DISEÑO DE PROCEDIMIENTO PARA LA PROTECCION DE ATAQUES POR INYECCION SQL A BASES DE DATOS SQL Y NOSQL

EYMAR SILVA MEZA
MARBIZ SAID DUCUARA AMADO

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA UNAD
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
YOPAL
2017

DISEÑO DE PROCEDIMIENTO PARA LA PROTECCION DE ATAQUES POR INYECCION SQL A BASES DE DATOS SQL Y NOSQL

EYMAR SILVA MEZA
MARBIZ SAID DUCUARA AMADO

Trabajo de grado para optar el título de
Especialista en Seguridad Informática

Asesor
Ing. Francisco Nicolás Solarte Solarte

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA UNAD
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
YOPAL
2017

Nota de aceptación

_____
_____
_____
_____
_____
_____
_____
_____

_____
Presidente del jurado

_____
Jurado

_____
Jurado

Yopal 05,12,2017

CONTENIDO

Pág.

LISTA DE FIGURAS

Pag.

LISTA DE TABLAS

Pag.

GLOSARIO

ATAQUE INYECCIÓN SQL: Consiste en la intrusión o alteración de comandos SQL para observar datos aparentemente ocultos con el fin de ingresar, sobrescribir, extraer datos o ejecutar comandos que pongan en riesgo una base de datos o el equipo que la contenga. Por lo general se basa en un inadecuado filtrado de la información que se envía por medio de los campos y/o variables que usa un sitio web. Puede ser explotada a través del método GET como del método POST. El atacante puede usar scripts automatizados con el fin de ejecutar varios comandos hasta lograr realizar la intrusión a la base de datos.

XSS - CROSS SITE SCRIPTING: Se basa en inyectar código llamados scripts atacando al cliente y no al servidor, la finalidad de este ataque es robar información, enviar amenazas, obtener las cookies de acceso o ataques de phishing. El desarrollo de estas amenazas se encuentra normalmente en lenguajes como JavaScript, HTML, VBScript, ActiveX o Flash.

NMAP: Es una herramienta de código abierto para escaneo de red y auditoría de seguridad, puede servir incluso para inventariar la red. Utiliza paquetes IP para establecer qué servicios se encuentran disponibles en una red, qué sistemas operativos se están ejecutando, qué tipo de filtrado de paquetes o cortafuegos se está utilizando, entre otras características.

SQLMAP: Herramienta de pruebas de penetración de código abierto que automatiza el proceso de detectar y explotar los errores de inyección SQL y de carga de los servidores de bases de datos. Soporte completo para MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB, HSQLDB e Informix. Soporte completo para seis técnicas de inyección SQL: boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries and out-of-band.

OWAS ZAP: Herramienta que permite analizar sitios web con el fin de buscar sus vulnerabilidades con el objetivo de aprender técnicas de penetración o mejorar la seguridad del sitio. Tiene en cuenta la lista de vulnerabilidades web de OWASP, por lo que incluye una gran cantidad de herramientas que pueden detectar casi cualquier vulnerabilidad que pueda existir en un sitio web.

VIRTUALBOX: es un software de virtualización x86 y AMD64 / Intel64 para uso empresarial y doméstico. Permite crear un sistema virtual en un único medio o anfitrión a través de un software que permite representarlo como si estuviera instalado físicamente. Existen diferentes tipos de virtualización entre los cuales están la de servidores, de escritorio, de aplicaciones y de almacenamiento entre

otros. Actualmente, VirtualBox funciona en los anfitriones de Windows, de Linux, de Macintosh, y de Solaris y apoya un gran número de sistemas operativos del huésped incluyendo pero no limitado a Windows (NT 4.0, 2000, XP, servidor 2003, Vista, Windows 7, Windows 8, Windows 10), DOS / Windows 3.x, Linux (2.4, 2.6, 3.xy 4.x), Solaris y OpenSolaris, OS / 2 y OpenBSD.

KALI LINUX: Es una distribución de Linux avanzada para pruebas de penetración y auditorías de seguridad. Es una reconstrucción de BackTrack Linux, se adhiere al estándar del entorno Debian y contiene más de 300 herramientas de pruebas de penetración.

AMENAZA: Causa potencial de un incidente no deseado, que puede provocar daños a un sistema o a la organización. (ISO/IEC 27000).

ATAQUES WEB: ejecutado contra una aplicación cliente y se origina desde un lugar en la Web, ya sea desde sitios legítimos atacados o sitios maliciosos que han sido creados para efectuar los ataques.

FIREWALL: es una aplicación de seguridad diseñada para bloquear las conexiones en determinados puertos del sistema.

SISTEMA DE DETECCIÓN DE INTRUSOS INTRUSION DETECTION System (IDS): Es una herramienta (hardware o software) que complementa a un esquema de seguridad de una organización utilizada principalmente para identificar ataques en tiempo real, guardar los registros y reportar al equipo de TI con el fin de tomar acciones al respecto.

WIRESHARK: es un analizador de protocolos, funciona en Windows y Linux, permite realizar un sniffing en la red donde se encuentre conectado.

RESUMEN

Al paso de los años las organizaciones han crecido y han aumentado su inversión en tecnología, con el auge de los computadores y el internet cada día se automatiza más sus actividades y buscan mejorar sus procesos y sus sistemas de información. Sin embargo, al avanzar en este campo también se debe tener en cuenta que las vulnerabilidades están presentes en los diferentes sistemas: llámese, redes, bases de datos, aplicaciones, sistemas operativos entre otros y que a partir de ese crecimiento surgen las amenazas sobre la integridad, confidencialidad e integridad de los datos.

El propósito del presente trabajo es identificar algunas de las principales técnicas o métodos para la protección frente a ataques por inyección SQL tanto en las transacciones como en los accesos de algunos motores de base de datos sql y bases de datos no sql, mediante la realización de test de penetración a través de entornos de prueba controlados que permitan explorar y explotar vulnerabilidades con la utilización de una herramienta muy conocida como lo es Kali Linux, que contiene una suite completa para realizar diferentes pruebas con el fin de determinar aquellas mejores prácticas que orienten los procedimientos y mecanismos de control para prevenir un evento que atente contra la confidencialidad, integridad y disponibilidad de la información salvaguardada en las bases de datos.

ABSTRACT

Over the years organizations have grown and increased their investment in technology with the rise of computers and the internet every day is more automated its activities and seek to improve their processes and information systems. However, in advancing in this field it should also be taken into account that vulnerabilities are present in the different systems networks, databases, applications, operating systems and more and that from that growth threats on the integrity, confidentiality and integrity of the data arise.

The purpose of this paper is to identify some of the main techniques or methods for protection against SQL injection attacks both in the transactions and in the accesses of some sql database engines and non-sql databases by conducting penetration testing through controlled test environments that allow the exploration and exploitation of vulnerabilities with the use of a well-known tool like Kali Linux which contains a complete suite to perform different tests in order to determine those best practices that guide the procedures and control mechanisms to prevent an event that violates the confidentiality, integrity and availability of information safeguarded in databases

INTRODUCCIÓN

En la actualidad la seguridad informática ha venido adquiriendo gran importancia tanto a nivel personal como dentro de la misión las entidades, siendo indispensable en este ultimo la elaboración de estrategias para el mejoramiento de los recursos tecnológicos en la implementación de un Sistema de gestión de seguridad de la información. En una entidad siempre prevalecerá la importancia de mantener sus activos seguros, una parte fundamental de los activos de una empresa es la información; datos sensibles que en manos equivocadas pueden acarrear serios problemas con consecuencias no deseadas. Los profesionales en el área de la seguridad informática deben contar con conocimientos bien fundados y ejercer sus funciones siempre enmarcadas en las normas vigentes y los procedimientos creados para ello.

Con el acceso a Internet es muy probable el enfrentarse a riesgos y amenazas cada vez más sofisticados, los atacantes también evolucionan y mejoran sus técnicas de ataque con el fin de obtener resultados eficaces y salir siempre exitosos. Implementar un esquema de seguridad en una organización es una tarea que requiere bastante planificación donde se involucran aspectos no solamente tecnológicos como software y hardware sino de recurso humano, tanto de las personas que trabajan en el área de TI como de las personas que laboran en la entidad. Los ataques hacia una organización dependerán de las intenciones y los fines que busca el atacante, no obstante, la información que maneja la organización también determina qué tipo de estructura y herramientas de seguridad se utilizaría para protegerla, teniendo en cuenta estos aspectos es así que el presente documento aborda algunos de los principales ataques que ocurren en las bases de datos mediante diversas técnicas que se han desarrollado a medida que la tecnología ha avanzado a través del tiempo. Los ciberdelincuentes se enfocan en el estudio de las vulnerabilidades que los DBMS puedan tener, a su vez que tienen en cuenta las formas de reacción de los usuarios permitiendo hacer más fácil su tarea de poder hacer una intrusión y alterar, dañar o sustraer los datos incidiendo contra la confidencialidad, integridad o disponibilidad de la información con el fin de obtener un beneficio por lo general de tipo monetario.

También se muestran algunas de las herramientas que permiten el análisis de vulnerabilidades a las bases de datos, así como otras que permiten realizar una evaluación de los controles implementados con el fin de identificar factores claves de éxito para blindarse contra este tipo de ataques y mejorar la seguridad en las bases de datos.

1. DEFINICIÓN DEL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

La inyección SQL cumple casi dos décadas desde su aparición y pese a los avances tecnológicos en seguridad, continúa siendo una de las vulnerabilidades más conocidas y explotadas. En la actualidad encabeza la lista del Top 10 de amenazas OWASP. El impacto negativo de los ataques a través de la historia ha demostrado que pueden abarcar entro otros: la pérdida de cifras millonarias, vulneración de datos personales de millones de usuarios, desprestigio, suplantación de identidad, etc.

En la actualidad se observan debilidades frente a las metodologías de identificación y puesta en práctica de técnicas o mecanismos que orienten sobre la protección frente a ataques por inyección SQL de las bases de datos, ni existe un compendio que abarque las mejores recomendaciones en los esquemas de seguridad en bases de datos. En la red se puede encontrar diversa información respecto a este tipo de ataques por inyección SQL, sin embargo, se requiere de tiempo para buscar y encontrar aquellas características que servirán como mecanismo de protección para una base de datos determinada. Los estudiantes de seguridad informática, especialistas de seguridad informática, especialistas en bases de datos, administradores de bases de datos, auditores informáticos, entre otros, son protagonistas del día a día, cuando se requiera de tiempo, espacio, paciencia y a veces contar con suerte cuando este tipo de ataques por inyección SQL se presentan y se debe esperar por la documentación de los sitios oficiales de los diversos motores de bases de datos o buscar la solución por sus propios medios descubriendo así una labor compleja, que demanda una duración considerable y dando ocasión a los ciberdelincuentes a que actúen y logren su cometido.

1.2. FORMULACIÓN DEL PROBLEMA

¿Cómo el estudio y la búsqueda de técnicas de ataque por inyección SQL en motores de bases de datos SQL y NOSQL permitirá mejorar y plantear mecanismos o procedimientos para su protección?

2. JUSTIFICACIÓN

La tarea desde el área de TI específicamente del equipo de seguridad informática, en una organización, es evitar que los ataques tengan éxito, para ello se debería proteger a la Base de Datos lo mejor posible, aunque algunas veces los esfuerzos parecen ser insuficientes, los atacantes dejaran de atacar si observan que están al frente de una Base de Datos bien protegida. En informática las bases de datos tienden a ser declaradas siempre como un objetivo de ataque, puede suceder tanto en una base de datos local como a una base de datos accedida remotamente a través de una red como Internet.

Contar con un procedimiento documentado con las técnicas o mecanismos de protección a bases de datos relacionales y no relacionales, permitirá reducir esa brecha de oportunidades potenciales que tiene el atacante y la eficacia de los ataques reales ejecutados. Los métodos de protección que se identifiquen dilucidaran que son efectivos y garantizarán que las bases de datos están diseñadas con un esquema de seguridad para soportar y afrontar los peligros existentes en la red y la Internet, impidiendo de una u otra forma ser víctimas de robo, desfalcos monetarios, suplantaciones, comercio de datos, incumplimientos de índole legal, como por ejemplo el cuidado con el tratamiento de datos personales o de publicación de datos confidenciales, entre otros tantos problemas que pueden ser evitados de una manera proactiva aplicando aquellos conocimientos que ofrece la misma tecnología a través de la utilización de diferentes técnicas que analizadas y probadas con anterioridad permitan adoptar metodologías convenientes, adecuadas y efectivas que blinden la arquitectura de seguridad en las bases de datos y que provea a estudiantes de seguridad informática, especialistas de seguridad informática, especialistas en bases de datos administradores de bases de datos, auditores informáticos, usuarios de bases de datos, un documento que oriente de una u otra forma las buenas prácticas y lecciones aprendidas frente a la protección de ataques a bases de datos por Inyección SQL.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Plantear un procedimiento para la protección de ataques por inyección SQL en bases de datos relacionales y no relacionales mediante el estudio y la búsqueda de técnicas aplicadas a partir de un escenario de pruebas.

3.2. OBJETIVOS ESPECÍFICOS

Realizar el levantamiento de información sobre Metodologías de ethical haking y mecanismos de seguridad para las bases de datos relacionales y no relacionales.

Determinar y aplicar las metodologías de ethical hacking en entorno de prueba con sistemas de gestión de bases de datos relacionales y no relacionales.

Presentar propuesta con los mecanismos o técnicas de protección para evitar los ataques a bases de datos relacionales y no relacionales.

4. MARCO REFERENCIAL

4.1. ANTECEDENTES

- En el año 2011 IVAN CAMILO GOMEZ, presenta trabajo de grado para adquirir el título de ingeniero en telecomunicaciones. El proyecto de grado titulado "DISEÑO DE METODOLOGIA PARA VERIFICAR LA SEGURIDAD EN APLICACIONES WEB CONTRA INYECCIONES SQL".

  Este documento presenta una serie de procedimientos para la adecuada verificación de seguridad en páginas web basado en un ciclo PHVA , esta revisión va encaminada a la mitigación de los riesgos en cuanto a ataques por inyección de código SQL

- En el año 2008 Angel prado montes, presenta tesis a la universidad PONTIFICIA COMILLAS Madrid – España, con el fin de alcanzar el título de ingeniero en informática. La tesis se llamó: "PERTRECHAMIENTO DE ATAQUES DE INYECCIÓN SQL CIEGA Y EXTRACCIÓN DE CONTENIDO EN SISTEMAS SQL SERVER MEDIANTE EXFILTRACIÓN DE DATOS VÍA DNS Y ENCAMINAMIENTO SOBRE LA ZONA DESMILITARIZADA".

- En el año 2012 Yury Daniel Zavaleta De la Cruz, presenta trabajo de grado a la universidad Privada del norte en Perú, con el fin de optar al título de ingeniera de sistemas. El proyecto se llamó: "IMPACTO DE ATAQUES SQL INJECTION, EN LOS PORTALES WEB INTERACTIVOS DE LAS EMPRESAS DEL SECTOR TI DE LA CIUDAD DE TRUJILLO"

4.2. MARCO TEORICO

En una entidad siempre prevalecerá la importancia de mantener sus activos seguros, una parte fundamental de los activos de una empresa es la información; datos sensibles que en manos equivocadas pueden acarrear serios problemas con consecuencias no deseadas. Existen dos campos que velan para la puesta en marcha este blindaje deseado: la seguridad informática y la seguridad de la información.

Seguridad Informática: Se define como la implementación de estrategias técnicas para que los recursos de un sistema de información garanticen la confidencialidad, integridad y disponibilidad de la información.

La diferencia entre estos dos campos es su misión específica; la seguridad informática busca que los recursos de un sistema sean capaces de brindar seguridad, teniendo un enfoque tecnológico y la seguridad de la información busca crear e implementar planes, protocolos y procedimientos que brinden seguridad, teniendo un enfoque estratégico.

Si bien, los dos campos son diferentes en sus conceptos, ambos trabajan de la mano para que la información sea siempre confidencial, integra y disponible en una entidad.

Dando alcance a los conceptos presentados anteriormente, se ve la necesidad de desarrollar una metodología que brinde herramientas para la protección del sitio donde se almacenan generalmente los datos: las bases de datos.

Para llevar a cabo el presente proyecto Inicialmente se realiza el levantamiento de información sobre las Metodologías de ethical haking y mecanismos de seguridad para las bases de datos relacionales y no relacionales que se en la actualidad se hayan desarrollado, paso así llevar a cabo un proceso de investigación más eficaz, administrando de forma correcta el tiempo de desarrollo.

Las variables que se tienen en cuenta son los tres pilares fundamentales de la información a saber, la confidencialidad integridad y disponibilidad, entiéndase la confidencialidad como aquella propiedad de la información de ser consultada y modificada con autorización explícita, la integridad como la propiedad de mantenerla exacta y completa desde su origen hasta disposición final, y la disponibilidad como la propiedad de estar disponible valga la redundancia, en el momento y lugar para quien este autorizado en consultarla y/o modificarla, de igual manera la seguridad informática aplica estos tres pilares con un enfoque de tipo tecnológico, como el software, hardware, redes, bases de datos, sistemas operativos, entre otros activos que es donde la información se genera, modifica, trasmite y elimina.

4.2.1. SQL. Sus siglas corresponden a Structured Query Language, es un lenguaje de acceso a bases de datos relacionales basado en la programación declarativa de alto nivel. Este lenguaje se caracteriza por el uso de algebra y calculo relacional como herramienta para la realización de las consultas. La utilización de manejo de registros permite una gran eficacia en su código y la orientación a objetos.

SQL se compone de sentencias, cada sentencia es una instrucción que se da a la base de datos incluyendo las actividades que requerimos y los datos para la base de datos.
Las sentencias se dividen en 3 grupos: DDL, DML y DCL.

DDL: lenguaje de definición de datos, sentencias que permiten la definición y la creación de objetos en las bases de datos
DML: Lenguaje de Manipulación de Datos, sentencias que operan los elementos de la base de datos
DCL: Lenguaje de Control de Datos, sentencias que trabajan la administración y control para la base de datos.

Las instrucciones básicas y las más utilizadas de SQL son las siguientes.

Tabla 1. Resumen de instrucciones básicas SQL

Instrucción	Descripción
CREATE	Creación de algún elemento en la base de datos.
SELECT	Utilizado para la consulta de registros de una base de datos.
INSERT	Permite el cargue de conjunto de datos
UPDATE	Permite la modificación de registros o campos de la base de datos
DELETE	Permite eliminar registros de la base de datos
DROP	Utilizado para la eliminación de tablas
FROM	Con esta instrucción se especifica la tabla a la cual se le seleccionaran los registros
GROUP BY	Se utiliza para separar los registros para ser agrupados
ORDER BY	Utilizado para la organización de los registros en un grupo
AND, OR y NOT	Operadores lógicos que especifican el “y”, “o” y la negación lógica, respectivamente

Fuente: el autor.

4.2.2. MYSQL. Es un sistema de gestión de bases de datos relacional de fuente abierta, multihilo y multiusuario; desarrollado por la compañía MySQL AB y comprada en el año 2010 por la compañía ORACLE corporation. Actualmente se considera como las más popular, teniendo gran importancia en las aplicaciones WEB. Usuado por grandes empresas en el desarrollo de sus aplicaciones como Facebook, google, Youtube, twitter, entre otras.

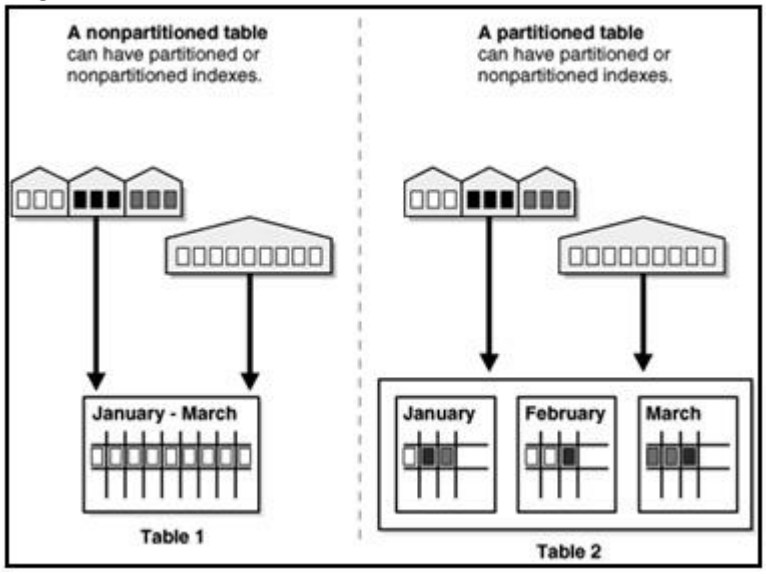
Se caracteriza por la operación de bases datos multiusuario en diferentes lenguajes de programación, igualmente se destacado por la alta velocidad en la búsqueda de datos y registros. Por su licencia de GNU, en MYSQL se puede desarrollar cualquier tipo de aplicación,

4.2.3. Oracle Database. Es un sistema de gestión de bases de datos objeto – relacional de tipo cliente / servidor, creado por Oracle Corporation. En la actualidad es considerado el sistema de bases de datos más robusto y completo, teniendo una gran acogida en el mercado por grandes empresas, permitiendo a las mismas gestionar y controlar grandes cantidades de información.

Oracle database puede ser ejecutado por múltiples plataformas, además desde pequeñas y grandes maquinas.

Su última versión Oracle Database 12c, cuenta con una nueva característica: Oracle Partitioning, que mejora sustancialmente en las áreas de rendimiento, administración de grandes cantidades de información y disminución de costos en almacenamiento. En la figura 1 se observa el particionado de tablas que se maneja en la versión de Oracle Database 12c.

Figura 1. Particionado de tablas en Oracle Database 12c



Fuente:http://www.oracle.com/technetwork/es/images/new-partitioning-1-2244547.jpg

4.2.4. Trigger. Los trigger o disparadores son objetos almacenados en las bases de datos, como su nombre lo indica, este objeto se dispara o se activa una vez se ejecuta una instrucción que modifique la tabla a la cual está definido. Las instrucciones por las cuales el trigger se ejecuta son las de INSERT, DELETE y UPDATE, siendo de gran importancia para garantizar la integridad de la información y su coherencia.

La sintaxis básica para la creación de un Trigger inicia con la instrucción “créate trigger” acompañado del nombre que se le asigne, seguido de la instrucción “ON”

acompañado del nombre de la tabla a la que se va a asociar el trigger, acontinuacion definimos los eventos para los cuales el trigger se activara: “for EVENTO – insert, delete o update”.

Existen dos tipos de trigger, los Row Triggers que se ejecutan una vez de llamen desde la tabla asociada y los Statement Trigger que sin importar las veces que se cumpla la condición este se ejecutara una sola vez.

4.2.5. PL/SQL. Es un lenguaje de procesamiento procedimental que ofrece un grupo de instrucciones de la programación estructurada, diseñado para la inclusión de sentencias SQL dentro su sintaxis, siendo un complemento y un medio por el cual se aumenta los alcances de SQL.

PL SQL se programa en bloques que se utilizan como funciones u operaciones, los bloques son guardados como objetos dentro de la base de datos

4.2.6. Inyección SQL. Es una de las técnicas más reconocidas en el área de la seguridad informática. Esta técnica se basa en la inyección de código para la modificación de una cadena de consulta de una base de datos, generalmente se realizan en los formularios donde busca usurpar las consultas que ya están definidas. Esta técnica tiene unos fines muy precisos como lo son: la modificación de información alojada en las bases de datos, la exposición de información guardada de manera oculta y/o la ejecución de comandos maliciosos; esta ultima una de las más peligrosas. El ataque de inyección SQL se presenta más frecuentemente en aplicaciones PHP y ASP.
Los comandos utilizados para el desarrollo de este tipo de ataques nos mas que la utilización de sintaxis SQL.

4.2.7. Ethical Hacking. Es el proceso por el cual, se utilizan las mismas técnicas y herramientas de un atacante malicioso para atacar a una organización de una manera controlada; esta filosofía resulta de la práctica probada: "para atrapar a un ladrón debes pensar como un ladrón".[1]

Las técnicas de ethical hacking se basan generalmente en pruebas de penetración controladas conocidas como pentest. Dichas pruebas se dividen en las siguientes:

Pentest con objetivo: se buscan vulnerabilidades en elementos seleccionados o priorizados por el usuario

[1] http://www.cibertec.edu.pe/pdf/division-de-alta-tecnologia/plataforma-tecnologica/ethical-hacking.pdf

Pentest sin objetivo: se buscan vulnerabilidades en todos los elementos del sistema o los sistemas de propiedad de la empresa usuaria

Pentest ciega: la prueba de penetración ciega, se basa en la simulación de un ataque real a la información pública que posee la empresa. Estas pruebas pueden ser con la información que da la empresa, con información publicada o dentro de la empresa para probar la eficacia de los controles existentes.
Una de la forma clásica actualmente de protección ante intrusión no autorizada es el uso de captcha, el cual busca identificar que quien este ingresando la información sea una persona y no una máquina, impidiendo que se pueda ejecutar automáticamente scripts.

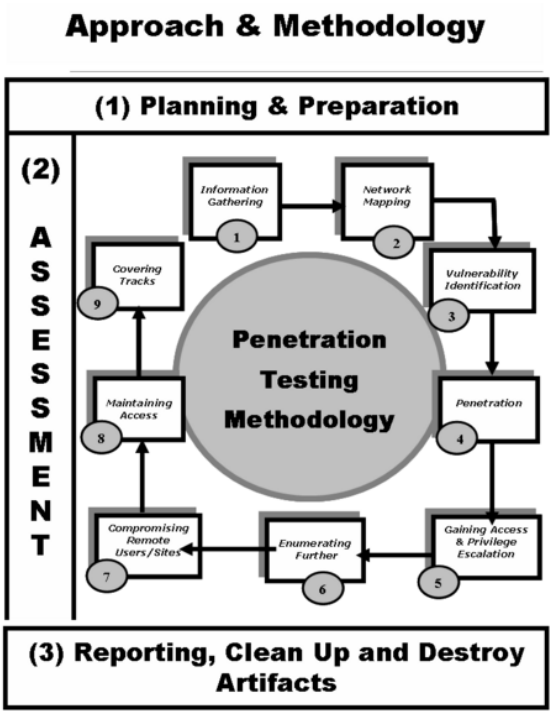
Otra forma de evitar el uso de la inyección SQL es mediante la limpieza de las variables antes de cada consulta, no permitiendo el uso de caracteres especiales (propio de este ataque). Haciendo esta limpieza de las variables, no se autorizará la ejecución o inclusión de otras consultas continuas a la inicial.

Igualmente, el uso de URL amigables o semánticas hará más complicado para el atacante la detección de nuestras variables y así impedirá que incluyan instrucciones para las mismas

4.2.8. Metodologías de ethical hacking. Al paso de los años las organizaciones han crecido y han aumentado su inversión en tecnología, con el auge de los computadores y el internet cada día se automatiza más y buscan mejorar sus procesos y sus sistemas de información. Sin embargo, al avanzar en este campo también se debe tener en cuenta que las vulnerabilidades están presentes en los diferentes sistemas: llámese, redes, bases de datos, aplicaciones, sistemas operativos entre otros y que a partir de ese crecimiento surgen las amenazas sobre la integridad, confidencialidad e integridad de los datos, una de ellas a través de la denominada intrusión a los sistemas informáticos de la organización. Al existir este tipo de amenazas surge la necesidad de realizar pruebas de accesos no autorizados de investigar y practicar métodos de intrusión que permita observar el comportamiento en la organización, es así que nace el ethical hacking, que en resumen es un servicio prestado por un profesional o grupo de profesionales enfocado hacia la seguridad informática o de la información mediante escaneos de vulnerabilidades y testeos de penetración. Existe sin embargo hoy en día una clasificación de estos profesionales conocidos también como hackers, que abarcan desde los inofensivos o buenos como hacker blanco, hasta los más peligrosos hacker negro por solo mencionar estos dos y que dependiendo de los principios y características en su modus operandi y muy significativamente por su ética al desarrollar el trabajo, se deriva el nombre de hacking ético pues el profesional en seguridad informática o de la información no debe comprometer ninguno de los activos de la organización.

4.2.9. ISSAF Information System Security Assessment Framework. Es un marco de trabajo desarrollado para la evaluación de Seguridad de Sistemas de Información, una metodología estructurada de análisis de seguridad en varios dominios y detalles específicos de test o pruebas conocidos como Criterios de Evaluación. Su objetivo es proporcionar procedimientos muy detallados para el testing de sistemas de información que reflejan situaciones reales con el fin de encontrar vulnerabilidades[2]. En la figura 2 se muestra el modelo general de las fases de planificación y preparación de un test de penetración.

Figura 2. Fases de test de penetración



Fuente: http://2.bp.blogspot.com/_NcZQ3njhqn8/SeSxbyfodYI/AAAAAAAAADc/nOW3hjQNfTE/s1600-h/453px-Image001.png

El enfoque incluye tres fases: Planificación y Preparación, Evaluación Reportes, Limpieza y Destrucción de Objeto. Los criterios de evaluación se componen de los siguientes elementos:

- Una descripción del criterio de evaluación
- Puntos y Objetivos a cubrir
- Los pre-requisitos para conducir la evaluación

[2]http://datateca.unad.edu.co/contenidos/233016/EXE_SAM/leccin_29_issaf.html

- El proceso mismo de evaluación
- El informe de los resultados esperados
- Las contramedidas y recomendaciones
- Referencias y Documentación Externa.

En la primera fase se planifica y prepara la prueba, se debe tener un acuerdo firmado entre las partes interesadas. De igual manera se especifica las personas que harán parte del equipo de trabajo, fechas, tiempos estimados de la prueba, definiendo los respectivos roles y responsabilidades.

En la fase de evaluación se lleva acabo el test de penetración y se desarrolla como se muestra en la imagen efectuando los siguientes pasos:

- Recolección de Información
- Mapeo de la red de trabajo
- Identificación de vulnerabilidades
- Penetración
- Obtener Acceso y escalada de privilegios
- Enumeración
- Comprometer usuarios remotos y sitios
- Mantener Acceso

En la última fase se presentan los reportes de los resultados en el trascurso de las pruebas de penetración, si llegase a detectarse un factor crítico, deberá ser informado para asegurar que la organización es comunicada y proceder analizar las posibilidades de corregir el problema identificado.

Después de finalizar la ejecución de los casos de prueba, se elabora un informe escrito con la descripción de los resultados detallados de las pruebas y las recomendaciones para la mejora. Algunas cosas que se incluyen en el informe son: Resumen de Gestión, Alcance del proyecto, Herramientas utilizadas (Incluyendo Exploits), Fechas y horas reales en las que se ejecutaron las pruebas en el sistema y la información de salida de las pruebas realizadas.

En esta metodología la información que se genera y/o se almacena en los sistemas de prueba deben ser eliminado. Cuando por algún motivo no es posible remover dicha información, deben relacionarse en el informe técnico con su respectiva

ubicación para que la organización pueda eliminarlos después de la formalización en la entrega del informe.

4.2.10. OSSTMM Open Source Security Testing Methodology Manual. Manual de la Metodología Abierta del Testeo de Seguridad, para pruebas y análisis de seguridad realizado siguiendo la metodología OML (Open Methodology License) Es uno de los más completos y comúnmente utilizados, la metodología OSSTMM se centra en los detalles técnicos de los elementos que necesitan ser comprobados, qué hacer antes, durante y después de las pruebas de seguridad, así como evaluar los resultados obtenidos[3].

La metodología se encuentra dividida en varias secciones, a su vez en una serie de módulos de testeo específicos, a través de los cuales se observan cada una de las dimensiones de seguridad, integradas con las tareas a llevar a cabo en los diferentes puntos de revisión:

Sección A - Seguridad de la Información
Sección B – Seguridad de los Procesos
Sección C – Seguridad en las tecnologías de Internet
Sección D – Seguridad en las Comunicaciones
Sección E – Seguridad Inalámbrica
Sección F – Seguridad Física

Actualmente el manual se encuentra en su tercera versión, no solo abarca ámbitos técnicos y de operación de seguridad, además regula aspectos tales como: las credenciales del profesional responsable de la prueba, la manera en que la prueba debe ser comercializada, la forma de mostrar los resultados de la prueba, las normas éticas y legales al formalizar la prueba, los intervalos de tiempo para desarrollar la prueba y un componente muy importante el análisis y evaluación de riesgos.

La metodología está compuesta por cuatro fases y estás a su vez por diecisiete módulos, estos módulos contienen tareas y procedimientos, los cuales varían dependiendo de que se está evaluando.

A. Fase de inducción. Determina la especificación de las pruebas, el alcance y sus restricciones. En esta parte se recopila información relacionada con las leyes,

[3]http://datateca.unad.edu.co/contenidos/233016/EXE_SAM/leccin_27_osstmm.html

ética, políticas, regulaciones industriales y la cultura que influyen en los requisitos de seguridad y privacidad de la organización a la que se aplicará las pruebas.

B. Fase de Interacción. Se define el ámbito de aplicación, son determinados los objetivos que serán probados. Se establecen los parámetros de Visibilidad, Acceso, Confianza y la verificación de los controles frente al no repudio, confidencialidad, privacidad e integridad.

C. Fase de Investigación. Busca identificar en qué medida se cumplen las políticas y reglas de seguridad con la realidad de las operaciones.

D. Fase de Intervención. Son verificados los controles de autenticación, indemnización y sometimiento. También se realizan pruebas para la medición de la consistencia de los controles para mantener la continuidad de las operaciones, así como su capacidad de recuperación. La parte final de toda la metodología consiste en la revisión de los logs, para la búsqueda de la evidencia de las pruebas, lo cual representa verificar el control de alarma.

4.2.11. OWASP Open Web Application Security Project. Proyecto abierto de seguridad de aplicaciones web, este marco de trabajo tiene por objetivo, resaltar áreas con carencias, ayudar a las organizaciones a comprobar sus aplicaciones web con el propósito de construir software fiable y seguro, aplicando las guías y listados de comprobación del OWASP [3] https://www.owasp.org/images/8/80/Gu%C3%ADa_de_pruebas_de_OWASP_ver_3.0.pdf

La metodología está compuesta por 2 partes, la primera contiene los siguientes puntos:

- Principios del testeo
- Explicación de las técnicas de testeo.
- Explicación general acerca del framework de testeo de OWASP.

En la segunda parte, se planifican las técnicas para realizar pruebas durante el ciclo de vida del desarrollo de software. Incorpora en su metodología de testeo (https://www.owasp.org/index.php/Main_Page), aspectos claves relacionados con el Ciclo de Vida del Desarrollo de Software o SDCL con el propósito de que se efectúe las pruebas antes de que la aplicación este en producción. Según el proyecto de la organización OWASP, el SDLC es un proceso bien conocido por los desarrolladores. Mediante la integración de la seguridad en cada fase del SDLC, permite un enfoque integral a la seguridad de aplicaciones que se apoya en los

procedimientos ya existentes en la organización. Se debe tener en cuenta que mientras los nombres de las fases pueden cambiar dependiendo del modelo SDLC usado por una organización, cada fase conceptual del arquetipo SDLC será usada para desarrollar la aplicación (es decir, definir, diseñar, desarrollar, implementar, mantener). Cada fase tiene implicaciones de seguridad que deberán formar parte del proceso existente, para asegurar un programa de seguridad rentable y exhaustivo.

Según lo anterior, se puede referenciar como un programa efectivo de testeo de aplicaciones web, donde se incluye como elementos a testear: Personas, Procesos y Tecnologías, definiendo conceptos claves a la vez que introduce un marco de trabajo específicamente diseñado para evaluar la seguridad de aplicaciones web a lo largo de su vida.

Paso 1 Antes de iniciar el desarrollo

- Revisión de Políticas y Estándares
- Desarrollo de un Criterio de Medidas y Métricas (Aseguramiento de la
- Trazabilidad)

Paso 2 Durante la definición y el diseño

- Revisión de los Requerimientos de Seguridad
- Revisión del diseño y/o Arquitectura
- Creación y Revisión de modelos UML
- Creación y Revisión de modelos y Amenazas

Paso 3 Durante el desarrollo

- Revisión de Código

Paso 4 Durante el desarrollo

- Testeo de Penetración sobre la Aplicación
- Testeo sobre la Administración y Configuración

Paso 5 Operación y mantenimiento

- Revisión Operacional
- Chequeos Periódicos
- Control de Cambios

4.2.12. Owas zap. Herramienta que permite analizar sitios web con el fin de buscar sus vulnerabilidades con el objetivo de aprender técnicas de penetración o mejorar la seguridad del sitio. Tiene en cuenta la lista de vulnerabilidades web de OWASP, por lo que incluye una gran cantidad de herramientas que pueden detectar casi cualquier vulnerabilidad que pueda existir en un sitio web.

Las principales características de OWASP ZAP son:

- Herramienta totalmente gratuita y de código abierto.
- Posibilidad de asignar un sistema de prioridades.
- Traducida a más de 12 idiomas, entre ellos, el español.
- Posibilidad de comprobar todas las peticiones y respuestas entre cliente y servidor.
- Posibilidad de localizar recursos en un servidor.
- Análisis automáticos.
- Análisis pasivos.
- Posibilidad de lanzar varios ataques a la vez.
- Capacidad para utilizar certificados SSL dinámicos.
- Soporte para utilizar tarjetas inteligentes (DNI-e, por ejemplo) y certificados personales.
- Análisis de sistemas de autenticación.
- Posibilidad de actualizar la herramienta automáticamente.

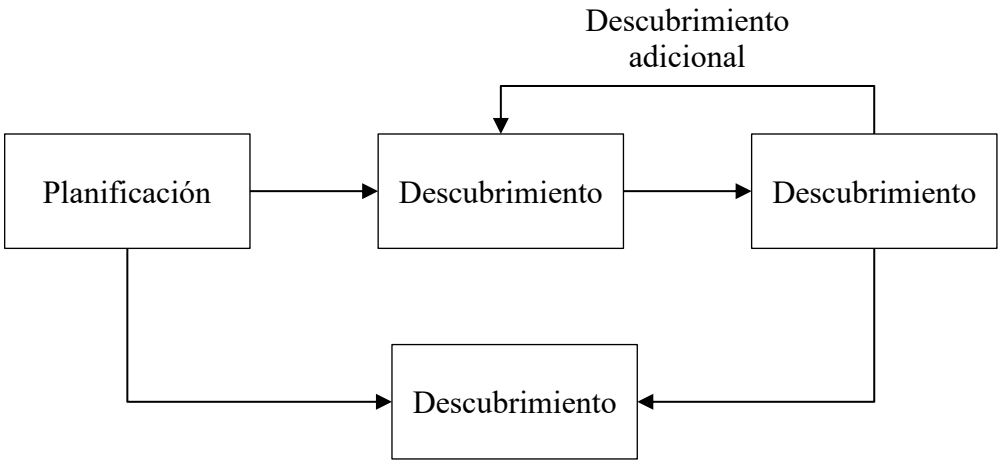
Dispone de una tienda de extensiones (plugins) con las que añadir más funcionalidades a la herramienta ([Velasco, Rubén. (2015) OWASP ZAP, herramienta para auditar la seguridad de una página web. Disponible en: https://www.redeszone.net/2015/04/25/seguridad-web-owasp-zap/).

4.2.13. NIST SP800-115 Technical Guide to Inf. Security Testing and Assessment. El propósito de la guía técnica de pruebas de seguridad de la información y evaluación del Instituto Nacional de Normas y Tecnología de Estados Unidos es proporcionar directrices para las organizaciones sobre la planificación y realización de pruebas técnicas y evaluaciones de la seguridad de la información, el análisis de los resultados y el desarrollo de estrategias de mitigación. Proporciona recomendaciones prácticas para diseñar, implementar y mantener información técnica relacionada con los procesos y procedimientos de pruebas y evaluación de seguridad, los cuales pueden ser utilizados para varios propósitos tales como encontrar vulnerabilidades en un sistema o red y verificar el cumplimiento de una política u otros requisitos. La guía no pretende presentar un programa exhaustivo de pruebas o evaluación de la seguridad de la información, sino más bien una visión general de los elementos clave de las pruebas y evaluaciones técnicas de seguridad con énfasis en técnicas específicas, sus beneficios y limitaciones y recomendaciones para su uso

(http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf)

El NIST 800-115 divide una prueba de penetración en 4 fases: planificación, descubrimiento, ataque y reporte.

Figura 3. Fases de la Metodología de la Prueba de Penetración del NIST SP 800115



Fuente: NIST SP800-115

4.2.13.1. Fase de Planificación. Consiste en determinar los objetivos y las reglas de la prueba de penetración, para enfocar el resultado lo más provechoso para la organización.

Las actividades en la elaboración de un plan para la gestión y administración del proyecto son:

- Especificar las necesidades y requerimientos que deben cumplir la evaluación de la seguridad en la organización.
- Delimitar el alcance de las pruebas. Determinar los límites de la prueba en términos de acciones y de los resultados esperados.
- Identificar los riesgos asociados a la realización de la prueba de penetración, así como las técnicas y medidas de mitigación que serán utilizados.
- Decidir el personal que hará parte del equipo en la prueba de penetración, así como los responsables de los activos que serán probados. Se puede incluir los datos de contacto de las personas si es necesario.

- Programar las pruebas incluyendo las pruebas críticas y sus etapas.

- Identificar los lugares internos o externos autorizados para realizar las pruebas de penetración.

- Autorizar los niveles de acceso tanto de usuario como administrador a los sistemas y/o red.

- Indicar la dirección IP desde donde se harán las pruebas de forma remota si es necesario.

- Especificar el hardware y software que se utilizará para efectuar las pruebas.

- Establecer el procedimiento y la periodicidad para informar sobre los avances y eventos ocurridos durante el desarrollo de la prueba de penetración.

- Documentar las acciones a realizar en situaciones donde alguna de las pruebas tenga un resultado negativo en la red o que un ataque se presente durante la ejecución de la prueba.

- Formalizar los sistemas y/o redes que se probarán y se excluirán durante las pruebas.

- Describir en detalle las actividades permitidas y no permitidas, incluyendo la metodología de la prueba de penetración a utilizar.

- Si la prueba incluye actividades de ingeniería social, es necesario que sean conocidas y aprobadas por el grupo directivo antes de ser aplicadas.

- Disponer el modo para la recolección, almacenamiento, transmisión y destrucción de los datos generados como resultado de las pruebas.

- Especificar los requisitos de la presentación y el informe de los resultados en la ejecución y finalización de las pruebas.

Los pormenores de las pruebas, así como cada una de las actividades a desarrollar en la prueba deberán documentarse formalmente con el fin de garantizar la legalidad y beneficio mutuo entre las partes interesadas.

4.2.13.2. Fase de descubrimiento. Esta fase se divide en dos partes: La primera parte consiste en la recopilación de información y la realización de escaneos; la segunda parte consiste en realizar un análisis de vulnerabilidades.

A. Recopilación de información
Es aconsejable conseguir información asociada con el nombre del host, la dirección IP puede conseguirse con algunos métodos como por ejemplo preguntas al DNS, realizar búsquedas en InterNIC (WHOIS) y haciendo sniffing a la red.

A través de la búsqueda en la página web de la organización o servidores de directorio activo se puede obtener los nombres de los empleados o información para contactarlos.

En algunas ocasiones, se puede utilizar técnicas como dumpster diving (búsqueda de información en la basura de la organización, zonas de reciclaje o reutilización de papel) y/o con un simple recorrido por las áreas de trabajo de los usuarios, con el fin de detectar datos relevantes incluso se puede encontrar contraseñas escritas en un papel.

Para la recolección de información se pueden tener en cuenta las siguientes actividades:

- Descubrimiento en la red: la intención es ver los equipos activos en la red. Existen dos clases de escaneos: el escaneo pasivo que se basa en utilizar un sniffer de red, para realizar el monitoreo de los equipos. Es aplicado de manera interna en la organización y requiere de más tiempo para su ejecución. El escaneo activo radica en enviar al blanco varios tipos de paquetes y dependiendo de la respuesta del equipo al tráfico generado, es posible determinar el tipo de sistema operativo. Este escaneo requiere de menos tiempo para obtener datos, pero es más susceptible a ser detectado.
- Identificación de puertos y servicios de red: se utiliza un analizador de puertos, con el fin de detectar los puertos abiertos de un equipo, así es posible identificar el tipo de sistema operativo, técnica llamada OS fingerprint. La versión de la aplicación se puede obtener utilizando banner grabbing, capturando la información transmitida por el banner cuando se inicia una conexión.

De acuerdo a lo anterior el objetivo es tener un listado de los equipos activos en la red, así como los puertos que tiene abiertos.

B. Análisis de Vulnerabilidades

Se fundamenta en la comparación de los servicios, aplicaciones y sistemas operativos de los hosts escaneados enfrentándolos con una base de datos de vulnerabilidades.

Apoya a la identificación de versiones obsoletas de software, parches no instalados, configuraciones erradas y valida el cumplimiento de las políticas de seguridad de la organización.

Con el conocimiento de las vulnerabilidades de los sistemas de la organización, es posible concretar la estrategia para realizar la prueba de penetración a los equipos de la red.

4.2.13.3. Fase de Ataque. Una vez se tienen identificadas las vulnerabilidades, el pentester o equipo de pruebas trata de comprender más sobre la estructura y comportamiento de la red con el fin de explotar las vulnerabilidades establecidas. Algunos exploits admiten elevar privilegios en el sistema o en la red para acceder recursos adicionales. Es recomendable moverse por el sistema en búsqueda de más información, esto permitirá efectuar un descubrimiento adicional en busca de más vulnerabilidades.

Si es posible explotar una vulnerabilidad, se puede utilizar más herramientas en el sistema para obtener información y/o accesos adicionales, con el fin de observar hasta donde un atacante puede acceder. Después de tener un acceso es importante cubrir las huellas utilizando rootkits y eliminando logs que hayan registrado la actividad realizada.

Se puede categorizar las vulnerabilidades explotadas de la siguiente manera:

- Errores de configuración.
- Errores del kernel.
- Código expuesto a ataques de Buffer Overflows.
- Debilidades en la validación de posibles entradas en una aplicación.
- Ataques de descriptor de archivos.
- Privilegios accedidos. Es posible aprovechar los privilegios superiores durante el tiempo que un programa o proceso está todavía en modo de ejecución.
- Inadecuada asignación de permisos a archivos y directorios.

Cuando se finaliza la fase de ataque, es primordial eliminar los archivos creados y desinstalar las herramientas utilizadas durante las pruebas, las vulnerabilidades encontradas serán corregidas por parte de la organización a partir del reporte generado.

4.2.13.4. Fase de Reporte. En esta fase se hace una descripción de las vulnerabilidades encontradas, se demuestra el procedimiento utilizado para su explotación y se entrega una guía de cómo mitigar las debilidades encontradas.

Es necesario precisar que los informes se deben diferenciar de acuerdo a las personas a quienes va dirigido, se puede hacer un informe de tipo técnico donde se incluya los resultados a nivel de detalle de las pruebas realizadas y otro de tipo gerencial donde se muestran los hallazgos principales y las recomendaciones a realizar por parte de la organización.

El NIST recomienda utilizar para las pruebas como herramienta BackTrack que para hoy en día fue reemplazada por Kali Linux, adicionalmente hace mención como buena alternativa la metodología de OSSTMM.

4.2.14. Aplicar las metodologías de ethical hacking.

Según las metodologías identificadas es preciso mencionar que existen tres mecanismos de evaluación a saber:

- La auditoría de seguridad informática

Es un proceso formal orientado a medir aspectos de alto nivel de la infraestructura de seguridad desde un enfoque organizacional. Por lo general, los auditores delimitan el alcance y no incluyen detalles técnicos de bajo nivel como por ejemplo fallas en protocolos de comunicación. Sin embargo, pueden existir auditorías con un mayor nivel de profundidad y alcance, con datos y recomendaciones técnicas detalladas. El proceso de auditoría generalmente se lleva a cabo por auditores EDP (Electronic Data Processing) certificados bajo las directrices de un comité, quién reporta a las directivas de la organización o a un ente externo mandado por la ley (KAPLAN, R. (1997) Penetration Testing: reward or ruin?. Computer Security Journal. Vol. XIII, No.1. pág.69-89).

- La evaluación de seguridad: Orientada a determinar mayores detalles técnicos de la seguridad de la infraestructura de una organización. Son menos formales y tienden a ofrecer mayor detalle y alcance que las auditorías. Este proceso

generalmente es llevado a cabo por un equipo de expertos en tecnología con la asistencia de un comité ad-hoc conformado por personal de la organización, más que por auditores certificados. La evaluación es considerada un arte, en comparación con el enfoque estructurado que desarrollan los auditores. Al igual que las auditorías, las evaluaciones siempre están limitadas en su alcance. Sin embargo, tienden a ser lo suficientemente flexibles para abarcar detalles de bajo nivel (KAPLAN, R. (1997) Penetration Testing: reward or ruin?. Computer Security Journal. Vol. XIII, No.1. pág.69-89).

- Las pruebas de penetración (PEN TEST): Son las pruebas de seguridad en el que las personas que los realizan imitan los ataques del mundo real en un intento por identificar los métodos para eludir las medidas de seguridad de una aplicación, sistema o red. Las pruebas de intrusión a menudo implican la realización de ataques a sistemas y datos, usando las mismas herramientas y técnicas utilizadas por los atacantes (Scarfone, 2008).

  Son pruebas de seguridad en la que especialistas explotan las vulnerabilidades de un sistema informático de forma controlada para evaluar los controles de seguridad de dicho sistema. (EC-Council,2010).

  Dependiendo de las necesidades de una organización es posible que se oriente en uno o varios tipos de evaluación, sin embargo, las tres prácticas tienen como objetivo primordial la seguridad informática y cada uno varia en alcance y profundidad, en conclusión, se orientan a necesidades e inquietudes diferentes pero complementarias.

Figura 4. Perspectivas de los métodos de evaluación de seguridad informática

Fuente: http://www.derechotecnologico.com/estrado/estrado003.html#4

4.3. MARCO LEGAL

Todo lo concerniente al manejo de información está reglamentado por un conjunto de leyes y normas que buscan que los todos los procedimientos que se planteen y desarrollen dentro de las organizaciones se llevan a cabo dentro de un marco de confidencialidad, integridad y disponibilidad de la información. A continuación, se citan algunos componentes legales.

4.3.1. LEY 1273 DE 2009. Norma por medio de la cual se modifica el código penal agregando a este la reglamentación para la protección de la información y los datos.

Se identifican los posibles delitos a los cuales los colombianos o extranjeros pueden incurrir en la manipulación indebida de la información.

Los artículos de esta ley que tienen relación con el desarrollo del proyecto son los siguientes.

Artículo 269A: Acceso abusivo a un sistema informático. El que, sin autorización o por fuera de lo acordado, acceda en todo o en parte a un sistema informático protegido o no con una medida de seguridad, o se mantenga dentro del mismo en contra de la voluntad de quien tenga el legítimo derecho a excluirlo, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1.000 salarios mínimos legales mensuales vigentes.
Artículo 269C: Interceptación de datos informáticos. El que, sin orden judicial previa intercepte datos informáticos en su origen, destino o en el interior de un sistema informático, o las emisiones electromagnéticas provenientes de un sistema informático que los transporte incurrirá en pena de prisión de treinta y seis (36) a setenta y dos (72) meses.

Artículo 269F: Violación de datos personales. El que, sin estar facultado para ello, con provecho propio o de un tercero, obtenga, compile, sustraiga, ofrezca, venda, intercambie, envíe, compre, intercepte, divulgue, modifique o emplee códigos personales, datos personales contenidos en ficheros, archivos, bases de datos o medios semejantes, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 salarios mínimos legales mensuales vigentes.

Artículo 269G: Suplantación de sitios web para capturar datos personales. El que con objeto ilícito y sin estar facultado para ello, diseñe, desarrolle, trafique, venda, ejecute, programe o envíe páginas electrónicas, enlaces o ventanas emergentes, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1.000 salarios mínimos legales mensuales vigentes, siempre que la conducta no constituya delito sancionado con pena más grave.

En la misma sanción incurrirá el que modifique el sistema de resolución de nombres de dominio, de tal manera que haga entrar al usuario a una IP diferente en la creencia de que acceda a su banco o a otro sitio personal o de confianza, siempre que la conducta no constituya delito sancionado con pena más grave.

5. MARCO METODOLOGICO

5.1. METODOLOGÍA DE INVESTIGACIÓN

De acuerdo a los tipos de investigación existentes y teniendo en cuenta que las características de los medios utilizados para obtener los datos del presente proyecto se realizan a través de fuentes de información electrónica la investigación tiene un enfoque de tipo descriptivo y cuantitativo.

5.2. METODOLOGÍA DE DESARROLLO

Las fases a desarrollar según la investigación se relacionan a continuación:

Investigadora: examina elementos de una base de conocimientos, compara aspectos del conocimiento con otros ya conocidos, estableciendo relaciones entre ambos.

Sistematización: análisis reflexivo, crítica de los elementos del conocimiento para comprobar su validez.

Expositiva: exactitud y orden del conocimiento adquirido, generación de ideas relacionadas sacando utilidad con los productos de fuentes documentales y la experiencia.

La investigación tiene como objetivo el obtener información para conocer, recolectar ideas o sugerencias, afinar una metodología para complementar temas de estudio, sin arribar a conclusiones definitivas o generales. Teniendo en cuenta estas tres fases se desarrollarán las etapas que a continuación se describen:

1) Elección y Delimitación del Tema:
El campo de aplicación es en el en el área de la Informática, en Seguridad Informática específicamente en lo relacionado con los ataques a bases de datos mediante inyección SQL en gestores conocidos como lo son MySql, Oracle, y gestores NoSql como MongoDB y CouchDB.

2) Definición del Tema y determinación de Subtemas
Se dividirá en sus posibles componentes, los factores que serán más importantes de conocer serán considerados como un subtema. Se buscará la definición de cada subtema relacionado con los motores de bases de datos MySql, Oracle, MongoDB y CouchDB, la explotación de vulnerabilidades y los posibles mecanismos de protección.

3) Selección de la Información
La selección o levantamiento de la información debe realizarse de acuerdo con los subtemas objetos de estudio. Las fuentes de información más comunes son:

- Material Bibliográfico. materiales impresos como libros, publicaciones de revistas científicas, tesis y artículos que provean información sobre el tema o algún subtema que se pretende estudiar.
- Material Hemerográfico. Son las publicaciones periódicas que salen en un determinado lapso de tiempo, sea diario, quincenal, mensual, trimestral o semestral que contengan alguna información sobre el tema o algunos de los subtemas a investigar.
- Material Videográfico. Se contemplan los videos que pueden aportar datos al tema que se pretende investigar.
- Material Iconográfico. Es aquel que se expone en imágenes y pinturas que proporcionen información sobre el objeto en estudio.
- Material de Internet. Es la información que se recupera de algún buscador o banco de datos sobre el objeto de estudio.

4) Organización de la Información. En ocasiones no es necesario leer toda la obra o material encontrado, únicamente lo que haga referencia a cada uno de los subtemas o aquellos aspectos que proporcionen la información más relevante. Se elaborará un resumen de los materiales a través de fichas bibliográficas donde se describirá el título de la obra o capítulo, el contenido del material estudiado y la fuente de donde se obtuvo la información.

5) Entorno de Prueba. Paralelo a la organización de la información se adecuará un entorno de prueba con el fin de realizar las diferentes pruebas sobre la inyección SQL y NoSQL a las bases de datos objeto de estudio.

5.3. TÉCNICAS DE RECOLECCIÓN DE DATOS

Se aplica la técnica de análisis documental, entendido este como un conjunto de operaciones encaminadas a representar un documento y su contenido bajo una forma diferente de su forma original, con la finalidad posibilitar su recuperación posterior e identificarlo. La finalidad última del análisis documental es la transformación de los documentos originales en otros secundarios, instrumentos de trabajo, identificativos de los primeros y gracias a los cuales se hace posible tanto la recuperación de éstos como su difusión. [4]
Otro factor que hace parte de estas técnicas, es el análisis documental de recursos en Internet, que permite dentro de la extensa gama de páginas web y de información existente poder encontrar y explorar los contenidos que fortalecerán el proceso de

[4] http://www.uv.es/macas/T5.pdf

investigación y el estudio de las metodologías existentes en la realización del diagnóstico de seguridad para las bases de datos.

En la norma UNE 50-113 Información y Documentación – Vocabulario – Parte 3ª Adquisición, identificación y análisis de documentos y datos, en el punto 3.3 incluye conceptos referidos al Análisis Documental. De entre estos los siguientes términos son algunos de los más empleados.

Análisis documental / Analysis / analyse : Operación que consiste en examinar un documento para encontrar sus elementos esenciales y las relaciones entre ellos.

Análisis de contenido / content analsys / analyse du contenu : Análisis que tiene por objeto facilitar los datos que caracterizan el contenido del documento de una forma clara y concisa.

Indización / indexing / indexation : Operación destinada a representar los resultados del análisis del contenido de un documento o de una parte del mismo, mediante elementos de un lenguaje documental o natural, generalmente para facilitar la recuperación.

Resumen / abstract / résumé : Representación breve del contenido del documento, sin interpretación ni crítica.

5.4. TÉCNICAS DE PROCESAMIENTO DE DATOS

Con respecto a esta técnica se ordenaron y clasificaron los datos de acuerdo al análisis de contenido como parte de la hermenéutica bajo un enfoque descriptivo y cuantitativo. Con el fin de comprender e integrar los elementos estudiados se acudió a la hermenéutica que en griego quiere decir interpretación.

5.5. POBLACIÓN Y MUESTRA

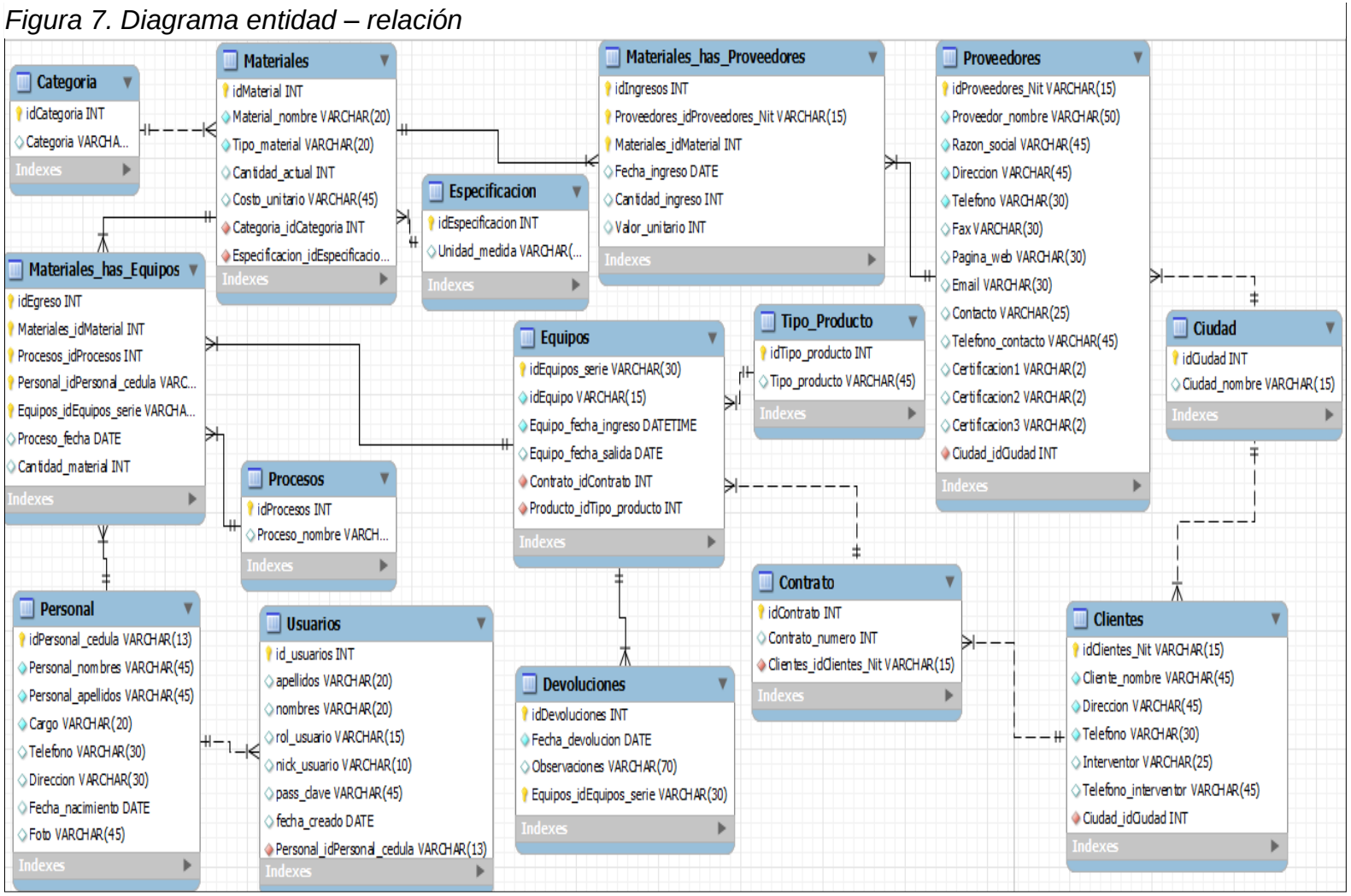
El presente proyecto tiene como población a cuatro motores de bases de datos que incluyen características de SQL y NOSQL. La muestra entonces como es una población finita y accesible se toma para la investigación la totalidad de la población para el desarrollo de las temáticas de estudio.

6. DESARROLLO DEL PROYECTO

6.1 ENTORNO DE PRUEBA EN MYSQL.

En este primer entorno de prueba se utilizó sistema operativo Linux y el gestor de bases de datos MySQL, también se utilizó MySQL Workbench que es una herramienta que permite modelar diagramas de entidad-relación para bases de datos MySQL, puede usarse para diseñar el esquema de una base de datos nueva o documentar una que ya existe. En la figura 7 se presenta el modelo entidad relación utilizado.

Figura 7. Diagrama entidad – relación



Fuente: el autor.

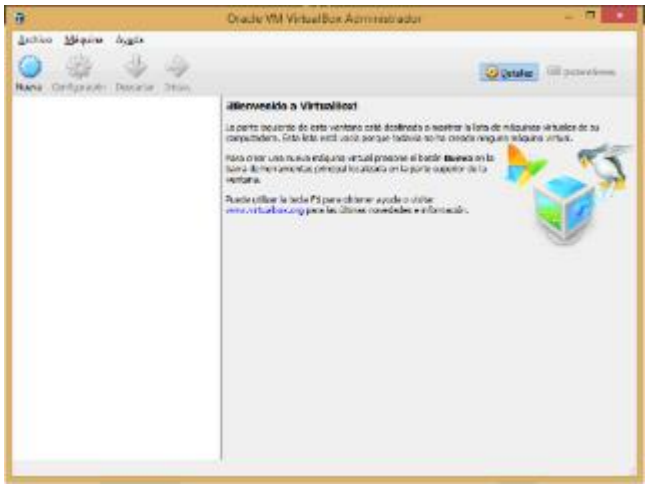
Para este este entorno de prueba se utilizó virtual box, el cual se descargó desde la página https://www.virtualbox.org/wiki/Downloads como lo muestra la Fig. 8. página de descarga virtual box.

Figura 8. Página de descarga virtual box

Fuente: https://www.virtualbox.org/wiki/Downloads

Se realizó la instalación y se ejecutó el programa como lo indica la Fig. 9. Pantalla inicial de Virtual Box se muestra la aplicación de virtualización instalada.

Figura 9. Pantalla inicial de Virtual Box



Fuente: El autor.

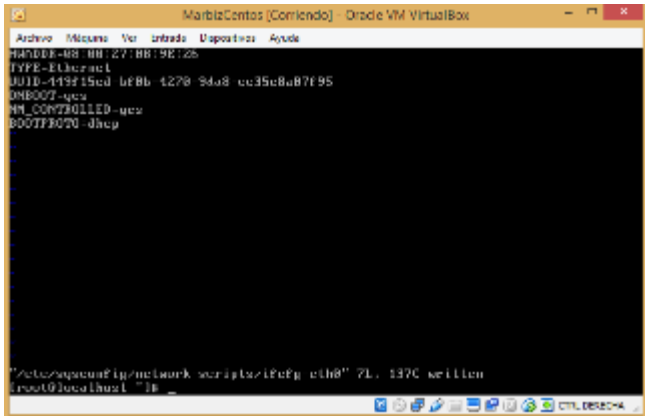
Se descargó la imagen ISO de la página http://centos.uniminuto.edu/6.8/isos/i386/ y se descargó CentOS-6.8-i386-minimal.iso de 354M. Al iniciar la instalación se seleccionó la primera opción, así como se muestra en la Figura 10. Pantalla inicial instalación de Centos.

Figura 10. Pantalla inicial instalación de Centos

Fuente: El autor.

Terminada la instalación se configura Centos para que pueda conectarse a Internet, para eso se ingresa al archivo de configuración de red y se habilita el modo ONBOOT a yes como lo indica la figura 11 donde se establece este parámetro.

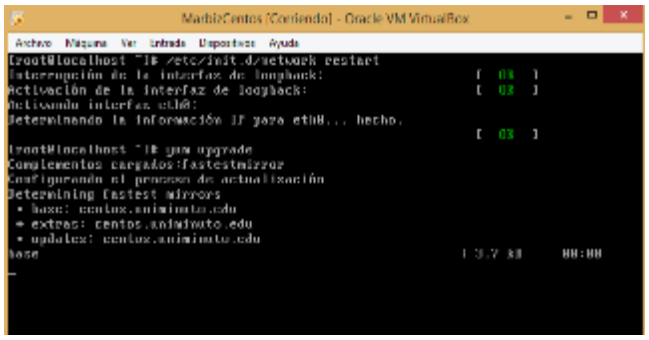
Figura 11. Editar interface eth0.



Fuente: El autor.

Una vez se realiza la configuración se reinicia el servicio de red. Luego se actualiza el sistema con el comando yum upgrade como lo indica la figura 12 digitando el anterior comando.

Figura 12. Reinicio servicio de red y actualización

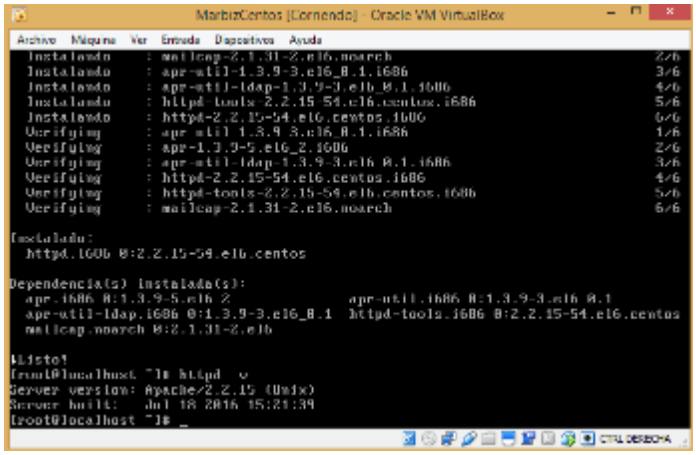


Fuente: El autor.

Instalación servicios Apache Php y MySQL. Para la configuración de apache requiere instalar http y luego iniciar el servicio.

En las siguientes imágenes se muestra la instalación, inicio de servidor Apache y configuración inicio automático. En la figura 13 se muestra la instalación de apache.

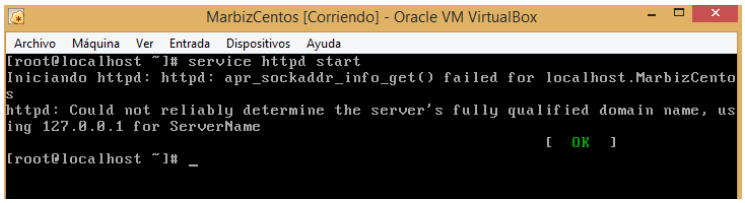
Figura 13. Instalación Servidor Apache



Fuente El autor.

En la figura 14 se indica el comando para iniciar el servicio de servidor apache con el comando service httpd start.

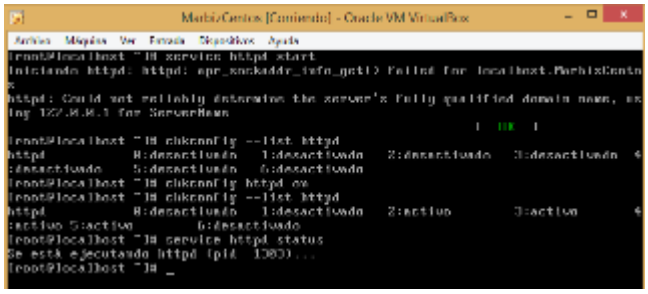
Figura 14. Inicio de servidor Apache



Fuente El autor.

En la figura 15 se indica como configurar el servicio de apache para que inicie de manera automática mediante el comando chkconfig –lst httpp.

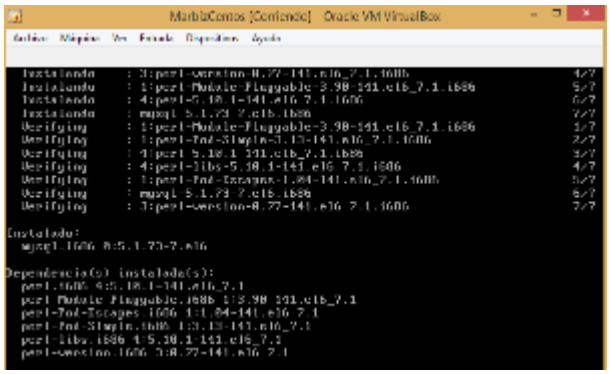
Figura 15. Configuración automática Apache



Fuente El autor.

Se instaló el gestor de bases de datos MYSQL versión 5.1.73, con el usuario root utilizando el comando # yum install mysql mysql-server. En la figura 16 se muestra el proceso de instalación del servidor después de aplicar el comando.

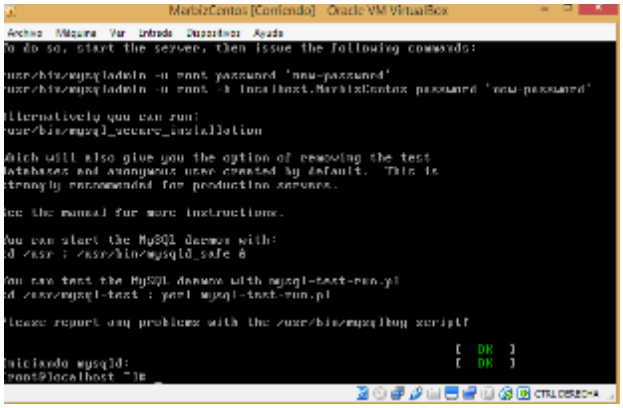
Figura 16. Instalación servidor mysql



Fuente El autor.

En la figura 17 se muestra el inicio del servicio de mysql mediante el comando de mysqld start.

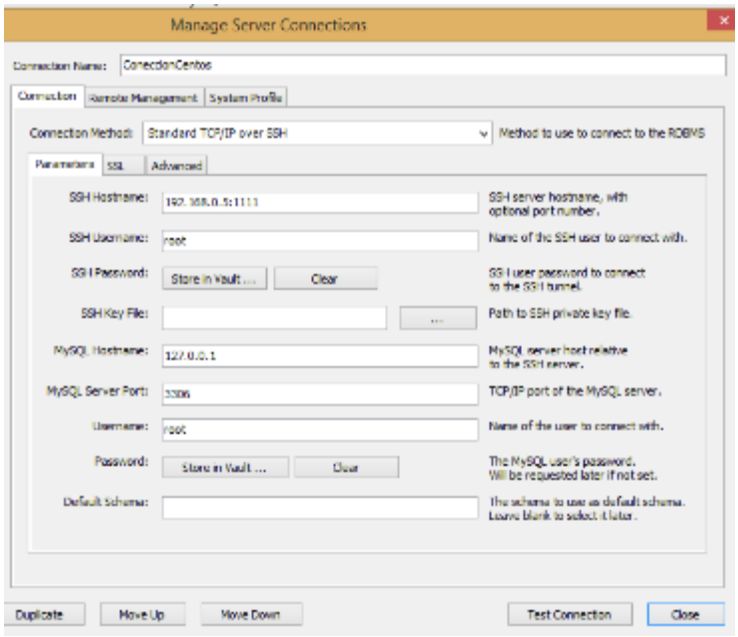
Figura 17. Inicio servicio de mysqld



Fuente El autor.

- Configuración de base de datos de prueba. Se realiza la conexión al esquema de mysql mediante conexión desde equipo anfitrión con la opción TCP/IP sobre SSH como se muestra en la imagen 18 con los parámetros de conexión.

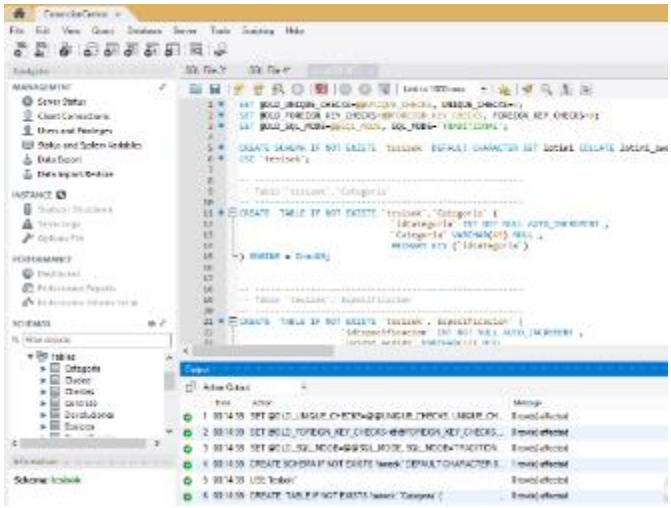
Figura 18. Conexión a servidor Centos sobre SSH



Fuente: El autor

En la figura 19 se muestra la ejecución del script generado con el modelo entidad relación desde el aplicativo MySQL Workbench.

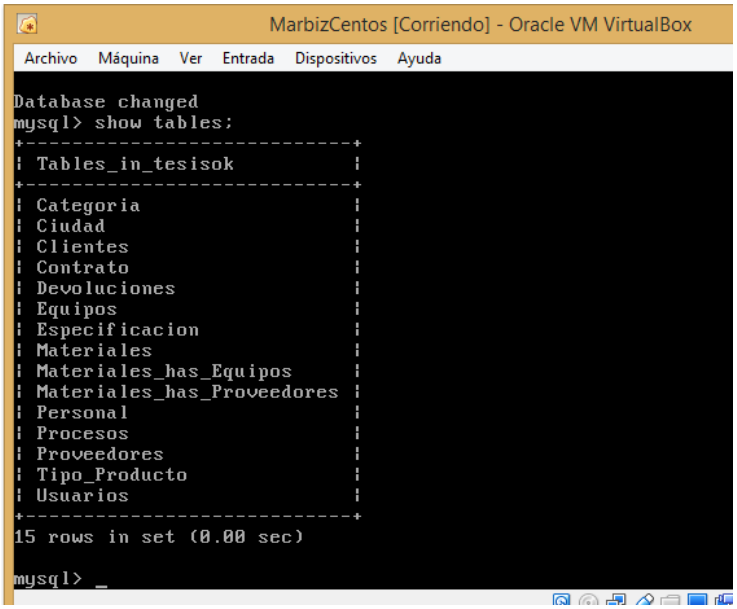
Figura 19. Creación de base de datos



Fuente: autor

En la figura 20 se muestra la base de datos de prueba creada desde la máquina virtual con sistema operativo Centos.

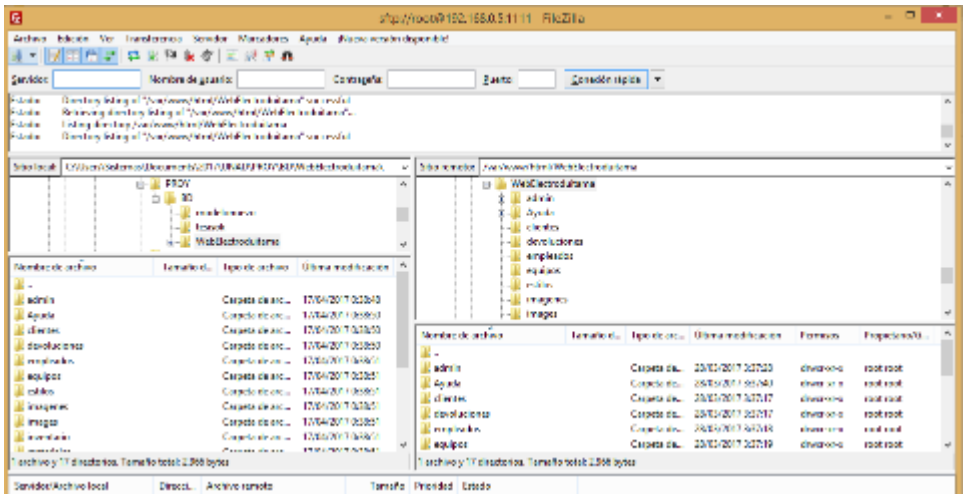
Figura 20. Base de datos creada en Centos



Fuente: El autor.

Se crean usuarios y se ingresan datos a la base de datos. Se carga la página web en servidor utilizando servicio de sftp de archivos mediante FileZilla como se muestra en la figura 21.

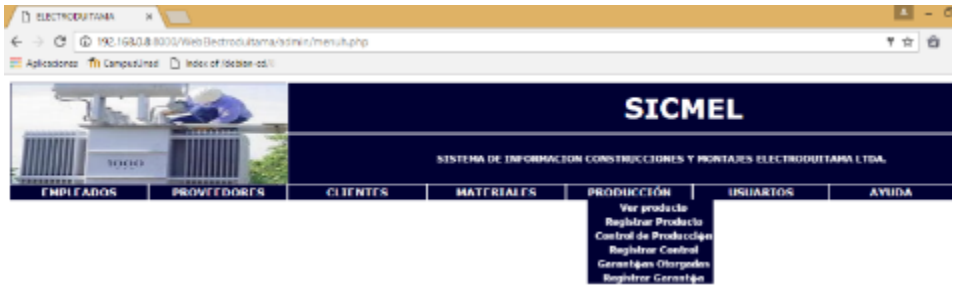
Figura 21. Carga de página web en servidor Centos



Fuente: El autor

En la figura 22 se muestra el acceso a la página web que interactúa con la base de datos creada digitando la dirección ip del servidor Centos..

Figura 22. Acceso a página web alojada en Centos



Fuente: El autor.

6.1.1 Pruebas de escaneo y vulnerabilidad sobre MySQL. Una vez instalado y configurado el servidor de Linux Centos y la base de datos en MySql se realiza la instalación en otra máquina virtual de Kali Linux, sistema que contiene un kit

completo de herramientas para realizar las pruebas de escaneo y de inyección a la base de datos.

Se preparó y se instaló máquina virtual con Kali Linux en VMware como se indica en la Figura 23 se seleccionó e instaló modo gráfico.

Figura 23. Parámetros para máquina virtual Kali Linux

Fuente: El autor

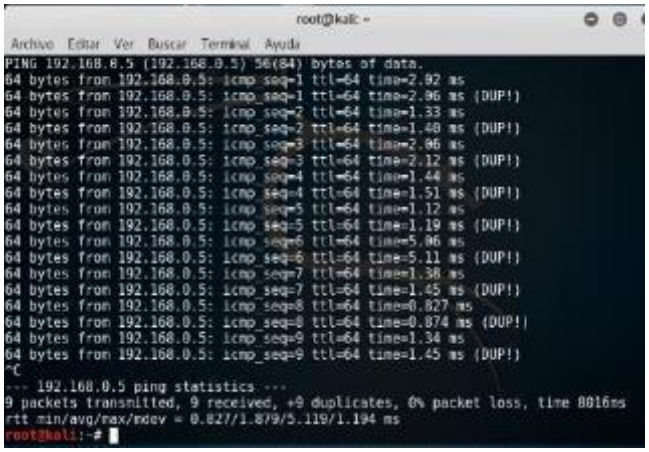
En la figura 24 se muestra la verificación de la IP de la máquina con Kali que este dentro de la red para poder realizar el ejercicio con la máquina Centos.

Figura 24. Dirección IP Kali Linux

Fuente: El autor

Se realiza prueba de ping desde Kali hacia servidor Centos que es donde está configurada la base de datos como se muestra en la figura 25.

Figura 25. Ping a Centos desde Kali

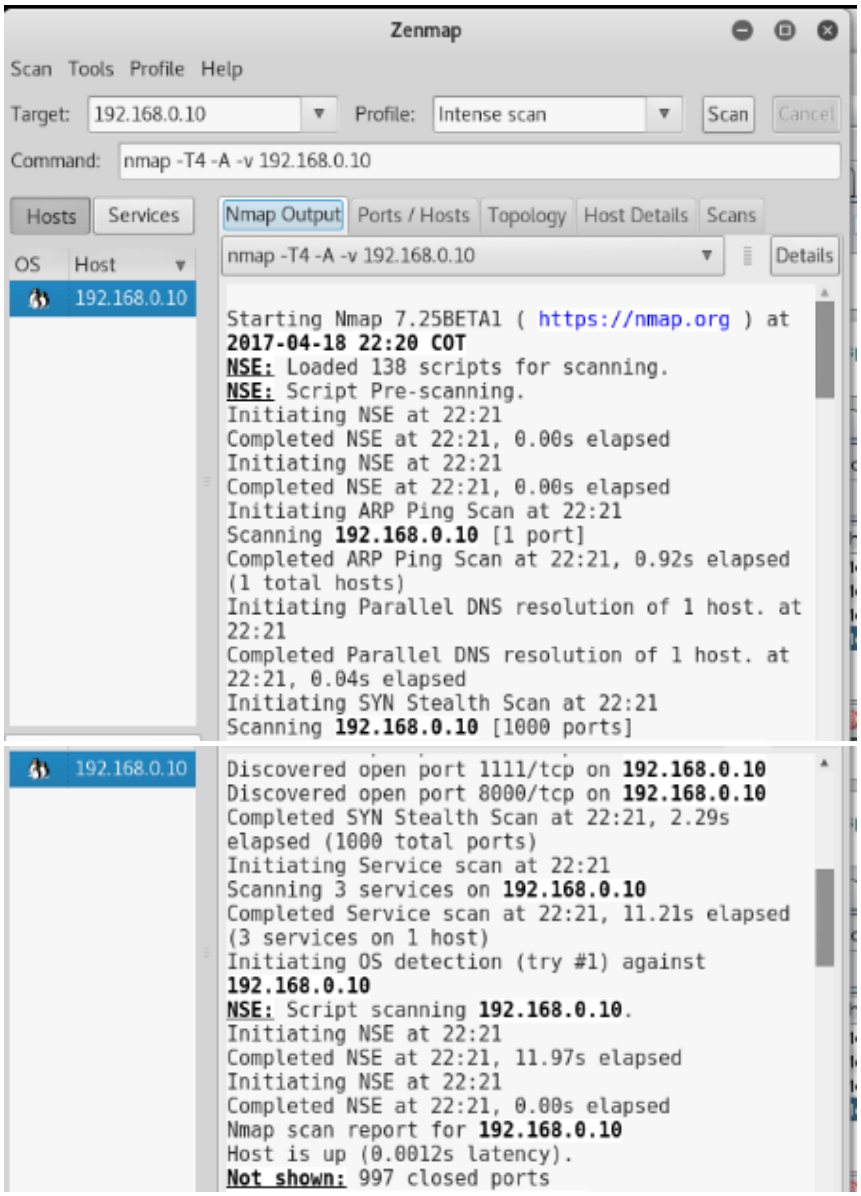


Fuente: El autor

6.1.1.1 ZENMAP sobre MYSQL

Se utilizó la herramienta Zenmap y se realiza escaneo de la máquina con el servidor web con el fin de detectar puertos abiertos. La herramienta provee información detallada, en la figura 26 se muestra el escaneo de 1000 puertos de los cuales 7 están abiertos y 993 cerrados.

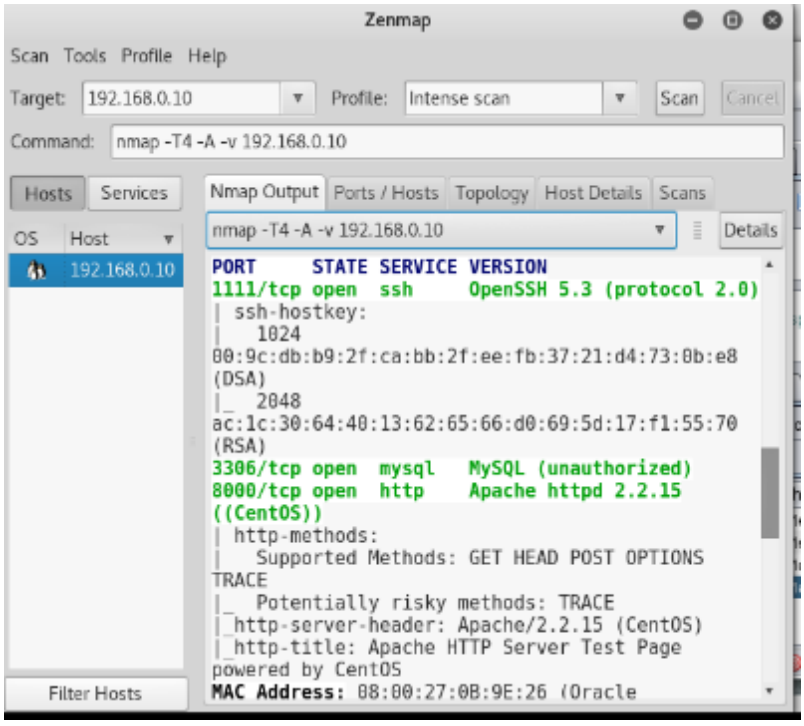
Figura 26. Escaneo con Zenmap sobre MySQL



Fuente: autor

De igual forma como se muestra en la figura 27 Zenmap ofrece información valiosa como es el nombre del servidor web en este caso apache, la versión, el empaquetador que fue instalado, la versión de PHP, los puertos que tienen el servicio.

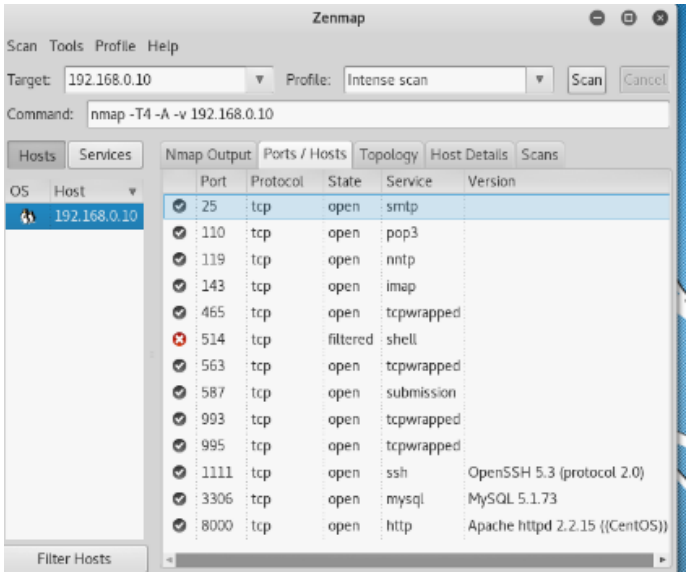
Figura 27. Puertos y servicios escaneados con Zenmap sobre maquina con MySQL



Fuente: El autor.

En la figura 28 se muestra un resumen de lo encontrado con el escaneo de la herramienta Zenmap, se puede observar que capturó puertos tcp como por ejemplo el 8000 del servidor Apache con su respectiva versión, el puerto 3306 de MySql donde está alojada la base de datos y el puerto 1111 para la conexión por SSH, estos y otros puertos cuando se encuentran abiertos y no se analiza el tráfico es una vulnerabilidad que puede ser explotada y pueden ser aprovechados para realizar un ataque.

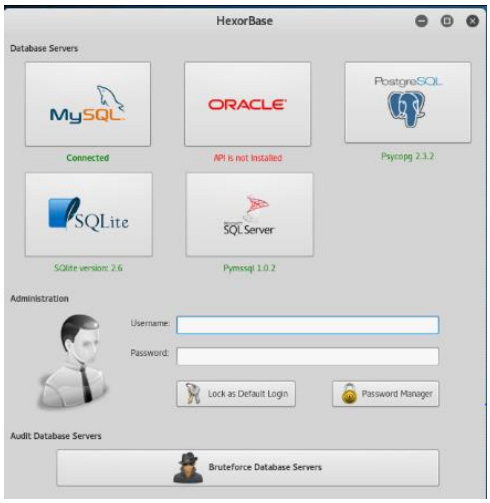
Figura 28. Resumen escaneo con Zenmap sobre maquina con MySQL



Fuente: El autor

6.1.1.2 HEXORBASE sobre MySQL. Es una aplicación de base de datos diseñada para la administración y la auditoria de varios servidores de bases de datos de forma simultánea desde una ubicación centralizada, es capaz de realizar consultas SQL y ataques de fuerza bruta contra servidores de bases de datos MySQL, SQLite, Microsoft SQL Server, Oracle y PostgreSQL[5]. En la figura 29 se muestra la interfaz del programa ejecutándose desde kali Linux.

Figura 29. Interfaz de Hexorbase sobre MySQL



[5] http://ehacking.com.bo/hexorbase-excelente-herramienta-de-auditoria-y-administracion-de-multiples-bases-de-datos/.

Fuente: El autor.

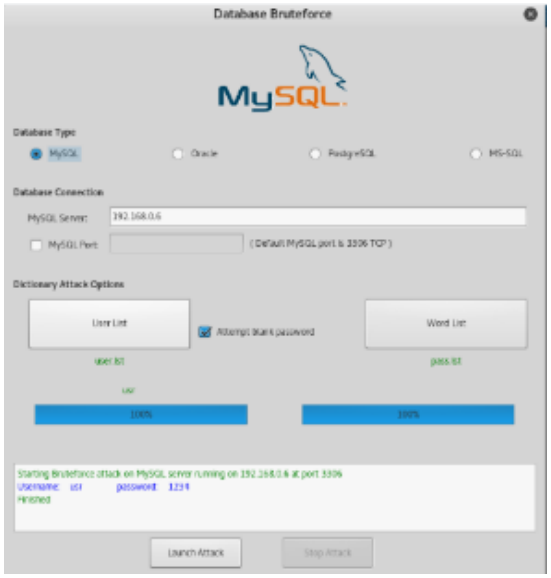
Se puede realizar un ataque de fuerza bruta para conseguir el usuario y la contraseña de acceso. Primero se selecciona MySql, se hace clic en User List y se carga diccionario de usuarios al igual que el Word List para cargar diccionario de contraseñas, se realiza el ataque y se observa como resultado que MySql tiene restringido el acceso a la base de datos por el puerto 3306 si no se tiene permiso respectivo. En la figura 30 se muestra el resultado del ataque después de haber colocado los parámetros indicados anteriormente.

Figura 30. Ataque de fuerza bruta con Hexorbase sobre MySQL

Fuente: El autor.

Es importante aclarar que el DBMS de MySql no permite todas las conexiones por este puerto, es un aspecto positivo de este gestor de base de datos. Para solucionar este inconveniente en la conexión y probar la herramienta hexorbase, desde MySql se asigna el respectivo permiso a esta dirección IP y se realiza nuevamente el ataque. En la figura 31 se observa el ataque con resultado positivo.

Figura 31. Ataque con Hexorbase resultado usuario y contraseña sobre MySQL



Fuente: El autor.

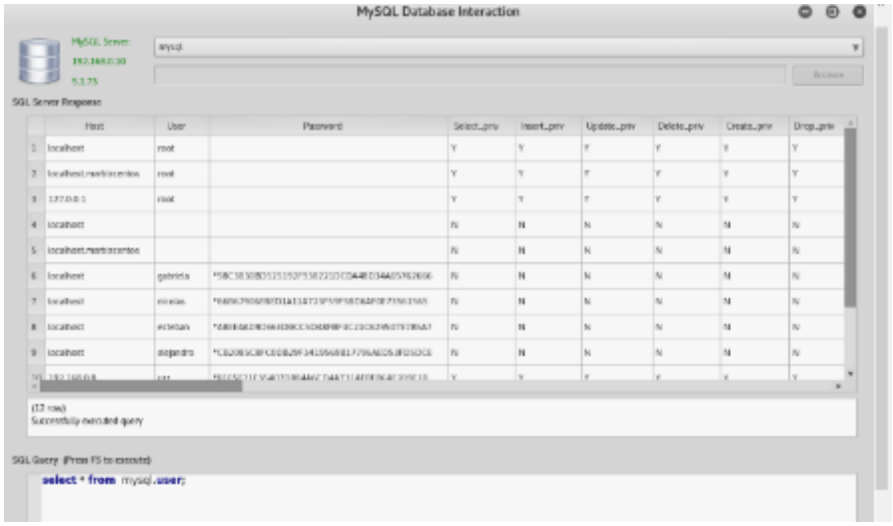
Con el usuario y contraseña se puede acceder desde Hexorbase a la base de datos y realizar consultas y transacciones en la base de datos como se muestra en la figura 32.

Figura 32. Ingreso a la base de datos MySql con Hexorbase

Fuente: El autor.

Así como se puede observar las tablas y sus datos, es posible seleccionar la lista de usuarios de mysql como se muestra en la figura 33 y ejecutar comandos de inyección, cambiar datos etc.

Figura 33. Listado de usuarios de MySql



Fuente: EL autor.

Hexorbase permite también generar reportes y exportarlos en html para una mejor visualización, así como lo muestra la figura 34 se generó un reporte sobre los datos encontrados.

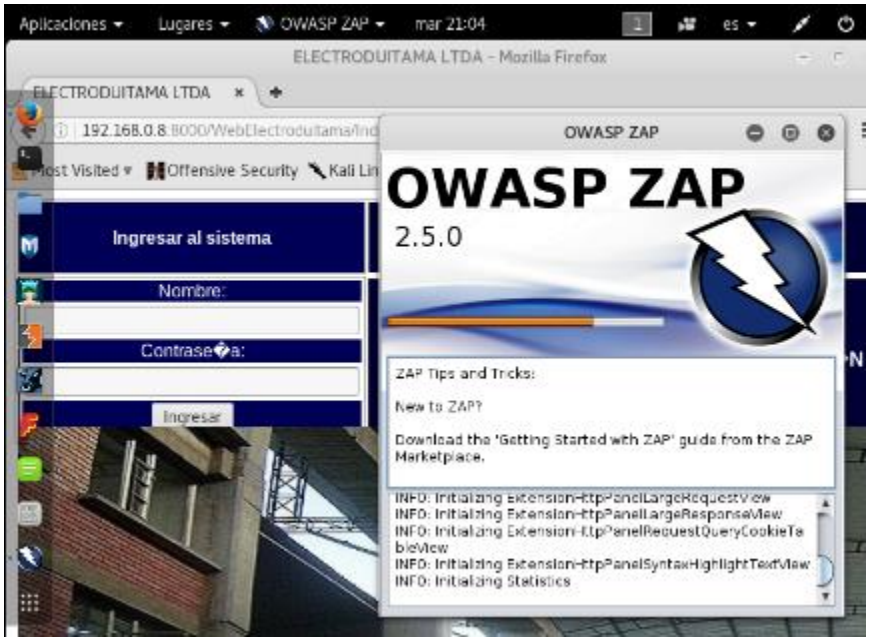
Figura 34. Reporte de Hexorbase en html sobre MySQL

Fuente: El autor.

6.1.2 Pruebas de Inyección SQL sobre MySQL. Una vez identificado el acceso al puerto de comunicación con MySQL, así como el resultado positivo de ataque con Hexorbase, se utilizaron cuatro herramientas para hacer las pruebas de ataque de inyección SQL: Owasp Zap, SQLmap, Paros y Jsql Injection obteniendo resultados positivos con excepción de Jsql Injection que después de varios intentos no mostró ningún resultado.

6.1.2.1 Owasp Zap sobre MySQL. Se ejecuta la herramienta owaspzap en kali, para realizar el ataque como lo indica la figura 35.

Figura 35. Ejecución de OWASP ZAP sobre MySQL

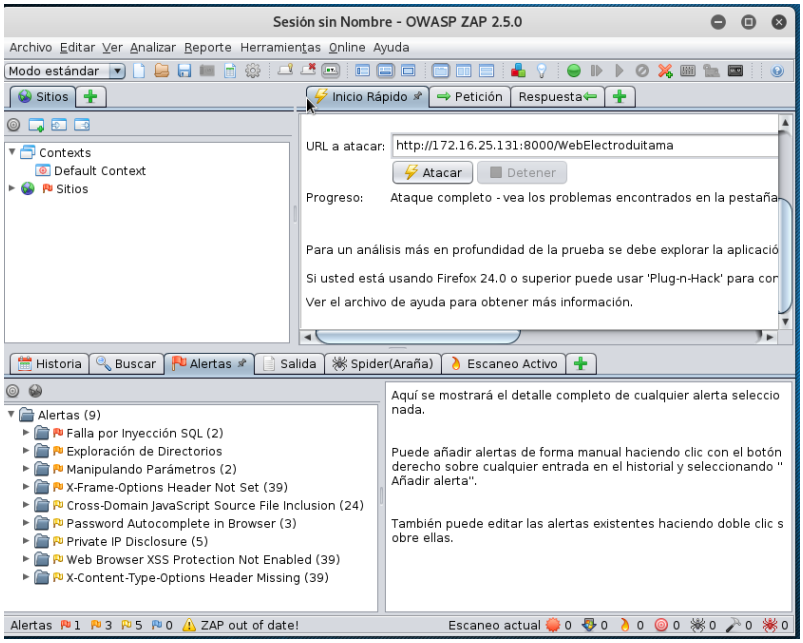


Fuente: El autor

La herramienta identificó nueve alertas agrupadas por el respectivo nombre del ataque, la descripción y la posible solución. En este caso para la página Web Electro

Duitama como se muestra en la figura 36, se encontraron cinco alertas entre las cuales están: Falla por inyección SQL, exploración de directorios, manipulación de parámetros, activación de opciones X-Frame que puede traer riesgos de ataques de clickjacking, el sitio tiene activo la opción de autocompletar cuando se ingresa password, se puede obtener la dirección del servidor de alojamiento de la página web, no se tiene activo en el sitio la protección contra ataques XSS entre otros.

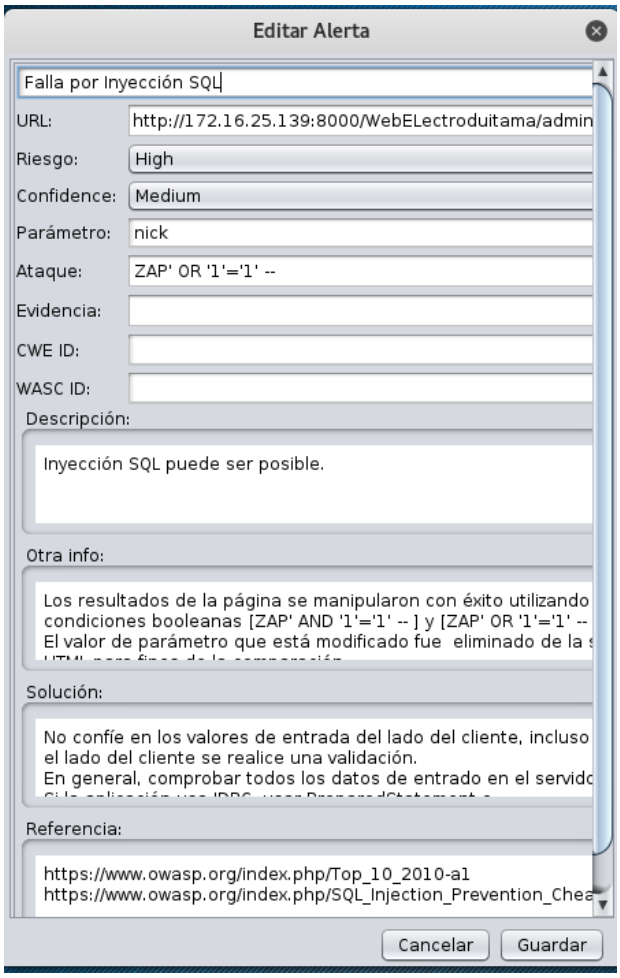
Figura 36. Resultado obtenido con OWASPZAP sobre MySQL



Fuente: El autor

En la figura 37 se muestra el resultado encontrado por falla de inyección SQL, se detectó un archivo en directorio admin, el cual permite que el parámetro nick pueda ser manipulado utilizando condiciones booleanas (ZAP’ AND ‘1’=’1’) y (ZAP’ OR ‘1’=’1’), es una vulnerabilidad que puede retornar datos dependiendo el código utilizado en el parámetro.

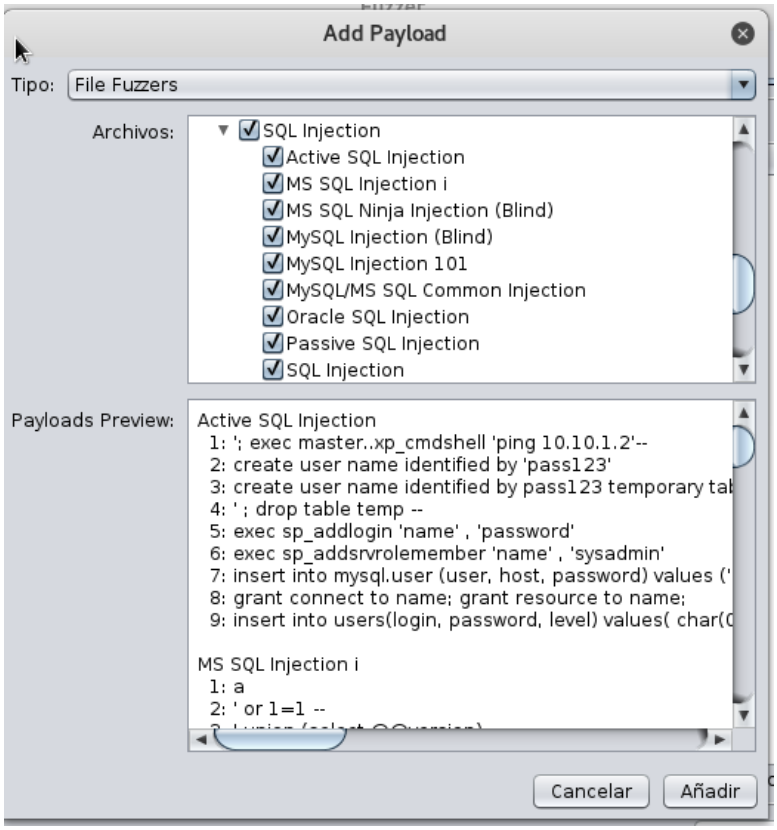
Figura 37. Fallas por Inyección SQL sobre MySQL



Fuente: el autor

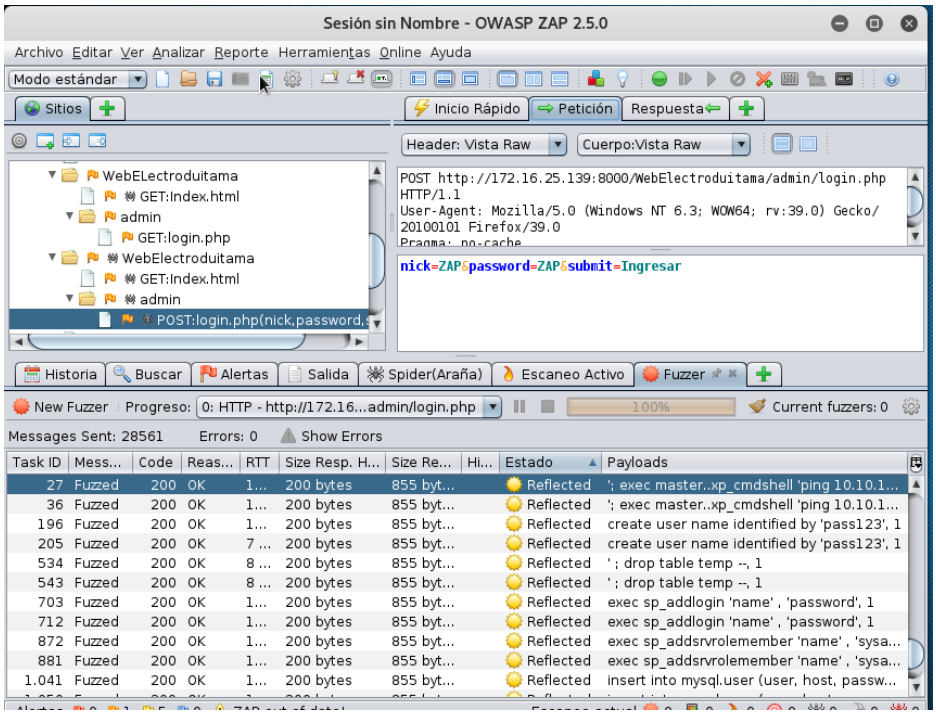
Teniendo en cuenta que se detectó como vulnerabilidad dos fallas por inyección SQL, se realizó un ataque de inyección utilizando la opción spider del owasp zap. Se cargaron los ficheros por defecto que trae la herramienta para hacer los ataques de inyección SQL o payloads como se indica en la figura 38.

Figura 38. Payload de inyección SQL de OWASP ZAP sobre MySQL



En la figura 39 se muestra el resultado obtenido al realizar el ataque con la opción fuzzer. En la parte izquierda de la figura se encuentra el árbol de consultas, allí se ubicó el archivo donde fue detectada la vulnerabilidad, en este caso sobre el archivo login.php, en la parte derecha se muestran las opciones de petición y respuesta hacia el servidor, indicando los valores por parámetro para el usuario y contraseña de acceso al aplicativo. El resultado del fuzzer muestra 28.561 sentencias SQL de las cuales fueron positivas 31 o con respuesta del servidor.

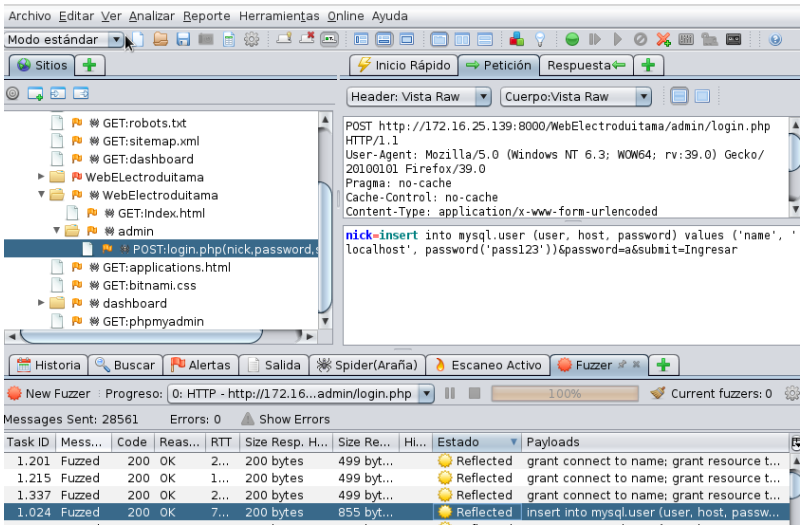
Figura 39. Resultados de ataque con inyección SQL sobre MySQL



Fuente: el autor.

En la figura 40 se muestra otro ejemplo de inyección SQL con los payloads cargados de la herramienta. Se envío una petición al servidor de inserción de datos a la base de datos en la tabla de usuarios.

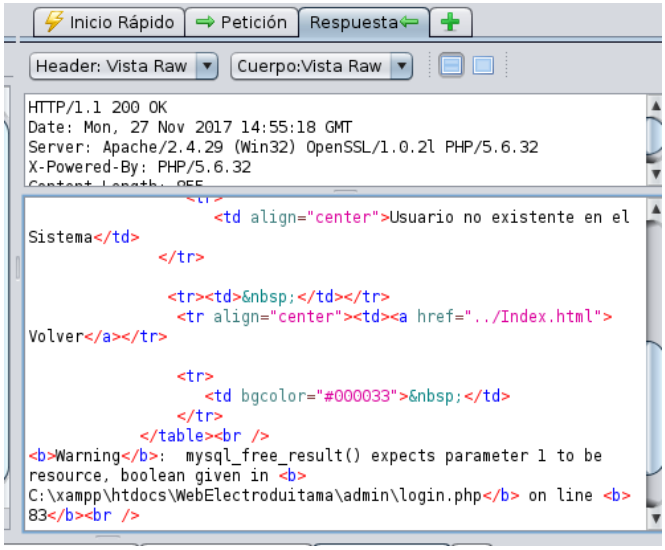
Figura 40. Inyección SQL inserción a la tabla usuarios sobre MYSQL



Fuente: el autor

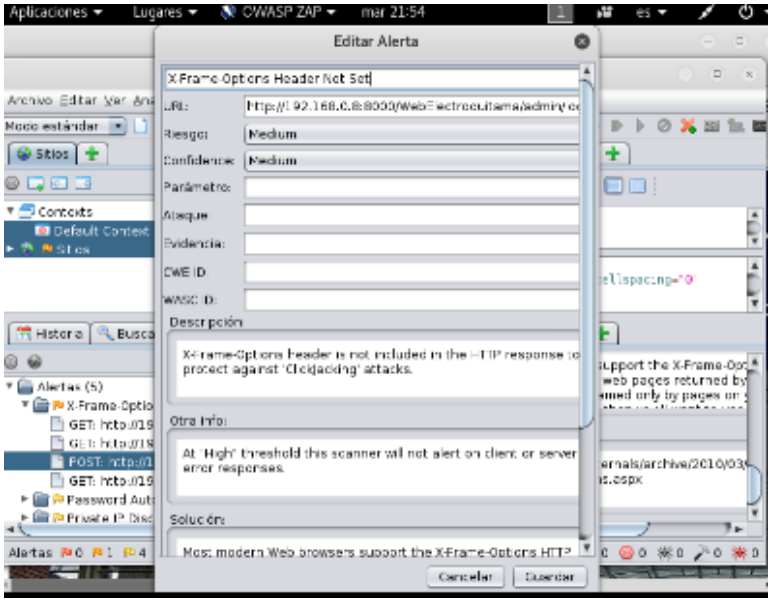
En la figura 41 se muestra la respuesta del servidor, aunque el parámetro se puede manipular y se envió una inserción de datos a una tabla no se realizó la transacción solicitada indicando que el usuario no existe en el sistema.

Figura 41. Respuesta a la inyección SQL por inserción de datos sobre MySQL



En la figura 42 se muestra otro tipo de vulnerabilidad encontrada sobre la falta de filtrado en las cabeceras de las páginas exponiéndose a un posible ataque de clic o clickjacking.

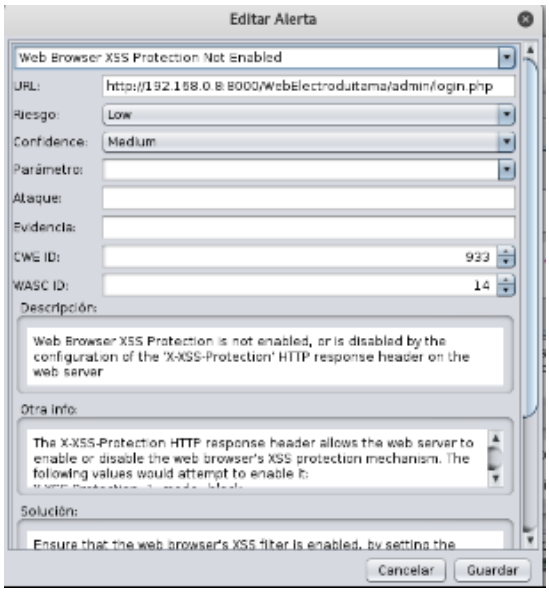
Figura 42. Vulnerabilidad ataque de clickjacking sobre MySQL



Fuente: el autor.

Otro problema encontrado en aplicación tiene que ver con la falta de protección frente al XSS, no están configurados los mecanismos de protección frente a esta vulnerabilidad como se indica en la figura 43.

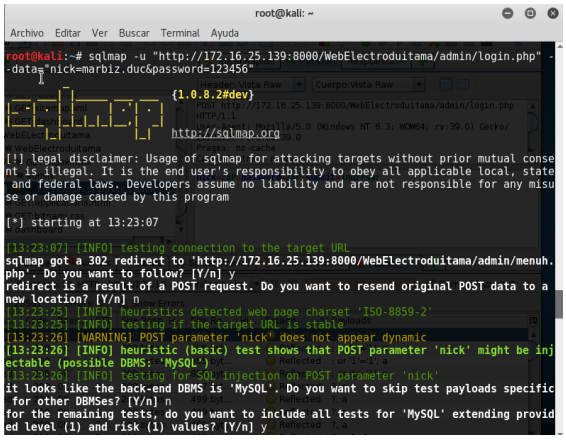
Figura 43. Vulnerabilidad ataque de XSS sobre MySQL



Fuente: El autor.

6.1.2.2 SQLMAP sobre MySQL. Otra herramienta importante de inyección SQL es SQLMAP. En la figura 44 se muestra la configuración para realizar la inyección SQL a través de SQLMAP, además se refleja el resultado de la posibilidad de realizar una inyección sobre el parámetro nick.

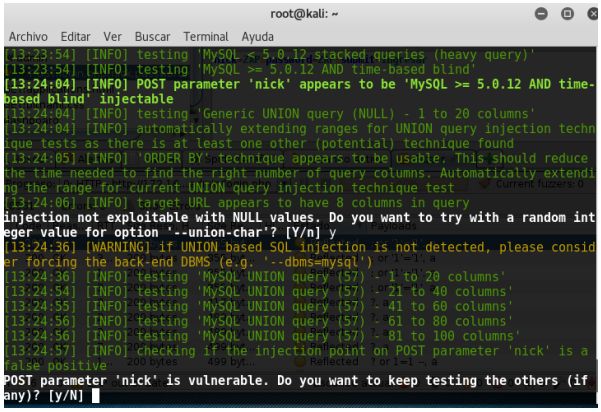
Figura 44. Ejecución de SQLMAP sobre MYSQL



Fuente: el autor

Una vez confirmado el parámetro nick como vulnerable, SQLMAP realiza otras sentencias con el fin de recuperar más datos sobre la base de datos MYSQL como se muestra en la figura 45.

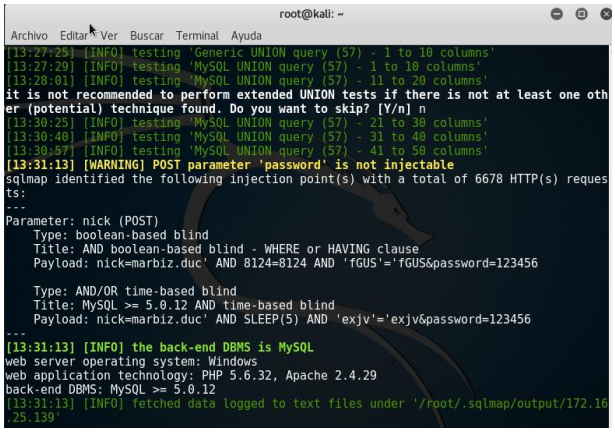
Figura 45. Ejecución de sentencias de inyección SQL sobre MySQL



Fuente: el autor

Finalmente, en la figura 46 se muestran los resultados de los puntos de inyección realizados, con un total de 6678, ataques a ciegas o Blind SQl Injection utilizando valores booleanos.

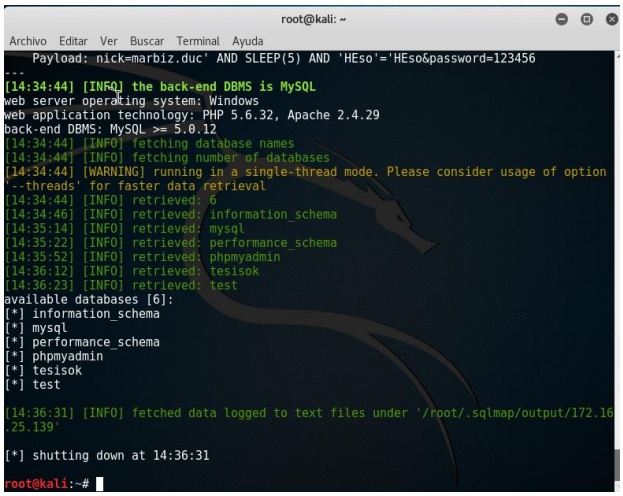
Figura 46. Resultados de la inyección SQL con SQLMAP sobre MySQL



Fuente: el autor

Con SQLMAP también se obtiene las bases de datos existentes, en la figura 47 se muestra seis bases de datos almacenadas en el DBMS de MySQL.

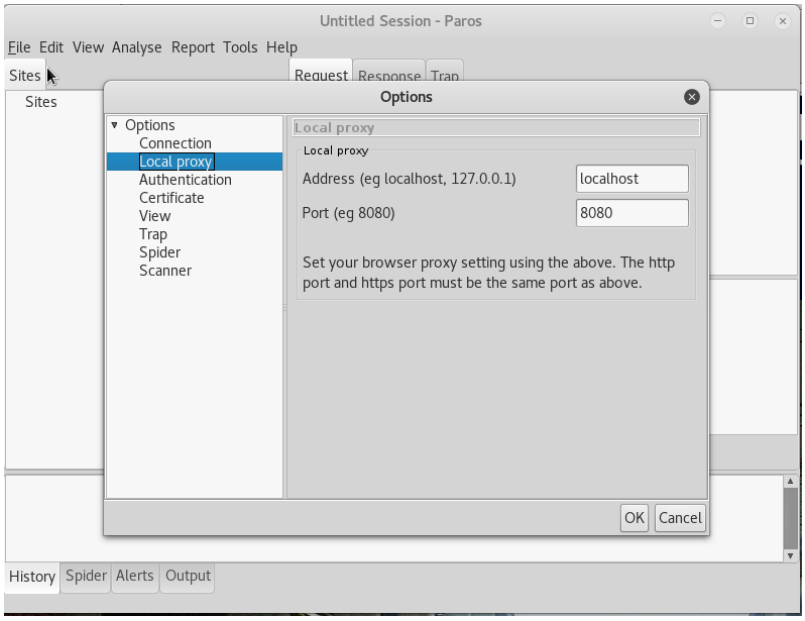
Figura 47. Bases de datos obtenidas con SQLMAP sobre MySQL



Fuente: el autor

6.1.2.3 Paros sobre MySQL. Se utilizó la herramienta Paros que funciona como un proxy HTTP / HTTPS basado en Java para evaluar la vulnerabilidad de las aplicaciones web y tiene otras características que incluyen arañas, certificado de cliente, cadena de proxy, escaneo inteligente para inyecciones XSS y la inyección SQL. En la figura 48 se muestra la configuración como proxy de la herramienta.

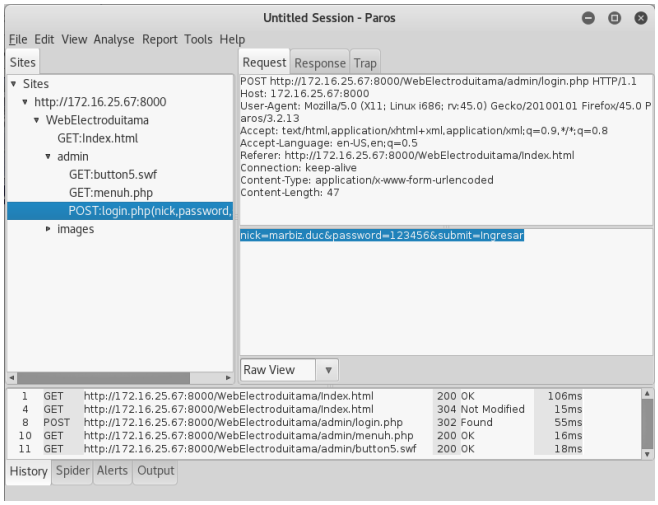
Figura 48. Configuración de Paros sobre MySQL



Fuente: el autor.

Una vez realizado el análisis se observa que tiene vulnerabilidad para ataque de inyección SQL sobre los parámetros de ingreso de sesión al sitio web, mediante la petición POST en los parámetros del formulario de captura de datos del usuario como se muestra en la figura 49.

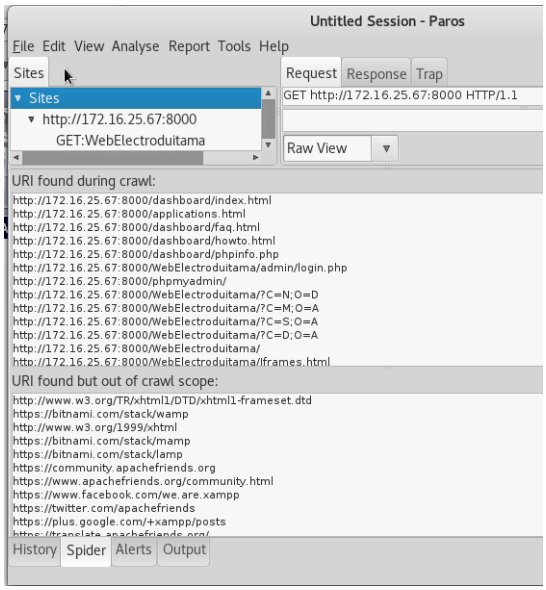
Figura 49. Ejecución de Paros sobre MySQL



Fuente: el autor.

La herramienta Paros también contiene una utilidad llamada spider que permitió visualizar otras páginas del sitio susceptibles a ataque por inyección SQL, en la figura 50 se muestra la opción spider reflejando los resultados.

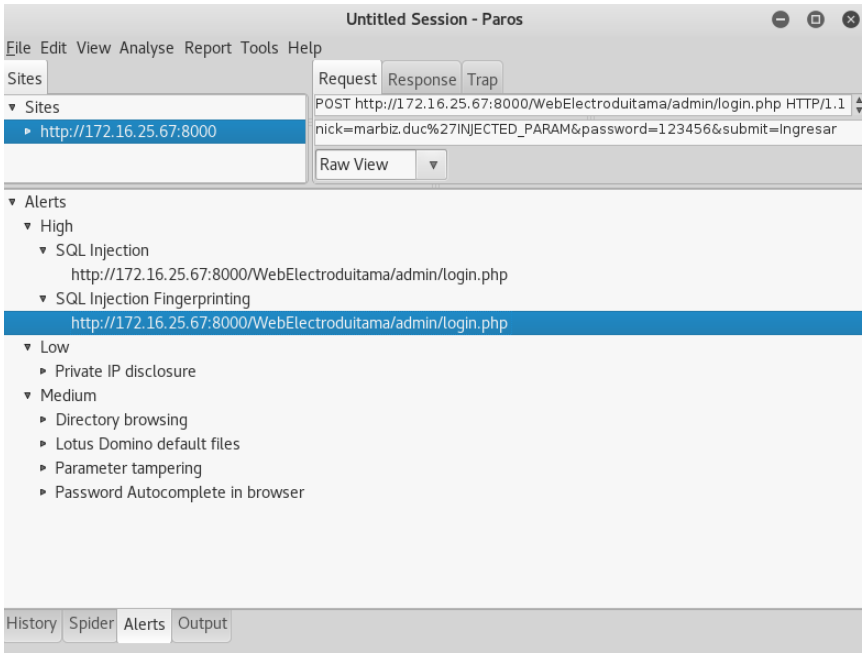
Figura 50. Resultados ejecución Spider de Paros sobre MySQL



Fuente: el autor.

En la figura 51 se muestra la pestaña alerta de la herramienta Paros, allí se pudo identificar de manera agrupada por nivel alto, medio y bajo las diferentes vulnerabilidades que tiene el sitio en especial la de inyección SQL.

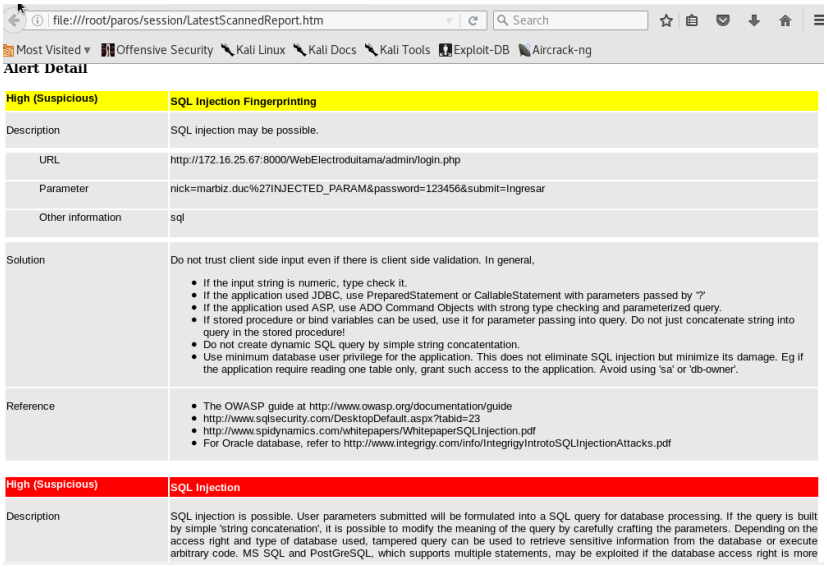
Figura 51. Resultados Inyección SQL de Paros sobre MySQL



Fuente: el autor.

En la figura 52 se muestra la opción que tiene Paros para generar reportes sobre el escaneo realizado y sobre la inyección SQL que se pudo realizar sobre MySQL.

Figura 52. Reporte de Paros sobre MySQL

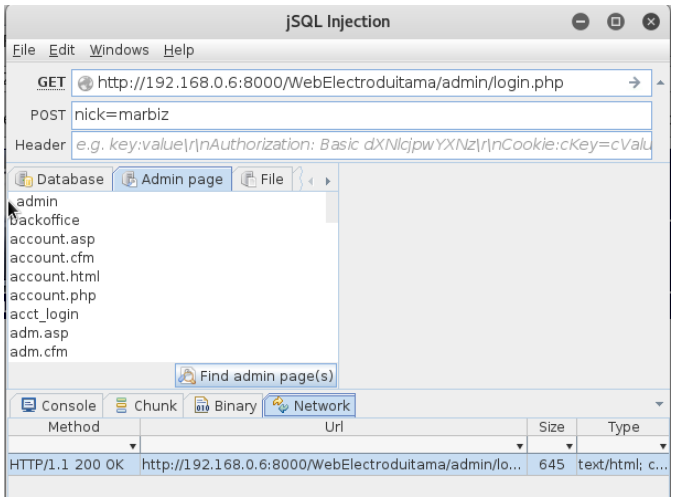


Fuente: el autor.

6.1.2.4 jSQL Injection sobre MySQL

Se utilizó la herramienta de Kali Linux jSQL Injection para realizar otro ataque a la base de datos de MySQL. En la figura 53 se observa el panel principal de la herramienta, en la parte superior se ingresa la dirección del sitio web a atacar, en la parte inferior se selecciona el tipo de base de datos, para este caso MySQL y se efectúa el proceso de inyección.

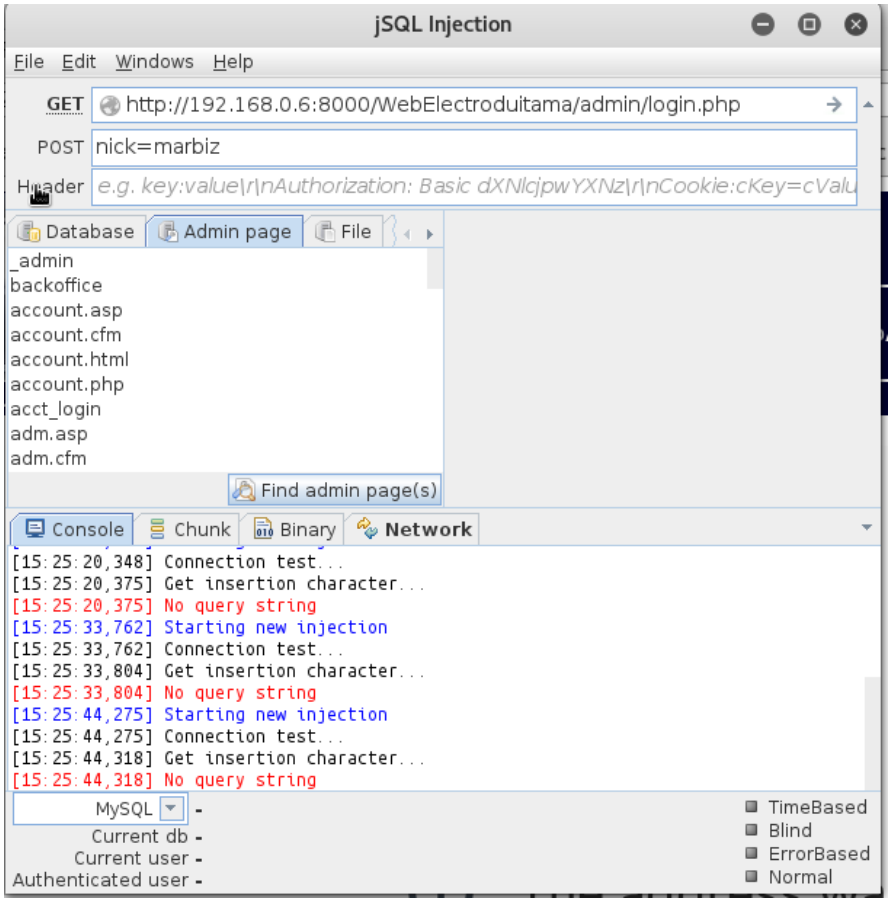
Figura 53. Ejecución de jSQL Injection sobre MySQL



Fuente: el autor.

Después de varios intentos con la herramienta, y los posibles comodines para inyectar código SQL la inyección falla como se muestra en la figura 54.

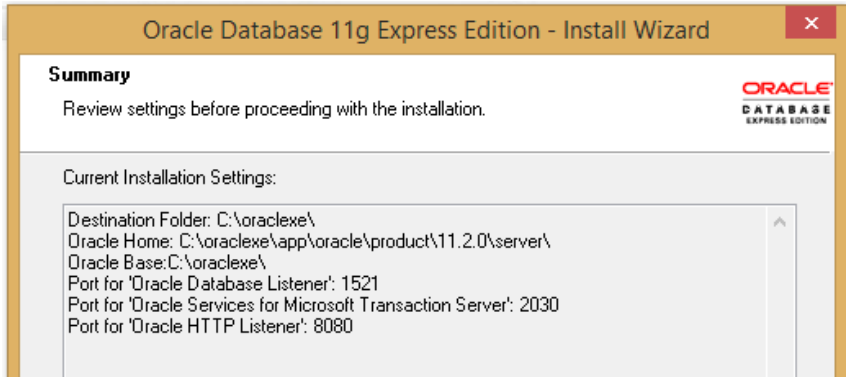
Figura 54. Resultado de ejecución de jSQL Injection sobre MySQL



Fuente: el autor.

6.2 ENTORNO DE PRUEBA EN ORACLE. Se descargó e instaló base de datos Oracle Database 11g Release 2 Express Edition de página oficial de Oracle. Cuando se inicia la instalación se asigna una contraseña que es utilizada para dos (2) usuarios para la administración por defecto, SYS y SYSTEM. En la figura 55 se muestra el resumen de la instalación de la versión Express de Oracle 11g.

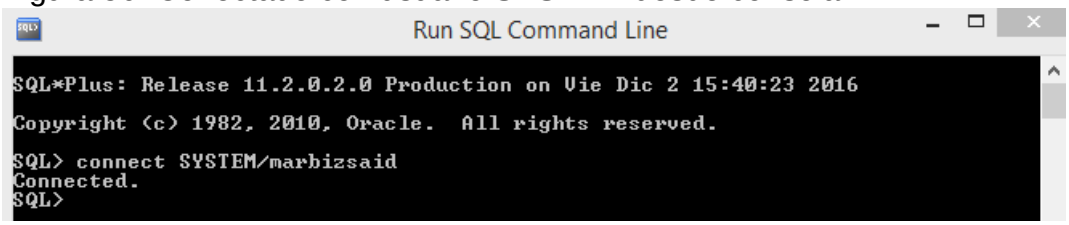
Figura 55. Resumen Instalación Oracle Database 11g Release 2 Express Edition



Fuente: El autor.

Una de las formas más sencillas de conexión es utilizando la consola de comandos donde se pueden ejecutar consultas sql como se muestra en la figura 56 accediendo por SQL PLUS.

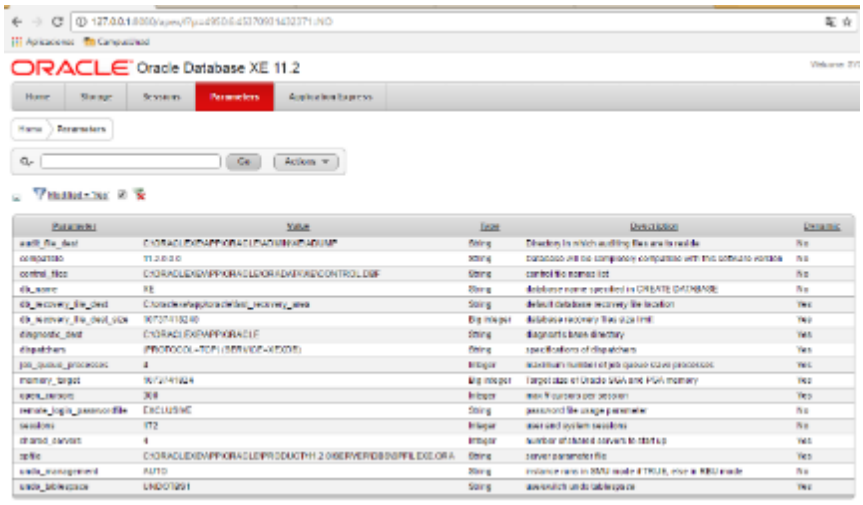
Figura 56. Conectado con usuario SYSTEM desde consola



Fuente: El autor

Otra forma de conexión de Oracle es en entorno web mediante el puerto 8080 que permite crear aplicaciones. En la figura 57 se muestra los parámetros de configuración en el entorno web.

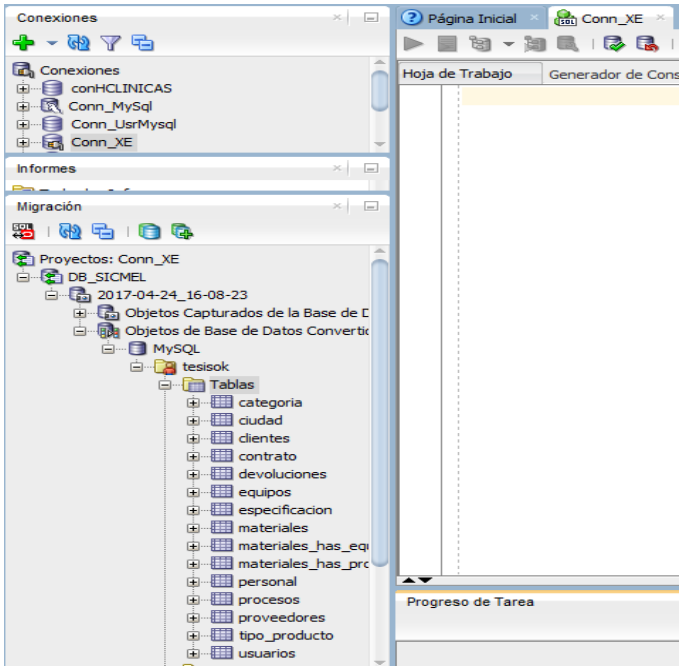
Figura 57. Conexión de Oracle en entorno web



Fuente: El autor

Con el fin de dar continuidad y observar las diferencias entre motores de base de datos se migró la base de datos del entorno de prueba uno de MySql a Oracle XE para eso se utilizó sqldeveloper como se indica en la figura 58, mediante esta herramienta se importa la base de datos.

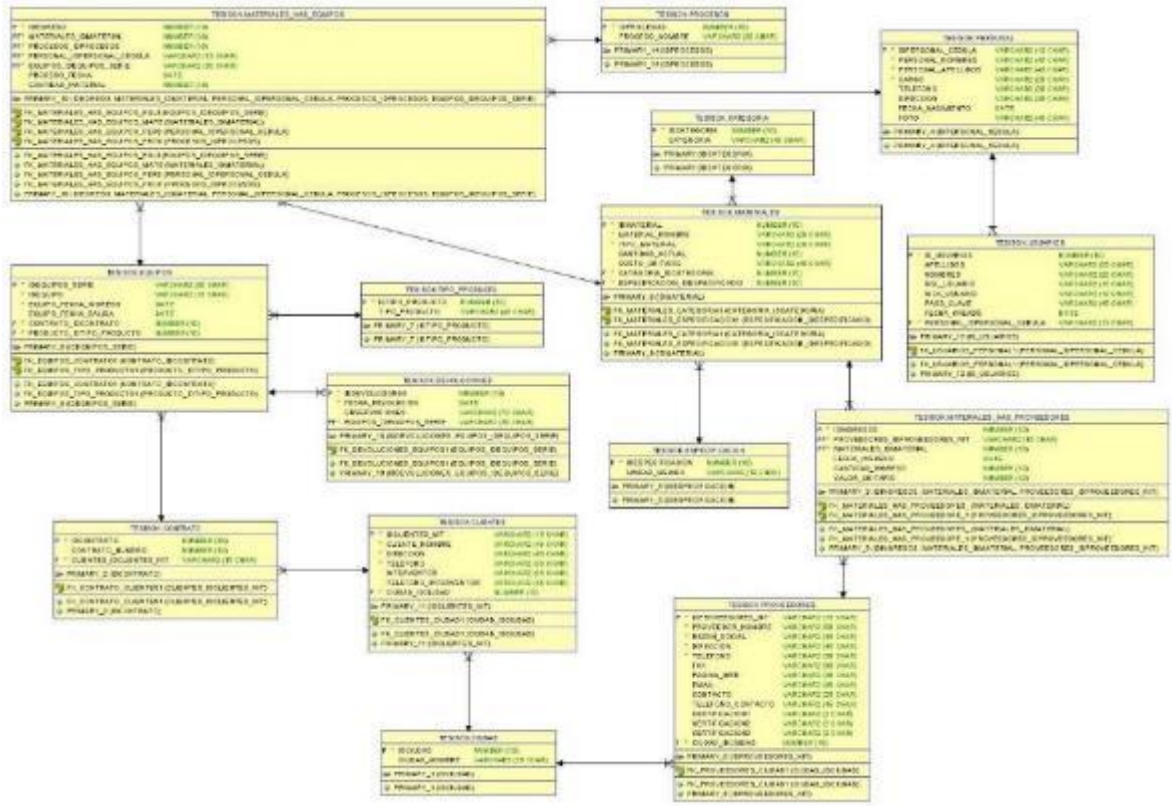
Figura 58. Migración de MySql a Oracle con SqlDeveloper



Fuente: El autor.

Una vez migrada la base de datos se genera y verifica el modelo entidad relación para la base de datos en estudio de la empresa Electroduitama. En la figura 59 se muestra el modelo generado en Oracle.

Figura 59. Modelo Entidad Relación en Oracle

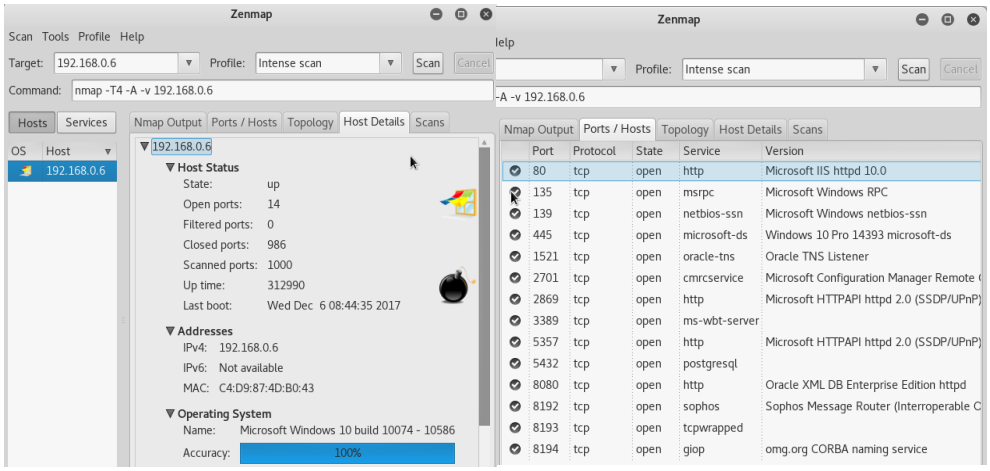


Fuente: el autor

6.2.1 Pruebas de escaneo y vulnerabilidad sobre Oracle. Una vez instalada y configurada la base de datos de Oracle se utilizó la instalación de la máquina virtual de Kali Linux, con las herramientas Zenmap, Nessus y Hexorbase para las pruebas de escaneo y vulnerabilidad a la base de datos.

6.2.1.1 ZENMAP sobre Oracle. Se utilizó nuevamente la herramienta Zenmap y se realiza escaneo de la máquina con el servidor web de Oracle con el fin de detectar puertos abiertos. La herramienta provee información detallada, en la figura 60 se muestra el escaneo de 1000 puertos de los cuales 14 están abiertos dentro de los cuales está el puerto 1521 del listener de Oracle y el puerto 8080 para la conexión httpd, de igual manera la herramienta reporta 986 puertos cerrados.

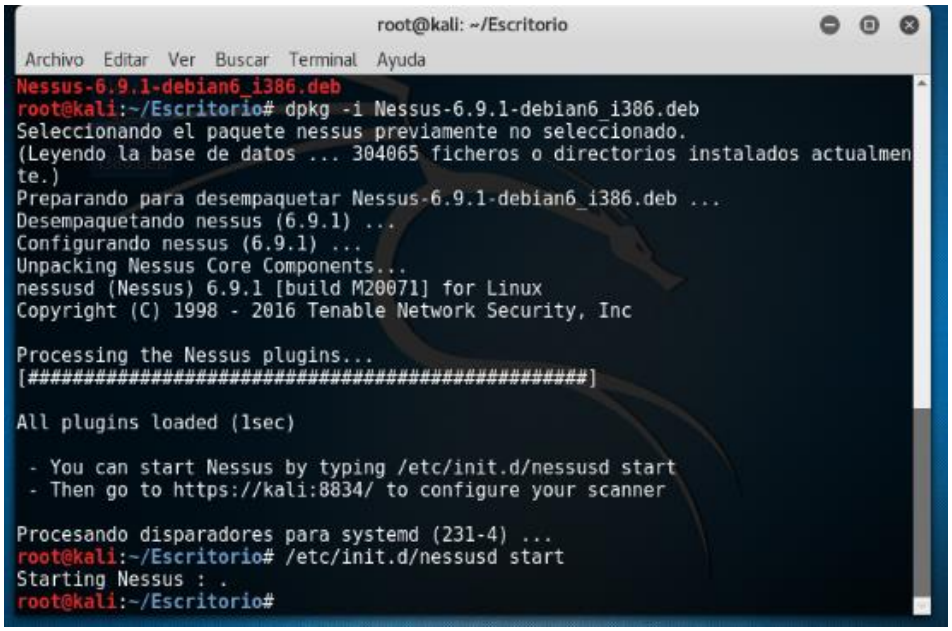
Figura 60. Escaneo con zenmap sobre maquina con Oracle



Fuente: el autor.

6.2.1.2 NESSUS sobre Oracle. Se instaló y configuró Nessus. Se descargó de la página oficial http://www.tenable.com/products/nessus/select-your-operating-system#tos y se instala desde la consola de kali, luego se inicia el servicio de Nessus como lo indica la figura 61 donde se muestra la instalación de la herramienta.

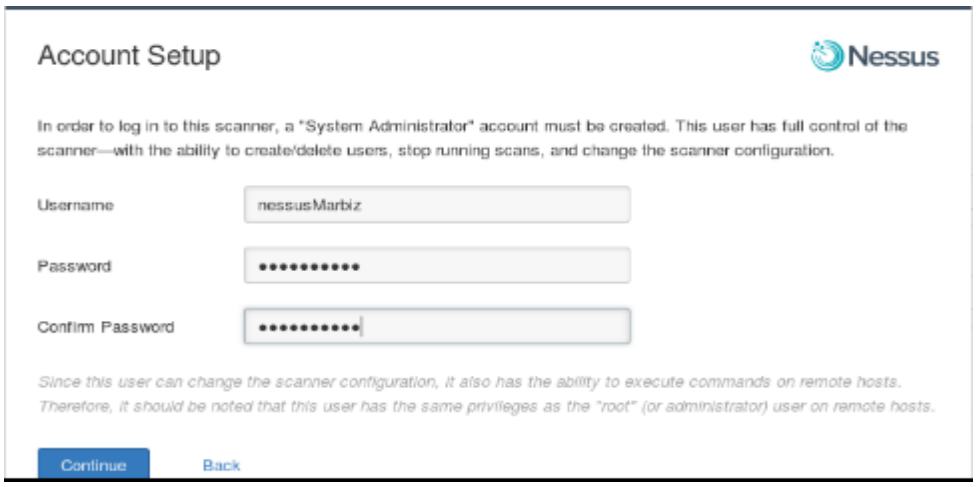
Figura 61. Instalación e inicio de servicio de Nessus



Fuente: el autor

En la figura 62 se muestra la creación del usuario nessusMarbiz, se registra el código de activación y se inicia Nessus.

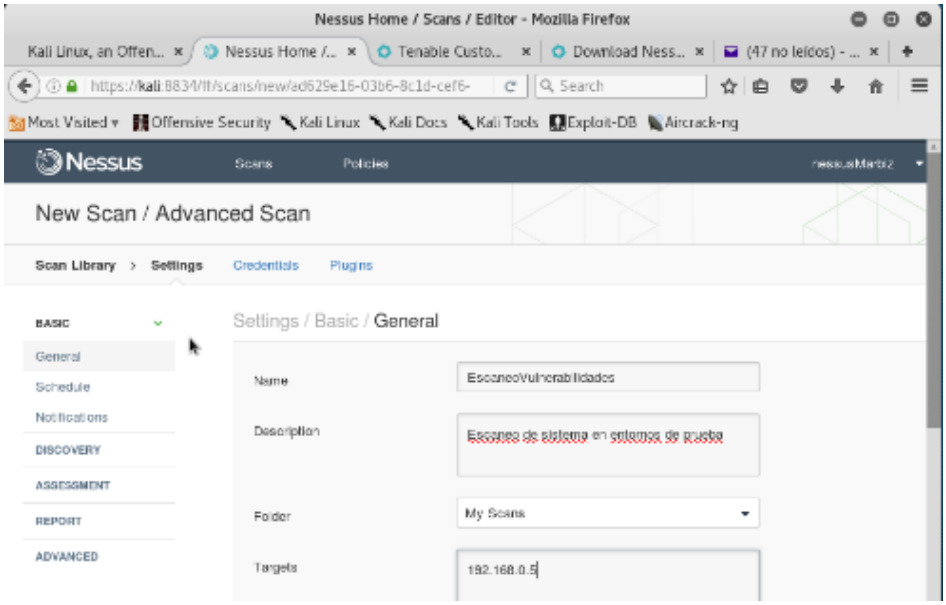
Figura 62. Instalación e inicio de servicio de Nessus



Fuente: el autor

En la figura 63 se muestra la configuración del escaneo con Nessus, el cual va dirigido hacia la máquina donde se encuentra instalado Oracle y donde se detectó el puerto 1521 por defecto de Oracle.

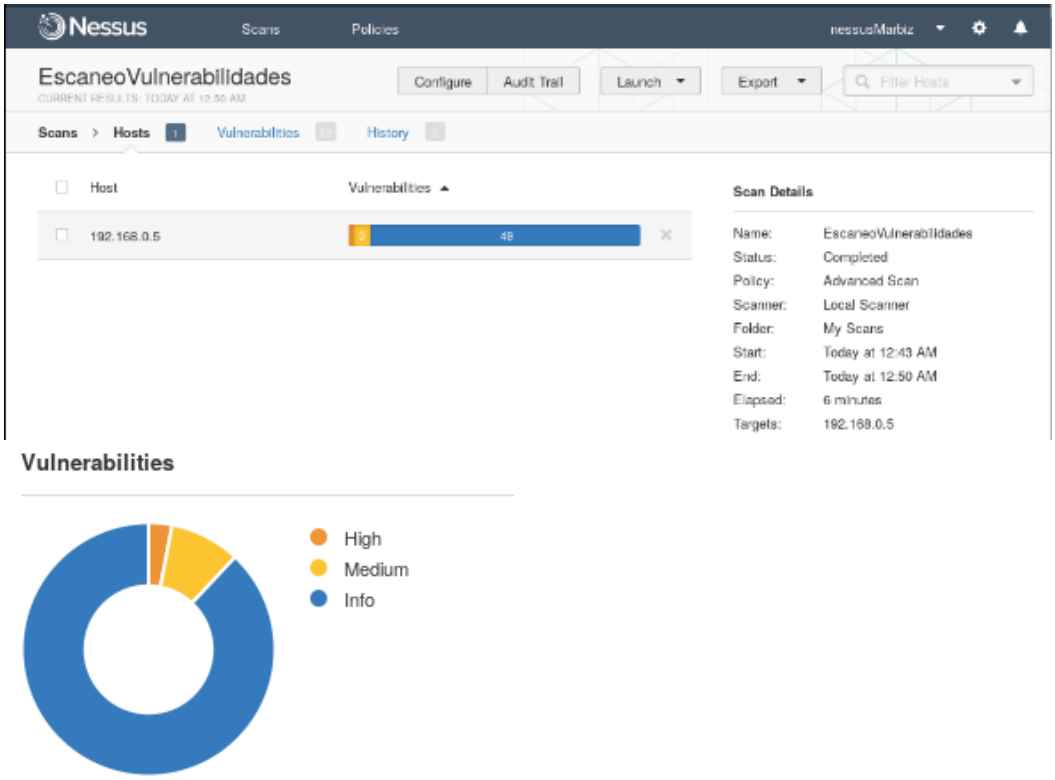
Figura 63. Preparación Nessus para escaneo



Fuente: el autor.

Se efectúa el análisis a la maquina con la base de datos de oracle arrojando como resultado 64 resultados de los cuales se detectan 1 vulnerabilidad alta, 3 vulnerabilidades medias y 49 a manera de información como se muestra en la figura 56.

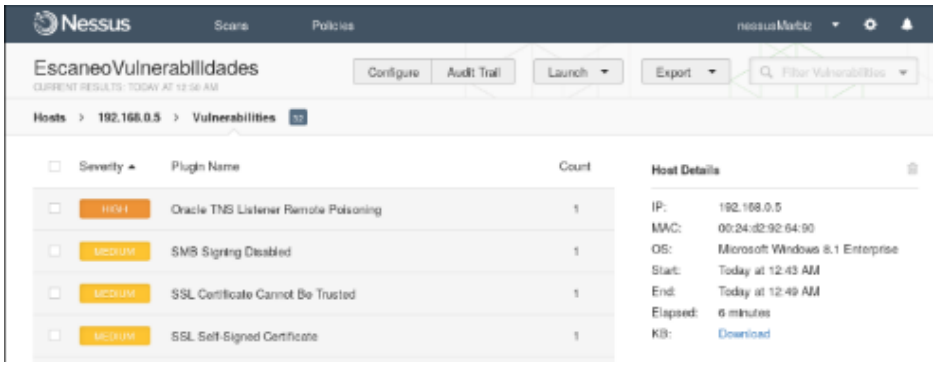
Figura 64. Preparación Nessus para escaneo



Fuente: el autor.

En la figura 65 se muestra el resumen de las vulnerabilidades encontradas y catalogadas por Nessus como altas y medias entre las cuales se destaca el servicio web de Oracle a través del puerto tcp/8080. Desde Nessus se pueden generar reportes en pdf, Excel, html con el resumen de los resultados obtenidos y con las posibles soluciones a implementar.

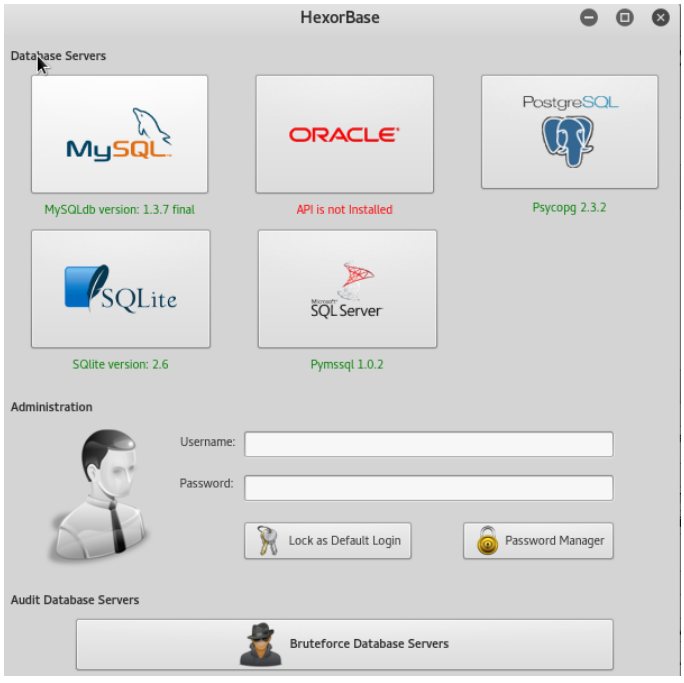
Figura 65. Resumen Vulnerabilidades detectadas con Nessus



Fuente: el autor

6.2.1.3 HEXORBASE sobre Oracle. Se utiliza nuevamente la herramienta Hexorbase de Kali Linux para realizar ataque de fuerza bruta a Oracle, en la figura 66 se observa la pantalla de inicio donde indica que la API de Oracle no se encuentra instalada, razón por la cual no se utilizó esta herramienta.

Figura 66. No disponible Hexorbase sobre Oracle

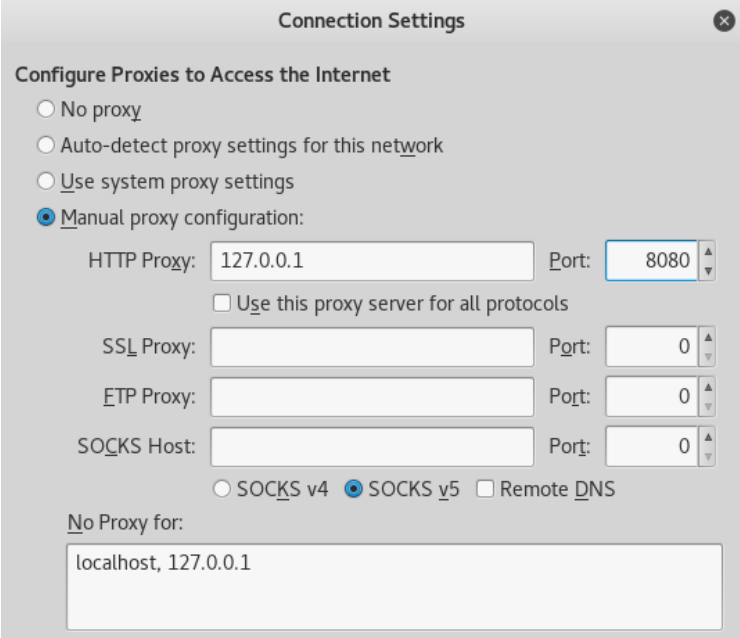


Fuente: el autor.

6.2.2 Prueba de Inyección SQL sobre Oracle. Una vez identificado el acceso al puerto de comunicación con MySQL, así como el resultado positivo de vulnerabilidades con Nessus, se utilizaron cuatro herramientas para hacer las pruebas de ataque de inyección SQL: Owasp Zap, SQLmap, Paros y Jsql Injection obteniendo resultados positivos con excepción de Jsql Injection que después de varios intentos no mostró ningún resultado.

6.2.2.1 OWASP ZAP sobre Oracle. Teniendo en cuenta las vulnerabilidades encontradas referentes al servicio web de Oracle ejecutándose, se utilizó el ataque pasivo de Owaspzap para Inyección SQL, para eso se configuró el navegador para conectarse mediante proxy como se muestra en la figura 67.

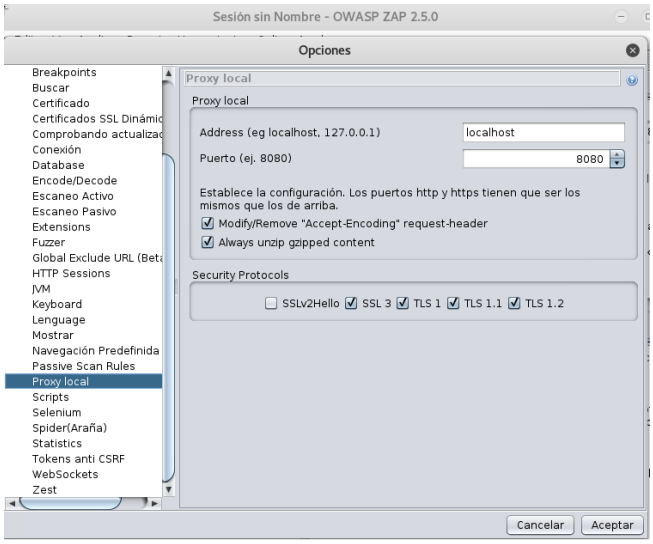
Figura 67. Configurar navegador con conexión a proxy sobre Oracle



Fuente: el autor.

El owaszap se configuró para funcionar como proxy local, en opciones de configuración como se muestra en la figura 68.

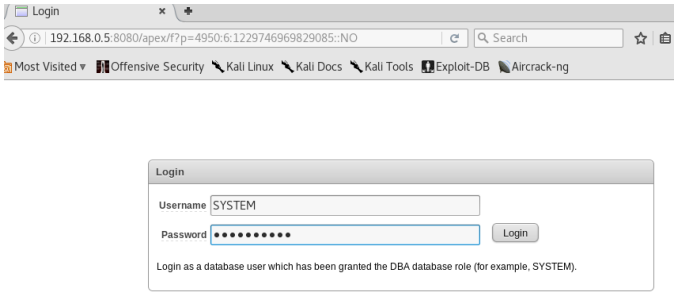
Figura 68. Ejecución de OWASP ZAP sobre Oracle



Fuente: el autor.

Al intentar ingresar al entorno web de Oracle se solicita usuario y contraseña, estos datos son ingresados con el fin de detectar los parámetros post o get del formulario web como se muestra en la figura 69.

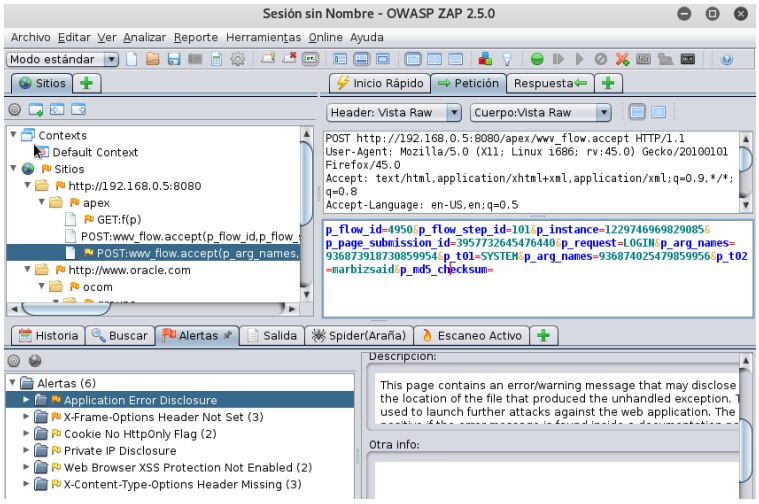
Figura 69. Ingreso a entorno web de Oracle



Fuente: el autor.

Una vez ingresados los datos al sitio web, owaszap detecta las posibles vulnerabilidades y muestra el resumen en el árbol de consulta y el resumen de las alertas encontradas agrupadas por nombre. En la figura 70 se muestra una posible vulnerabilidad por inyección SQL mediante el método POST, en donde se están recibiendo los argumentos de respuesta del servidor para el usuario y contraseña.

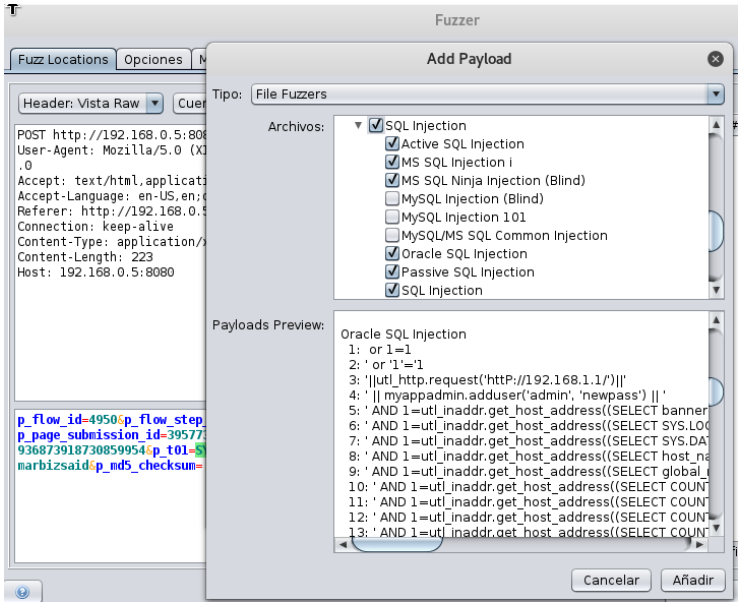
Figura 70. Resultado obtenido con OWASP ZAP sobre Oracle



Fuente: el autor.

Se realiza la prueba de inyección SQL utilizando los parámetros de usuario y contraseña, configurando el payload por defecto de Owaspzap como se indica en la figura 71.

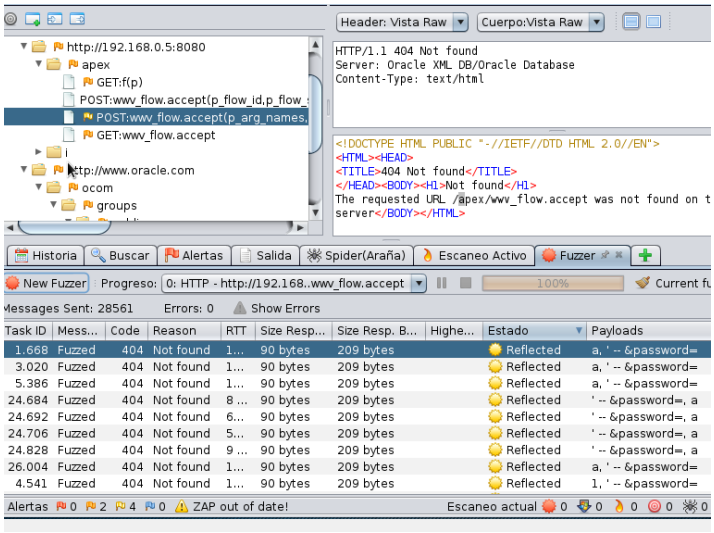
Figura 71. Payload de inyección SQL de OWASP ZAP sobre Oracle



Fuente: el autor.

Se ejecutó el ataque de manera satisfactoria y se obtuvo como resultado 28.561 sentencias ejecutadas de las cuales fueron positivas 13 como se indica en la figura 72.

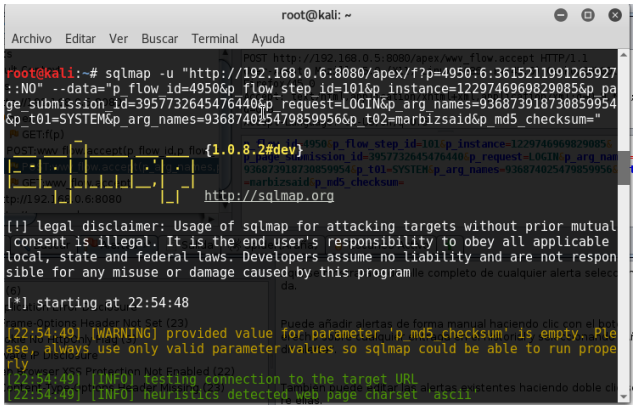
Figura 72. Resultados de ataque con Inyección SQL sobre Oracle



Fuente: el autor.

6.2.2.2 SQLMAP sobre Oracle. Con el fin de confirmar y aprovechar las vulnerabilidades de inyección SQL detectadas, al igual que se hizo en el entorno de MySQL, se utilizó SQLMAP. En la figura 73 se muestra la configuración para realizar la inyección SQL a través de SQLMAP, además se refleja el resultado de la no posibilidad de realizar una inyección, aunque se atacó mediante la inserción de los parámetros detectados con owaspzap, la base de datos no permitió la explotación de la vulnerabilidad.

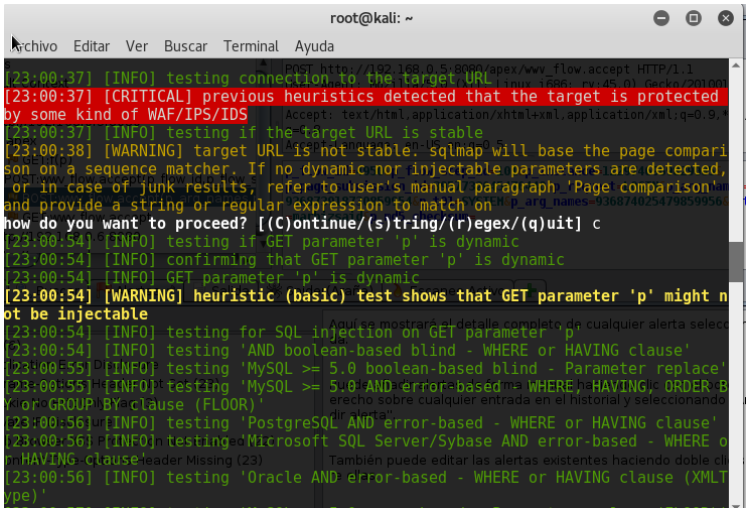
Figura 73. Ejecución de SQLMAP sobre Oracle



Fuente: el autor.

Una vez confirmado los parámetros insertados como no vulnerables, SQLMAP realiza otras sentencias con el fin de recuperar más datos sobre la base de datos Oracle como se muestra en la figura 74.

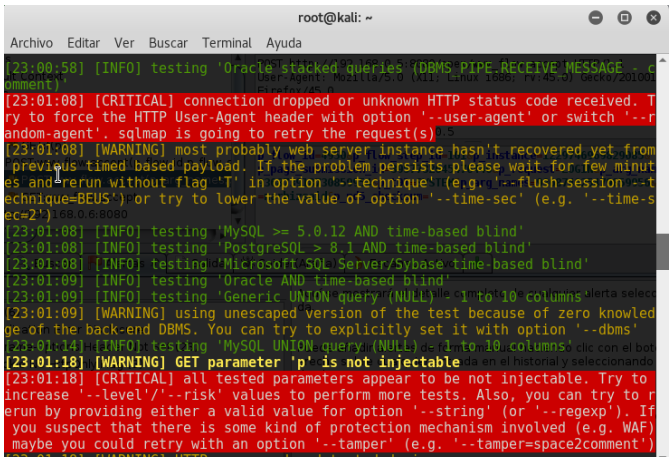
Figura 74. Ejecución de sentencias de inyección SQL con SQLMAP sobre Oracle



Fuente: el autor.

Finalmente, en la figura 75 se muestran los resultados de la inyección realizada indicando que los parámetros insertados no son inyectables, terminando así la sesión de SQLMAP sobre Oracle.

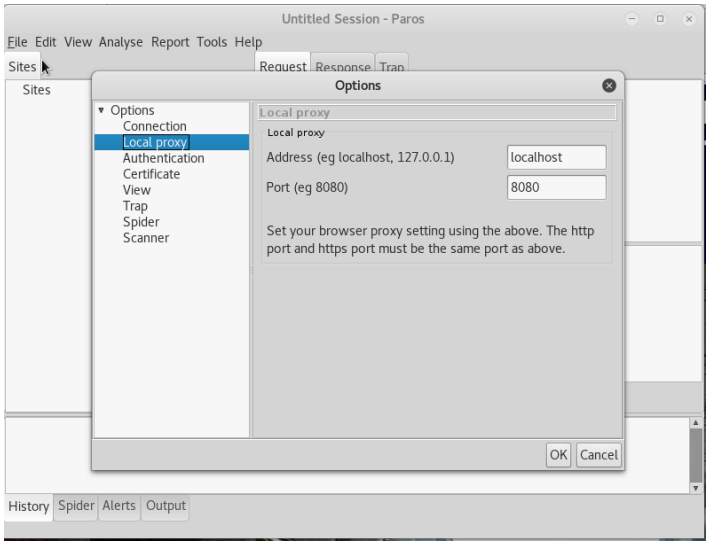
Figura 75. Resultados de la ejecución de inyección SQL con SQLMAP sobre Oracle



Fuente: el autor.

6.2.2.3 Paros sobre Oracle. Se utilizó la herramienta Paros también sobre Oracle, en la figura 76 se muestra la configuración como proxy de la herramienta.

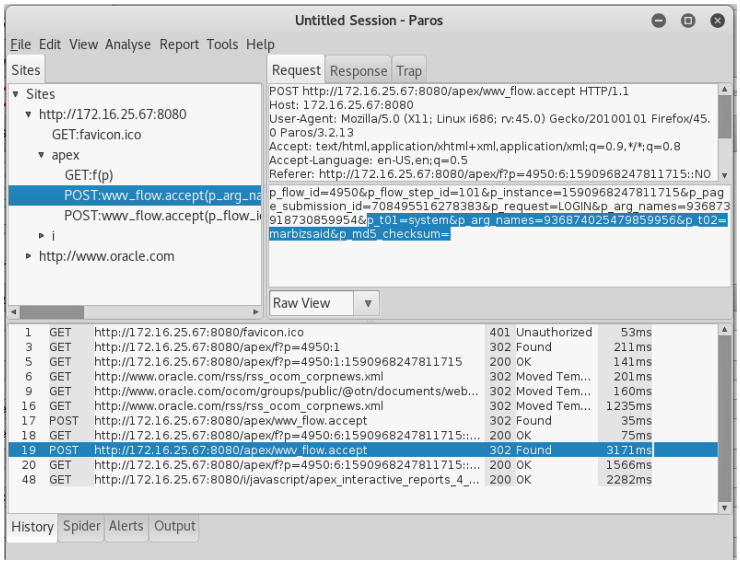
Figura 76. Configuración de Paros sobre Oracle



Fuente: el autor.

Una vez realizado el análisis se observa que tiene vulnerabilidad para ataque de inyección SQL sobre los parámetros de ingreso de sesión al sitio web, mediante la petición POST en los parámetros del formulario de captura de datos del usuario como se muestra en la figura 77.

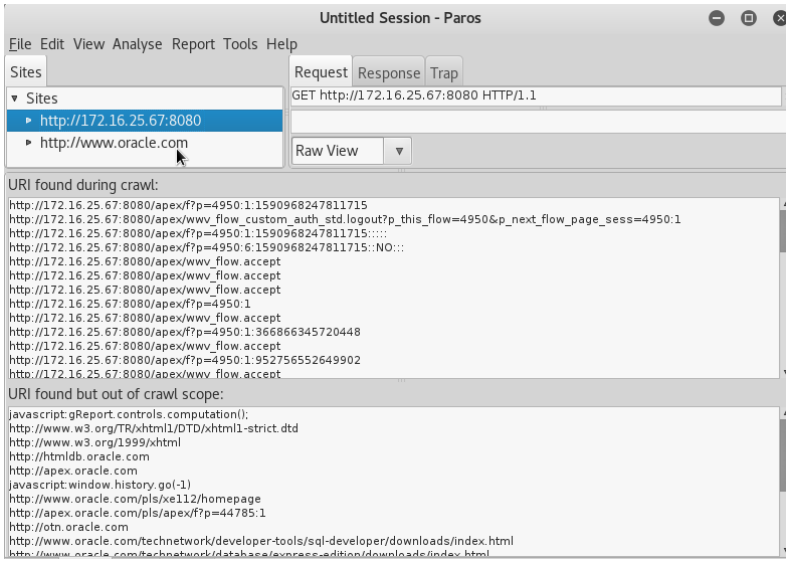
Figura 77. Ejecución de Paros sobre Oracle



Fuente: el autor.

La herramienta Paros también contiene una utilidad llamada spider que permitió visualizar otras páginas del sitio susceptibles a ataque por inyección SQL, en la figura 78 se muestra la opción spider reflejando los resultados.

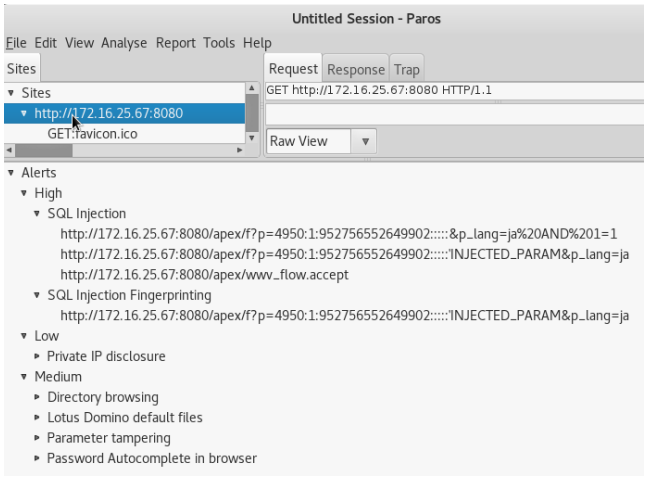
Figura 78. Resultados ejecución Spider de Paros sobre Oracle



Fuente: el autor.

En la figura 79 se muestra la pestaña alerta de la herramienta Paros, allí se pudo identificar de manera agrupada por nivel alto, medio y bajo las diferentes vulnerabilidades que tiene el sitio en especial la de inyección SQL.

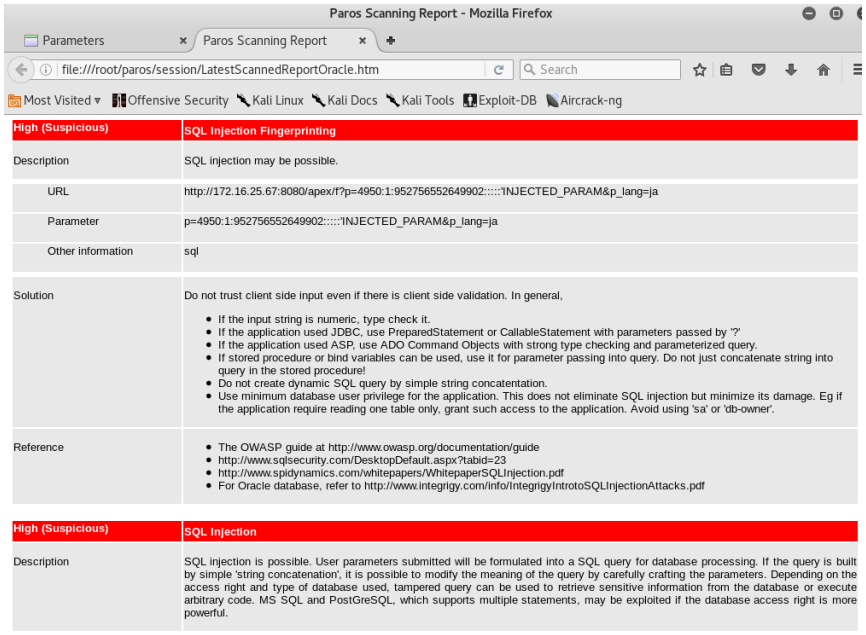
Figura 79. Resultados Inyección SQL de Paros sobre Oracle



Fuente: el autor.

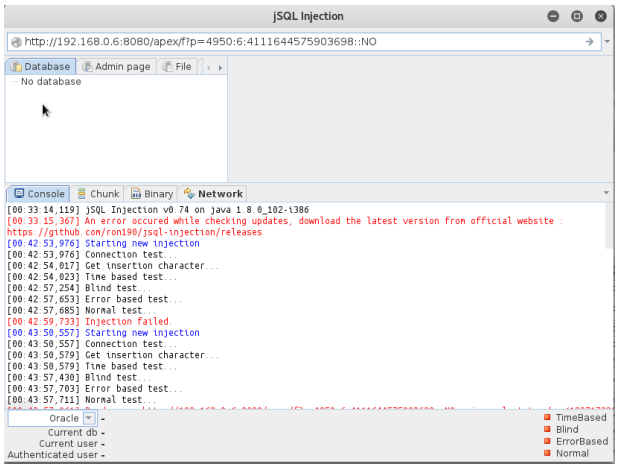
En la figura 80 se muestra la opción que tiene Paros para generar reportes sobre el escaneo realizado y sobre la inyección SQL que se pudo realizar sobre Oracle.

Figura 80. Reporte de Paros sobre Oracle



6.2.2.4 JSQL INJECTION sobre Oracle. Se utilizó también la herramienta de KaliLinux jSQL Injection para realizar otro ataque a la base de datos de Oracle. En la figura 81 se observa el panel principal de la herramienta, en la parte superior se ingresa la dirección del sitio web a atacar, en la parte inferior se selecciona el tipo de base de datos, para este caso Oracle y se efectúa el proceso de inyección.

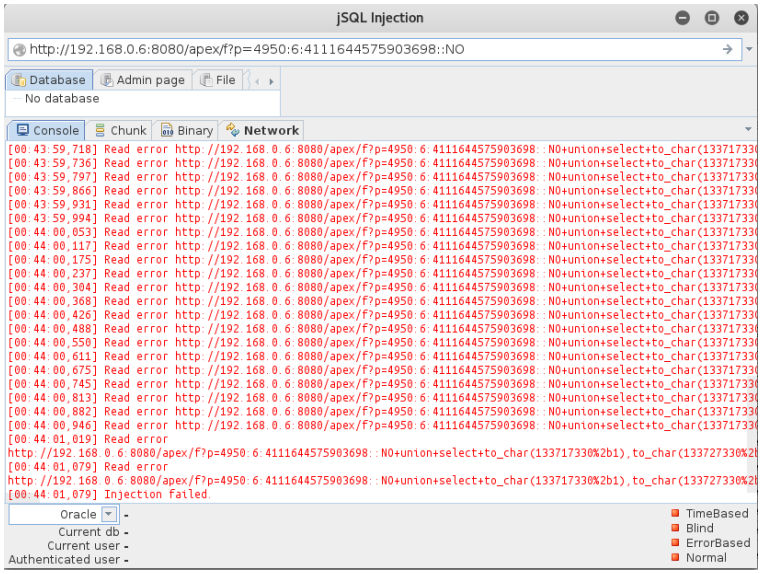
Figura 81. Ejecución de jSQL Injection sobre Oracle



Fuente: el autor.

Después de varios intentos con la herramienta, y los posibles comodines para inyectar código SQL la inyección falla como se muestra en la figura 82.

Figura 82. Resultado de ejecución de jSQL Injection sobre Oracle



Fuente: el autor.

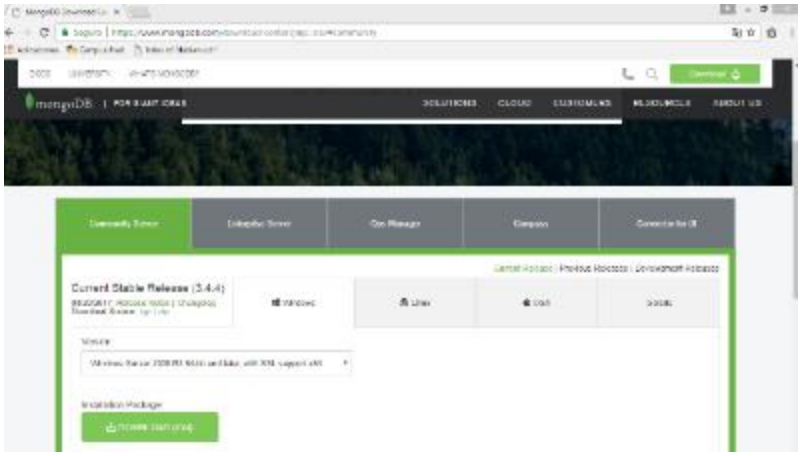
6.3 ENTORNO DE PRUEBA BASE DE DATOS NOSQL MONGODB

Mongodb es una base de datos ágil que permite a los esquemas cambiar rápidamente cuando las aplicaciones evolucionan, proporcionando siempre la funcionalidad que los desarrolladores esperan de las bases de datos tradicionales, tales como índices secundarios, un lenguaje completo de búsquedas y consistencia estricta, ha sido creado para brindar escalabilidad, rendimiento y gran disponibilidad, escalando de una implantación de servidor único a grandes arquitecturas complejas de centros multidatos. MongoDB brinda un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria (in-memory). La replicación nativa de MongoDB y la tolerancia a fallos automática ofrece fiabilidad a nivel empresarial y flexibilidad operativa. (https://www.mongodb.com/es).

MongoDB es un motor de base de datos NoSQL de código abierto, escrito en C++, se encuentra orientado al almacenamiento de documentos y es multiplataforma. Algunas empresas que usan esta base de datos son CISCO, MTV, The New York Times, Forbes, entre otras.

Se descargó la última versión para Windows de la página oficial https://www.mongodb.com/download-center?jmp=nav#community como se indica en la figura 83.

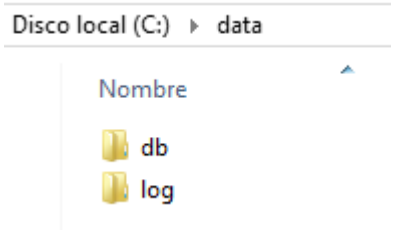
Figura 83. Descarga mongodb de página oficial



Fuente: el autor

El asistente de instalación guía el proceso, una vez finaliza se crea en disco c una carpeta llamada data con dos subcarpetas llamadas db y log respectivamente, son fundamentales para guardar las bases de datos y los eventos que ocurren en mongodb, como se muestra en la figura 84.

Figura 84. Carpetas para base de datos mongodb

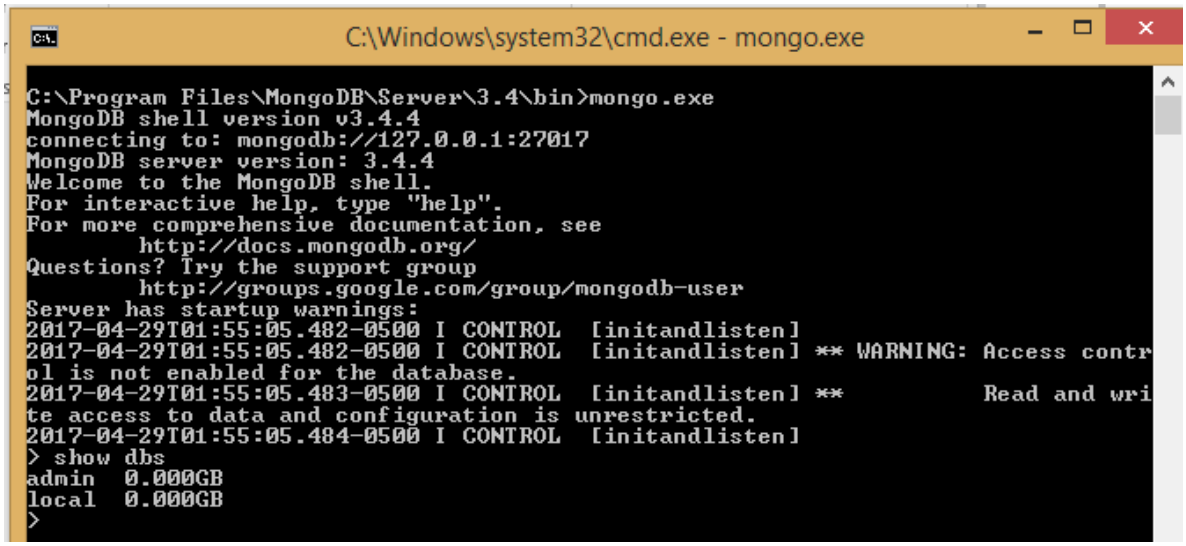


Fuente: el autor

Se ingresa a los archivos de instalación en el disco c que por lo general tendrá la siguiente ruta C:\Program Files\MongoDB\Server\3.4\bin, para ejecutar mongo.exe y mongod.exe, se ingresa por consola y se ejecutan el archivo mongo.exe como lo indica la figura 85 iniciando mongo versión 3.4.4 conectando de manera local por el puerto 27017, se puede apreciar las bases de datos por defecto admin y local, sin

embargo en estas bases no hay nada, es necesario crear las bases de datos e ingresar las colecciones de datos que se necesiten dependiendo la necesidad o proyecto a implementar y sobre todo implementar la autenticación porque mongodb no tiene opciones de seguridad cuando se instala.

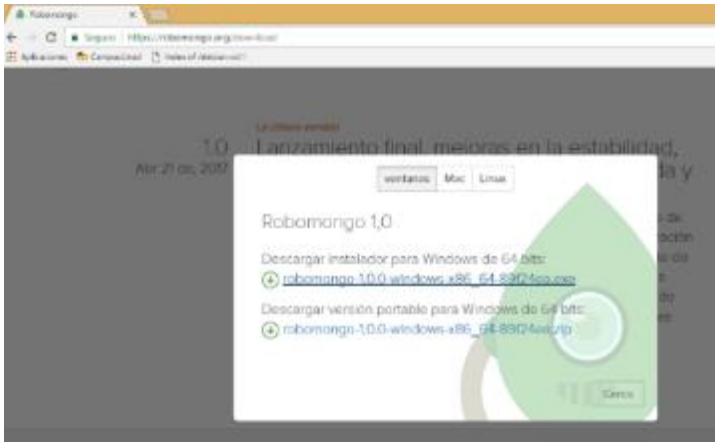
Figura 85. Ejecución mongodb



Fuente: el autor

Una forma de realizar la administración de mongodb es a través del Shell o línea de comandos, o también se tienen varias opciones de interfaz gráfica. En este caso se utilizó Robomongo que es una herramienta multiplataforma con la que se puede administrar gráficamente las bases de datos creadas en mongo como se muestra en la figura 86.

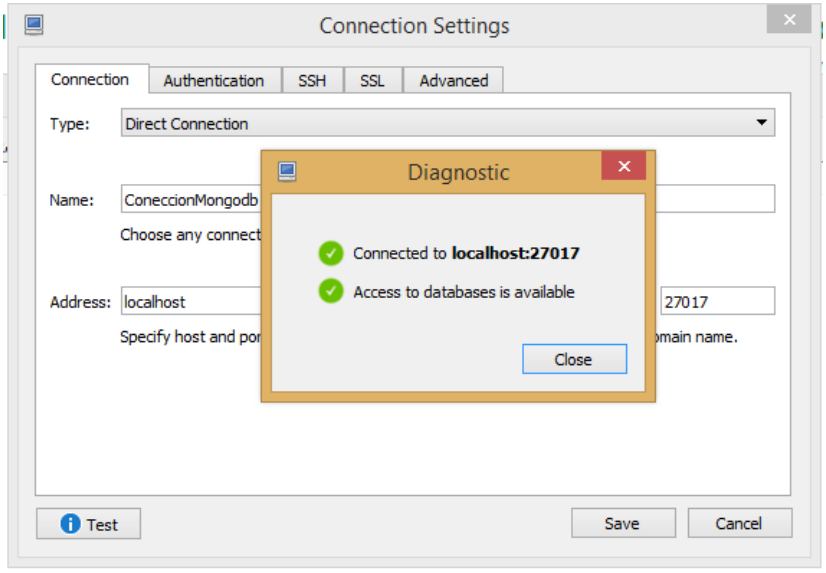
Figura 86. Ejecución mongodb



Fuente: el autor.

Se instala Robomongo, se ejecuta la aplicación se crea una conexión nueva y se realiza una prueba de conexión como se muestra en la figura 87 la conexión correcta con la base de datos.

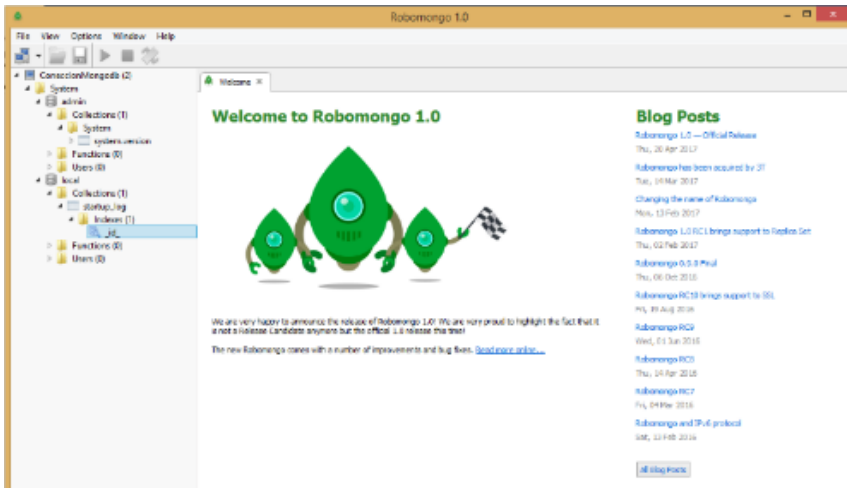
Figura 87. Conectado a la base de datos



Fuente: el autor.

En la figura 88 se muestra el acceso a Robomongo donde se puede apreciar las bases de datos admin y local, pero al desplegar las opciones de exploración está completamente vacía.

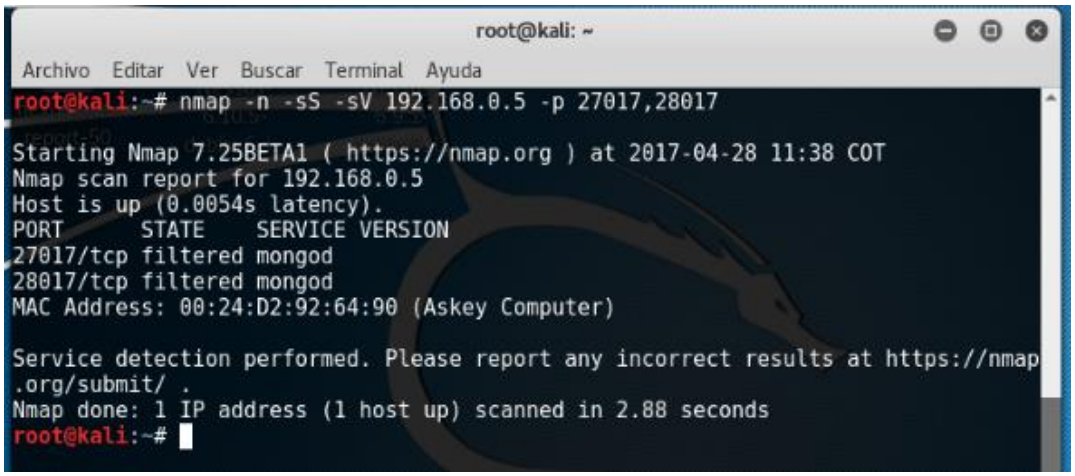
Figura 88. Base de datos detectada y conectada con Robomongo



Fuente: el autor

En la instalación de mongodb no se tiene predeterminado ningún usuario ni aplicado ningún rol con privilegios, esta tarea debe ser realizada antes de iniciar a implementar un proyecto. En la figura 89 se utilizó nmap para penetrar filtrando los puertos por defecto detectando el servicio activo.

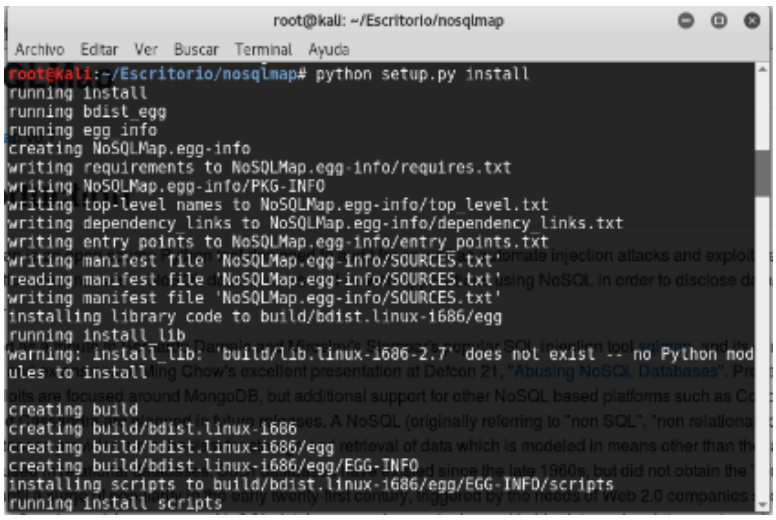
Figura 89. Prueba de escaneo a mongodb con nmap



Fuente: el autor

6.3.1 Inyección NoSQL utilizando NoSQLMap. Se ingresa a https://github.com/tcstool/nosqlmap y se descarga el paquete de instalación, se ejecuta la instalación con el comando python setup.py install como se muestra en la figura 90.

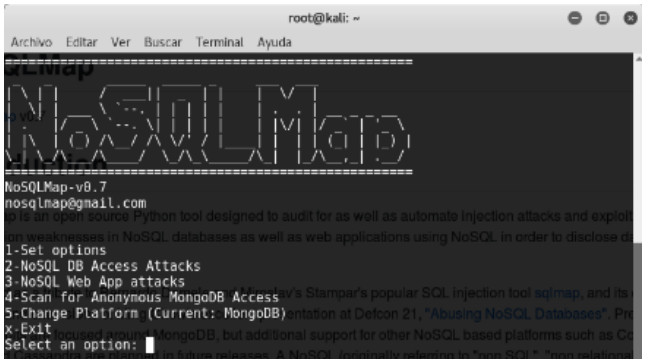
Figura 90. Instalando NoSQLMap



Fuente: el autor.

Una vez instalado NoSQLMap se puede ejecutar mediante el comando nosqlmap.py y ejecutará la ventana con las opciones para realizar la inyección NoSQL sobre bases de datos NoSql en este caso hacia mongodb. En la figura 91 se muestra la pantalla principal de Kali con la herramienta activa de NoSQLMap.

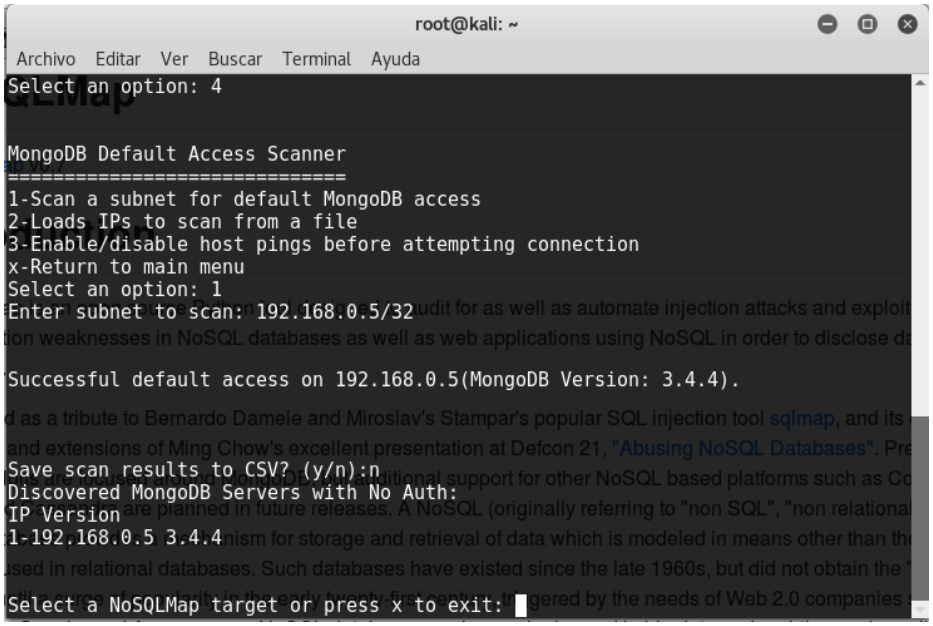
Figura 91. Inyección NoSQL con NoSQLMap a MONGODB



Fuente: el autor.

Para el primer ataque se selecciona la opción 4 para escanear la maquina a atacar, se puede escanear toda la red, para el ejercicio se coloca únicamente la ip de la víctima como se muestra en la figura 92 allí se observa como resultado que encuentra la maquina con mongodb y la versión que para este caso es la 3.4.4.

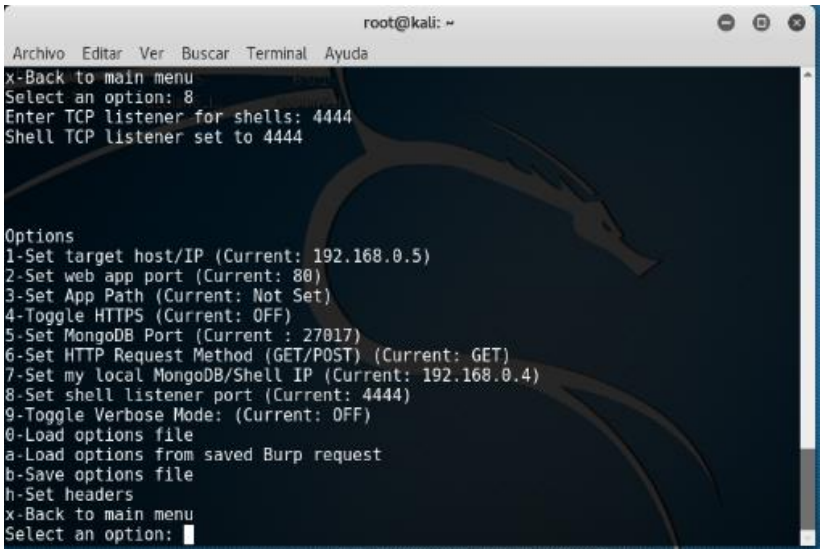
Figura 92. Descubriendo la base de datos MongoDB



Fuente: el autor.

Luego se retorna a las opciones y se configura la ip de la maquina a atacar y la ip del atacante y puerto por defecto que puede ser el 4444, como se muestra en la figura 93.

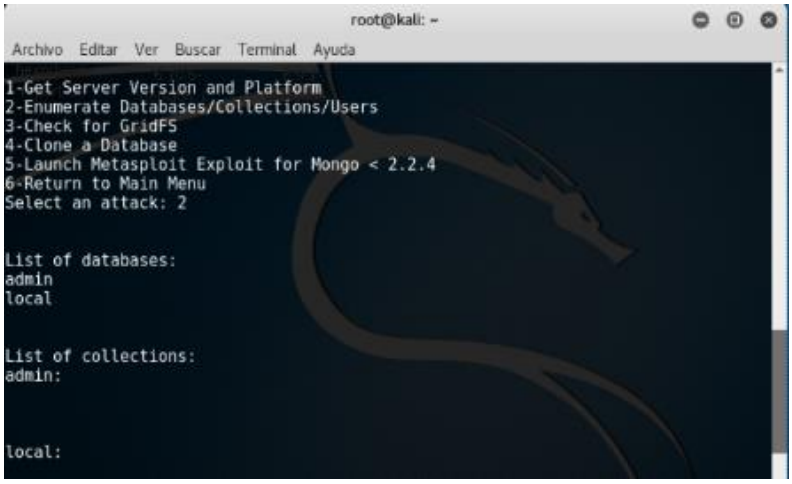
Figura 93. Configuración IP y puerto



Fuente: el autor.

Una vez configurada la maquina a atacar se ejecuta el ataque de inyección con la opción 2 y se observan los resultados, como que pudo ingresar sin credenciales y las posibles opciones de ataque. En la figura 94 se muestra con la opción 2 se pueden observar las bases de datos en mongodb.

Figura 94. Bases de datos encontradas con NoSQLMap



Fuente: el autor.

Con la opción 4 se puede clonar las bases de datos de mongodb como se muestra en la figura 95.

Figura 95. Clonación de base de datos

Fuente: el autor

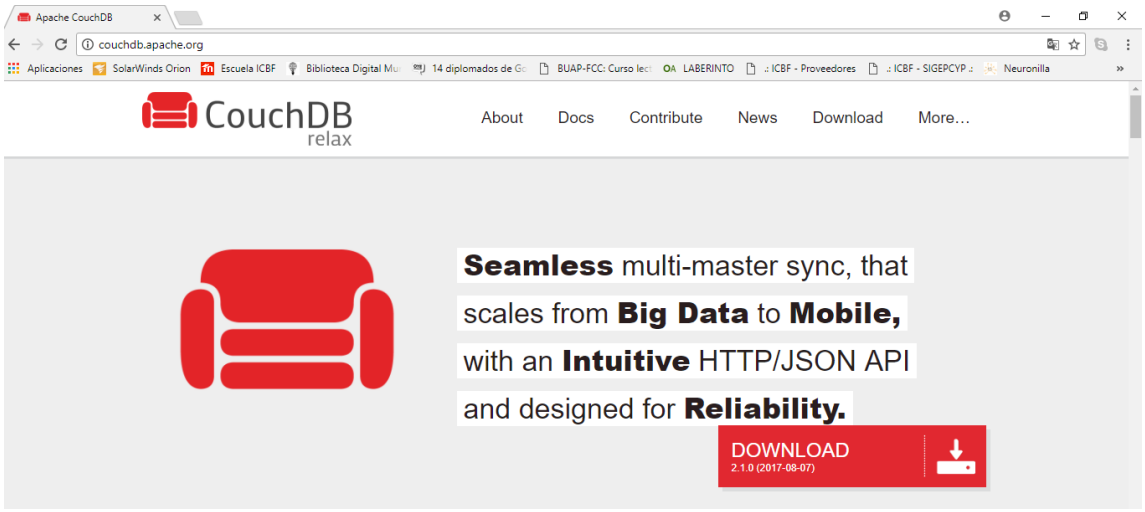
6.4 ENTORNO DE PRUEBA BASE DE DATOS NOSQL COUCHDB

CouchDB es una herramienta de gestión de bases de datos no relacionales de código abierto. Una de las principales características de este gestor de bases de datos es la búsqueda de facilitar las actividades de la programación y administración, garantizando funciones adecuadas para el desarrollo de páginas WEB completas.

Además de la anterior característica, apache couchDB también es muy práctico a la hora de gestionar errores dentro del desarrollo; si se llegase a presentar un error en una consulta, couchDB no detiene totalmente el servidor, se centra en el problema como tal y la misma herramienta brinda apoyo para resolución de dicho inconveniente. La herramienta presenta algunas restricciones en la programación frente a la escalabilidad de un sistema, esto está pensado para que una vez se presente un evento de este tipo, se haga de manera correcta evitando futuros inconvenientes para el programador.

Se descarga instalador desde página oficial http://couchdb.apache.org/ para el entorno de prueba se escogió la versión de Windows 64 como se muestra en la figura 96.

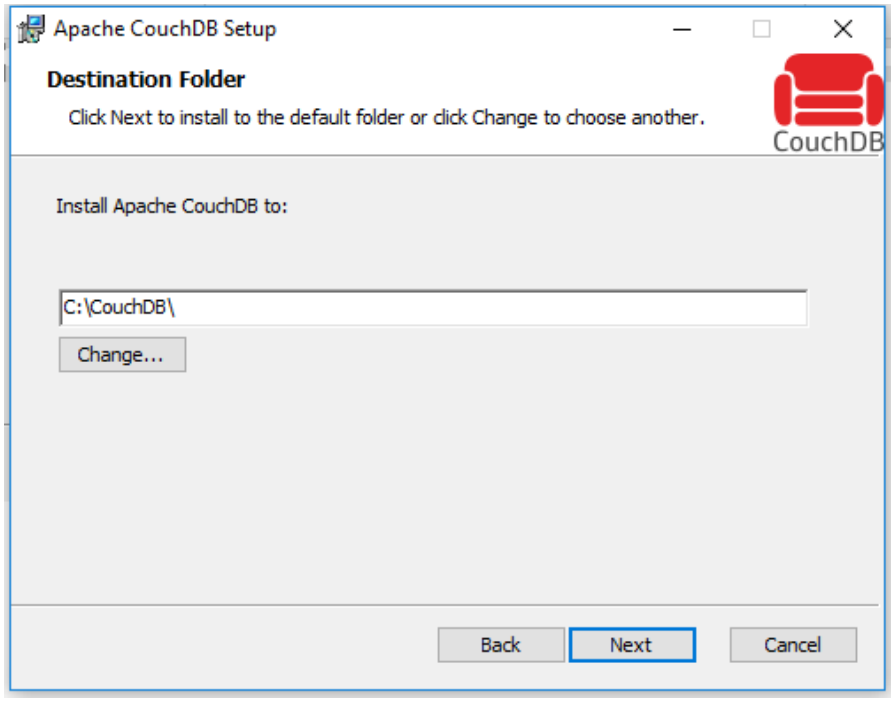
Figura 96. Descarga de apache CouchDB desde la página oficial



Fuente: el autor

Se inicia el asistente de la instalación aceptando los términos de licencia, hasta llegar a la ruta de creación de Couchdb que se deja por defecto en el disco C como se muestra en la figura 97.

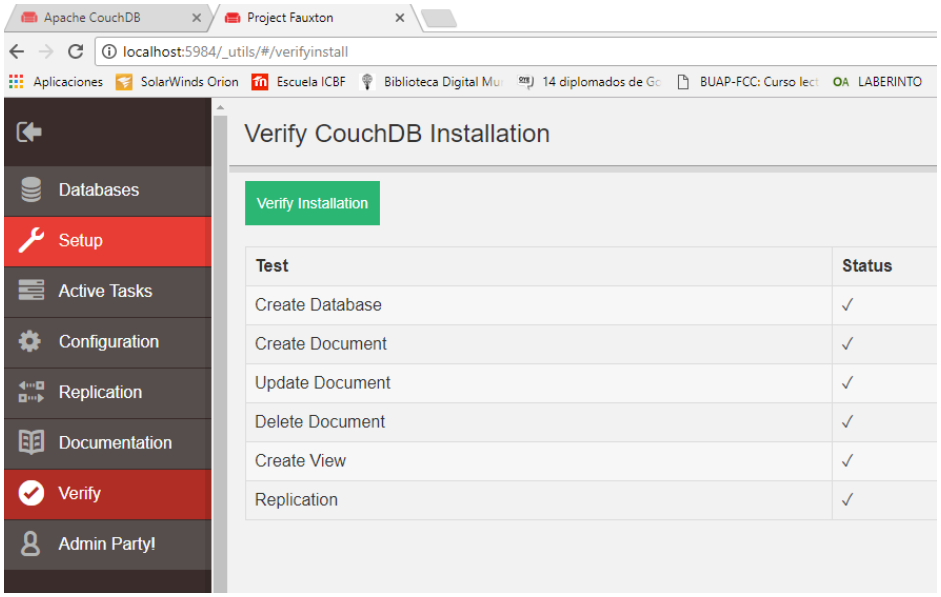
Figura 97. Inicio de instalación de CouchDB



Fuente: el autor

Después de la instalación, para ingresar al módulo de administración de las utilidades de Couchdb, se ingresa desde un navegador a la ruta http://localhost:5984/_utils/ como se indica en la figura 98.

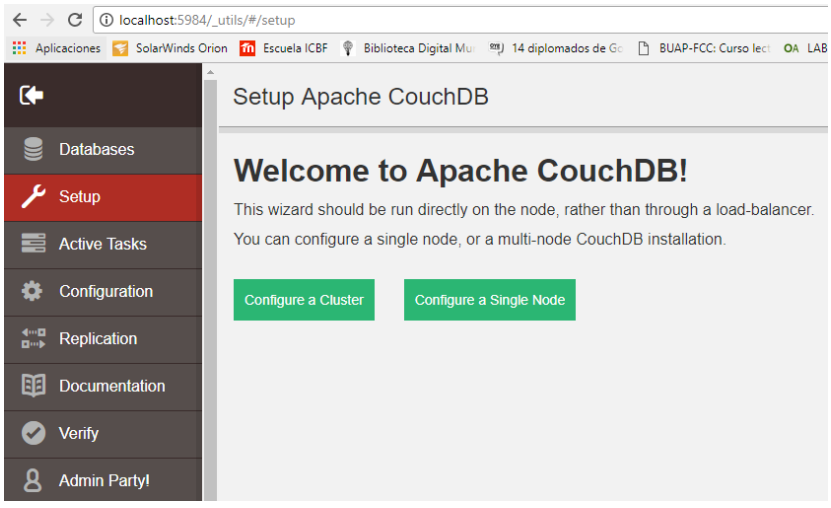
Figura 98. Módulo de administración de couchDB



Fuente: el autor

En el menú de opciones se ingresa al setup y se selecciona nodo simple para la creación de la base de datos así como se indica en la figura 99.

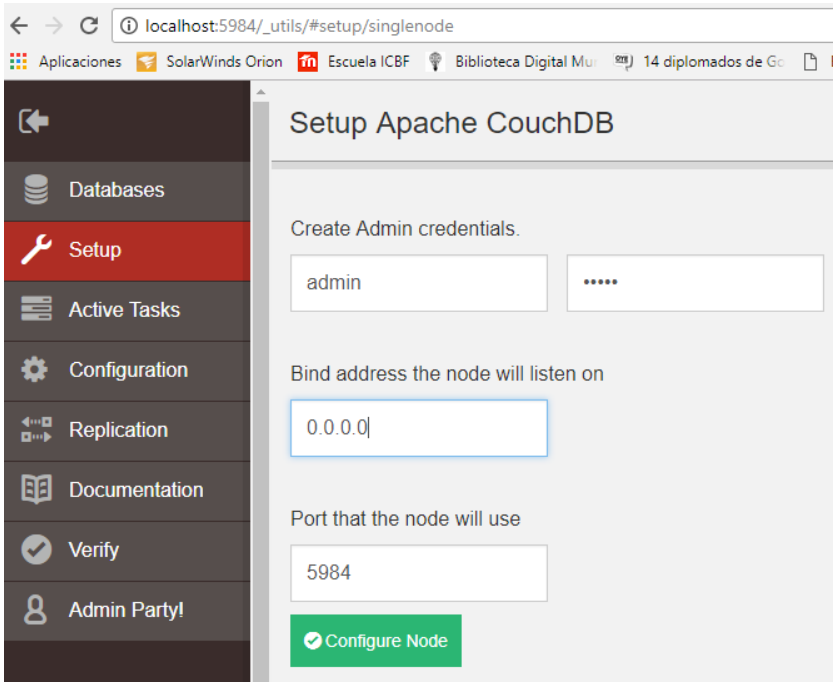
Figura 99. Área de setup de couchDB



Fuente: el autor

Seguidamente se configura el usuario, contraseña, dirección ip y puerto para comunicación de la base de datos, figura 100.

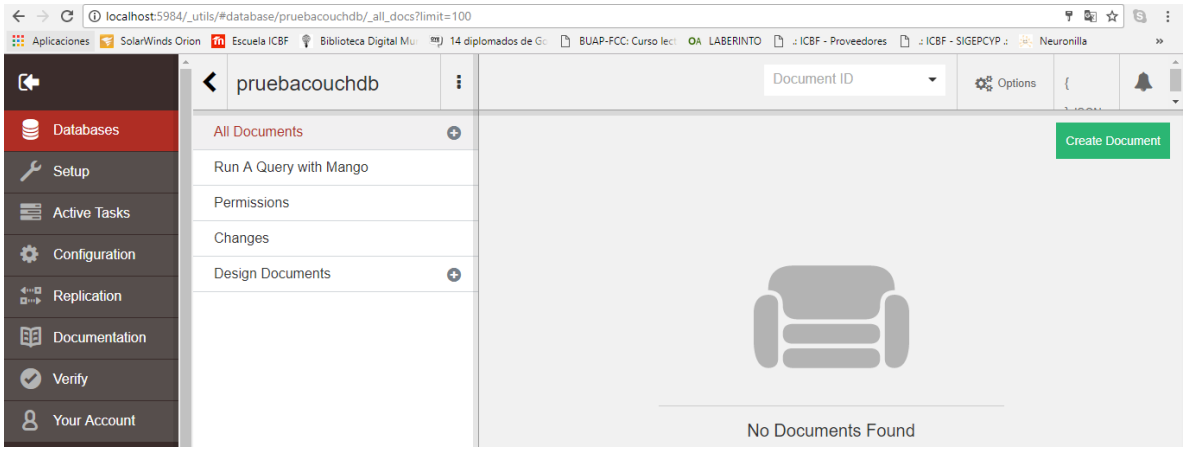
Figura 100. Configuración de nodo simple para couchDB



Fuente: el autor

Una vez configurado el nodo se crea la base de datos la cual se llamó pruebacouch, como se muestra en la figura 101.

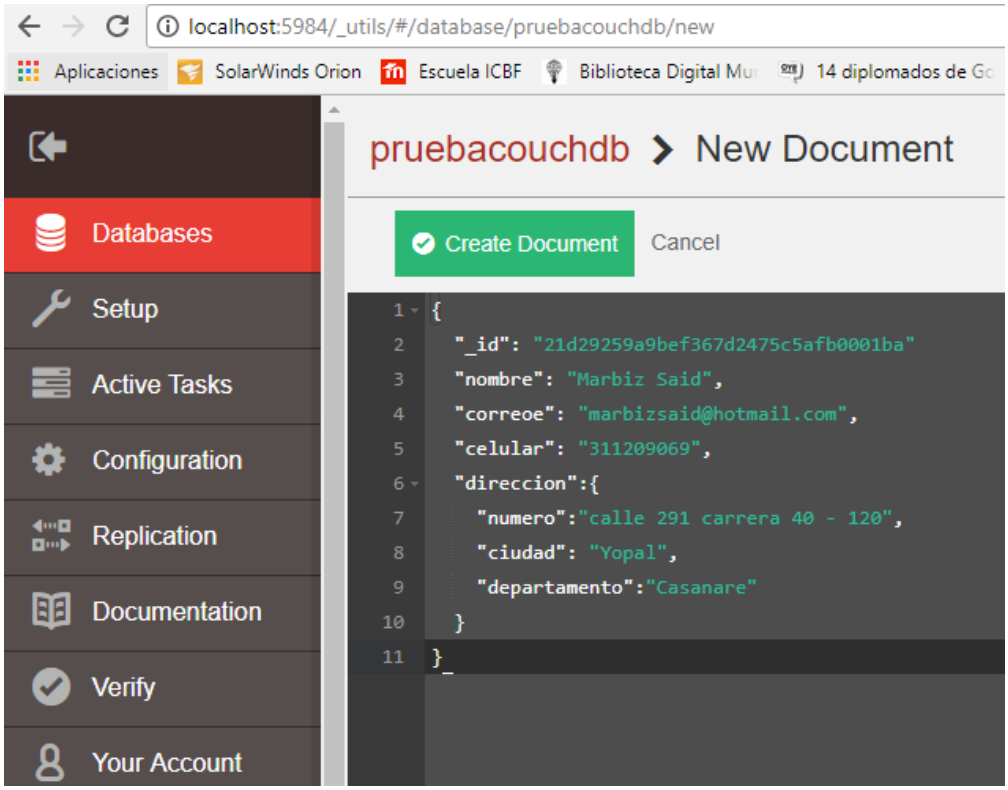
Figura 101. Creación de base de datos de prueba



Fuente: el autor

Se crea el documento para la base de datos pruebacouchdb, haciendo clic en el botón y generando el documento con estructura JSON como se muestra en figura 102.

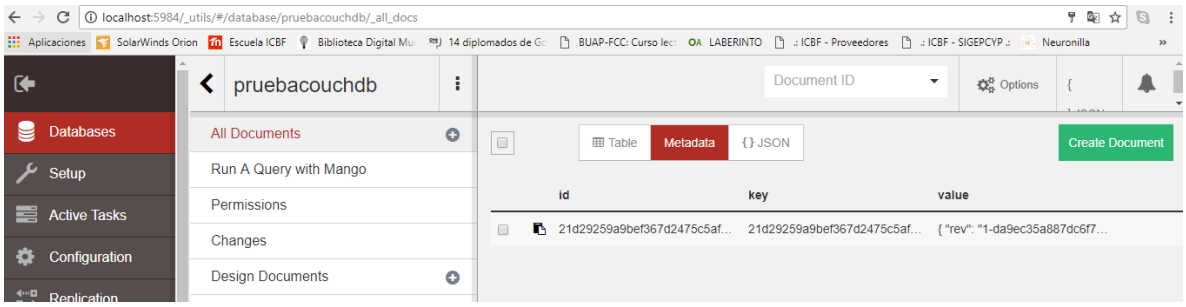
Figura 102. Creación de nuevo documento en CouchDB



Fuente: el autor

Una vez creado el documento apacheCouchDb muestra el id, la clave y el número de revisión del documento como se muestra en la figura 103.

Figura 103. Verificación de nuevo documento creado



Fuente: el autor

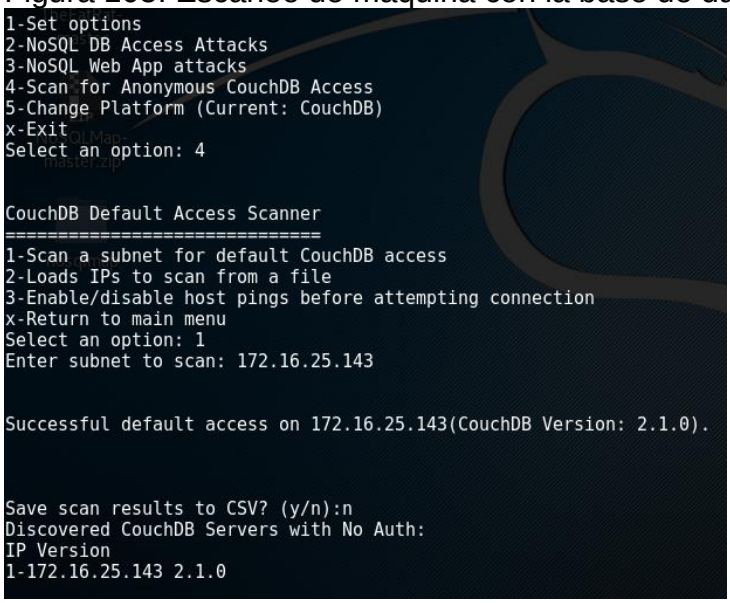
Teniendo ya instalada la herramienta de inyección NoSQLMap en Kali Linux, se ejecuta mediante el comando nosqlmap.py. una vez inicie NoSQLMap se visualiza la ventana con las opciones para realizar las pruebas de penetración sobre bases de datos NoSql, en este caso dirigido a CouchDB. En la figura 104 se muestra la pantalla principal de Kali Linux con la herramienta activa de NoSQLMap.

Figura 104. Inyección NoSQL con NoSQLMap a COUCHDB

Fuente: el autor

Previo al inicio del ataque se selecciona la opción 4 para escanear la maquina a atacar, para el presente se registra la ip de la víctima como se muestra en la figura 105 en el cual se puede observar como resultado la identificación de la maquina en donde apache CouchDB se encuentra instalado y su versión correspondiente.

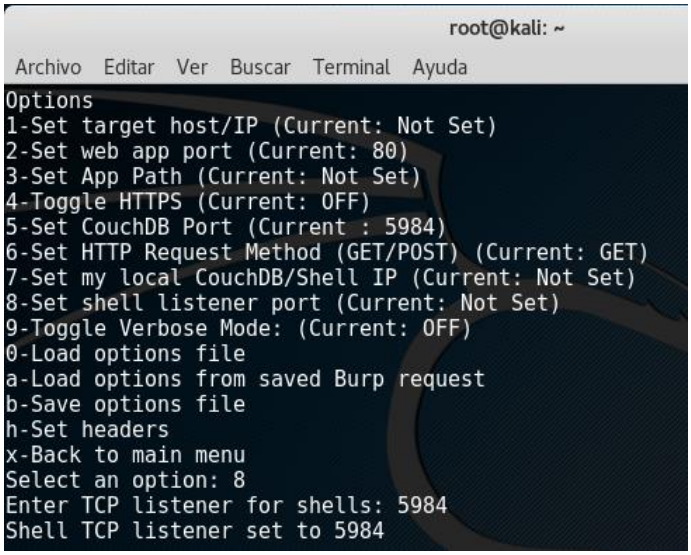
Figura 105. Escaneo de máquina con la base de datos



Fuente: el autor

Luego se retorna a las opciones y se configura en la opción 1 “Set target host/IP” como se muestra en la figura 106.

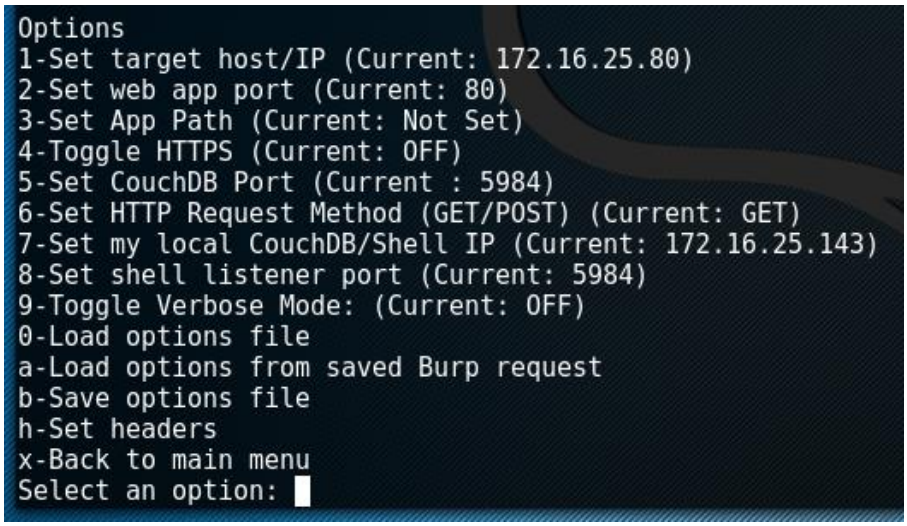
Figura 106. Opciones sin configuración



Fuente: el autor

Al configurar la ip de la maquina a atacar y la ip del atacante y puerto por defecto que puede ser el 5984, está listo para realizar el ataque como se indica en la figura 107.

Figura 107. Opciones luego de la configuración (atacante, víctima, puerto)



Fuente: el autor

Luego de la configuración con la opción “X” se regresa al menú de principal de opciones y se realiza el ataque con opción 2

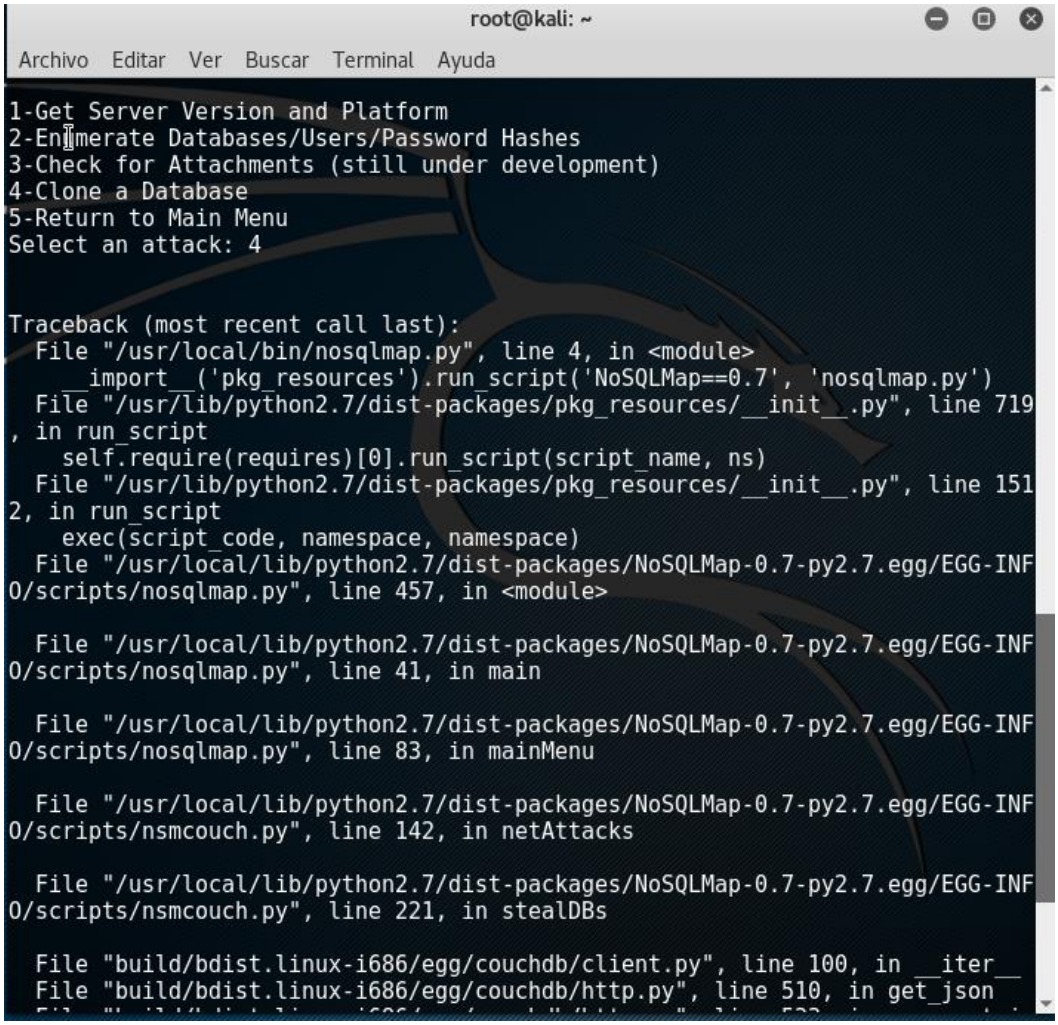
En el ejercicio con apache CouchDB se pudo evidenciar que el gestor de base de datos cuenta con mayor seguridad frente a la infiltración evadiendo el proceso de autenticación. Lo anterior se evidencia en las figuras 108 mostrando mensaje de error al intentar obtener el listado de base de datos de CouchDB.

Figura 108. Resultado de ejecución de ataque

Fuente: el autor

Se intentó otro ataque con la herramienta NoSqlMap y fue el de clonar la base de datos, contrario a MongoDB que dejó clonar la base de datos, en la figura 109 se muestran los errores de intento de ingreso a los paquetes de archivos de CouchDB.

Figura 109. Resultado clonación de base de datos



Fuente: el autor

7 RESULTADOS

Se identificaron algunos de los principales ataques (fuerza bruta para romper contraseñas, sniffing, spoofing, inyección SQL, inyección nosql) que ocurren en las bases de datos mediante la utilización de software que puede ser desarrollado y/o descargado de diversas fuentes y que a medida que la tecnología ha avanzado a través del tiempo los atacantes mejoran sus niveles de ataque. De igual manera se identificaron y probaron algunas de las herramientas de la suite de Kali Linux que permitieron el análisis de vulnerabilidades a las bases de datos, así como otras que permiten realizar una evaluación de los controles implementados con el fin de comprobar puntos críticos que deben ser protegidos para garantizar la seguridad en las bases de datos. Por otra parte, se logró establecer los mecanismos y actividades que convienen ser adoptados por las organizaciones para proporcionar los lineamientos, soporte, componentes y metodologías para la protección de los sistemas informáticos de nivel crítico para la operación misional especial lo relacionado con bases de datos SQL y NOSQL.

En la siguiente tabla se muestra el resumen de las vulnerabilidades encontradas para los motores de bases de datos SQL y los motores de bases de datos NoSQL.

Tabla 2. Resultados vulnerabilidades SQL y NoSQL

DBMS	**Vulnerabilidad**
MYSQL	Inyección SQL
	Sin uso de Cabecera X-Frame-Options, permite la exploración de Directorios, Sin manejo de declaración de errores, permite manipulación de Parámetros
	Sin uso de filtro Cabecera X-XSS-Protection para evitar ataques XSS en IE y Chrome, autocompletado del formulario activado, sin uso de Cabecera X-Content-Type-Options para evitar problemas de Seguridad, permite la inclusión de secuencias de comandos entre sitios (XSSI), las Cookies sin el conjunto de banderas HttpOnly
ORACLE	Inyección SQL
	Sin uso de la Cabecera X-Frame-Options, permite la manipulación de parámetro, sin manejo de la declaración de errores
	Sin filtro de la Cabecera X-XSS-Protection para evitar ataques XSS en IE y Chrome, sin uso de la Cabecera X-

DBMS	Vulnerabilidad
	Content-Type-Options para evitar problemas de Seguridad, las Cookies sin el conjunto de banderas HttpOnly
MONGODB	Inyección NoSQL, Sin asignación de usuario, sin rol y sin privilegios, extracción de datos con facilidad.
	Puerto abierto permite escaneo por red
COUCHDB	Ninguna
	No se permite Inyección NoSQL, restricciones de acceso al puerto, configuraciones de usuario y privilegios.

Fuente: El autor.

7.1 MECANISMOS O TÉCNICAS DE PROTECCIÓN PARA EVITAR LOS ATAQUES A BASES DE DATOS SQL Y NOSQL

Las fallas de inyección como SQL, OS, y LDAP, NOSQL ocurren cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta. Los datos del atacante pueden engañar al intérprete en ejecutar comandos no intencionados o acceder datos no autorizados[6].

7.1.1. MySQL. En la siguiente tabla se muestra el resumen de los resultados de las inyecciones SQL encontradas y otras vulnerabilidades en MYSQL y las medidas preventivas para evitarlo o para hacer del proceso de transacciones a una base de datos con menos probabilidades de explotar una vulnerabilidad.

Tabla 3. Mecanismos para evitar la Inyección SQL en MYSQL

Vulnerabilidad	Descripción	Mecanismo/Actividad
ZAP' OR '1'='1'	Los resultados de la página se manipularon con éxito utilizando las condiciones booleanas que permiten recuperar datos de la base de datos.	comprobar todos los datos de entrada en el servidor. Si la aplicación usa JDBC, usar PreparedStatement o CallableStatement, con parámetros pasados por '?' Si la aplicación utiliza ASP, usar ADO Command Objects con una

[6] https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Espa%C3%B1ol.pdf

Vulnerabilidad	Descripción	Mecanismo/Actividad
		fuerte comprobación de tipos de consultas y parámetros. Si la Base de Datos permite Procedimientos Almacenados, usarlos. NO concatenar cadenas en las consultas en el procedimiento almacenado, o utilizar ‘exec’, ‘exec immediate’, o su funcionalidad equivalente. No crear consultas SQL dinámicas usando una sencilla concatenación de cadenas. Aplicar una lista blanca de caracteres permitidos, o una lista negra de caracteres no permitidos en la entrada del usuario. Aplicar el privilegio mínimo posible al usuario de la base de datos de los privilegios usados. En particular evitar el uso de los usuarios de base de datos 'root' o 'db-owner'. Esto no elimina la inyección SQL, pero minimiza su impacto. Conceder el mínimo acceso de base de datos que es necesario para la aplicación.
$result = mysql_query("select * from	Este código puede ser fácilmente vulnerable porque directamente	Transferir a PHP y MySQL la tarea de verificar la sentencia SQL. Se hace uso del método prepare de la

Vulnerabilidad	Descripción	Mecanismo/Actividad
usuarios where nick_usuario='$ nickN'");	agrega el valor de la variable 'nick' a la sentencia SQL. Esto permitirá a un atacante agregar lo que desee y comprometer los datos.	clase PDO (marcadores de posición) para preparar la sentencia y de execute para ejecutarla. $result = $db->prepare(select * from usuarios where nick_usuario= :nick'); // Indicando los datos $result->execute(array(':nick' => $clean['nick']));
Funciones mysql_* (mysql_connect, mysql_query	Se están utilizando funciones que se consideran inseguras, no está recomendado su uso, se consideran obsoletas y se han eliminado de PHP 5.6 en adelante	Usar método SQLi que tiene dos interfaces: una procedural y otra orientada a objetos. La interfaz procedural es muy parecida a mysql_*, la orientada objetos ofrece una mejor alternativa al aplicar el ocultamiento de los datos.
Autocompletado en los formularios activo	Esta característica puede ser un problema de privacidad para los usuarios, algunos navegadores permiten a los usuarios inhabilitarlas, sin embargo, estos están usualmente habilitados por defecto.	Inhabilitar el método autocompletar en el formulario. <form method="post" action="/form" autocomplete="off"> […] </form>
Privilegios de usuario no controlado	En la base de datos se crean los usuarios y se deja por defecto todos los privilegios sobre el esquema de MYSQL.	A nivel de base de datos controlar los privilegios de los usuarios creados y restringir al máximo los permisos sobre los objetos de la base de datos.
Cabecera X-Frame-Options sin activar	La cabecera X-Frame-Options puede ser abierta en un frame, o	Teniendo en cuenta que el servidor es Apache, se puede usar el fichero

Vulnerabilidad	Descripción	Mecanismo/Actividad
	iframe. De esta forma se pueden efectuar ataques de clickjacking sobre el sitio web.	.htaccess, agregando el siguiente código: Header always append X-Frame-Options SAMEORIGIN Otra opción sería especificar siempre en enmarcado denegado, o que solo pueda ser enmarcada desde el mismo origen (SAME-ORIGIN). Estos son los valores que acepta: DENY: La página no podrá ser mostrada en un frame/iframe. SAMEORIGIN: Solo podrá ser mostrada en un frame/iframe desde su propio dominio. ALLOW-FROM uri: Solo podrá ser mostrada en un frame/iframe desde las url's indicadas.

Cabecera X-XSS-Protection no está activada	La cabecera X-XSS-Protection no se está utilizando, no tiene activo el filtro XSS, el cual puede ser aprovechado para realizar ataques XSS	Una forma de agregar esta cabecera sería adicionando unas líneas de código a los archivos de funciones php del tema que se esté usando. Este archivo se puede ubicar en la ruta wp-content/themes/NOMBRE_TEMA, en donde NOMBRE_TEMA es el nombre del tema que se encuentre activado.
		
		add_action('send_headers', 'add_header_xxssprotection');
		function add_header_xxssprotection() {
		header('X-XSS-Protection: 1;mode=block');

Vulnerabilidad	Descripción	Mecanismo/Actividad
		} Este código genera la siguiente cabecera de respuesta: X-XSS-Protection: 1; mode=block Otra posibilidad para habilitar la cabecera en un servidor web como Apache, sería agregando el siguiente código en el fichero .htaccess: Header set X-XSS-Protection "1; mode=block"

Fuente: el autor.

7.1.2. ORACLE. En la siguiente tabla se muestra las actividades que se puede realizar para mitigar las vulnerabilidades encontradas con el entorno de prueba de la base de datos en Oracle.

Tabla 4. Mecanismos para evitar la Inyección SQL en ORACLE

Vulnerabilidad	Descripción	Mecanismo/Actividad
Manipulación de parámetro login y contraseña de usuario p_flow_id=4950 &p_flow_step_id =101&p_instanc e=12297469698 29085&p_page_ submission_id= 3957732645476 440&p_request =LOGIN&p_arg _names=93687	Al ingresar a la página los parámetros que permiten la conexión a la instancia de la base de datos XE, como son el LOGIN y la CONTRASEÑA se pueden manipular devolviendo datos del servidor permitiendo hacer inyecciones SQL con el fin de obtener información de la base de datos.	Evitar utilizar SQL dinámico y concatenación de caracteres que generalmente se usa en Forms, Reports procedimientos almacenados y casi en la mayoría de herramientas de desarrollo Oracle. Si es necesario su uso entonces utilizar variables de binding. Otra alternativa es utilizar el paquete DBMS_ASSERT, que contiene una serie de funciones que pueden ayudar a limpiar las

Vulnerabilidad	Descripción	Mecanismo/Actividad
3918730859954&p_t01=SYSTEM&p_arg_names=936874025479859956&p_t02=marbizsaid&p_md5_checksum		variables de entrada y evitar que se inyecte código SQL malicioso. También se debería monitorear los privilegios de usuario evitando que se tengan privilegios excesivos.
Cookie sin el conjunto de banderas HttpOnly HTTP/1.1 302 Found Server: Oracle XML DB/Oracle Database Set-Cookie: LOGIN_USERNAME_COOKIE=system Set-Cookie: WWV_CUSTOM-F_10_4950=6E6CEF8770ADE812; path=/	Si el sitio web intenta establecer una cookie de HttpOnly, la bandera HttpOnly será ignorado por el navegador, generando una secuencia de comandos de la Cookie tradicional accesible. Como resultado, la cookie por lo general la de manejo de sesión, se vuelve vulnerable al robo de modificación mediante código malicioso.	Si la bandera HttpOnly está incluida en el encabezado de respuesta HTTP, la cookie no se puede acceder a través de script del lado del cliente, incluso si existe un cross-site scripting (XSS), falla, y un usuario accede accidentalmente a un enlace que explota esta falla, el navegador no revelan la cookie a un tercero[7].
69552 - Oracle TNS Listener Remote Poisoning tcp/1521	El receptor remoto de Oracle TNS permite el registro de servicio desde un host remoto. Un atacante puede explotar este problema para Desviar datos desde un servidor de base de datos legítima o un	Añadir el Oracle Advanced Security SSL / TLS para la licencia de base de datos Oracle Standard Edition cuando se utiliza con los Real Application Clusters.

[7] https://es.wikibooks.org/wiki/HTTP/Estado_y_Seguridad/Cookies_HttpOnly

Vulnerabilidad	Descripción	Mecanismo/Actividad
	cliente a un sistema especificado por el atacante. Si un ataque es exitoso se podrá manipular las instancias de la base de datos, facilitando potencialmente un man-in-the-middle, secuestro de sesión o ataques de denegación de servicio en un servidor de base de datos legítimo	
22073 - Oracle Database Detection	El host remoto está ejecutando un servidor de base de datos Oracle. Puede ser posible extraer el número de versión del receptor TNS remoto enviando una "VERSIÓN" al servicio de escucha TNS que opera en este puerto.	Restringir el acceso a la base de datos sólo a los IP permitidos.
64712 - Oracle Application Express (Apex) CVE-2011-3525 tcp/8080 URL: http://192.168.0.7:8080/apex/	Una vulnerabilidad no especificada en las versiones 3.2 y 4.0 del componente Application Express (Apex) del servidor de bases de datos Oracle permite a los usuarios remotos y autenticados afectar la confidencialidad, la integridad y la disponibilidad en relación con el usuario desarrollador de Apex.	Actualizar la Aplicación Express al menos la versión 4.1.

Vulnerabilidad	Descripción	Mecanismo/Actividad
22964 - Service Detection tcp/8080	Servicio web corriendo sobre este puerto. Nessus pudo identificar el servicio remoto por el banner o viendo el mensaje de error que envía cuando recibe una solicitud HTTP.	No tiene ningún riesgo. Siempre y cuando se verifique quien accede al puerto.
10107 - HTTP Server Type and Version tcp/8080 Base de datos Oracle XML DB / Oracle	Este complemento intenta determinar el tipo y la versión del servidor web remoto.	No representa riesgo mientras se controle el puerto.

Fuente: el autor

7.1.3 NOSQL MONGODB. En proyectos de este tipo es indispensable primero definir con mucha precisión que es lo que se quiere hacer, identificar correctamente las especificaciones o requerimientos que determine si se requiere tecnología NoSQL. Es recomendable evaluar las herramientas disponibles en el mercado para determinar la mejor opción de implementación, en estos motores es diferente lograr hacer una migración entre uno y otro. En la siguiente tabla se muestran las medidas básicas de seguridad para MONGODB.

Tabla 5. Mecanismos para evitar la Inyección NoSQL en MONGODB

Vulnerabilidad	Descripción	Mecanismo/Actividad
Autenticación de usuarios inactiva.	No se activó la autenticación de usuario.	Considerar los parámetros que pueden ser útiles para la mayoría de instalaciones y se configuran en el fichero de configuración principal /etc/mongod.conf security: authorization: “enabled”

Vulnerabilidad	Descripción	Mecanismo/Actividad
Conexión abierta desde cualquier ip	Se permite la conexión desde cualquier ip, permitiendo realizar una inyección NoSQL.	Permitir conexiones únicamente desde la ip que se requiera y cambiar puerto por defecto. Ejemplo desde ip local: net: bindIp: 127.0.0.1 port: 27019 (el que se requiera)
Acceso vía http a la base de datos	Se tiene habilitado el acceso vía HTTP permitiendo manipular datos.	Deshabilitar cualquier acceso vía http tanto a la parte de administración como a la API Rest. http: enabled: false RESTInterfaceEnabled: false
Inyección NOSQL	No se establecieron los parámetros básicos de seguridad ni se creó usuario para el acceso a la base de datos con privilegios específicos. Se puede ejecutar inyección NOSQL.	Crear usuario administrador con el rol de “root” en la base de datos principal. Crear usuario con el rol de “dwOwner” en la base de datos. Modificar el fichero .config con la configuración y reiniciar el servicio de Mongo

Las anteriores son unas orientaciones básicas de seguridad que deberían ser complementadas con una correcta configuración de, Firewall, Auditoría de logs, Política de backups, Shared key para configuración de réplica sets, Activar tráfico SSL entre el cliente y el servidor MongoDB.

7.1.4. NoSQL CouchDB. CouchDB es una alternativa que según los medios en la red la muestran como la mejor entre las opciones de los gestores NoSql, permitiendo flexibilidad, escalabilidad y seguridad.

Auditar constantemente los controles de seguridad implementados, autenticación, validación de entradas, validación de datos, acceso basado en roles y controles sobre el sistema operativo y las redes.

- Autenticación: algunas de las bases de datos no tienen implementado el esquema de autenticación.
- Validación de entradas: aunque depende del controlador usado y la programación realizada se debe validar la entrada con el fin de evitar código malicioso, aunque no hay código sql se puede utilizar otras técnicas de inyección como java script.
- Validación de datos: los tiempos que el motor tenga que hacer al intentar validar los tipos de datos puede ocasionar denegaciones de servicio.
- Acceso basado en roles: estos sistemas se iniciaron a implementar en entornos controlados, luego ahora con el uso de internet se debe desarrollar la implementación de roles.
- Activar los logs, para monitorear los eventos que se ejecutan en la base de datos.
- Analizar y auditar actividades sospechosas o no autorizadas.
- Analizar y auditar la información sensible.
- Enmascarar la información.
- Aplicar configuraciones de seguridad propias del motor de base de datos.

7.2 OTRAS MEDIDAS PREVENTIVAS

- Filtrar las entradas de los usuarios evitando que se puedan pasar caracteres especiales como \ / “ ‘ o cualquier otro que pueda causar inconvenientes.
- Limitar al máximo los permisos de acuerdo a los perfiles de usuario que ejecuta las sentencias de SELECT, DELETE, UPDATE, INSERT entre otros.
- Se debería validar los datos que introduce el usuario como por ejemplo la longitud de los campos y el tipo de datos aceptados.
- Evitar conectarse como un usuario administrador o súper usuario o como propietario de la base de datos remotamente. Utilizar usuarios personalizados con privilegios muy limitados.
- Existe una amplia rama de controles que se pueden utilizar para prevenir ataques de inyección SQL, pero dependerá de otros factores como el motor de base de datos, arquitectura utilizada en el diseño de base de datos, así como el diseño web, servidor de alojamiento, entre otros factores.

Otros aspectos a tener en cuenta para prevenir una inyección es mantener los datos sensibles separados de los comandos y consultas.

La opción más recomendada es utilizar una API segura la cual evite el uso de intérpretes por completo o supla una interface con parámetros definidos. Al utilizar

las APIs, como los procedimientos almacenados, se debe tener cuidado porque aun así se pueden introducir inyecciones en el motor del interprete.

Si una API con parámetros no está disponible, se debería codificar los caracteres especiales, usando la sintaxis de escape específica del intérprete. OWASP ESAPI ofrece muchas de estas rutinas de codificación.

También se recomienda la validación de entradas, aunque no es una defensa integral porque muchas aplicaciones requieren caracteres especiales en sus entradas. Si esto es así, se debe aplicar las anteriores recomendaciones. La ESAPI de OWASP contien una librería extensible de rutinas de validación.

La ESAPI de OWASP es una colección gratis y abierta de todos los métodos de seguridad que un desarrollador necesita para construir una aplicación Web segura. Los desarrolladores pueden usar solo las interfaces y construir su propia implementación usando la infraestructura de la organización. También se puede usar la implementación de referencia como un punto de inicio. En concepto, la API es independiente del lenguaje. Sin embargo, los primeros entregables del proyecto son una API Java y una referencia de implementación Java. Esfuerzos para construir ESAPI en .NET y PHP están en marcha.

Exposición de Datos Sensibles

En su gran mayoría las aplicaciones web no protegen adecuadamente los datos sensibles que administra, pueden ser números de tarjetas de crédito, datos de autenticación entre otros. Los atacantes pueden sustraer o alterar dichos datos para llevar a cabo fraudes, suplantación de identidad u otros delitos. Cuando se trabaja con datos sensibles se requieren métodos de protección adicionales como por ejemplo el cifrado de datos, de igual manera se requiere medidas de seguridad cuando se intercambia datos con el navegador.

Algunas de las medidas preventivas que se pueden tener en cuenta son las siguientes:

- Determinar las amenazas internas o externas de las cuáles se protegerán los datos, asegurar el cifrado de los datos sensibles almacenados o en tránsito.
- No almacenar datos sensibles innecesariamente. Prescindir tan pronto como sea posible.
- Aplicar algoritmos de cifrado fuertes y estandarizados, así como claves robustas

gestionándolas de forma segura. Utilizar módulos criptográficos validados como FIPS 140 del gobierno de Estados Unidos que incluye componentes software y hardware.

- Las claves deberían almacenarse con un algoritmo para protegerlas como ser bcrypt, PBKDF2 o scrypt. 5.
- Deshabilitar el autocompletado en los formularios que recolectan datos sensibles.
- Deshabilitar el almacenamiento cache de páginas que contengan datos sensibles.
- No guardar contraseñas cuando el navegador lo solicita.

La norma FIPS 140-2 define cuatro niveles de seguridad.

- Nivel 1, normalmente utilizado en productos de cifrado de software, contiene requisitos de seguridad limitados.
- Nivel 2 requiere autenticación basada en el usuario. Necesita la capacidad para detectar la intrusión física mediante sistemas de bloqueo físico o precintos de seguridad.
- Nivel 3 agrega resistencia a la intrusión física dificultando al máximo los ataques. Si una intrusión es detectada, el dispositivo puede eliminar los parámetros de seguridad críticos. También incluye protección criptográfica eficaz y administración de claves, autenticación basada en la identidad y separación física o lógica entre las interfaces.
- •Nivel 4 incluye protección avanzada contra intrusiones y está diseñado para productos que operan en entornos desprotegidos físicamente.

Plan de Respaldo

A continuación, se relacionan algunas actividades con el fin atender las posibles eventualidades y prevenir o mitigar el impacto de eventos que puedan presentarse en un ataque a base de datos.

- Realizar copias de seguridad de la información y documentos de los discos duros de los equipos de cómputo. Puede hacerse una copia de seguridad en la nube, o si se tiene un servidor de archivos o en su defecto en discos duros extraíbles a cargo del área de sistemas.
- Efectuar copias de seguridad de los sistemas de información y Bases de Datos.
- Tener disponible como mínimo un repositorio de software de instalación para restaurar los archivos del sistema operativo y aplicaciones de un computador o servidor en caso de falla o virus. Sistema operativo (Windows, Linux, etc.) Paquetes de ofimática (Office) Bases de datos (SqlServer,Oracle, MySql, etc.) Drivers y utilitarios de impresoras, redes, switches, router, etc.

- Programar y realizar mantenimientos preventivos de equipos de cómputo y comunicación, energía (UPS, Planta eléctrica), extintores, alarmas y sistemas contra incendio.
- Actualizar las contraseñas de acceso a las aplicaciones y bases de datos. Mínimo cada tres meses o cuando se requiera por el usuario o por vacaciones, traslados, encargos o reemplazos del cargo.
- Tener como respaldo equipos de cómputo, repuestos, consumibles, para su reemplazo inmediato en caso de alguna falla y de aquellos que se consideren críticos para Seguridad.
- Redundancia en el canal de comunicaciones y dispositivos de red para evitar la interrupción de la prestación del servicio.

7.3 POLÍTICAS Y BUENAS PRÁCTICAS

- Evitar abrir correos de remitentes desconocidos.
- NO abrir cadenas de correos.
- Leer antes de descargar un software o un ejecutable de alguna aplicación para estar seguros de que se trata y que se está descargando.
- Verificar las extensiones de los archivos antes de descargarlos de una web o del correo electrónico.
- Mantener el paquete de actualizaciones de Windows al día.
- Utilizar software licenciado y si es libre verificar que se obtiene de las páginas oficiales.
- Mantener antivirus al día ojalá licenciado y no solo uno, tener antimalware o anti espía.
- Tener dentro de la infraestructura tecnológica una barrera perimetral que proteja la red ante posibles infiltraciones de código malicioso.
- Realizar periódicamente backups de la información identificando y clasificando los activos de información críticos para el desempeño y cumplimiento de la misión de la organización.

7.4 ALGUNAS RECOMENDACIONES DE SEGURIDAD

Las recomendaciones se basan en algunos de los principales ataques que ocurren en las bases de datos mediante diversas técnicas que se han desarrollado a medida que la tecnología ha avanzado a través del tiempo. Los ciberdelincuentes se enfocan en el estudio de las vulnerabilidades que los DBMS puedan tener a su vez que tienen en cuenta las formas de reacción de los usuarios permitiendo hacer más fácil su tarea de poder hacer una intrusión y alterar, dañar o sustraer los datos con el fin de obtener un beneficio por lo general de tipo monetario.

Ataque de fuerza bruta

- Los dba de las bases de datos pueden restringir el número de intentos de autenticación no mayor a 3 veces, impidiendo que se siga realizando el ataque.
- Realizar auditorías a las bases de datos o inclusive al servidor, al intentar conectarse a la base de datos quedaran guardados en los logs del servidor y esto puede permitir alertar del ataque realizado.

Ataque por sniffing

- Se puede prevenir estos ataques utilizando cifrado HTTPS en toda página web si se tiene un acceso a una base de datos remota.
- No utilizar la URL como identificador de sesión, en cambio se puede utilizar únicamente las cookies.
- El Test ICMP Ping de Latencia, consiste en hacer ping al posible sospechoso de sniffing y se toma nota del Round Trip Time (RTT, retardo de ida y vuelta). Se genera entonces una cantidad de falsas conexiones TCP en la red, en un período de tiempo muy breve. Si existe el sniffer resuelve estos paquetes y se analiza si el tiempo de latencia incrementa. Se hacemos ping nuevamente y se compara el RTT con el primer valor obtenido. Luego de varios tests se puede concluir si un sniffer está realmente dentro de la red.
- Utilizar VLANs, que mejoran notablemente la seguridad. Es necesario tener en cuenta que algunos tipos de configuración VLANs, pueden ser aprovechadas para ser atacadas mediante ARP Spooofin.
- Algunos routers / switch implantan medidas de seguridad adicionales anti spoofing mediante reglas.

DNS spoofing

- Deshabilitar la opción de gestión remota de los routers. En caso de requerirse es necesario establecer una contraseña fuerte.
- Mantener el sistema operativo actualizado, instalar los últimos parches disponibles. En el caso de Java, actualizar los navegadores, así como también la propia versión de Java.
- Verificar las conexiones a sitios que utilizan cifrado. Observar bien al protocolo, si un sitio utiliza HTTPS y cuando se accede utiliza HTTP, el usuario podría estar accediendo a un sitio falso réplica del original.

7.5 HERRAMIENTAS DE SOFTWARE

También se cuenta con algunas herramientas que permiten el análisis y explotación de vulnerabilidades a las bases de datos, así como otras que permiten realizar una evaluación de los controles implementados con el fin de identificar factores claves de éxito para mejorar y garantizar las seguridades en las bases de datos.

Tabla 6. Herramientas de software para análisis y explotación de vulnerabilidades

Herramienta	Descripción	Proveedor / Licencia
SQLMAP	Herramienta de pruebas de penetración de código abierto que automatiza el proceso de detectar y explotar los errores de inyección SQL y de carga de los servidores de bases de datos. Soporte completo para MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB, HSQLDB e Informix. Soporte completo para seis técnicas de inyección SQL: boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries and out-of-band.[8]	Copyright © 2006-2016: Bernardo Damele Assumpcao Guimaraes y Miroslav Stampar. Todos los derechos reservados. Este programa es software libre; puede redistribuirlo y / o modificarlo bajo los términos de la Licencia Pública General de GNU publicada por la Fundación para el Software Libre
NMAP (MAPEADOR DE REDES) Y SUS FUNCIONES	Es una herramienta de código abierto para escaneo de red y auditoría de seguridad, puede servir incluso para inventariar la red. Utiliza paquetes IP para establecer qué servicios se encuentran disponibles en una red, qué sistemas operativos se están ejecutando, qué tipo de filtrado de paquetes o cortafuegos se está utilizando, entre otras características [9].	Licencia Pública General de GNU publicada por la Fundación para el Software Libre

[8] http://sqlmap.org/

Herramienta	Descripción	Proveedor / Licencia
	Nmap muestra como resultados un listado de servicios y objetos analizados, uno de los principales datos es la tabla de puertos, donde se muestra el número de puerto y protocolo, el nombre más común del servicio, y su estado. El estado puede ser open (abierto), filtered (filtrado), closed (cerrado), o unfiltered (no filtrado). Filtrado señala que un cortafuego o cualquier otro elemento en la red están bloqueando el acceso a dicho puerto. Nmap presenta información de los protocolos IP soportados, incluyendo información adicional como el nombre de DNS según la resolución inversa de la IP, un listado de sistemas operativos posibles, los tipos de dispositivo, y direcciones MAC	
KALI LINUX	Es una distribución de Linux avanzada para pruebas de penetración y auditorías de seguridad. Kali es una reconstrucción de BackTrack Linux, se adhiere al estándar del entorno Debian y contiene más de 300 herramientas de pruebas de penetración. [9]	Código abierto y el árbol de desarrollo está disponible para todos y todas, las fuentes están disponibles para aquellos que desean modificar y reconstruir paquetes.

Fuente: el autor.

En la siguiente tabla se muestran algunas de las principales herramientas de Kali Linux.

[9] http://es.docs.kali.org/introduction-es/que-es-kali-linux

Tabla 7. Herramientas de la suite de KaliLinux

Herramienta	Función
Information Gathering	
ACCCHECK	Diseñada como una herramienta de ataque de diccionario para contraseña, dirigido a la autentificación de Windows a través del protocolo SMB. Intenta conectarse con el IPC$ o ADMIN$ dependiendo de la configuración seleccionada, usando una combinación de nombres de usuario y contraseñan.
GOLISMERO	Es un framework opensource para realizar auditorías de seguridad y pentesting, principalmente está orientado hacia la detección de vulnerabilidades en entornos web. Desarrollado en Phyton permite la creación de plugins de forma fácil, interpreta datos obtenidos con otros aplicativos tales como OpenVAS, SQLMAP, y es multiplataforma.
Sniffing & Spoofing	
WIRESHARK	Es un potente analizador de protocolos de red que permite ver todo el tráfico en la red local. Útil para testear el funcionamiento de la red y la seguridad.
DNSCHEF	Escrita en Python permite de manera sencilla crear un proxy DNS (Fake DNS) con varias posibilidades de configuración. Útil cuando no se puede utilizar un servidor proxy (como HoneyProxy por ejemplo), ya sea porque el aplicativo o dispositivo objetivo ignoran el proxy HTTP o porque se utiliza uno para su correcto funcionamiento etc.
Exploitation Tools	
SQLMAP	Herramienta de pruebas de penetración de código abierto que automatiza el proceso de detectar y explotar los errores de inyección SQL y de carga de los servidores de bases de datos. Soporte completo para MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB, HSQLDB e Informix. Soporte completo para seis técnicas de inyección SQL: boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries and out-of-band.
SET	El kit de herramientas de Social-Engineer es un marco de pruebas de penetración de código abierto diseñado para la ingeniería social. Permite automatizar tareas como por ejemplo el envío de SMS (mensajes de texto) falsos, con el fin de suplantar el número

Herramienta	Función
	telefónico que envía el mensaje, también se puede clonar cualquier página web y poner en marcha un servidor para hacer phishing en cuestión de segundos. El kit de herramientas SET está especialmente diseñado para realizar ataques avanzados contra el elemento humano.
Web Applications	
SQLNINJA	Específica para explotar vulnerabilidades de inyección SQL en una aplicación web que utiliza Microsoft SQL Server. Su objetivo principal es proporcionar un acceso remoto en el servidor de base de datos.
SQLSUS	De código abierto escrito en Perl, median una interfaz de línea de comandos, se puede recuperar la estructura de base de datos, inyectar consultas SQL, descargar archivos desde el servidor web, rastrear el sitio web de directorios, cargar y controlar una puerta trasera, clonar la base de datos entre otros. Utilizado para hacer pruebas de penetración en motores MySQL.
Password Attacks	
SQLDICT	Herramienta de ataque de diccionario para SQL Server. Una forma sencilla de hackear estas bases es intentar un ataque mediante fuerza bruta sobre el usuario 'sa', buscando servidores sql dentro de la instalación utilizando por ejemplo sqlscan y crackeando el usuario 'SA' mediante sqldict, solamente teniendo la IP del servidor se agrega el diccionario de datos con los password posibles y se ejecuta el programa.
NCRACK	Herramienta de autenticación, fue construida para ayudar a las organizaciones a asegurar las redes mediante pruebas a los dispositivos de red identificando contraseñas débiles. Fue diseñado utilizando un enfoque modular, una sintaxis de línea de comando similar a Nmap y un motor dinámico que puede adaptar su comportamiento basado en la retroalimentación de la red. Soporta protocolos como RDP, SSH, HTTP, SMB, pop3, VNC, FTP y telnet.

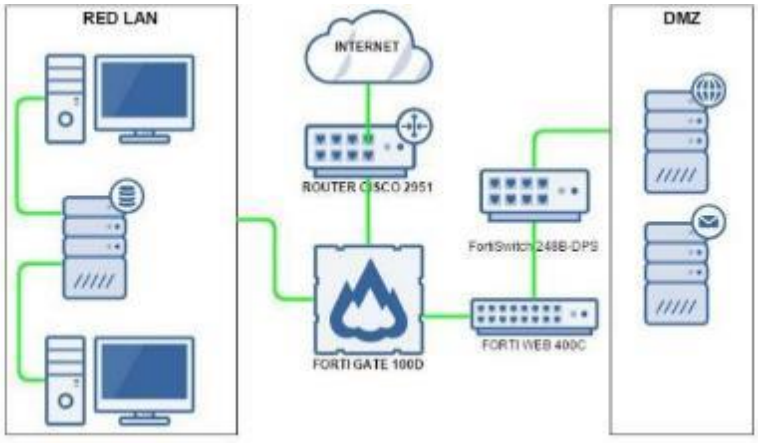
Fuente: el autor.

7.6 ZONA DESMILITARIZADA SEGURA DMZ

Otro aspecto a tener en cuenta cuando de bases de datos se trata es las redes e internet, aquí la seguridad también juega un papel importante, existe un mecanismo que permite separar los servicios públicos como las páginas web, el correo electrónico, servicios FTP, de la red interna o red LAN de la organización, esto consiste en establecer un perímetro de la red denominada DMZ, es un segmento de red o zona diseñada mediante la utilización de un cortafuego que puede filtrar el tráfico de entrada como de salida, lo anterior con el objetivo de proteger la red interna de posibles intrusiones así como de amenazas externas a los equipos con acceso a la red pública. El cortafuego funciona a través de políticas permisivas y restrictivas, la primera permite todo tipo de tráfico menos el que se quiera denegar un tipo de tráfico específico y la segunda que deniega todo el tráfico y acepta únicamente el explícitamente permitido o que se requiera en la organización. Existen en general cuatro tipos de cortafuegos: de pasarela que funciona para Telnet, FTP, Web, de capa de red que permiten configuraciones con dirección IP y puerto de origen y destino, de aplicación para analizar protocolos HTTP y los cortafuegos personales para los equipos de escritorio.

Con el fin de comprender la estructura para implementar el DMZ, se muestra en la figura 110 como puede ser su diseño. En la parte izquierda se encuentra la red interna o LAN de la organización, separada de los servicios web a través del DMZ. Para maximizar la seguridad se puede optar por la inclusión de un cortafuego para aplicaciones web con el fin de prevenir determinados ataques específicos en Internet y se controlan las transacciones al servidor web de la organización.

Figura 110. Diseño DMZ con WAF y Switch



Fuente: El autor

7.6.1. Elementos que intervienen en un DMZ. La implementación de un esquema DMZ dependerá en gran manera de las necesidades de una organización, se debería tener en cuenta las políticas de cambios y del Sistema de Seguridad de la información si se encuentran implementados, así como los servidores existentes, bases de datos y la arquitectura de red establecida.

Para una adecuada implementación de un DMZ aparte de una fase de planeación donde se incluyan los aspectos como personal, dispositivos, documentación entre otros se requieren los siguientes elementos:

- Revisión de información y requerimientos entregados por parte de Seguridad Segura. Es importante garantizar el cumplimiento, aprobación y/o entrega por parte de la entidad de los prerrequisitos definidos.
- El firewall Perimetral, que realizará enrutamiento a DMZ.
- El FortiGate-100D es una solución de seguridad para las pequeñas y medianas empresas o sucursales remotas de las grandes redes. Combina firewall, IPSec y SSL VPN, control de aplicaciones, prevención de intrusiones, antivirus, antimalware, antispam, seguridad P2P, y filtrado web en un solo dispositivo (Productos Fortinet, http://fortinet.globalgate.com.ar/ver.php/mod/productos/categoria/191). Este dispositivo esta entre los USD 1700 y USD 2000.

Figura 111. Imagen Fortigate 100D

Fuente: http://fortinet.globalgate.com.ar/imagenes/FG-100D.jpg

- Un WAF Web Application Firewall, dispositivo que puede ser hardware o software y analiza el tráfico web protegiendo de ataques como inyección SQL, Cross Site Scripting, entre otros. FortiWeb-400C proporciona la seguridad de las aplicaciones con una interfaz de usuario web fácil y las opciones de implementación flexibles, FortiWeb-400 ofrece una completa función de firewall que ayuda a proteger contra las amenazas de OWASP Top 10 y la dirección PCI DSS 6.6 (Productos Fortinet, http://fortinet.globalgate.com.ar/ver.php/mod/productos/categoria/191). Este dispositivo se encuentra en el mercado por valor de USD 7.800.

Figura 112. Imagen FortiWeb 400C

Fuente: http://fortinet.globalgate.com.ar/imagenes/FWB-400C_Ft.png

- Un switch para la DMZ. FortiSwitch-248B-DPS es un switch de alto rendimiento 10/100/1000 Ethernet con quad doble velocidad 1/10 puertos SFP + fibra uplink, que permite hasta 40 Gbps de tráfico core. Se garantiza la compatibilidad con los actuales puertos de 1 GbE de fibra y su futuro 10 GbE de fibra, sin costo adicional (Productos Fortinet, http://fortinet.globalgate.com.ar/ver.php/mod/productos/categoria/191). El precio de este switch se encuentra en USD 9.900.

Figura 113. Imagen FortiSwitch-248B-DPS

Fuente: http://fortinet.globalgate.com.ar/imagenes/FSW-248B-DPS_Lt.png

- Router Cisco 2951 Integrated Services Router (ISR) proporciona datos de alta seguridad, voz, vídeo y servicios de aplicaciones. Entre USD 400 y USD 500.

Figura 114. Imagen Cisco 2951

Fuente: http://www.cisco.com/c/en/us/products/routers/2951-integrated-services-router-isr/index.html#

- Si se requiere implementar servidores se tendría la posibilidad de servidores que pueden ser DELL o HP que oscilan entre USD 3000 y USD 5000, o se puede aprovisionar servidores en clúster, con su respectiva conexión clúster a la DMZ.

Otros sistemas de seguridad para la red

IDS: El Sistema De Detección De Intrusos - Intrusion Detection System (Ids), es una herramienta que puede complementar el esquema de Seguridad de una organización, es utilizada principalmente para identificar ataques en tiempo real, guardar los registros y reportar al equipo de TI con el fin de tomar acciones al respecto. Se puede decir que se caracteriza por servir como una bitácora y pueden ser el punto de partida para determinar el origen de los ataques. Los IDS han evolucionado al sistema de detección y prevención de intrusos IDPS que tienen la capacidad de bloquear la conexión si se detecta que el evento es peligroso de manera similar a los cortafuegos.

SNORT: basado en NIDS, contiene una base de detección de eventos y escaneo de puertos que permite registrar, analizar tráfico y alertar en tiempo real, puede efectuar análisis de protocolos, búsqueda/identificación de contenido y puede ser usado para detectar ataques y pruebas, como por ejemplo buffer overflows, escaneos indetectables de puertos, intentos de reconocimientos de sistema operativos entre otros. Disponible en la página http://www.snort.org/ bajo licencia GPL, funciona bajo plataformas Windows y UNIX/Linux. Es uno de los más utilizados y contiene gran cantidad de filtros o patrones ya predefinidos, así como actualizaciones frente a eventos o ataques, que se hayan detectado a través de diferentes boletines de seguridad.

Sniffer: Es un software que puede analizar los paquetes transmitidos a través de las tramas de una red. Tiene como funcionalidad monitorear la red para detectar y examinar fallos, aunque se puede emplear para fines delictivos, mediante la utilización de un programa de sniffing y estando en la red el atacante puede interceptar el tráfico de los diferentes protocolos de red incluyendo usuarios y contraseñas si ha ingresado o iniciado sesión a páginas web, correo electrónico, bases de datos entre otros.

WireShark: es un analizador de protocolos, funciona en Windows y Linux, permite realizar un sniffing en la red donde se encuentre conectado. Una vez ejecutado el programada se captura el tráfico y se pude hacer filtrado por direcciones IP, por protocolos HTTP, SMTP, por Dominio hacia páginas como Facebook, google o cualquier otro sitio, si se tiene una base de datos remota o local allí aparecerá en el analizador de tráfico.

CONCLUSIONES

Los escenarios configurados para las pruebas de penetración y ataques de inyección SQL y NoSQL son técnicas que permitieron comprobar la seguridad de las bases de datos, así como detectar vulnerabilidades en la red y aplicaciones web, que por lo general son aprovechadas por un intruso para explotarlas.

Aunque las pruebas de penetración son eficaces en la seguridad en la red, no es lo mismo en las aplicaciones web, aún más en entornos y conexiones a bases de datos, en razón a que son básicamente hechas a la medida. Existen herramientas para realizar pruebas de penetración que automatizan el proceso, sin embargo, su eficacia está determinada por la complejidad de la aplicación desarrollada.

La metodología desarrollada permitió identificar y plantear una propuesta con algunos de los principales métodos para proteger las bases de datos sql y no sql estudiadas, con el ánimo de brindar un documento que beneficie a los administradores de bases de datos, desarrolladores, auditores y usuarios, para que puedan consultar y aplicar de la mejor manera los procedimientos existentes acorde a sus necesidades.

RECOMENDACIONES

Aplicar los principios de ingeniería en el desarrollo de bases de datos evitará potenciales fallos y permitirá hacer un mejor control en su desarrollo, minimizando riesgos y mejorando el mantenimiento y calidad.

La guía de pruebas OWASP, así como sus referentes y demás procedimientos brindan un excelente mecanismo para establecer criterios válidos cuando se planifica diseña, desarrolla y mantiene una aplicación. Se abarca de manera acertada, aspectos tanto técnicos como organizativos, involucrando diseñadores, desarrolladores, programadores, analistas de seguridad y personal directivo en una organización.

BIBLIOGRAFÍA

ARELLANEZ HERNANDEZ, Jorge Luis (2012), Metodología de la investigación en ciencias sociales, http://metodologiaencienciassociales.blogspot.com.co/p/vi-fases-de-la-investigacion-documental.html, (vi: 10.11.2016)

ECURED, Metodología de la investigación documental, https://www.ecured.cu/Metodolog%C3%ADa_de_la_investigaci%C3%B3n_documental, (vi: 8.11.2016)

GEOTALLERES, Conceptos básicos de SQL. Internet: http://geotalleres.readthedocs.io/es/latest/conceptos-sql/conceptos_sql.html

GOMEZ POSADA, Susana (2009), Técnicas de Investigación, Pereira Colombia, http://datateca.unad.edu.co/contenidos/100104/100104_EXE/leccin_8_investigacin_terica__investigacin_emprica_investigacin_documental.html, (vi:14.11.2016)

ORACLE, Oracle Database 12c. internet: http://www.oracle.com/technetwork/es/articles/database-performance/oracle-partitioning-en-database-12c-2244565-esa.html

ORACLE, Construyendo con Bloques en PL/SQL, internet: http://www.oracle.com/technetwork/es/articles/sql/construyendo-con-bloques-parte-1-1549135-esa.html

PALENCIA AVENDAÑO, María Luisa (2013). Metodología de la Investigación, http://datateca.unad.edu.co/contenidos/100103/100103_2013_1/Metodologia_de_la_Investigacion_MODULO-1.pdf, (vi:14.11.2016)

TUTORIALES PROGRAMACION YA, Disparadores (Trigger), internet: http://www.tutorialesprogramacionya.com/sqlserverya/temarios/descripcion.php?cod=147&punto=&inicio=

Universidad del Cauca, Facultad en Ingeniería Electrónica y Telecomunicaciones Grupo en Ingeniería Telemática, http://univirtual.unicauca.edu.co/moodle/pluginfile.php/20629/mod_resource/content/0/Materiales/Libro_Ingeniero_Serrano/Modelo_de_Investigacion_Documental.pdf, (vi:10.11.2016)

OWASP Top 10 Proactive Controls 2016. Disponible en: https://www.owasp.org/images/5/57/OWASP_Proactive_Controls_2.pdf
[Proyecto OWASP API de seguridad empresarial (ESAPI). Disponible en: https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API/es

REPORTE OWASP TOP 10
https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Espa%C3%B1ol.pdf

FORTINET, Garantía de servicios y calidad de productos. Disponible en: http://fortinet.globalgate.com.ar/index.php/modo/home

R. Pressman. Ingeniería del Software un enfoque práctico séptima edición. (2010). Disponible en: http://artemisa.unicauca.edu.co/~cardila/Libro_Pressman_7.pdf

Vela Fernando, Andrade Roberto. Guía de pruebas OWASP 4.0. Escuela Politécnica Quito Ecuador.

RESUMEN ANALÍTICO ESPECIALIZADO R.A.E.

RAE No. 1		**Fecha de elaboración:** 12 de diciembre de 2017
Tipo Publicación: Investigación	**Páginas: 3 (908 palabras)**	**Año:** diciembre de 2017
Título y datos complementarios: Diseño de procedimiento para la protección de ataques por inyección sql a bases de datos SQL y NoSQL.		
Autor (es): Marbiz Said Ducuara Amado, Eymar Silva Meza		
Palabras Claves: ethical haking, sql, nosql, MySql, Oracle, MongoDB, CouchDB, Vulnerabilidad, Amenaza, Riesgo, Firewall, IDS, DMZ, WAF, Kali Linux, SSH, File Zilla, Wireshark, SQLMPAP, NMAP, NOSQLMAP, Inyección SQL, Ataque fuerza bruta.		
Descripción General o Resumen: Al paso de los años las organizaciones han crecido y han aumentado su inversión en tecnología, con el auge de los computadores y el internet cada día se automatiza más y buscan mejorar sus procesos y sus sistemas de información. Sin embargo, al avanzar en este campo también se debe tener en cuenta que las vulnerabilidades están presentes en los diferentes sistemas: llámese, redes, bases de datos, aplicaciones, sistemas operativos entre otros y que a partir de ese crecimiento surgen las amenazas sobre la integridad, confidencialidad e integridad de los datos. El propósito del presente trabajo es diagnosticar la seguridad en las transacciones y acceso en algunos motores de base de datos sql y bases de datos no sql, mediante la realización de pruebas de penetración y de inyección SQL y NoSQL a través de entornos de prueba controlados que permitan explorar y explotar vulnerabilidades con la utilización de herramientas conocidas como Kali Linux, con el fin de determinar aquellas mejores prácticas que orienten los procedimientos y mecanismos de control para prevenir un evento que atente contra la información salvaguardada en las bases de datos.		
Objetivo General: Plantear un procedimiento para la protección de ataques por inyección SQL en bases de datos relacionales y no relacionales mediante el estudio y la búsqueda de técnicas aplicadas a partir de un escenario de pruebas.		
Objetivos específicos: Realizar el levantamiento de información sobre Metodologías de ethical hacking y mecanismos de seguridad para las bases de datos relacionales y no relacionales. Determinar y aplicar las metodologías de ethical hacking en entorno de prueba con sistemas de gestión de bases de datos relacionales y no relacionales. Presentar propuesta con los mecanismos o técnicas de protección para evitar los ataques a bases de datos relacionales y no relacionales.		
Áreas del Conocimiento: El campo de aplicación es en el en el área de la Informática, en Seguridad Informática específicamente en lo relacionado con los ataques a bases de datos mediante inyección SQL y NOSQL. Se abordaron		

temáticas relacionadas con los motores de bases de datos MySql, Oracle, MongoDB y CouchDB, así como la explotación de vulnerabilidades y los posibles mecanismos de protección.

Metodología: De acuerdo a los tipos de investigación existentes y teniendo en cuenta que las características de los medios utilizados para obtener los datos del proyecto se realizan a través de fuentes de información electrónica la investigación tiene un enfoque de tipo descriptiva y cuantitativa.

Resultados: se identificaron algunos de los principales ataques (fuerza bruta, XSS, inyección sql, inyeccion nosql) que ocurren en las bases de datos mediante diversas técnicas que se han desarrollado a medida que la tecnología ha avanzado a través del tiempo. De igual manera se identificaron y probaron algunas herramientas que permiten el análisis de vulnerabilidades a las bases de datos, así como otras que permiten realizar una evaluación de los controles implementados con el fin de identificar puntos críticos con el fin de mejorar y garantizar la seguridad en las bases de datos. Por otra parte, se logró establecer los mecanismos y actividades que convienen ser adoptados por las organizaciones para proporcionar los lineamientos, soporte, componentes y metodologías para la protección de los sistemas informáticos de nivel crítico para la operación misional especial lo relacionado con bases de datos SQL y NOSQL.

Conclusiones:

Los escenarios configurados para las pruebas de penetración y ataques de inyección SQL y NoSQL son técnicas que permitieron comprobar la seguridad de las bases de datos, así como detectar vulnerabilidades en la red y aplicaciones web, que por lo general son aprovechadas por un intruso para explotarlas.

Los escenarios configurados para las pruebas de penetración y ataques de inyección SQL y NoSQL son técnicas que permitieron comprobar la seguridad de las bases de datos, así como detectar vulnerabilidades en la red y aplicaciones web, que por lo general son aprovechadas por un intruso para explotarlas.

La guía de pruebas OWASP, así como sus referentes y demás procedimientos brindan un excelente mecanismo para establecer criterios válidos cuando se planifica diseña, desarrolla y mantiene una aplicación. Se abarca de manera acertada, aspectos tanto técnicos como organizativos, involucrando diseñadores, desarrolladores, programadores, analistas de seguridad y personal directivo en una organización.

Bibliografía:

TUTORIALES PROGRAMACION YA, Disparadores (Trigger), internet: http://www.tutorialesprogramacionya.com/sqlserverya/temarios/descripcion.php?cod=147&punto=&inicio=

Universidad del Cauca, Facultad en Ingeniería Electrónica y Telecomunicaciones Grupo en Ingeniería Telemática,

http://univirtual.unicauca.edu.co/moodle/pluginfile.php/20629/mod_resource/content/0/Materiales/Libro_Ingeniero_Serrano/Modelo_de_Investigacion_Documental.pdf, (vi:10.11.2016)

San Murugesan. (2005). Web Engineering: Introduction and Perspectives. Disponible en: http://cic.javerianacali.edu.co/wiki/lib/exe/fetch.php?media=materias:webengineeringintro.pdf

R. Pressman. Ingeniería del Software un enfoque práctico séptima edición. (2010). Disponible en: http://artemisa.unicauca.edu.co/~cardila/Libro_Pressman_7.pdf

Vela Fernando, Andrade Roberto. Guía de pruebas OWASP 4.0. Escuela Politécnica Quito Ecuador.

Velasco, Rubén. (2015) OWASP ZAP, herramienta para auditar la seguridad de una página web. Disponible en: https://www.redeszone.net/2015/04/25/seguridad-web-owasp-zap/).

Comentarios del Investigador: Los autores agradecen a la Universidad Nacional Abierta y a Distancia UNAD, por el apoyo brindado y al Ingeniero Francisco Solarte Solarte, director del proyecto por la colaboración en la realización del presente trabajo.

Autores del RAE: Marbiz Said Ducuara Amado, Eymar Silva Meza