personales de bases de datos bibliográficas: conoce usted qué es y cómo se maneja el Procite,” *ACIMED*, vol. 12, no. 2, pp. 1–1, 2004.

[27] CRAI Universitat de Barcelona, “Mendeley: gestor de referencias y citas

bibliográficas | CRAI UB,” 2014. [Online]. Available: http://crai.ub.edu/es/que-ofrece-el-crai/citaciones-bibliograficas/mendeley.

[28] B. U. A. al A. y la Investigación, “Biblioguías. Cómo buscar en las bases de datos de forma eficaz. Qué son las bases de datos.”

[29] G. D. Rubio Cynthia, Barajas Luis, “BASES DE DATOS ACADEMICAS Y CIENTIFICAS de cynthia paola carlos en Prezi,” 2012. [Online]. Available: https://prezi.com/oscwgpf3q6gc/bases-de-datos-academicas-y-cientificas/.

[30] Universitaria Pascual Bravo, “Bases de datos de libre acceso,” 2010. [Online]. Available: http://www.pascualbravo.edu.co/index.php/biblioteca-cti/basesdedatos/bases-de-datos-gratuitas.

[31] S. J. Guirao Goris, “Utilidad y tipos de revisión de literatura,” *ENE, Rev. Enferm.*, vol. 9, no. 2, pp. 1–7, 2015.

[32] Hart and Chris, “Doing a Literature Review,” 2000.

[33] H. Aveyard, *Doing a literature review in health and social care: a practical guide*. 2014.

[34] M. Coughlan, F. Ryan, and P. Cronin, *Doing a literature review in nursing, health and social care*. SAGE, 2013.

[35] M. J. Grant and A. Booth, “A typology of reviews: an analysis of 14 review types and associated methodologies.,” *Health Info. Libr. J.*, vol. 26, no. 2, pp. 91–108, Jun. 2009.

[36] P. Cronin, F. Ryan, and M. Coughlan, “Undertaking a literature review: a step-by-step approach.,” *Br. J. Nurs.*, vol. 17, no. 1, pp. 38–43.

[37] R. Whittemore, A. Chao, M. Jang, K. E. Minges, and C. Park, “Methods for knowledge synthesis: an overview.,” *Heart Lung*, vol. 43, no. 5, pp. 453–61.

[38] R. Whittemore, A. Chao, M. Jang, K. E. Minges, and C. Park, “Methods for knowledge synthesis: An overview,” *Hear. Lung J. Acute Crit. Care*, vol. 43, no. 5, pp. 453–461, Sep. 2014.

[39] B. Kitchenham, “Procedures for performing systematic reviews,” *Keele, UK, Keele Univ.*, vol. 33, no. TR/SE-0401, p. 28, 2004.

[40] T. N. Aung and S. S. Khaing, “Genetic and Evolutionary Computing,” *Towar. SQL Inject. Attacks Detect. Mech. Using Parse Tree*, vol. 388, pp. 405–411, 2016.

[41] H. Shahriar, K. Weldemariam, M. Zulkernine, and T. Lutellier, “Effective detection of vulnerable and malicious browser extensions,” *Comput. Secur.*, vol. 47, pp. 66–84, 2014.

[42] H. Omotunde and R. Ibrahim, “Mitigating SQL injection attacks via hybrid threat modelling,” *2015 IEEE 2nd Int. Conf. InformationScience Secur. ICISS 2015*, pp. 15–18, 2016.

[43] T. Y. Wu, J. S. Pan, C. M. Chen, and C. W. Lin, “Towards SQL injection attacks detection mechanism using parse tree,” 2015, vol. 329, pp. 371–380.

I. Anexos

1. Anexo 1. Artículos revisados

Tabla 25. Artículos Revisados

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
1	Performance analysis for extended TLS with mutual attestation for platform integrity assurance	Abd Aziz, Norazah Udzir, Nur Izura, Mahmod Ramlan	Institute of Electrical and Electronics Engineers Inc.	2014
2	Integration of heterogeneous policies for trust management	Abdi Samane	Institute of Electrical and Electronics Engineers Inc.	2014
3	Toward an access control model for IOTCollab	Adda Mehdi, Abdelaziz Jabril, McHeick Hamid, Saad Rabeb	Elsevier	2015
4	Industrial and business systems for Smart Cities	Amaba Ben	Association for Computing Machinery, Inc	2014
5	Semantic Web application framework for teaching SPARQL/SPARUL injection attacks and defense mechanisms	Asghar Hira, Anwar Zahid, Latif Khalid	Elsevier Ltd	2016
6	A framework for resilient remote monitoring	Atighetchi Michael, Adler Aaron	Institute of Electrical and Electronics Engineers Inc.	2014
7	Context-aware usage control for web of things	Bai Guangdong, Yan Lin, Gu Liang, Guo Yao, Chen Xiangqun	John Wiley and Sons Inc.	2014
8	Developing a usability framework to support online rapid urban information discovery and interrogation	Barton John E. Pettit Christopher	CEUR-WS	2014

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
9	Towards a legislation driven framework for access control and privacy protection in public cloud	Belaazi Maherzia, Rahmouni Hanen, Boussi Bouhoula Adel.	SciTePress	2014
10	A secure and flexible data infrastructure for the VPH-share community	Benkner Siegfried, Kaniovskyi Yuriy, Borckholder Chris, Bubak Marian, Nowakowski Piotr, Lopez Dario, Ruiz Wood Steven	IEEE Computer Society	2014
11	Formal Specification of the Framework for NSSA	Bhandari Pardeep, Singh Manpreet	Elsevier	2016
12	Securing service in remote healthcare	Bhattasali, Tapalina; Chaki, Rituparna; Chaki, Nabendu; Saeed, Khalid	Springer Verlag	2016
13	Negotiation based framework for Attribute-Based Access Control policy evaluation	Caprin, Edward; Zhang, Yan	Association for Computing Machinery	2014
14	Spatial-temporal based integrated management for smart city: Framework, key techniques and implementation	Chen, Nengcheng; Du, Wenying	IEEE Computer Society	2016
15	A framework and language support for dynamic security policy in service-oriented architecture	Chi, W. U Lee; Hwang, Gwan Hwan	Institute of Information Science	2014
16	Modeling and formal verification of smart environments	Corno, Fulvio; Sanaullah, Muhammad	John Wiley and Sons Inc.	2014
17	Automated dynamic firmware analysis at scale: A case study on embedded web interfaces	Costin, Andrei; Zarras, Apostolis; Francillon, Aurélien	Association for Computing Machinery, Inc	2016

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
18	CONCEPT: A service framework for secure Ad-Hoc environments	De Azevedo, Nuno Solinho; Costa, Antonio Duarte; Macedo, Joaquim; Nicolau, Maria João	Springer Verlag	2014
19	NodeSentry: Least-privilege library integration for server-side JavaScript	De Groef, Willem; Massacci, Fabio; Piessens, Frank	Association for Computing Machinery	2014
20	A framework of cloud-based virtual phones for secure intelligent information management	Ding, Jiun Hung; Chien, Roger; Hung, Shih Hao; Lin, Yi Lan; Kuo, Che Yang; Hsu, Ching Hsien; Chung, Yeh Ching	Elsevier Ltd	2014
21	A context-aware recommendation framework in e-learning environment	Do, Phung; Nguyen, Hung; Nguyen, Vu Thanh; Dung, Tran Nam	Springer Verlag	2015
22	EMF-REST: Generation of RESTful APIs from models	Ed-Douibi, Hamza; Izquierdo, Javier Luis Cánovas; Gómez, Abel; Tisi, Massimo; Cabot, Jordi	Association for Computing Machinery	2016
23	Security evaluation of the OAuth 2.0 framework	Ferry, Eugene; Raw, John O.; Curran, Kevin	Emerald Group Publishing Ltd.	2015
24	RT-SPDM: Real-time security, privacy and dependability management of heterogeneous systems	Fysarakis, Konstantinos; Hatzivasilis, George; Askoxylakis, Ioannis; Manifavas, Charalampos	Springer Verlag	2015

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
25	WSACd - A usable access control framework for smart home devices	Fysarakis, Konstantinos; Konstantourakis, Charalampos; Rantos, Konstantinos Manifavas, Charalampos Papaefstathiou, Ioanni	Springer Verlag	2015
26	CSSXC: Context-sensitive Sanitization Framework for Web Applications against XSS Vulnerabilities in Cloud Environments	Gupta, Shashank; Gupta, B. B.	Elsevier	2016
27	Alleviating the proliferation of JavaScript worms from online social network in cloud platforms	Gupta, Shashank; Gupta, B. B.	Institute of Electrical and Electronics Engineers Inc.	2016
28	Revitalizing the Afghan educational assessment paradigm through ICT	Hamidullah, Sokout; Rashid, Ahmadi; Abdul Hanif, Gharanai Mohammad; Samiullah, Paracha	Institute of Electrical and Electronics Engineers Inc.	2015
29	Flexible access control for JavaScript	Hammer, Christian	Gesellschaft fur Informatik (GI)	2014
30	Enhancing secure web mashups development in enterprise environments	He, Wu;	Institute of Electrical and Electronics Engineers Inc.	2014
31	APSIM - Evolution towards a new generation of agricultural systems simulation	Varios autores	Elsevier Ltd	2014
32	Enhanced hospital information system by cloud computing: SHEFA'A	Varios autores	Springer Verlag	2014

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
33	Design and implementation of multilevel security subsystem based on XACML and WEB services	Jarmakiewicz, Jacek; Podlasek, Tomasz	Institute of Electrical and Electronics Engineers Inc.	2015
34	A design of web log integration framework using NoSQL	Jeong, Huijin; Choi, Junho; Choi, Chang; You, Ilsun; Kim, Pankoo	Springer Verlag	2014
35	Web simulation training environment for aircraft resource planning in wildfire events	Jove, Jaume; Figueras; Petit, Antoni Guasch; Casanovas-Garcia, Josep	Institute of Electrical and Electronics Engineers Inc.	2016
36	A comprehensive survey on variants and its extensions of BIG DATA in cloud environment	Varios autores	Association for Computing Machinery	2015
37	An inductive and semantic model of constraints for master data management under cloud computing	Kikuchi, Shinji		2014
38	Seamless qoe support for mobile cloud services using ieee802.21 mih and the geni future internet framework	Kim, Gijeong; Lee, Sungwon; Lee, Seung Gwan	World Scientific Publishing Co. Pte Ltd	2014
39	Design and implementation of an easy-setup framework for personalized cloud device	Koo, Bonhyun; Ahn, Taewon; Kong, Simon; Cho, Hyejung	Springer Verlag	2015
40	Query monitoring and analysis for database privacy - a security automata model approach	Kumar, Anand; Ligatti, Jay; Tu, Yi Cheng	Springer Verlag	2015

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
41	A framework to integrate public information into runtime safety analysis for critical systems	Li, Guoqi	Maruzen Co., Ltd.	2014
42	A Decentralized Locality-Preserving Context-Aware Service Discovery Framework for Internet of Things	Li, Juan; Zaman, Nazia; Li, Honghui	Institute of Electrical and Electronics Engineers Inc.	2015
43	Research on semantic++ computing based on big data environment	Liang, Ye; Zhang, Guigang; Xing, Chunxiao; Zhang, Yong; Li, Chao	Institute of Electrical and Electronics Engineers Inc.	2016
44	A generic trust framework for large-scale open systems using machine learning	Liu, Xin; Tredan, Gilles; Datta, Anwitaman	Blackwell Publishing Inc.	2014
45	Mobile-based medical data accessibility in mHealth	Lomotey, Richard K.; Deters, Ralph	IEEE Computer Society	2014
46	Facilitating the design/evaluation process of web-based geographic applications: A case study with WINDmash	Luong, The Nhan; Marquesuzaà, Christophe; Etcheverry, Patrick; Nodenot, Thierry; Laborie, Sébastien	Springer Verlag	2015
47	Policy-based access control for body sensor networks	Varios autores	Springer Verlag	2014
48	Multi-Modal Medical Data Analysis Platform (3MDAP) for analysis and predictive modelling of cancer trial data	Manikis, Georgios C.; Maniadi, Evangelia; Tsiknakis, Manolis Marias, Kostas	Institute of Electrical and Electronics Engineers Inc.	2014
49	Novel security conscious evaluation criteria for web service composition	Varios autores		2014

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
50	Addressing identity and location privacy of things for Indoor—case study on internet of everything's (IoE)	Nadargi, Ajay; Thirugnanam, Mythili	Springer Science and Business Media Deutschland GmbH	2016
51	Trust enhanced distributed authorisation for web services	Nagarajan, Aarthi; Varadharajan, Vijay; Tarr, Nathan	Academic Press Inc.	2014
52	CloudArmor: Supporting Reputation-Based Trust Management for Cloud Services	Noor, Talal H.; Sheng, Quan Z; Yao, Lina; Dustdar, Schahram Ngu, Anne H H.	IEEE Computer Society	2016
53	Assessing the security of web service frameworks against Denial of Service attacks	Oliveira, Rui André; Laranjeiro, Nuno; Vieira, Marco	Elsevier Inc.	2015
54	Self-protecting multi-cloud applications	Ortiz, Antonio M.; Rios, Erkuden; Mallouli, Wissam; Iturbe, Eider; De Oca, Edgardo Montes	Institute of Electrical and Electronics Engineers Inc.	2015
55	Security analysis and proposal of new access control model in the Internet of Thing	Ouaddah, Aafaf; Bouij-Pasquier, Imane; Abou Elkalam, Anas; Ait Ouahman, Abdellah	Institute of Electrical and Electronics Engineers Inc.	2015
56	A Framework for Anomaly Diagnosis in Smart Homes Based on Ontology	Pardo, Etienne; Espes, David; Le-Parc, Philippe	Elsevier	2016
57	A novel methodology towards a trusted environment in mashup web applications	Patel, Ahmed; Al-Janabi, Samaher; Alshourbaji, Ibrahim; Pedersen, Jens	Elsevier Ltd	2015
58	A security extension for securing the feedback & rating values in TIDE framework	Pranata, Ilung; Skinner, Geoff	Institute of Electrical and Electronics Engineers Inc.	2014

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
59	Mobile cloud computing: A survey, state of art and future directions	Varios autores	Kluwer Academic Publishers	2014
60	Automatic acceptance testing of the web application security with ITU-T X.805 framework	Rathod, Paresh; Julkunen, Viljami; Kaisti, Tero; Nissila, Janne	Institute of Electrical and Electronics Engineers Inc.	2015
61	Semantic security against web application attacks	Varios autores		2014
62	Using commercial web services to build Automated Test Equipment cloud based applications	Reitze, Dale D.	Institute of Electrical and Electronics Engineers Inc.	2014
63	Pervasive geo-security - a lightweight triple-A approach to securing distributed geo-service infrastructures	Resch, Bernd; Schulz, Bernhard; Mittlboeck, Manfred; Heistracher, Thomas	Taylor and Francis Ltd.	2014
64	A robust and light weight authentication framework for hadoop file system in cloud computing environment	Sarvabhatla, Mrudula; Reddy, M. Chandra Mouli; Vorugunti, Chandra Sekhar	Association for Computing Machinery	2015
65	Protection against web 2.0 client-side web attacks using information flow control	Sayed, Bassam; Traore, Issa	IEEE Computer Society	2014
66	3drepo.io: Building the next generation web3D repository with AngularJS and X3DOM	Scully, Timothy; Dobos, Jozef; Sturm, Timo; Jung, Yvonne	Association for Computing Machinery, Inc	2015
67	CBRAIN: A web-based, distributed computing platform for collaborative neuroimaging research	Varios autores	Frontiers Research Foundation	2014

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
68	Trust but verify: Authorization for web services	Skalka, Christian; Wang, X. Sean	Association for Computing Machinery, Inc	2015
69	How to manage cloud risks based on the BMIS model	Song, Youjin; Pang, Yasheng	Korea Information Processing Society	2014
70	An efficient cloud based key aggregate data sharing	Srinivas Kumar, P. Thirumala Rao, B.	Asian Research Publishing Network	2016
71	Remote Surveillance Robot System-A Robust Framework Using Cloud	Sundaram, Aditya; Gupta, Monel; Rathod, Vinod Chandrasekaran, K.	Institute of Electrical and Electronics Engineers Inc.	2016
72	Securing integration of cloud services in cross-domain distributed environments	Suzic, Bojan	Association for Computing Machinery	2016
73	Situation awareness in airport environment based on Semantic Web technologies	Varios autores	IEEE Computer Society	2014
74	Towards privacy-preserving reputation management for hybrid broadcast broadband applications	Tormo, Ginés Dólera; Mármol, Félix Gómez; Pérez, Gregorio Martínez	Elsevier Ltd	2015
75	Overall security solutions for OPC UA based monitoring and control application	Tu, Nguyen Thi Thanh; Thang, Huynh Quyet	Springer Verlag	2014
76	Test as a service: A framework for web security TaaS service in cloud environment	Tung, Yuan Hsin; Lin, Chen Chiu; Shan, Hwai Ling	IEEE Computer Society	2014
77	DPWSec: Devices profile for Web Services Security	Unger, Sebastian; Timmermann, Dirk	Institute of Electrical and Electronics Engineers Inc.	2015

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
78	PaaSword: A holistic data privacy and security by design framework for cloud services	Varios autores	SciTePress	2015
79	A webgis-based system for analyzing and visualizing air quality data for shanghai municipality	Wang, Manyi; Liu, Chaoshun; Gao, Wei	SPIE	2014
80	Semantic-level fusion of heterogenous sensor network and other sources based on Bayesian network	Wu, Kui; Tang, Wenyin; Mao, K. Z; Ng, Gee Wah; Mak, Lee Onn	Institute of Electrical and Electronics Engineers Inc.	2014
81	Towards SQL injection attacks detection mechanism using parse tree	Wu, Tsu Yang; Pan, Jeng Shyang; Chen, Chien Ming; Lin, Chun Wei	Springer Verlag	2015
82	Designing a risk assessment system for China's third-party mobile payments	Xu, Lei; Zhuo, Wuyang	Springer Verlag	2014
83	Enabling collaborative MapReduce on the Cloud with a single-sign-on mechanism	Zhao, Jiaqi; Tao, Jie; Streit, Achim	Springer-Verlag Wien	2016
84	A survey on Information Flow Control mechanisms in web applications	De Paiva, Oscar Zibordi; Ruggiero, Wilson Vicente	Institute of Electrical and Electronics Engineers Inc.	2015
85	On the use of security analytics for attack detection in vehicular ad hoc networks	Gantsou, Dhavy	Institute of Electrical and Electronics Engineers Inc.	2015
86	WEB Applications and services security: On preventing language - based attacks	Idowu, Sunday; Maitanmi, Sola	Academic Conferences Limited	2014

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
87	MTBAC: A mutual trust based access control model in Cloud computing	Lin, Guoyuan; Wang, Danru; Bie, Yuyu; Lei, Min	Editorial Department of China Communicatios	2014
88	SecureCyber: Risk-[43]based optimization through common vulnerability systems scoring over energy smart grid	Majed, Shahir; Ibrahim, Suhaimi; Shaaban, Mohamed	Association for Computing Machinery	2014
89	A sound framework for dynamic prevention of Local File Inclusion	Tajbakhsh, Mir Saman; Bagherzadeh, Jamshid	Institute of Electrical and Electronics Engineers Inc.	2015
90	A web-based screening model for climate risk to water supply systems in the northeastern United States	Whateley, Sarah Walker, Jeffrey D. Brown, Casey	Elsevier Ltd	2015
91	A study on Web security incidents in China by analyzing vulnerability disclosure platforms	Cheng Huang, JiaYong Liu, Yong Fang, Zheng Zuo	Computers & Security	2016
92	Security Busters: Web browser security vs. rogue sites	Nikos Virvilis, Alexios Mylonas, Nikolaos Tsalis, Dimitris Gritzalis	Computers & Security	2015
93	Ontology for attack detection: An intelligent approach to web application security	Abdul Razzaq, Zahid Anwar, H. Farooq Ahmad, Khalid Latif, Faisal Munir	Computers & Security	2014
94	Security knowledge representation artifacts for creating secure IT systems	Jose Fran. Ruiz, Marcos Arjona, Antonio Maña, Carsten Rudolph	Computers & Security	2016
95	The simulated security assessment ecosystem: Does penetration testing need standardisation?	William Knowles, Alistair Baron, Tim McGarr	Computers & Security	2016

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
96	Novel efficient techniques for real-time cloud security assessment	Varios autores	Computers & Security	2016
97	A survey of security solutions for distributed publish/subscribe systems	Anton V. Uzunov	Computers & Security	2016
98	A problem shared is a problem halved: A survey on the dimensions of collective cyber defense through security information sharing	Florian Skopik, Giuseppe Settanni, Roman Fiedler	Computers & Security	2016
99	Shaping intention to resist social engineering through transformational leadership, information security culture and awareness	Waldo Rocha Flores, Mathias Ekstedt	Computers & Security	2016
100	Exploratory security analytics for anomaly detection	Fabio Pierazzi, Sara Casolari, Michele Colajanni, Mirco Marchetti	Computers & Security	2016
101	Information security policy compliance model in organizations	Nader Sohrabi Safa, Rossouw Von Solms, Steven Furnell	Computers & Security	2016
102	A review of cyber security risk assessment methods for SCADA systems	Varios autores	Computers & Security	2016
103	The information systems' security level assessment model based on an ontology and evidential reasoning approach	Kresimir Solic, Hrvoje Ocevcic, Marin Golub	Computers & Security	2015

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
104	Information security conscious care behaviour formation in organizations	Varios autores	Computers & Security	2015
105	Security solution frames and security patterns for authorization in distributed, collaborative systems	Anton V. Uzunov, Eduardo B. Fernandez, Katrina Falkner	Computers & Security	2015
106	SecKit: A Model-based Security Toolkit for the Internet of Things	Neisse, Ricardo; Steri, Gary Fovino, Igor Nai; Baldini, Gianmarco	Computers & Security	2015
107	Security of Software Defined Networks: A survey	Izzat Alsmadi, Dianxiang Xu	Computers & Security	2015
108	Improving the information security culture through monitoring and implementation actions illustrated through a case study	Adéle da Veiga, Nico Martins	Computers & Security	2015
109	Triangular data privacy-preserving model for authenticating all key stakeholders in a cloud environment	Razaque, Abdul; Rizvi, Syed S.	Computers & Security	2016
110	Information security knowledge sharing in organizations: Investigating the effect of behavioral information security governance and national culture	Waldo Rocha Flores, Egil Antonsen, Mathias Ekstedt	Computers & Security	2014
111	A framework for generating realistic traffic for Distributed Denial-of-Service attacks and Flash Events	Bhatia, Sajal; Schmidt, Desmond; Mohay, George; Tickle, Alan	Computers & Security	2014

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
112	Effective detection of vulnerable and malicious browser extensions	Shahriar, Hossain; Weldemariam, Komminist; Zulkernine, Mohammad; Lutellier, Thibaud	Computers & Security	2014
113	Evaluation model for knowledge sharing in information security professional virtual community	Tamjidyamcholo, Alireza; Bin Baba, Mohd Sapiyan; Shuib, Nor Liyana Mohd; Rohani, Vala Ali	Computers & Security	2014
114	Framework and principles for active cyber defense	Denning, Dorothy E.	Computers & Security	2014
115	Information security knowledge sharing in organizations: Investigating the effect of behavioral information security governance and national culture	Rocha Flores, Waldo; Antonsen, Egil; Ekstedt, Mathias	Computers & Security	2014
116	Permission based Android security: Issues and countermeasures	Fang, Zheran; Han, Weili; Li, Yingjiu	Computers & Security	2014
117	Security and compliance challenges in complex IT outsourcing arrangements: A multi-stakeholder perspective	Bachlechner, Daniel Thalmann, Stefan Maier, Ronald	Computers & Security	2014
118	SoNeUCONABC, an expressive usage control model for Web-Based Social Networks	González-Manzano, Lorena; González-Tablas, Ana I.; De Fuentes, José M. Ribagorda, Arturo	Computers & Security	2014
119	A framework for metamorphic malware analysis and real-time detection	Alam, Shahid Horspool, R. Nigel Traore, Issa Sogukpinar, Ibrahim	Computers & Security	2015

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
120	A survey of information security incident handling in the cloud	Ab Rahman, Nurul Hidayah; Choo, Kim-Kwang Raymond	Computers & Security	2015
121	An anomaly analysis framework for database systems	Vavilis, Sokratis; Egner, Alexandru; Petković, Milan; Zannone, Nicola	Computers & Security	2015
122	Automatic generation of HTTP intrusion signatures by selective identification of anomalies	Garcia-Teodoro, P.; Diaz-Verdejo, J.E.; Tapiador, J.E.; Salazar-Hernandez, R.	Computers & Security	2015
123	Relationship-based federated access control model for EPC Discovery Service	Liu, Bing; Chu, Chao-Hsien	Computers & Security	2015
124	SENTINEL: Securing Legacy Firefox Extensions	Varios autores	Computers & Security	2015
125	A network based document management model to prevent data extrusion	Morovati, Kamran; Kadam, Sanjay; Ghorbani, Ali	Computers & Security	2016
126	Causality reasoning about network events for detecting stealthy malware activities	Zhang, Hao; Yao, Danfeng (Daphne); Ramakrishnan, Naren; Zhang, Zhibin	Computers & Security	2016
127	Exploring infrastructure support for app-based services on cloud platforms	Varios autores	Computers & Security	2016
128	From old to new: Assessing cybersecurity risks for an evolving smart grid	Langer, Lucie; Skopik, Florian; Smith, Paul; Kammerstetter, Markus	Computers & Security	2016

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
129	Graph similarity metrics for assessing temporal changes in attack surface of dynamic networks	Bopche, Ghanshyam S.; Mehtre, Babu M.	Computers & Security	2016
130	Information assurance techniques: Perceived cost effectiveness	Varios autores	Computers & Security	2016
131	MVPSys: Toward practical multi-view based false alarm reduction system in network intrusion detection	Li, Wenjuan; Meng, Weizhi; Luo, Xiapu; Kwok, Lam For	Computers & Security	2016
132	Novel session initiation protocol-based distributed denial-of-service attacks and effective defense strategies	Tas, Ismail Melih; Ugurdogan, Bahar; Baktir, Selcuk	Computers & Security	2016
133	Social engineering attack examples, templates and scenarios	Mouton, Francois; Leenen, Louise; Venter, H.S.	Computers & Security	2016
134	Toward the design of adaptive selection strategies for multi-factor authentication	Dasgupta, Dipankar; Roy, Arunava; Nag, Abhijit	Computers & Security	2016
135	PCA-based multivariate statistical network monitoring for anomaly detection	Varios autores	Computers & Security	2016
136	Decentralized detection of network attacks through P2P data clustering of SNMP data	Cerroni, Walter; Moro, Gianluca; Pasolini, Roberto; Ramilli, Marco	Computers & Security	2015
137	Evaluating and comparing the quality of access control in different operating systems	Varios autores	Computers & Security	2014

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
138	Revocation and update of trust in autonomous delay tolerant networks	Djamaludin, C.I.; Foo, E.; Camtepe, S.; Corke, P.	Computers & Security	2016
139	An adversary model to evaluate DRM protection of video contents on iOS devices	D'Orazio, Christian; Choo, Kim-Kwang Raymond	Computers & Security	2016
140	Enhancing the detection of metamorphic malware using call graphs	Elhadi, Ammar Ahmed E.; Maarof, Mohd Aizaini; Barry, Bazara I.A; Hamza, Hentabli	Computers & Security	2014
141	Cyber situational awareness – A systematic review of the literature	Franke, Ulrik; Brynielsson, Joel	Computers & Security	2014
142	CooPeD: Co-owned Personal Data management	Varios autores	Computers & Security	2014
143	A formal proximity model for RBAC systems	Gupta, Aditi; Kirkpatrick, Michael S.; Bertino, Elisa	Computers & Security	2014
144	SQLiGoT: Detecting SQL injection attacks using graph of tokens and SVM	Kar, Debabrata; Panigrahi, Suvasini; Sundararajan, Srikanth	Computers & Security	2016
145	Authentication graphs: Analyzing user behavior within an enterprise network	Kent, Alexander D.; Liebrock, Lorie M.; Neil, Joshua C.	Computers & Security	2015
146	A game of Droid and Mouse: The threat of split-personality malware on Android	Maier, Dominik; Protsenko, Mykola; Müller, Tilo	Computers & Security	2015

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
147	EFM: Enhancing the performance of signature-based network intrusion detection systems using enhanced filter mechanism	Meng, Weizhi; Li, Wenjuan; Kwok, Lam-For	Computers & Security	2014
148	GARS: Real-time system for identification, assessment and control of cyber grooming attacks	Michalopoulos, Dimitrios; Mavridis, Ioannis; Jankovic, Marija	Computers & Security	2014
149	Trusted Online Social Network (OSN) services with optimal data management	Park, Joon S; Kwiat, Kevin A; Kamhoua, Charles A.; White, Jonathan; Kim, Sookyung	Computers & Security	2014
150	Towards a distributed secure in-vehicle communication architecture for modern vehicles	Patsakis, Constantinos; Dellios, Kleanthis; Bouroche, Mélanie	Computers & Security	2014
151	Mobile malware detection through analysis of deviations in application network behavior	Varios autores	Computers & Security	2014
152	Reconciling user privacy and implicit authentication for mobile devices	Shahandashti, Siamak F.; Safavi-Naini, Reihaneh; Safa, Nashad Ahmed	Computers & Security	2015
153	Intrusion alert prioritisation and attack detection using post-correlation analysis	Varios autores	Computers & Security	2015
154	Selecting a trusted cloud service provider for your SaaS program	Tang, Changlong; Liu, Jiqiang	Computers & Security	2015

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
155	A survey on touch dynamics authentication in mobile devices	Teh, Pin Shen; Zhang, Ning; Teoh, Andrew Beng Jin; Chen, Ke	Computers & Security	2016
156	Towards privacy-preserving reputation management for hybrid broadcast broadband applications	Tormo, Ginés Dólera; Mármol, Félix Gómez; Pérez, Gregorio Martínez	Computers & Security	2015
157	Toward protecting control flow confidentiality in cloud-based computation	Wang, Yongzhi; Wei, Jinpeng	Computers & Security	2015
158	Vulnerabilities and mitigation techniques toning in the cloud: A cost and vulnerabilities coverage optimization approach using Cuckoo search algorithm with Lévy flights	Zineddine, Mhamed	Computers & Security	2015
159	Design strategies for a privacy-friendly Austrian eID system in the public cloud	Zwattendorfer, Bernd; Slamanig, Daniel	Computers & Security	2015
160	Detecting seam carving based image resizing using local binary patterns	Yin, Ting; Yang, Gaobo; Li, Leida; Zhang, Dengyong; Sun, Xingming	Computers & Security	2015
161	Representation and querying of unfair evaluations in social rating systems	Varios autores	Computers & Security	2014
162	Survey of certificate usage in distributed access control	Kortesniemi, Yki; Särelä, Mikko	Computers & Security	2014
163	Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon	Kokolakis, Spyros	Computers & Security	2015

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
164	Styx: Privacy risk communication for the Android smartphone platform based on apps' data-access behavior patterns	Bal, Gökhan; Rannenberg, Kai; Hong, Jason I.	Computers & Security	2015
165	Workload modelling for mix-based anonymity services	Fuchs, Karl-Peter; Herrmann, Dominik; Federrath, Hannes	Computers & Security	2015
166	Leakage-resilient password entry: Challenges, design, and evaluation	Yan, Qiang; Han, Jin; Li, Yingjiu; Zhou, Jianying; Deng, Robert H.	Computers & Security	2015
167	Incorporating attacker capabilities in risk estimation and mitigation	Ben Othmane, Lotfi Ranchal, Rohit; Fernando, Ruchith; Bhargava, Bharat; Bodden, Eric	Computers & Security	2015
168	A privacy-preserving encrypted OSN with stateless server interaction: the Snake design	Barenghi, Alessandro; Beretta, Michele; DiFederico, Alessandro; Pelosi, Gerardo	Computers & Security	2016
169	Cyber resilience recovery model to combat zero-day malware attacks	Tran, Hiep; Campos-Nanez, Enrique; Fomin, Pavel; Wasek, James	Computers & Security	2016
170	Efficiently Computing the Likelihoods of Cyclically Interdependent Risk Scenarios	Muller, Steve; Harpes, Carlo; Le Traon, Yves; Gombault, Sylvain	Computers & Security	2016
171	Detection of malicious PDF files and directions for enhancements: A state-of-the art survey	Nissim, Nir; Cohen, Aviad; Glezer, Chanan; Elovici, Yuval	Computers & Security	2015
172	Hypervisor-based malware protection with AccessMiner	Fattori, Aristide; Lanzi, Andrea; Balzarotti, Davide; Kirda, Engin	Computers & Security	2015

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
173	A multi-level approach to understanding the impact of cyber-crime on the financial sector	Lagazio, Monica; Sherif, Nazneen; Cushman, Mike	Computers & Security	2014
174	Enhanced template update: Application to keystroke dynamics	Pisani, Paulo Henrique; Giot, Romain; De Carvalho, André C.P.L.F;	Computers & Security	2016
175	HTTP attack detection using n-gram analysis	Oza, Aditya; Ross, Kevin; Low, Richard M.; Stamp, Mark	Computers & Security	2014
176	On fingerprinting probing activities	Bou-Harb, Elias; Debbabi, Mourad; Assi, Chadi	Computers & Security	2014
177	BankSealer: A decision support system for online banking fraud analysis and investigation	Carminati, Michele; Caron, Roberto; Maggi, Federico	Computers & Security	2015
178	Taxonomy of intrusion risk assessment and response system	Shameli-Sendi, Alireza; Cheriet, Mohamed; Hamou-Lhadj, Abdelwahab	Computers & Security	2014
179	Andro-Dumpsys: Anti-malware system based on the similarity of malware creator and malware centric information	Jang, Jae-wook; Kang, Hyunjae; Woo, Jiyoung; Mohaisen, Aziz; Kim, Huy Kang	Computers & Security	2016
180	Automated feature engineering for HTTP tunnel detection	Davis, Jonathan J.; Foo, Ernest	Computers & Security	2016
181	PRIDE: A practical intrusion detection system for resource constrained wireless mesh networks	Hassanzadeh, Amin; Xu, Zhaoyan; Stoleru, Radu; Gu, Guofei	Computers & Security	2016

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
182	Evaluation of TFTP DDoS amplification attack	Sieklik, Boris; Macfarlane, Richard; Buchanan, William J.	Computers & Security	2016
183	Distributed Semantic Discovery for Web-of-Things Enabled Smart Buildings	Bovet, Gerome; Hennebert, Jean	IEEE	2014
184	Cloud-based testbed for simulation of cyber attacks	Varios autores	IEEE	2014
185	Toward Increasing Awareness of Suspicious Content through Game Play	Hale, M.; Gamble, R.	IEEE	2014
186	A Security Oriented Design (SOD) Framework for eHealth Systems	Yu, Weider D.; Davuluri, Lavanya; Radhakrishnan, Monica; Runiassy, Maryam	IEEE	2014
187	IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios	Cirani, Simone; Picone, Marco; Gonizzi, Pietro; Veltri, Luca; Ferrari, Gianluigi	IEEE	2015
188	Application of MEMS as the hardware sensor on network access intrusion detection system using ALBA framework	Navarro, Eric; Paglinawan, Arnold C.	IEEE	2014
189	Secure Session on Mobile: An Exploration on Combining Biometric, TrustZone, and User Behavior	Feng, Tao; DeSalvo, Nicholas; Xu, Lei; Zhao, Xi; Wang, Xi; Shi, Weidong	IEEE	2014
190	Cloud-Based Semantic Data Management for the VPH-Share Medical Research Community	Varios autores	IEEE	2014

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
191	A sound framework for dynamic prevention of Local File Inclusion	Tajbakhsh, Mir Saman; Bagherzadeh, Jamshid	IEEE	2015
192	Self-Adaptive Volunteered Services Composition through Stimulus- and Time-Awareness	Elhabbash, Abdessalam; Bahsoon, Rami; Tino, Peter; Lewis, Peter R.	IEEE	2015
193	A Remote Code Editing Framework for AMRITA Remote Triggered WSN Laboratory	Varios autores	IEEE	2015
194	Robustness of computational intelligent assurance models when assessing e-Commerce sites	Mayayise, Thembekile O.; Osunmakinde, Isaac O.	IEEE	2015
195	Semantic modelling and automated reasoning of non-functional requirement conflicts in the context of softgoal interdependencies	Xiang, Hong; Ma, Qi Feng, Yong; Tan, Yong; Hu, Haibo Fu, Chunlei; Zhang, Tingting	IEEE	2015
196	W TaaS: An architecture of website analysis in a cloud environment	Mungekar, Shraddha; Toradmalle, Dhanashree	IEEE	2015
197	Secure integrated framework for business processes	Bhandari, Rajat; Suman, Ugrasen	IEEE	2015
198	Access control management with provenance in healthcare environments	Ma, Taotao; Wang, Hua; Cao, Jinli; Yong, Jianming; Zhao, Yueai	IEEE	2016
199	Web simulation training environment for aircraft resource planning in wildfire events	Jove, Jaume Figueras; Petit, Antoni Guasch; Casanovas-Garcia, Josep	IEEE	2015

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
200	Authorization mechanism for MQTT-based Internet of Things	Varios autores	IEEE	2016
201	Demo Abstract: SURE: An Experimentation and Evaluation Testbed for CPS Security and Resilience	Varios autores	IEEE	2016
202	SLACM: Heterogeneous multilevel service aggregation access control model based situation logic	Wang, Min; Wang, Yongbin	IEEE	2016
203	Enterprise Application Security in Android Devices Using Short Messaging Service under Unified Communication Framework	Chowdhury, Souvik; Ghosal, Prasun	IEEE	2016
204	Access control framework for API-enabled devices in smart buildings	Bandara, Syafril Yashiro, Takeshi Koshizuka, Noboru; Sakamura, Ken	IEEE	2016
205	On the security of text-based 3D CAPTCHAs	Nguyen, Vu Duc; Chow, Yang-Wai; Susilo, Willy	Computers & Security	2014
206	Determining employee awareness using the Human Aspects of Information Security Questionnaire (HAIS-Q)	Parsons, Kathryn; McCormac, Agata; Butavicius, Marcus; Pattinson, Malcolm; Jerram, Cate	Computers & Security	2014
207	Colluding browser extension attack on user privacy and its implication for web browsers	Saini, Anil; Gaur, Manoj Singh; Laxmi, Vijay; Conti, Mauro	Computers & Security	2016

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
208	Complexity is dead, long live complexity! How software can help service providers manage security and compliance	Thalmann, Stefan; Bachlechner, Daniel; Demetz, Lukas; Manhart, Markus	Computers & Security	2014
209	Information security incident management: Current practice as reported in the literature	Tondel, Inger Anne; Line, Maria B.; Jaatun, Martin Gilje	Computers & Security	2014
210	Improving mobile device security with operating system-level virtualization	Wessel, Sascha; Huber, Manuel; Stumpf, Frederic; Eckert, Claudia	Computers & Security	2015
211	A comprehensive approach for network attack forecasting	GhasemiGol, Mohammad; Ghaemi-Bafghi, Abbas; Takabi, Hassan	Computers & Security	2016
212	On the adoption of anomaly detection for packed executable filtering	Varios autores	Computers & Security	2014
213	Implementing a database encryption solution, design and implementation issues	Shmueli, Erez; Vaisenberg, Ronen; Gudes, Ehud; Elovici, Yuval	Computers & Security	2014
214	Input extraction via motion-sensor behavior analysis on smartphones	Shen, Chao; Pei, Shichao; Yang, Zhenyu; Guan, Xiaohong	Computers & Security	2015
215	When Mice devour the Elephants: A DDoS attack against size-based scheduling schemes in the internet	Serwadda, Abdul; Phoha, Vir V.	Computers & Security	2015
216	Are mobile botnets a possible threat? The case of SlowBot Net	Farina, Paolo; Cambiaso, Enrico; Papaleo, Gianluca; Aiello, Maurizio	Computers & Security	2016

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
217	Framework and principles for active cyber defense	Denning, Dorothy E	Computers & Security	2014
218	Identifying cyber risk hotspots: A framework for measuring temporal variance in computer network risk	Awan, Malik Shahzad Kaleem; Burnap, Pete; Rana, Omer	Computers & Security	2016
219	A flexible e-voting scheme for debate tools	López García, D.A.	Computers & Security	2016
220	Static analysis based invariant detection for commodity operating systems	Zhu, Feng; Wei, Jinpeng	Computers & Security	2014
221	Combating advanced persistent threats: From network event correlation to incident detection	Friedberg, Ivo; Skopik, Florian; Settanni, Giuseppe; Fiedler, Roman	Computers & Security	2015
222	Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk	Kim, Eunhyun; Park, Kyungwon; Kim, Hyoungshick; Song, Jaeseung	Computers & Security	2015
223	An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems	Almalawi, Abdulmohsen; Yu, Xinghuo; Tari, Zahir; Fahad, Adil; Khalil, Ibrahim	Computers & Security	2014
224	CPBAC: Property-based access control model for secure cooperation in online social networks	Jung, Youna; Joshi, James B.D.	Computers & Security	2014
225	Towards privacy-preserving reputation management for hybrid broadcast broadband applications	Tormo, Ginés Dólera; Mármol, Félix Gómez; Pérez, Gregorio Martínez	Computers & Security	2014

N	Artículo	Autor/es	Lugar de publicación	Año de publicación
226	Editorial: Special issue on trust in cyber, physical and social computing	Zheng Yan, Guojun; Wang, Valtteri; Niemi, Robert H Deng	Computers & Security	2014

2. *Anexo 2.* Correo de confirmación de la participación en la séptima edición de las Jornadas de Ingeniería en Sistemas Informáticos y de Computación JISIC-2016

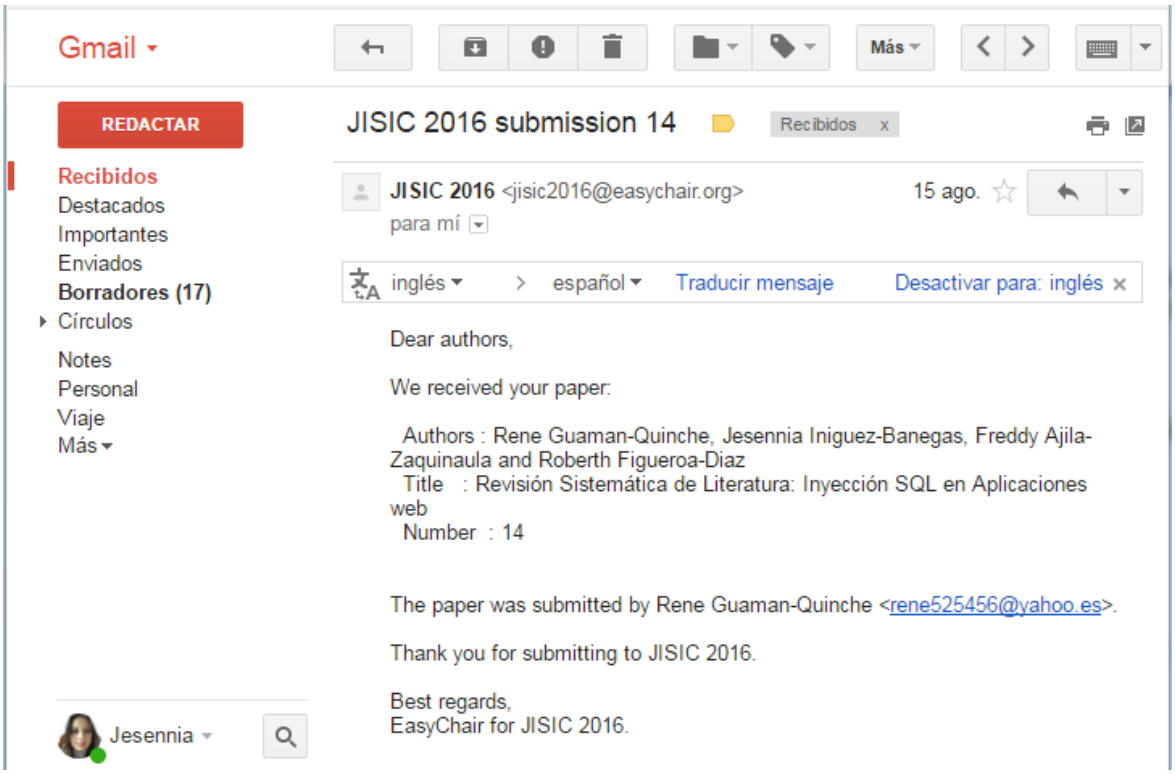

Gráfico 9. *Correo de confirmación de participación en JISIC-2016*

Revisión Sistemática de Literatura: Inyección SQL en Aplicaciones web

Jesennia Iñiguez-Banegas, Rene Guamán-Quinche

***Resumen*— La inyección SQL es una vulnerabilidad de seguridad que afecta a las aplicaciones web. Esto ocurre cuando se inserta una consulta SQL (código malicioso), por medio de las entradas de una interfaz de cliente permitiendo leer y modificar la información. El presente artículo detalla el proceso de la revisión sistemática de literatura sobre estudios primarios que plantean propuestas y solución acerca de inyección SQL. Se siguió el protocolo propuesto por Bárbara Kitchenham y se revisó un total de 9 estudios de varias revistas y conferencias. Las investigaciones sobre inyecciones SQL es todavía un tema abierto, se ha obtenido propuestas para la prevención y detección de la misma. Una de ellas es Hibrid Modeling Framework que hace frente a las vulnerabilidades de inyección SQL en la fase de diseño. Las soluciones expuestas son muchas y diversas, enfocadas en la prevención y detección de vulnerabilidades de inyección SQL.**

***Palabras clave:* mecanismos de seguridad, inyección SQL, frameworks de desarrollo, ataques inyección SQL, seguridad de aplicaciones web.**

***Abstract*— SQL injection is a security vulnerability that affects web applications. This occurs when a SQL (malicious code) query is inserted through the inputs of a client interface allowing you to read and modify information. This article details the process of systematic review of literature on primary studies that raise proposals and solution about SQL injection. Barbara Kitchenham proposed protocol was followed and a total of 9 studies of various journals and conferences was reviewed. Research on SQL injections is still an open issue, it has been obtained proposals for the prevention and detection of it. One is Hibrid Modeling Framework that addresses SQL injection vulnerabilities in the design phase. Exposed solutions are many and diverse, focused on prevention and detection of SQL injection vulnerabilities.**

***Keywords:* security mechanisms, SQL injection, development frameworks, SQL injection attacks, web application security**.

Este artículo fue enviado para su revisión el 15 de agosto de 2016

J. Iñiguez-Banegas egresada de la Carrera de Ingeniería en sistemas de la Universidad Nacional de Loja (e-mail: jesenniaib@gmail.com)

R. Guaman-Quinche, Docente Investigador de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja (e-mail: rguaman@unl.edu.ec)

INTRODUCCIÓN

LA seguridad del software es una inquietud cada vez más significativa para las instituciones del sector público o privado. Sin embargo, pocos programadores abordan este carácter de calidad de forma estratégica [1].

Los arquitectos y desarrolladores continuamente ponen un énfasis mayor en satisfacer los requerimientos prácticos y funcionales, y la seguridad usualmente es aplicada como un "adicional" para arreglar una vulnerabilidad durante o después de que la aplicación ha sido desarrollada [2].

Desarrollar código encaminado a la seguridad es una tarea que pocos la realizan por ser compleja y [3], por ello, asiduamente se recurre a la adopción y uso de frameworks que se orientan en satisfacer distintas áreas de la seguridad como por ejemplo el control de acceso a los distintos sistemas, el cifrado de la información y la validación de entradas, entre las más importantes [4].

Para mejorar la seguridad en los framework en el diseño de arquitectura, se consideran tres enfoques:

- *Ninguna adopción*: cuando la seguridad no se considera para el diseño de la arquitectura, sino, soluciones adicionales para cubrir aspectos puntuales;
- *Adopción a medias:* se usan frameworks de seguridad luego del diseño inicial de la arquitectura; y
- *Adopción total:* considera la seguridad desde el inicio dentro del diseño de la arquitectura e influye en todo el proyecto [5] [6].

Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado, por lo tanto se depende de lo sólido y flexible del mismo a la vez lo cual es un problema para el programador si no lo sabe utilizar [7][8]. Siendo así que el 80% de problemas de seguridad de sistemas web se debe que los programadores no configuran los mecanismos de seguridad de frameworks [9].

Las vulnerabilidades de aplicaciones Web se han convertido, en los últimos años, en una gran amenaza para la seguridad de sistemas informáticos [10]. Esta situación se explica por el aumento de la complejidad de tecnologías de la Web [11], por la evolución frecuente de estas tecnologías, por los ciclos cortos de desarrollo de aplicaciones Web durante el cual las actividades de prueba

y validación son limitados, y también, en algunos casos, por la falta de seguridad habilidades y cultura de los desarrolladores [12].

Según Owasp en el año 2014 el 98% de las aplicaciones web son vulnerables, lo que da como resultados un promedio de vulnerabilidades por aplicación del 20%. El servicio de almacenamiento en la nube FireHost reporta que el número de ataques de inyección de código SQL fue de cerca del 69% en el año 2012. Según un reporte, los servidores localizados en centros de datos alrededor de Europa y EUA registraron al menos medio millón de estos ataques en abril y junio de 2012; menos de 300 mil fueron registrados durante el primer trimestre [13] [14].

De las estadista descritas sobre los ataque de Inyección SQL, es necesario que se generen investigaciones para contribuir a que la información de los usuario tenga los principios de integridad, disponibilidad y confidencialidad, por ello, el propósito de este artículo es mostrar el resultado de la revisión sistemática de literatura, que fue orientada en estudios actuales en inyección SQL, la sección II presenta la metodología para desarrollar la revisión sistemática y extracción de información basada en [15][16]. En la sección III presenta los resultados obtenidos en tablas de estudios relevantes y en la sección IV se discute los principales hallazgos y la sección V se define las conclusiones del presente artículo.

METODOLOGÍA

Basado en la metodología de revisiones sistemáticas de Bárbara Kitchenham se elaboró un esquema para la revisión, selección y extracción de información quedando de la siguiente manera:

a. Pregunta de investigación.
b. Palabras clave.
c. Método de revisión.
- Fuentes y estrategias de búsqueda
- Cadenas de búsqueda,
- Criterios de selección de estudios.
- Extracción de información.

d. Estudios incluidos y excluidos

Además, se utiliza Mendeley, como gestor bibliográfico para almacenar y organizar los estudios y sus referencias.

Pregunta de investigación.

Se dirigió el alcance de este trabajo sobre artículos relacionados a mecanismos de seguridad aplicando frameworks de desarrollo. La pregunta de investigación planteada es:

¿Qué tipos de estudios primarios existen sobre mecanismos de seguridad para inyección SQL en frameworks de desarrollo?

Palabras clave.

Se realizó una revisión de literatura previa, que consistió en analizar algunos documentos relacionados al tema que facilitan identificar las palabras claves obtenidas de los títulos, resúmenes e introducción.

En la tabla 1 se detalla la lista de palabras obtenidas a través del Keywords.

TABLA 1. REVISIÓN PRELIMINAR Y TÉRMINOS.

Cód.	Titulo	Palabras clave
R01	Towards SQL Injection Attacks Detection Mechanism Using Parse Tree	SQL injection attacks, parse tree, detection, web environments.
R02	Securing Web Applications from Injection and Logic Vulnerabilities: Approaches and Challenges	SQL injection, Cross-site scripting, Business logic vulnerabilities, Application logic vulnerabilities, Web application security, Injection flaws
R03	Effective detection of vulnerable and malicious browser extensions	Browser extensions, Web security, Malware, Hidden Markov Model, JavaScript
R04	Mitigating SQL Injection Attacks Via Hybrid Threat Modelling	SQL Injection Attacks, Software Security, SDLC, SSDL, Hybrid Threat Modeling, Attack Trees, Misuse Cases, State Machines
R05	Securing Web Applications from Injection and Logic Vulnerabilities: Approaches and Challenges	SQL injection, Cross-site scripting, Business logic vulnerabilities, Application logic vulnerabilities, Web application security, Injection flaws

Una vez obtenidas las palabras claves descritas en la tabla 1, se puede realizar la construcción de la cadena de búsqueda.

Método de revisión

1) Fuentes y estrategias de búsqueda

- SCOPUS Library: https://www.scopus.com
- SCIENCEDIRECT Library: http://www.sciencedirect.com
- IEEEXPLORE Library: http://ieeexplore.ieee.org/

2) Cadenas de búsqueda

A partir de la pregunta de investigación, se definieron palabras clave para las búsquedas: Security mechanisms, SQL injection, development frameworks, SQL injection attacks, web application security.

Para generar la cadena de búsqueda se utilizaron los operadores lógicos “OR” y “AND”, quedando:
(Security mechanisms and SQL injection or SQL injection attacks and development frameworks or web application security).

Criterios de inclusión.

Es necesario aclarar que se consideró los siguientes criterios de búsquedas:
- Considerar sólo publicaciones desde el año 2014 en adelante.
- Los resultados de la búsqueda solo sean en el área de

Ciencias y Computación.

- Las producciones científicas sean estudios primarios (artículos de conferencia, artículos de revista).
- La búsqueda por su relevancia científica será en el idioma inglés.
- Los estudios deben tener información relevante a la pregunta de investigación.

Criterios de exclusión.

Los estudios que no han sido relevantes en este estudio se han excluido mediante los siguientes criterios:

- Publicaciones informales que no siguen una metodología científica.
- Todas las que no cumplan con los criterios de inclusión.

Las cadenas de búsquedas(C) utilizadas fueron las siguientes:

Biblioteca digital de SCOPUS Library:

C01: TITLE-ABS-KEY (security mechanisms AND sql injection OR sql injection attacks AND development frameworks OR web application security) AND (LIMIT-TO (PUBYEAR , 2016) OR LIMIT TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014)) AND (LIMIT-TO (SUBJAREA , "COMP"))

Biblioteca digital de SCIENCEDIRECT Library:

C02: ALL (Security mechanisms and SQL injection or SQL injection attacks and development frameworks or web application security) AND LIMIT-TO (yearnav, "2016, 2015, 2014") AND LIMIT-TO (cids, "271887","Computers & Security") AND LIMIT-TO (contenttype, "JL, BS","Journal")

Biblioteca digital de IEEEXPLORE Library:

C01: (Security mechanisms and SQL injection or SQL injection attacks and development frameworks or web application security)

3) Criterios de selección de estudios

Obtenidos los resultados de las búsquedas es conveniente describir el criterio a seguir para la selección de estudios primarios, considerando los siguientes:

- Presenten en el resumen, información actual de mecanismos de seguridad para inyección SQL en frameworks de desarrollo.
- Contener información relevante para la revisión en la introducción o conclusión.

4) Extracción de información

Los criterios de selección de estudios establecen la pauta de extracción de información relevante para este trabajo. Por cada artículo seleccionado, se sintetizará al menos uno de los siguientes elementos:

- Propuestas o modelos para prevenir inyecciones SQL
- Resultados
- Conclusiones relevantes.

Estudios incluidos y excluidos

El criterio utilizado para la selección de artículos fue que aportaran sobre la existencia de mecanismos de seguridad en los framework.

Las búsquedas realizadas generaron 24 artículos, de los cuales se registraron 5 coincidencias, es decir el número de artículos revisados fueron 19, de los cuales se seleccionaron 9 artículos de acuerdo al criterio ya mencionado.

TABLA 2. ARTÍCULOS INCLUIDOS Y EXCLUIDOS

	Artículos			
Base de Datos	Encontrados	Coincidencias	Revisados	Seleccionados
Scopus	10	2	8	6
ScienceDirect	13	3	10	2
IEEE	1	0	1	1
Total	24	5	19	9

RESULTADOS

Las siguientes tablas muestra la información relevante extraída de cada uno de los artículos seleccionados.

TABLA 3. RESULTADOS DEL ARTÍCULO SA01.

Mecanismos de seguridad	Se exhibe un esquema novedoso que transforma automáticamente las aplicaciones web, haciéndolas seguras contra ataques de inyección SQL. Esta técnica analiza dinámicamente el tamaño resultado de la consulta desarrollador destinados a cualquier entrada, y detecta los ataques de comparar esto contra el resultado de la consulta real.
Resultados	La evaluación empírica demostró que IDL (Injection Detector Libraries) consume más tiempo para la detección de los algoritmos existentes, ya que incluye muchos pasos que se consideran métodos sintácticos para dar resultados más precisos. Sin embargo, este es un método que puede detectar con mayor precisión que SQLIA.
Conclusiones relevantes	Mediante el uso de una variable de entrada de sustitución y desinfección basada en el tamaño de la consulta, es posible detectar y prevenir las consultas SQL que incluyen vulnerabilidades de inyección.

TABLA 4. RESULTADOS DEL ARTÍCULO SA02.

Mecanismos de seguridad	Una nueva técnica o método para detectar SQLIA mediante el modelado de las consultas SQL como una gráfica de tokens y el uso de la medida de centralidad de los nodos para entrenar a una máquina de vectores soporte (SVM).
Resultados	Los resultados experimentales demuestran que esta técnica puede identificar con eficacia las consultas SQL maliciosas con sobrecarga de rendimiento insignificante.
Conclusiones relevantes	El sistema no requiere la construcción de un modelo de uso normal de las consultas, ni requiere el acceso al código fuente.

TABLA 5. RESULTADOS DEL ARTÍCULO SA03.

Mecanismos de seguridad	Propone una metodología de la vulnerabilidad y ataque de inyección para SQLi y XSS se puede aplicar a una variedad de configuraciones y tecnologías. Se basa en la idea de que podemos evaluar diferentes atributos de los mecanismos de seguridad de aplicaciones web

	existentes mediante la inyección de vulnerabilidades realistas en una aplicación web y atacar de forma automática.
Resultados	Los resultados muestran que la inyección de vulnerabilidades y ataques es de hecho una forma eficaz para evaluar los mecanismos de seguridad y para señalar no sólo sus debilidades, si no también formas para su mejora.
Conclusiones relevantes	Concluye que aproximadamente la mitad de las vulnerabilidades SQLi provienen de la explotación de los campos numéricos.

TABLA 6. RESULTADOS DEL ARTÍCULO SA04.

Mecanismos de seguridad	Un mecanismo de detección concreta basado en DSD (Dynamic Detección SQLIAs) se plantea para detectar SQLIAs mediante el uso de árbol de análisis sintáctico. La principal ventaja de la propuesta es que no requiere acceder al código fuente de las aplicaciones si no que es incorporado para entornos web existentes.
Resultados	Los resultados experimentales demostraron que el mecanismo tiene una mayor precisión (99.9%), menor tasa de falsos positivos (2%) y falsos negativos cuando se detecta SQLIAs. Por lo tanto, es un eficiente mecanismo de detección SQLAS para entornos web.
Conclusiones relevantes	El mecanismo propuesto no requiere acceder al código fuente de las aplicaciones. Esto significa que DSD se puede aplicar directamente a aplicaciones web existentes. Por lo tanto, es un eficiente mecanismo de detección de SQLIAs para entornos web.

TABLA 7. RESULTADOS DEL ARTÍCULO SA05.

Mecanismos de seguridad	Propone una amenaza Hybrid Modeling Framework, polilla, para hacer frente a vulnerabilidades de inyección SQL en la fase de diseño, una fase de desarrollo temprana del SDLC (ciclo vital del desarrollo/diseño de sistemas).
Resultados	Como resultado de los puntos de entrada ampliado, los investigadores han desplazado los engranajes de los enfoques reactivos prevalentes de SQLIAs la prevención de una estrategia de gestión de riesgos proactiva llamada de amenaza de modeling un ejercicio realizado en la fase de diseño del SDLC.
Conclusiones relevantes	La seguridad es un proceso continuo que debe ser integrado en las aplicaciones desarrolladas de solicitud a través de la liberación de mantenimiento.

TABLA 8. RESULTADOS DEL ARTÍCULO SA06.

Mecanismos de seguridad	Un enfoque basado en la técnica HMM (Modelo oculto de Markov) para detectar las extensiones del navegador vulnerable y malicioso, ampliando y complementando las técnicas existentes. Estas técnicas se centran principalmente en el análisis de flujo de información.
Resultados	Se implementa en una herramienta de prototipo y evaluó utilizando un número de 387 extensiones de Mozilla Firefox. Indican que el enfoque no sólo detecta extensiones vulnerables y maliciosos conocidas, sino que también identifica previamente no detectados, extensiones con una sobrecarga de rendimiento insignificante. La precisión de falso positivo 97,68%.
Conclusiones relevantes	El número de muestras utilizadas durante la evaluación es pequeño para apoyar la eficacia de HMM, nuestro enfoque se puede utilizar como una técnica complementaria a los enfoques existentes.

TABLA 9. RESULTADOS DEL ARTÍCULO SA07.

Mecanismos de seguridad	Una revisión de la literatura que resume el estado actual de la técnica para asegurar las aplicaciones web de los principales defectos tales como errores de inyección y lógicas. Aunque existen diferentes tipos de errores de inyección, el alcance se limita a la inyección de SQL (SQLI) y Cross-site scripting (XSS), ya que son calificados como los mejores
	entre la mayoría de las amenazas de los diferentes consorcios de seguridad.
Resultados	Se necesita más investigación en el área de los defectos de fijación en el código fuente de las aplicaciones. La mayoría de los artículos se centran en la detección de los defectos y la prevención de ataques contra las aplicaciones web.
Conclusiones relevantes	A pesar de que varios enfoques están disponibles para asegurar las aplicaciones web de SQLI y XSS, son todavía muy extendido debido a su impacto y la gravedad. Este articulo proporciona una revisión integral de los recientes avances en la obtención de las vulnerabilidades de inyección y la lógica de negocio de las aplicaciones web, y señala los problemas no resueltos que deben abordarse.

TABLA 10. RESULTADOS DEL ARTÍCULO SA08.

Mecanismos de seguridad	Marco de seguridad adopción de Cloud Computing (CCAF) adecuada para negocios nubes. Se basa en el desarrollo y la integración de las tres principales tecnologías de seguridad: firewall, gestión de identidad y cifrado basado en el desarrollo de la empresa, sincronización de archivos y las tecnologías de Acciones.
Resultados	Los resultados en la primera hora y 24 horas pruebas mostraron que una CCAF protección de seguridad completa de múltiples capas podría bloquear y evitar la inyección de SQL para MySQL y MongoDB. En las pruebas de penetración, de seguridad en capas múltiples CCAF podría detectar y bloquear el 99,95%
Conclusiones relevantes	Una protección de seguridad multicapa completa CCAF podría bloquear toda inyección SQL y proporcionar una protección real a los datos.

TABLA 11. RESULTADOS DEL ARTÍCULO SA09.

Mecanismos de seguridad	Nueva metodología, basada en técnicas de agrupamiento de las páginas web, que está dirigido a identificar las vulnerabilidades de una aplicación web después de un análisis de cuadro negro de la aplicación de destino.
Resultados	Este enfoque también condujo al desarrollo de un nuevo escáner de vulnerabilidad denominada Wasapy.
Conclusiones relevantes	Enriquecer las gramáticas implementadas en Wasapy para permitir la generación de una variedad más grande de inyecciones que cubren las vulnerabilidades incluidos hasta el momento, así como las nuevas vulnerabilidades.

DISCUSIÓN

Principales hallazgos

SA01: Se describe una técnica de adulteración positivo que caracterizan IZES el proceso de desinfección mediante el modelado de la forma en que una aplicación procesa los valores de entrada. En base al uso de una variable de entrada de sustitución y desinfección basada en el tamaño de la consulta, es posible detectar y prevenir las consultas SQL que incluyen vulnerabilidades de inyección. La evaluación empírica demostró que IDL (Injection Detector Libraries) a pesar de consumir más tiempo en la detección de algoritmos, es eficaz contra el conjunto de pruebas SQLIA. Los IDL tienen tres pasos principales: Paso 1 comprueba los patrones de ataque contra las expresiones regulares. Si ninguna regla coincide, entonces ese patrón de ataque se envía al sistema de detección. Paso 2 analiza el

patrón de ataque utilizando nuestra base de datos interna, llamado un "conjunto de reglas," para clasificar la vulnerabilidad. Para evaluar la cadena de consulta, el IDL utiliza un analizador de SQL para dividirla en una secuencia de símbolos que corresponden a palabras clave de SQL, operadores y literales. El IDL luego itera a través de los tokens y comprueba si las que no son literales contienen exclusivamente los datos de confianza. Si todas las fichas pasan esta comprobación, la consulta se considera segura y se puede ejecutar. Por último, el paso 3 se sustituye caracteres o cadenas vulnerables en la consulta SQL. En particular, esto incluye funciones que eliminan o sustituyen ciertos caracteres o cadenas de su entrada.

SA02: Muestra una nueva técnica para detectar SQLIA mediante el modelado de las consultas SQL como una gráfica de tokens y el uso de la medida de centralidad de los nodos para entrenar a una máquina de vectores soporte (SVM). El enfoque fue diseñado para trabajar en la capa de base de datos y servidor de seguridad se implementó en un prototipo llamado SQLiGoT, además utiliza las funciones disponibles en la mayoría de los lenguajes de programación modernos y puede ser portado a otras plataformas sin necesidad de grandes modificaciones. El sistema fue probado exhaustivamente el uso de grafos no dirigidos y dirigidas con dos diferentes métodos de borde de ponderación. Proponen diseños alternativos de la SVM clasificador, que consiste en simples y múltiples, probados y comparados. Los resultados experimentales obtenidos en cinco aplicaciones web totalmente vulnerables confirman la eficacia que tiene. El sistema no requiere la construcción de un modelo de uso normal de las consultas, ni requiere el acceso al código fuente.

SA03: La metodología propuesta ofrece un entorno práctico que se puede utilizar para probar mecanismos de contramedida (como los sistemas de detección de intrusos (IDS), escáneres de vulnerabilidades de aplicaciones web, paredes de fuego de aplicaciones web, analizadores de código estático, etc.), y el tren evaluar los equipos de seguridad, ayudar a estimar las medidas de seguridad (como el número de vulnerabilidades presentes en el código), entre otros. Esta evaluación de herramientas de seguridad puede realizarse en línea mediante la ejecución del inyector de ataque, mientras que la herramienta de seguridad también está en marcha; o fuera de línea mediante la inyección de un conjunto representativo de las vulnerabilidades que se pueden utilizar como banco de pruebas para evaluar una herramienta de seguridad. La metodología se llevó a cabo en una vulnerabilidad de hormigón y herramienta del inyector Ataque (Vait) para aplicaciones web. El primero en evaluar la eficacia de Vait en la generación de un gran número de vulnerabilidades realistas para la evaluación en línea de herramientas de seguridad, en particular los escáneres de vulnerabilidades de aplicaciones web. El segundo para mostrar cómo se puede explotar las vulnerabilidades inyectadas para lanzar ataques, permitiendo la línea evaluación de la eficacia de los mecanismos de contramedida instalados en el sistema de destino, en particular un sistema de detección de intrusos. Estos experimentos ilustran cómo propuesta se puede utilizar en la práctica, no sólo para descubrir debilidades existentes de las herramientas analizadas, sino también para ayudar a mejorarlas.

SA04: Propone un interesante mecanismo de detección concreta basada en DSD, se plantea para detectar SQLIAs mediante el uso de árbol de análisis sintáctico. La principal ventaja de la propuesta es que no requiere acceder al código fuente de las aplicaciones si no que es incorporado para entornos web existentes. El DDS consiste en cinco unidades: Collector1, Collector2, Repositorio1, Repositorio2, y Agente y SQLIAs. El mecanismo consta de dos fases: de clasificación y detección. Cuando un usuario envía una solicitud HTTP a una aplicación, la fase de clasificación está involucrado para identificar la solicitud si se trata de primer acceso en tiempo o tiempo de acceso no primero. Después de eso, la fase de detección proporciona una detección SQLIA para esta aplicación en los dos casos anteriores. La exactitud de este mecanismo es más de 99.9% para cada tipo de aplicación típica, la tasa de falsos positivos es menos de 2% para cada tipo de aplicaciones.

.

SA05: Estudio de amenaza Hybrid Modeling Framework, polilla, para hacer frente a vulnerabilidades de inyección SQL en la fase de diseño, una fase de desarrollo temprana del SDLC, el modelado de amenazas implica el descubrimiento de la superficie de ataque explotable del activo de software mediante el examen de todos los límites de confianza, el flujo de datos, incluyendo los caminos de entrada y salida de todos los puntos de entrada. Los casos de mal uso (CUG), árboles de ataque (ATS) y máquinas de estado de comportamiento (MAN) se combinan en una técnica híbrida para diseñar un modelo de amenazas necesaria para proporcionar los requisitos de seguridad optimas y activos de software, concluyendo que la seguridad es un proceso continuo que debe ser integrado en las aplicaciones desarrolladas de solicitud a través de la liberación de mantenimiento.

SA06: Se enfoca en técnica basada en HMM (Modelo oculto de Markov) para detectar las extensiones del navegador vulnerable y malicioso, ampliando y complementando las técnicas existentes. Estas técnicas se centran principalmente en el análisis de flujo de información, para capturar los flujos de datos sospechosos, imponer la restricción de privilegios de llamadas a la API de extensiones maliciosos, aplicar firmas digitales para supervisar las actividades del proceso y el nivel de memoria, y permitir a los usuarios del navegador especificar las políticas con el fin de restringir las operaciones de extensiones. Se implementa en una herramienta de prototipo y evaluó utilizando un número de 387 extensiones de Mozilla Firefox. Indican que el enfoque no sólo detecta extensiones vulnerables y maliciosos conocidas, sino que también identifica previamente no detectados, extensiones con una sobrecarga de rendimiento insignificante. La precisión de falso positivo 97,68%.

SA07: Presenta una revisión sistemática de la literatura de los recientes avances en la obtención de las vulnerabilidades de inyección de las aplicaciones web. El objetivo de este estudio es resumir el estado actual de la técnica para asegurar las aplicaciones web de los principales defectos tales como errores de inyección y lógicas. Se exploran principalmente los siguientes puntos:

- Se analizan diversos tipos de vulnerabilidades y ataques que explotan estas vulnerabilidades en aplicaciones web.
- Se analizan los pros y los contras de los enfoques de mitigación para proteger las aplicaciones web de inyección y de negocios vulnerabilidades lógicas
- Proporciona información sobre las capacidades de los escáneres de vulnerabilidad existentes.
- Se destacan las aplicaciones web de código abierto que pueden utilizarse para la prueba y evaluación.

A pesar de que varios enfoques están disponibles para asegurar las aplicaciones web de SQLI y XSS, son todavía muy extendido debido a su impacto y la gravedad

SA08: framework de seguridad de varios niveles adopción de Cloud Computing (CCAF) adecuada para negocios en las nubes. Se basa en el desarrollo y la integración de las tres principales tecnologías de seguridad: firewall, gestión de identidad y cifrado basado en el desarrollo de la empresa, sincronización de archivos y las tecnologías de Acciones. Describe la tecnología de seguridad básica de Empresa sincronización de archivos y Compartir, la arquitectura y componentes en capas, y las tecnologías y resultados básicos de varias capas de experimentos a gran escala para las pruebas de penetración, inyección SQL y escaneo de datos. Dando como resultado en las pruebas de penetración que podría detectar y bloquear el 99,95% los virus troyanos y mantener un 85%, por encima de bloquear durante 100 horas continúas de ataques. Una protección de seguridad multicapa completa CCAF podría bloquear toda inyección SQL que proporciona una protección real a los datos. CCAF seguridad multicapa tenía tasa de 100% de no informar de falsa alarma.

SA09: nueva metodología que permite identificar automáticamente las vulnerabilidades residuales de una aplicación web a partir del análisis de la aplicación específica, siguiendo un enfoque cuadro negro puesto que no requiere detalles de la implementación del código fuente de la página Web. Está diseñado para poner de relieve los posibles escenarios de ataque, incluyendo la explotación de vulnerabilidades varios sucesivos que no son necesariamente independientes. Utilizan una herramienta Wasapy (Evaluación de la Seguridad Web de aplicaciones en Python.) de software utilizando el lenguaje Python, lo que facilita enormemente el manejo de conceptos HTTP (cookies, configuraciones, etc.). Se implementa un escáner de vulnerabilidades Web que contribuyen a enriquecer las gramáticas implementadas en Wasapy para permitir la generación de una variedad más grande de inyecciones que cubren las vulnerabilidades incluidos hasta el momento, así como las nuevas vulnerabilidades.

No hay una solución única hasta el momento que pueda eliminar las vulnerabilidades y prevenir ataques SQL. Por lo tanto, una serie de técnicas de mitigación debe ser empleado para frenar la propagación de los ataques y eliminar las vulnerabilidades SQL.

La solución más ideal es eliminar vulnerabilidades SQL de la raíz, es decir, el código fuente. Sin embargo, en las aplicaciones web en el mundo real, la obtención del código fuente o parches puede ser difícil. Por lo tanto, las técnicas de análisis estático son más útiles durante el desarrollo de la aplicación y antes del despliegue. Las técnicas de análisis dinámicos como técnica de pruebas de penetración pueden ser utilizadas para explotar las aplicaciones web durante el tiempo de ejecución con el fin de determinar si todavía son vulnerables a ataques SQL.

CONCLUSIONES Y TRABAJOS FUTUROS

El trabajo se centra en 9 artículos relacionados con la investigación de inyección SQL. Se identifica las soluciones, métodos, técnicas propuestas en los estudios. Las soluciones propuestas son muchas y diversas, en su mayoría se enfocan en la prevención de ataques de inyección SQL y detección de vulnerabilidades. Solo el estudio SA05, discute la eliminación de vulnerabilidades de inyección SQL a partir del código fuente, lo que es importante para prevenir ataques y ahorrar recursos en post-implementación. Los artículos SA01, SA02, SA08, hacen un análisis sintáctico a través de tokens, SVM (máquina de vectores soporte), IDL (Injection Detector Libraries) para la detección de SQLIA. Los documentos SA03, SA04, SA06, SA09, realizan análisis semánticos de diferentes mecanismos de seguridad como DSD (Detección SQLIAs Dinámico), IDS (Sistema de detección de intrusos), HMM (Modelo oculto de Markov), CCAF (Framework de adopción de Cloud Computing) y Wasapy en la detección de vulnerabilidades. Y un estudio proporciona una revisión integral de los recientes avances en la obtención de las vulnerabilidades de inyección. Todos los estudios hacen frente a los problemas relacionados con inyecciones SQL para eliminarlos. Pero los ataques son cada vez más frecuentes, así que la seguridad debe ser tratada en todas las fases de desarrollo considerando la seguridad desde el principio y en todo el ciclo de vida de la aplicación y usar framework para soportarla puede dar excelentes resultados. El análisis de estudios primarios indica la facilidad con la que puede ser vulnerada una aplicación cuando no se les asigna una prioridad adecuada a los controles de seguridad en las distintas etapas de desarrollo.

En el futuro, compararemos nuestro estudio con, los estudios propuestos y publicados sobre ataques de inyección SQL, para ver cuál ha sido el avance y eficacia de los mecanismos de protección. También fortalecer la investigación con, el estudio de los diferentes tipos de escáneres de vulnerabilidades existentes para mitigar la Inyección SQL y las diferentes vulnerabilidades de las aplicaciones web.

ARTÍCULOS SELECCIONADOS EN LA REVISIÓN SISTEMÁTICA.

SA01.-Y. S. Jang and J. Y. Choi, "Detecting SQL injection attacks using query result size," Comput. Secur., vol. 44, pp. 104–118, 2014.

SA02.-D. Kar, S. Panigrahi, and S. Sundararajan, "SQLiGoT: Detecting SQL Injection Attacks using Graph of Tokens and SVM," Comput. Secur., 2016.

SA03.-J. Fonseca, N. Seixas, M. Vieira, and H. Madeira, "Analysis of field data on web security vulnerabilities," IEEE Trans. Dependable Secur. Comput., vol. 11, no. 2, pp. 89–100, 2014.

SA04.-T. N. Aung and S. S. Khaing, "Genetic and Evolutionary Computing," Adv. Intell. Syst. Comput., vol. 388, pp. 405–411, 2016.

SA05.-H. Omotunde and R. Ibrahim, "Mitigating SQL injection attacks via hybrid threat modelling," 2015 IEEE 2nd Int. Conf. InformationScience Secur. ICISS 2015, pp. 15–18, 2016.

SA06.-H. Shahriar, K. Weldemariam, M. Zulkernine, and T. Lutellier, "Effective detection of vulnerable and malicious browser extensions," Comput. Secur., vol. 47, pp. 66–84, 2014.

SA07.-G. Deepa and P. S. Thilagam, "Securing web applications from injection and logic vulnerabilities: Approaches and challenges," Inf. Softw. Technol., vol. 74, pp. 160–180, 2016.

SA08.-V. Chang, Y. H. Kuo, and M. Ramachandran, "Cloud computing adoption framework: A security framework for business clouds," Futur. Gener. Comput. Syst., vol. 57, pp. 24–41, 2016.

SA09.-R. Akrout, E. Alata, M. Kaaniche, and V. Nicomette, "An automated black box approach for web vulnerability identification and attack scenario generation," J. Brazilian Comput. Soc., vol. 20, no. 1, p. 4, 2014.

REFERENCIAS

R. A. Oliveira, N. Laranjeiro, and M. Vieira, "Assessing the security of web service frameworks against Denial of Service attacks," J. Syst. Softw., vol. 109, pp. 18–31, 2015.

M. Castro-león, F. Boixader, M. Taboada, D. Rexachs, E. Universitaria, and T. Cerdá, "Servicios y Seguridad, un enfoque basado en estrategias de ataque y defensa," pp. 39–48, 2015.

D. CAMACHO, G. MARTINEZ, and D. BIANCHA, "Diseño De Framework Web Para El Desarrollo Dinámico De Aplicaciones," no. 44, pp. 178–183, 2010.

M. D. P. Salas-Zárate, G. Alor-Hernández, R. Valencia-García, L. Rodríguez-Mazahua, A. Rodríguez-González, and J. L. López Cuadrado, "Analyzing best practices on Web development frameworks: The lift approach," Sci. Comput. Program., vol. 102, pp. 1–19, 2015.

H. Cervantes, R. Kazman, and J. Ryoo, "Seguridad y uso de Frameworks _ SG." p. SG # 47, 2015.

A. R. Sartorio, G. L. Rodríguez, and M. Vaquero, "Investigación en el diseño y desarrollo para el enriquecimiento de un framework colaborativo web sensible al contexto," XIII Work. Investig. en Ciencias la Comput., pp. 1–5, 2011.

C. García, R. Hervás, and P. D. A.-/9 L. B.-G. Gervás, "Una Arquitectura Software para el Desarrollo de Aplicaciones de Generación de Lenguaje Natural," Soc. Española para el Proces. del Leng. Nat. Proces. Leng. Nat., vol. 33, pp. 111–118 ST – Una Arquitectura Software para el De, 2004.

G. Martínez Villalobos, G. D. Camacho Sánchez, and D. A. Biancha Gutiérrez, "Diseño de Framework web para el desarrollo dinámico de aplicaciones," Sci. Tech., vol. XVI, no. 44, pp. 178–183, 2010.

H. T. Quinche, René Guamán, "Seguridad en Entornos Web para Sistemas de Gestión Académica," pp. 1–47.

R. Akrout, E. Alata, M. Kaaniche, and V. Nicomette, "An automated black box approach for web vulnerability identification and attack scenario generation," J. Brazilian Comput. Soc., vol. 20, no. 1, p. 4, 2014.

A. María Reina Quintero, "Separación avanzada de conceptos en entornos WEB.," pp. 3–16.

G. Deepa and P. S. Thilagam, "Securing web applications from injection and logic vulnerabilities: Approaches and challenges," Inf. Softw. Technol., vol. 74, pp. 160–180, 2016.

Owasp, "OWASP Top 10 - 2013," OWASP Top 10, p. 22, 2013.

J. I. Calderón, "Seguridad en Aplicaciones Web."

S. E. Group and R. Unido, "Directrices para la realización sistemática de la literatura críticas en Ingeniería de Software Sección de Control de Documentos," 2007.

B. Kitchenham, "Procedures for performing systematic reviews," Keele, UK, Keele Univ., vol. 33, no. TR/SE-0401, p. 28, 2004.

Jesennia Iñiguez, Egresada de la Carrera de Ingeniera en Sistemas de la Universidad Nacional de Loja. Líneas de interés: seguridad web, redes y telecomunicaciones. Docente en Unidad Educativa Calasanz (septiembre 2013 – enero 2014). Instructora de computación en Compucenter Technology (Sept. 2014 – noviembre 2014). Técnico y operados de cibercafé en Gyg@net (enero 2012 – noviembre 2014). Ciudad Loja, Ecuador, 2016

Rene Guamán Quinché, Docente Investigador de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja. Líneas de interés en tecnologías web y móviles, sistemas distribuidos y paralelos. Máster en Sistemas Informáticos Avanzados en la Universidad del País Vasco. Ciudad Loja, Ecuador, 2016

3. Anexo 3. **Correo de confirmación de la aceptación del artículo en la séptima edición de las Jornadas de Ingeniería en Sistemas Informáticos y de Computación JISIC-2016**

Imprimir https://es-mg42.mail.yahoo.com/neo/launch?.ra...

Asunto: Publicación de artículos

De: TORRES OLMEDO JENNY GABRIELA (jenny.torres@epn.edu.ec)

Para:

Fecha: Miércoles 19 de octubre de 2016 21:42

Estimado,

Nos es grato comunicarle que su artículo ha sido seleccionado para su publicación en las JISIC 2016. Para que esta publicación se efectúe, usted debe realizar una presentación oral de 30 minutos sobre su trabajo. Por favor programe 25 minutos de presentación y 5 para preguntas del público, para el día jueves 17 de noviembre de 14.30 a 16.30.

Para mayor información http://jisic.epn.edu.ec. Las inscripciones ya se encuentran abiertas.

De tener dudas o inquietudes sobre su presentación, ni dude en contactarnos.

Saludos,

Dra. Jenny Torres
JISIC 2016

Gráfico 10. *Correo de aceptación del artículo*