

PENTESTING PARA WEB

ANGELA DEL PILAR LOPEZ MOLINA

**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA – UNAD
ESCUELA DE CIENCIAS BASICAS, TECNOLOGIA E INGENIERIA
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA**

BOGOTA D.C

2019

PENTESTING PARA WEB

ANGELA DEL PILAR LOPEZ MOLINA

Proyecto de Grado para optar por el título:

Especialista en Seguridad informática

Director de Monografía:

Ing. DANIEL FELIPE PALOMO LUNA

**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA – UNAD
ESCUELA DE CIENCIAS BASICAS, TECNOLOGIA E INGENIERIA
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
BOGOTÁ D.C**

2019

Nota de Aceptación:

Firma del Presidente del
Jurado

Firma del Jurado

Firma del Jurado

Bogotá, 15 de febrero de 2019

DEDICATORIA

Dedico este Proyecto de monografía a mi familia, de los cuales he recibido apoyo para sacar adelante la especialización para dar un valor agregado a mi Carrera y actividades laborales.

AGRADECIMIENTOS

A mi familia como apoyo y principal motivación y motor para continuar aprendiendo y logrando crecimiento profesional.

Agradezco a los docentes y director de proyecto que intervinieron en la realización de la monografía aportando para una adecuada construcción y brindando apoyo constante para la realización de la monografía.

CONTENIDO

	pág.
GLOSARIO	12
1. TITULO	14
3. DEFINICION DEL PROBLEMA	16
4. ALCANCE Y DELIMITACION DE PROYECTO	18
5. JUSTIFICACION	19
6. OBJETIVOS GENERAL Y ESPECIFICOS	20
6.1 OBJETIVO GENERAL	20
6.2 OBJETIVOS ESPECIFICOS.....	20
7. MARCO REFERENCIAL	21
7.1 MARCO TEORICO	21
7.1.1. QUE ES PENTESTING.....	21
7.1.2 PERFIL DEL PENTEST.....	21
7.1.2.1 Hackers de Sombre Blanco.....	21
7.1.2.2 Hackers de Sombre Negro.....	21
7.1.3 TIPOS DE PRUEBAS DE PENETRACION	22
7.1.4 PRUEBAS DE PENETRACION A APLICACIONES WEB.....	22
7.1.5 METODOS DE PENTESTING	24
7.1.6 FASES DE TEST DE PENETRACION	24
7.1.7 HERRAMIENTASDE PRUEBAS DE <i>PENTESTING</i>	25
7.2 MARCO CONCEPTUAL	27
7.2.1 Guía de Owasp.....	27
7.2.2 Capitulo 4 – Pruebas de Intrusión de Aplicaciones Web.....	27
7.2.3 Fases de las pruebas. Modo Pasivo:.....	27
7.2.4 Definición de las pruebas en subcategorías según OWASP V4	28
7.2.5 Recolección de Información.	28
7.2.6 Kali Linux.....	29
7.3 MARCO LEGAL.....	29

7.3.1 LEYES EN COLOMBIA DE LA PROTECCION DE DATOS.....	29
7.3.1.1 Capítulo 1: De la protección de la información y de los datos.	29
7.3.1.2 Capítulo2: De los atentados informáticos y otras infracciones.	30
8 PROPUESTA METODOLOGICA.....	31
8.1 METODOLOGIAS	31
8.1.1 Metodología OSSTMM.	31
8.1.2 Metodología ISSAF	32
8.1.3 OWASP Top 10.....	32
8.1.3.1 Resumen de los Factores de riesgo Top 10	33
9. DESARROLLO DE PRUEBAS	34
9.1 Configuración de Ambientes	34
9.2 Ejecución de las pruebas.....	34
10. ADECUACION DE AMBIENTES	35
10.1 INSTACION DEL SOFTWARE REQUERIDO.	35
10.1.2 Instalación de <i>Sugar CRM</i> en Windows 8	37
10.1.3 Configuración de <i>SugarCRM</i>	44
11. EJECUCION DE LAS PRUEBAS DE PENTESTING	49
11.1 Pruebas para gestionar la configuración y la implementación.....	50
11.1.1 Prueba de los métodos HTTP (OTG-CONFIG-006).	50
11.1.2 Resultado Prueba de los métodos HTTP (OTG-CONFIG-006).....	52
11.2 Pruebas de Administración de Identidad	53
11.2.1 Prueba proceso de registro de usuarios (OTG-IDENT-002).....	53
11.2.2 Resultado de la prueba proceso de registro de usuarios (OTG-IDENT-002).....	57
11.3 Pruebas de seguridad al proceso de autenticación	57
11.3.1 Pruebas para determinar un mecanismo de bloqueo débil (OTG-AUTHN-003).....	57
11.3.2 Resultado de las pruebas para determinar un mecanismo de bloqueo débil (OTG-AUTHN-003).	61
11.4 Pruebas de seguridad al proceso de autorización	62
11.4.1 Pruebas para determinar el escalamiento de privilegios (OTG-AUTHZ003).	62
11.4.2 Resultado de la Prueba para determinar el escalamiento de privilegios (OTG-AUTHZ003).....	66
11.5 Pruebas de seguridad al proceso de gestión de sesiones	66

11.5.1 Prueba para atributos de cookies (OTG-SESS-002)	66
11.5.2 Resultado de la prueba para atributos de cookies (OTG-SESS-002)	72
11.6 Pruebas de seguridad a la validación de entrada	73
11.6.1 Pruebas de Inyecciones de SQL (OTG-INPVAL-005)	73
11.6.1.1 Resultado de las pruebas de Inyecciones de SQL (OTG-INPVAL-005)	75
11.6.2 Pruebe la saturación del Búfer (OTG-INPVAL-014)	75
11.6.2.1 Resultado de la prueba de saturación del Búfer (OTG-INPVAL-014)	77
11.6.3 Pruebas de Cross Site Scripting almacenados (OTG-INPVAL-002)	77
11.6.3.1 Resultado de la prueba de Cross Site Scripting almacenados (OTG-INPVAL-002)	79
11.7 Pruebas para Criptografía débil	80
11.7.1 Pruebas para el envío de información sensible por canales sin encriptar (OTG-CRYPST-003)	80
11.7.2 Resultado Pruebas para el envío de información sensible por canales sin encriptar (OTG-CRYPST-003)	81
12. CONSOLIDADO DE RESULTADOS	82
13. CONCLUSIONES	84
14. RECOMENDACIONES	85
BIBLIOGRAFIA	86

LISTA DE TABLAS

pág.

TABLA 1- HERRAMIENTAS PARA PENTESTING.....	26
TABLA 2. OWASP RECOPIACIÓN DE INFORMACIÓN	28
TABLA 4. EJECUCIÓN DE PRUEBAS PENTESTING	49
TABLA 5.RESULTADO DE LAS PRUEBAS	82

LISTA DE FIGURAS

pág.

FIGURA 1 COMPARATIVO TOP 10 OWASP.....	33
FIGURA 2 INSTALACIÓN KALI LINUX	35
FIGURA 3.FINALIZA LA INSTALACIÓN.....	36
FIGURA 4 DESCARGA DE SUGARCRM.....	37
FIGURA 5 RENOMBRAR EL ARCHIVO DESCARGA SUGAR	37
FIGURA 6 CREACIÓNBDSUGAR	38
FIGURA 7 INSTALACIÓN SUGAR CRM.....	39
FIGURA 8 ACEPTAR TÉRMINOS Y CONDICIONES SUGAR.....	39
FIGURA 9 SELECCIÓN DE OPCIÓN DE INSTALACIÓN SUGAR	40
FIGURA 10 CONFIRMACIÓN INSTALACIÓN SUGAR	41
FIGURA 11 CONFIGURACION SUGARCRM	41
FIGURA 12 VERIFICACION DB CREDENTIALS	42
FIGURA 13 CREACIÓN DE USUARIO ADMINISTRADOR.....	42
FIGURA 14 CONFIRMACIÓN DE CONFIGURACIÓN SUGAR.....	43
FIGURA 15 CONFIRMACIÓN DE LA CREACIÓN TABLAS BD	44
FIGURA 16 INGRESO A SUGARCRM.....	44
FIGURA 17 WELCOMESUGAR	45
FIGURA 18 CAMBIO DE LOGO SUGAR	46
FIGURA 19 DATOS DE CONFIGURACIÓN	46
FIGURA 20 CONFIGURACIÓN SMTP SUGAR.....	47
FIGURA 21 DATOS ADMINISTRADOR	48
FIGURA 22 FINALIZACIÓN DE CONFIGURACIÓN.....	48
FIGURA 23. COMPLEMENTO RESTCLIENT	50
FIGURA 24 RESULTADOHEADER.....	51
FIGURA 25. RESULTADO CURL	51
FIGURA 26. EJECUCIÓN CURL EN UBUNTU	52
FIGURA 27. INGRESO SUGARCRM DESDE MOZILLA.....	53
FIGURA 28. INFORMACIÓN DEL COMPLEMENTO CON EL LOGUEO.....	54
FIGURA 29. INFORMACIÓN DE USUARIO Y PASSWORD.....	55
FIGURA 30. OPCIÓN DE CREACIÓN USUARIOS	55
FIGURA 31. CONFIRMACIÓN DEL REGISTRO DE USUARIO FUENTE: EL AUTOR... 56	56
FIGURA 32. INFORMACIÓN NUEVO USUARIO	56
FIGURA 33. INGRESO A SUGARCRM CON DATOS ERRADOS	58
FIGURA 34. ARCHIVO CON LISTA DE USUARIOS/CLAVES.....	58
FIGURA 35.CONFIGURACIÓN DE HYDRA - TARGET	59
FIGURA 36. CONFIGURACIÓN HYDRA -PASSWORDS	60
FIGURA 37. CONFIGURACIÓN HYDRA - SPECIFIC.....	60
FIGURA 38. ATAQUE DE FUERZA BRUTA CON HYDRA	61
FIGURA 39 APERTURA WEBS CARAB	62

FIGURA 40 CONFIGURACIÓN WEBCARAB -PROXY	63
FIGURA 41 CONFIGURACIÓN NAVEGADOR COMO PROXY	63
FIGURA 42 INICIO DE WEBCARAB	64
FIGURA 43 PÁGINA DE LOGUEO SUGARCRM	64
FIGURA 44 LOGUEO A LA APLICACIÓN SUGARCRM	65
FIGURA 45 DATOS DE CUENTA DE SUGARCRM	65
FIGURA 46 CONFIGURACIÓN PROXY DE OWASP -ZAP	67
FIGURA 47 CONFIGURACIÓN EN EL NAVEGADOR MOZILLA	67
FIGURA 48 INGRESO A LA APLICACIÓN SUGARCRM CON INTERCEPTACIÓN PROXY	68
FIGURA 49 APERTURA DE LA APLICACIÓN SIN LOGUEO	68
FIGURA 50 INFORMACIÓN TAG FORMA Y PASSSWOR D	69
FIGURA 51 INFORMACIÓN DESPUÉS DE LOGUEO	69
FIGURA 52 RESPONSE DE SCRIPT, SETCOOKIE FUENTE: EL AUTOR	70
FIGURA 53 INFORMACIÓN TAG FORM HIDDEN	70
FIGURA 54 INFORMACIÓN DE ALERTA PARA X-FRAME-OPTIONHEADER	71
FIGURA 55 ALERTA COOKIE NO HTTPONLY	72
FIGURA 56 DESHABILITA FIREWALL WINDOWS FUENTE: EL AUTOR	73
FIGURA 57 ATAQUE SQL INJECTION FUENTE: EL AUTOR	74
FIGURA 58 ATAQUE DENEGADO SQL INJECTION	74
FIGURA 59 ATAQUE DOS	75
FIGURA 60 ATAQUE DOS -METASPLOIT	76
FIGURA 61 LUEGO DE DETENER ATAQUE DOS	77
FIGURA 62 HERRAMIENTA XSS SNIPER	78
FIGURA 63 RESULTADO DE ESCANE O	79
FIGURA 64 LOGUEO SIN CIFRADO	80

GLOSARIO

ATAQUES INFORMATICOS: en la actualidad existen varios tipos de ataques informáticos, consiste en tomar el control o causar daño a una aplicación informática. Los ataques se pueden presentar por fallos en los diseños de internet, problemas de configuración en protocolos y sistemas operativos.

AMENAZAS: las amenazas presentadas a los sistemas informáticos se pueden definir como una situación que puede causar daño o pérdida. Cuando las amenazas se materializan se pueden presentar los ataques.

CLICKJACKING: Es un tipo de ataque ejecutado desde un navegador donde se engaña al usuario con el fin de direccionarlo a enlaces, cuadros de diálogos o botones no validos con funciones no reales para obtener la información confidencial o tener el control de su PC.

CROSS-SITE SCRIPTING (XSS): es un tipo de ataque presentado por temas de inseguridad informática o agujero de seguridad frecuente en las aplicaciones Web, que permite a un atacante inyectar información con script en páginas web visitadas.

CRM (*Customer relationship management*): administración basada en la relación con los clientes, automatizar la gestión de toda la organización basada en la satisfacción del cliente.

ESCAPADO: proceso que describe la importancia de que las variables deben estar formateadas para ser usadas en las consultas, de manera que se evite que sean corrompidas las mismas.

HACKER: se define a la persona que cuenta con los conocimientos de informática y seguridad con la capacidad para encontrar vulnerabilidades. Las pruebas realizadas por un Hacker ético son conocidas como pruebas de penetración o *pentesting*.

HTTPS: *hypertext transfer protocolsecure*, es la versión segura de http, usa un certificado digital para codificar la sesión y evitar que sea interceptada la información. Los certificados son conocidos como los SSL y una página con HTTPS se identifica en un navegador con un candado en la parte superior.

INTERNET: es una red global donde se interconectan todas las redes que usan protocolos TCP/IP, permite el intercambio de información y una comunicación eficaz entre todos las maquinas conectadas.

LOCAL FILE INCLUSION: tipo de ataque de inclusión de archivos locales, este ataque se puede presentar en páginas web con vulnerabilidades, el ataque se presenta cuando se da entrada al envío o cargue de archivos al servidor local.

OWASP: (open web application security project) es una organización a nivel mundial dedicada a mejorar la seguridad de las aplicaciones y del software.

El proyecto OWASP define los estándares y herramientas para identificar y corregir fallas. De igual manera generar documentos para brindar ayuda en las organizaciones para mejorar capacidad de producir código seguro.

PAGINAS WEB: es un documento donde se puede encontrar video, texto, imágenes y programas, están desarrolladas en diversos lenguajes lo que permite que sean interpretados por los navegadores, también facilitan la navegación con los link o hipervínculos.

RFI (Remote File Inclusión): tipo de ataque de inclusión de ficheros remotos que pueden contener código malicioso, este ataque se aplica sobre las páginas web, para obtener el control del servidor web.

RIESGO: se define como la posibilidad de que las amenazas sean materializadas, los análisis de riesgos permite tomar los controles necesarios para la protección del sistema. Para que se presente un riesgo se debe tener una amenaza y una vulnerabilidad explotada por esa amenaza.

SITIOS WEB: corresponde a un sitio virtual, donde se puede tener archivos electrónicos y se tiene un conjunto de páginas web.

SPIDERING: es un programa que inspecciona las páginas del *World Wide Web* de forma metódica y automatizada.

SPOOFING: corresponde a la falsificación o engaño de los datos en una comunicación con uso malicioso o investigativo.

SQL INJECTION: ataque generado a una base de datos presentado por la mala filtración de las variables, el ataque inyecta un código creado por el atacante al propio código fuente de la base de datos.

SSL/TLS (Secure Sockets Layer/TransmissionLayer Security): son protocolos usados para enviar paquetes cifrados en internet. El protocolo SSL hace uso de certificados digitales para las conexiones seguras en internet. En la actualidad el TLS ha sustituido el SSL y ambos son compatibles.

VULNERABILIDAD: se define como una debilidad que puede presentar los sistemas de la información, que pueden ser aprovechadas para causar daño.

1. TITULO

PENTESTING WEB

2. INTRODUCCION

En esta monografía se describe el tema correspondiente a *pentest* enfocado a páginas web, *pentest* se define como pruebas de penetración para la seguridad de la información protegiendo los sitios web de las vulnerabilidades a las que estén expuestas.

El tema de la monografía fue elegido por ser una de las actividades principales que se deben llevar a cabo en los temas relacionados con la seguridad de la información, que implica el uso de metodologías para llevar a cabo un correcto proceso de pruebas de penetración. De igual manera es un tema donde se requiere el conocimiento y manejo de herramientas que facilitan el escaneo y monitoreo de vulnerabilidades.

Las pruebas de penetración se deben efectuar y llevar a cabo en todas las organizaciones para la protección de la información teniendo en cuenta las amenazas a que están expuestas, de manera que sean implementadas como controles en la seguridad.

Como parte de las pruebas de penetración se hace importante comprobar que un sitio es seguro en el mundo real, por lo anterior se debe probar las herramientas y métodos que usan los ciber criminales en la página, en un entorno controlado actuando como un hacker ético.

En la monografía se plantea la ejecución de pruebas de *pentesting* sobre la aplicación *SugarCRM*, herramienta usada para la administración de los clientes en varios tipos de empresas, este software maneja una versión libre con la que se va a trabajar para aplicar pruebas de penetración.

3. DEFINICION DEL PROBLEMA

3.1 PLANTEAMIENTO DEL PROBLEMA

Los hackers tienen diferentes motivos para atacar un sitio web, puede ser para obtener información con diferentes fines, facilitar el ataque con otro objetivo como puede ser un secuestro de la información, de manera que se cumpla con la finalidad de alterar el sitio web ya sea conocido o independiente de que este protegido.

Las herramientas que escanean Vulnerabilidades en Aplicaciones Web esperan encontrar configuraciones inseguras de servidores y otras vulnerabilidades de seguridad conocidas que pueden generar ataques como por ejemplo: XSS, inyecciones de SQL y accesos a directorios entre los más comunes.

Cuando un sitio web tiene vulnerabilidades conocidas como debilidades, se hace posible que los hackers se aprovechen de estas. El problema de la investigación consiste en identificar el concepto de *Pentesting* de manera que se muestre la importancia de su aplicación en la seguridad de las páginas web.

En las pruebas de *pentesting* se debe pensar mal, como lo hacen los usuarios ilegítimos para atacar y de esta manera cerrar las puertas a las fallencias de seguridad, este es el concepto principal del *pentesting* como metodología para la revisión de seguridad de la empresa.

3.2 FORMULACION DEL PROBLEMA

En la actualidad las aplicaciones web son muy importantes ya que nos presenta posibilidades ilimitadas y facilitan la vida con aplicaciones como los correos, foros, chats, blogs entre otros que permiten una mejor comunicación e interrelación.

Las amenazas en internet se incrementan día a día por lo que tener un sitio web seguro es muy importante para las organizaciones, teniendo en cuenta que pueden ser blanco de muchos ataques desde la web, algunos de estos que se pueden presentar son: *spoofing*, *SQL injection*, *cross-siteScripting (XSS)* y problemas de usuarios.

Las amenazas de seguridad evolucionan tan rápido como la tecnología, por esa razón la importancia de realizar *pentesting* para encontrar vulnerabilidades a las que pueden estar expuestas las aplicaciones web y conocer los motivos que llevan a realizar un ataque a un sitio web, como puede estar el obtener información, alterar o fisgonear entre otros.

En el mercado existen herramientas libres, usada por varias empresas y personas para facilitar las actividades, una de ellas es *SugarCRM* utilizada para manejar toda la administración de clientes, esta es una herramienta Web que puede contener vulnerabilidades y sobre la que se pueden aplicar pruebas de *pentesting*.

Con el incremento en el uso de aplicaciones web y todas las funcionalidades que ofrece, es necesario crear aplicaciones con altos niveles de seguridad, confidencialidad e integridad. De igual manera la seguridad de las aplicaciones permite proteger los datos y la pérdida de información que es considerado como uno de los activos más importantes en la actualidad. Teniendo en cuenta las descripciones anteriores, se establece el siguiente interrogante a tener en cuenta.

¿Cuál sería la manera más eficaz de identificar vulnerabilidades a un sitio web donde se cuente con la herramienta *SugarCRM*?

4. ALCANCE Y DELIMITACION DE PROYECTO

El alcance para este proyecto es presentar y dar a conocer los conceptos sobre las pruebas de *pentesting* aplicadas a sitios web, concretamente a la herramienta de relación con clientes *SugarCRM*. Para lo anterior, se presentan los conceptos correspondientes a metodologías como *OWASP* que definen una guía para ejecutar las pruebas de manera correcta.

Como parte del alcance para este proyecto se aplicarán pruebas de *pentesting* sobre toda la herramienta *SugarCRM* con versión 6.5.26, con el objetivo de usar una aplicación real que está en el mercado y es usada por algunas empresas tomando en consideración su licenciamiento libre.

Para la ejecución de las pruebas de *pentesting* es requerida la adecuación de un ambiente controlado que permita la ejecución de las mismas. Se requiere contar con la instalación y adecuación de kali Linux, que integra varias herramientas para las pruebas a ejecutar, también se debe adecuar el servidor Web donde se instalara la aplicación de *SugarCRM*.

5. JUSTIFICACION

Se debe tener en cuenta que en la actualidad, la información juega un papel muy importante y es considerado el activo más valioso en todas las organizaciones, lo cual ha generado que se dé mayor atención a la disponibilidad, confidencialidad e integridad de los sistemas informáticos para así garantizar una fluidez de información segura y sistemas protegidos.

En la actualidad las organizaciones utilizan aplicaciones orientadas a la web, lo cual implica una mayor responsabilidad y la aplicación de temas de seguridad. Para lo cual existen varias formas de hacerlo, algunas son de carácter preventivo otras de carácter reactivo, obteniendo una respuesta inmediata a un ataque.

Existen también mecanismos conocidos como el perímetro, que consiste en poner controles a los accesos de información y a los accesos del flujo de datos, otras son simulaciones de los ataques para poder identificar y medir los riesgos que se tienen desde fuera de las redes de la empresa.

SugarCRM es reconocida en el mercado y considerada como líder en herramientas CRM utilizada por más de dos millones de usuarios en más de 120 países en medianas y grandes empresas, es por esto que la presente monografía es de vital importancia teniendo en cuenta la cantidad de usuarios con que cuenta actualmente la herramienta *SugarCRM*, dado esto resulta primordial identificar el nivel de vulnerabilidad a que se ven expuestos estos usuarios.

Una aplicación web con altos niveles de seguridad, es un elemento diferenciador, generador de confianza y valor para las empresas. Diariamente se genera evolución de las TIC lo que genera nuevos retos que permitan poder aplicar políticas de seguridad necesarias para evitar intrusión en los sistemas que ocasionen pérdidas de información.

Con las especificaciones anteriores se explican las razones por las que se debe realizar pruebas de *pentesting* web, como una medida de seguridad que permita encontrar las vulnerabilidades, previniendo o cerrando brechas que pongan en riesgo la información. Esta es una de las razones más importantes de realizar pruebas de *pentesting* para una web y la razón por la cual se eligió el tema para la monografía expuesta.

El tema seleccionado quiere dar a conocer y profundizar sobre los conceptos de *pentesting* web, para lograr la aplicabilidad en temas de seguridad de la información y como mecanismo de protección de la misma. Por lo anterior, se expone el proceso y funcionalidad que hacen parte del *pentesting* web y se presentan ejemplos de pruebas *pentesting* ejecutadas sobre la aplicación web *SugarCRM*.

6. OBJETIVOS GENERAL Y ESPECIFICOS

6.1 OBJETIVO GENERAL

Dar a conocer el método de *pentesting* para combatir las vulnerabilidades o brechas de seguridad en las aplicaciones web, logrando a través de las pruebas de *pentesting* web evaluar los controles y la seguridad de nuestras aplicaciones web.

6.2 OBJETIVOS ESPECIFICOS.

- Establecer los pasos predeterminados en las pruebas de *pentesting* web para obtener mejores resultados.
- Aplicación de las fases establecidas para realizar *pentesting* web con las actividades correspondientes en cada una de ellas para lograr los resultados esperados.
- Profundizar en el manejo y funcionalidad de las herramientas para pruebas de *pentesting* web, de manera que se realice uso adecuado de estas con resultados óptimos.
- Investigar sobre la conceptualización de los *pentesting* y su aplicación en las organizaciones de manera que pueda adoptarse como práctica de seguridad.
- Conocer la importancia sobre la aplicación de *pentesting* como política de seguridad en las organizaciones y la finalidad de las pruebas de *pentesting*.
- Presentar una explicación sobre las herramientas en el mercado usadas para llevar a cabo las labores de *pentesting*.

7. MARCO REFERENCIAL

7.1 MARCO TEORICO

7.1.1. QUE ES PENTESTING. Los "Test de penetración" son un conjunto de técnicas usadas para medir, auditar y evaluar la seguridad de redes y aplicaciones web de igual manera todo lo relacionado con los sistemas de información.

Las técnicas de *pentesting* permiten probar o simular los ataques de los ciber delincuentes con el objetivo de identificar las vulnerabilidades, conocer las brechas de seguridad a las que se puede estar expuesto para aplicar los controles necesarios.

7.1.2 PERFIL DEL PENTEST. Los profesionales en pruebas de penetración deben pensar de manera maliciosa, pero de forma profesional para hacer el bien sobre las aplicaciones web a las que se le ejecuten pruebas de penetración. Estos profesionales requieren tener conocimientos sobre tecnologías, aplicaciones, servicios y sistemas operativos. Normalmente a los encargados de las pruebas de *pentesting* se les llaman Hacker con una clasificación diferente para estos, que se mencionara posteriormente.

7.1.2.1 Hackers de Sombre Blanco: Son hacker con sentido ético, encargado de investigaciones de seguridad con fines no maliciosos. Permiten encontrar vulnerabilidades para indicar y demostrar la falta de seguridad presentada en una organización o para probar sus propios códigos.

7.1.2.2 Hackers de Sombre Negro: Son hacker que violan la seguridad para obtener provecho personal, se puede decir que son los causantes de los virus *malware*, *ransomware* etc. Son expertos en seguridad que se aprovechan de las vulnerabilidades para robar información, modificarla o secuestrarla.

7.1.2.3 Hacker de sombrero gris: Son hacker que pueden actuar impulsados con dos intenciones, una de ellas puede ser para fines personales o también para ayudar en la seguridad de una organización y de acuerdo con el objetivo de las pruebas de penetración, se presenta la clasificación del hacker.

Para realizar pruebas de penetración en las aplicaciones web es importante y requerido conocer sobre las tecnologías web y lograr un análisis desde la perspectiva del desarrollador o administrador web, para analizar cuáles podrían ser los errores cometidos por desarrolladores, administradores u operadores del sistema.

7.1.3 TIPOS DE PRUEBAS DE PENETRACION: A continuación, se describen los diferentes tipos de pruebas de penetración.

7.1.3.1 Caja Negra: Este tipo de pruebas realizan una evaluación de la seguridad y las pruebas que se realizan no requieren el conocimiento sobre la estructura o infraestructura, es decir, que no se conoce el funcionamiento del sistema. El objetivo de estas pruebas es que se realice las pruebas simulando un posible atacante externo.

7.1.3.2 Caja Blanca: Se realizan por un equipo interno, para este tipo de pruebas se cuenta con total conocimiento de la funcionalidad del sistema interno, sistemas operativos, arquitectura etc. Se observa que se tiene acceso al código fuente se puede revisar y reportar vulnerabilidades encontradas.

7.1.3.3 Caja Gris: En este tipo de pruebas, los *pentesters* cuentan con conocimientos de aspectos funcionales e internos del sistema. El objetivo de estas pruebas es simular un empleado para encontrar problemas o vulnerabilidades que pueden aprovechar los usuarios internos.

7.1.4 PRUEBAS DE PENETRACION A APLICACIONES WEB

7.1.4.1 Aplicaciones WEB: En la actualidad las Aplicaciones Web se convierten en uno de los principales objetivos de ataques por su gran utilización debido a todas las funcionalidades que ofrecen y también por al crecimiento de Internet que es imparable, por ser fuentes de información.

Las aplicaciones web son usadas en varios escenarios, como lo son: almacenar, gestionar y acceder a datos sensibles y manejo de información personal, ofreciendo beneficios como portabilidad, y variedad de funcionamiento en todos los sistemas operativos.

Con el gran uso de las nuevas tecnologías que día a día se van desarrollando, los sitios web son cada vez más usados, contando que las aplicaciones web cuentan con varias funcionalidades, lo cual las hace que sean vectores de ataque.

En la actualidad las empresas se centran en realizar pruebas sobre la funcionalidad de las aplicaciones desarrolladas para la web del negocio, es decir, pruebas de funcionalidad, pero raramente realizan pruebas de seguridad.

7.1.4.2 Pruebas de *Pentesting*: Las pruebas de penetración de aplicaciones web, buscan validar que estas sean seguras, para lograr su función simulan el acceso de

usuarios no autorizados y audita toda la aplicación de manera que el sitio web no esté en riesgo de ataques sobre la información. Como buena práctica es necesaria que las pruebas de penetración sean ejecutadas con el uso de una metodología válida, aprobada y explicable.

Por otro lado, para la realización de las pruebas de penetración en aplicaciones web es necesario tener conocimientos sobre tecnologías web, conceptos de desarrollo, infraestructura, seguridad en servidores y sistemas operativos. Esto con el fin de una correcta y acertada planeación de las pruebas.

7.1.4.3 Alcance de las pruebas de *Pentesting* en Aplicaciones Web. Se presenta a continuación una breve lista con las actividades comprendidas en las pruebas:

- Comprobar que las medidas de seguridad sean implementadas de forma correcta.
- Con las pruebas de penetración se espera identificar vulnerabilidades que pueda presentar la versión 6.5.26 de *SugarCRM*, con la finalidad de documentarlas y contribuir con su mitigación.
- Lograr la catalogación de los riesgos teniendo en cuenta las vulnerabilidades encontradas.
- Demostrar y comprobar los ataques que se pueden presentar en las aplicaciones web.
- Realiza pruebas de SQL Injection, Cross Site Scripting (XSS) y otras vulnerabilidades.
- En las pruebas de penetración se valida todas las vulnerabilidades Web y ataques que se pueden presentar, como por ejemplo: Encontrar directorios con permisos débiles, detecta aplicaciones web populares y versiones vulnerables, detecta si existen métodos de HTTP peligrosos habilitados en el servidor web.

7.1.5 METODOS DE PENTESTING: A continuación, se relacionan los métodos existentes

Manual: Es llevada a cabo de manera directa para conocer y comprender el ataque. Se hace uso de scripts sencillos y herramientas. Al ser un método manual es más lento comparado con otros métodos.

Automático: Es un método que utiliza herramientas para llevar a cabo el escaneo de un sitio y devolver las vulnerabilidades encontradas de una manera rápida y ágil sin mayor intervención del usuario o *pentester*.

Híbrido: Con este método se integran los dos anteriores, utiliza métodos manuales y automáticos. Utiliza herramientas para el escaneo de la aplicación y en paralelo se van realizando revisiones manuales.

7.1.6 FASES DE TEST DE PENETRACION: Las fases para las pruebas de *pentesting* son las siguientes:

7.1.6.1 Fase de reconocimiento: En esta fase se hace uso de herramientas de análisis para obtener toda la información del objetivo. En esta fase se deben realizar las siguientes actividades:

- Se deben definir objetivos para las pruebas de *pentesting*.
- Recopilar toda la información de manera que pueda ser usada en las demás fases.
- La información consultada abarca: nombres, direcciones de correos, topología de red, direcciones IP.

7.1.6.2 Fase de enumeración: En esta fase solo se realiza actividades de investigación, aun no se lleva a cabo ningún ataque. Las actividades desarrolladas en esta fase son:

- Se debe hacer un checklist o inventario de los elementos existentes como: sistemas operativos, servicios y aplicaciones.
- Se identifican las direcciones IP, usuarios y claves y *tokens* de sesión.
- Se debe comprender el flujo de la aplicación y cómo interactúan los demás elementos.

7.1.6.3 Fase de análisis: En esta fase se da inicio a la interacción y análisis con los sistemas encontrados, El análisis se realiza para encontrar vulnerabilidades, consultadas a nivel de la infraestructura, los sistemas operativos, los servicios disponibles o las

aplicaciones existentes. El objetivo en esta fase es llevar a cabo la planificación del ataque.

7.1.6.4 Fase de Explotación: En esta fase se inicia la intrusión en el sistema para obtener evidencia de las actividades y pasos ejecutados en las pruebas de intrusión realizada.

Con la evidencia que soporta la intrusión, posteriormente se debe generar toda la documentación.

La explotación se realiza sobre aquellas vulnerabilidades analizadas y detectadas en las fases anteriores.

7.1.6.5 Fase de Documentación: Las actividades de esta fase corresponden a realizar la documentación correspondiente al ataque realizado, en la documentación se debe tener en cuenta lo siguiente:

- Documentar todos los descubrimientos encontrados en el sistema.
- Realizar informes de las intrusiones
- Diseñar presentación de los resultados.
- Indicar los puntos donde se generen los problemas más relevantes.
- Como buena práctica se debe documentar con cada intrusión realizada.

7.1.7 HERRAMIENTAS DE PRUEBAS DE *PENTESTING*: Existen varias herramientas que apoyan las tareas para las pruebas de *pentesting*, estas herramientas permiten realizar actividades como:

- Escanear puertos: Herramienta que permite realizar escaneo de puertos con información de puertos abiertos y cerrados, para ubicar vulnerabilidades en el host o red.
- Descifrar contraseñas: Son herramientas que permite adivinar o recuperar contraseñas que puedan estar almacenadas en un servidor, con el objetivo de comprobar la seguridad de sistemas y plataformas.
- Realizar ataques por fuerza bruta: Con los ataques de fuerza bruta se logra conocer páginas o contenidos no seguros o autorizados en un servidor web, al igual que es el método más usado para el craqueo de contraseñas.
- Herramientas de *sniffing*: Estas herramientas permiten la captura de paquetes o tramas, ofreciendo diversos usos como pueden ser: Revisión del tráfico de una red

para identificar fallas y permitir la medición del mismo. Detectar intrusos o analizar información real que viaja en la red.

A continuación, en la tabla 1 se mencionan las más conocidas en el mercado.

Tabla 1- Herramientas para *pentesting*

Nombre	Descripción - Funcionalidad
SQLmap	Herramienta Open Source que a través de línea de comando permite la explotación de vulnerabilidades como por ejemplo SQL injection utilizado para obtener información de las bases de datos.
Nessus	Herramienta usada para pruebas de <i>pentesting</i> en aplicaciones Web, permite realizar escaneo para encontrar vulnerabilidades, genera alertas y sugiere las posibles soluciones a las alertas encontradas.
Burp Suite	Herramienta grafica usada para probar la seguridad de las aplicaciones web, funciona como un servidor proxy que permite analizar las peticiones y hacer test de intrusión.
OWASP Zed	Herramienta gratuita utilizada para monitorear la seguridad en aplicaciones web, hace parte de los proyectos de OWASP
Wireshark	Es una herramienta de código abierto usada para analizar y auditar una red local, con el objetivo de encontrar vulnerabilidades.
John TheRipper	Herramienta usada para descifrar o encontrar contraseñas, aplicando ataques de fuerza bruta.
THC Hydra	Herramienta que permite realizar acceso remoto a los sistemas por medio de los protocolos de red.

Fuente el autor

7.2 MARCO CONCEPTUAL

A continuación, se detallan los conceptos de la guía de OWASP, relacionados con el tema de seguridad en Web. Esta es una guía donde se describe lo relacionado a pruebas de penetración a aplicaciones Web en concreto.

7.2.1 Guía de Owasp. El objetivo del proyecto de OWASP es lograr ayudar en el entendimiento del por qué, cuándo, dónde y cómo se deben realizar las pruebas de las aplicaciones web

7.2.2 Capitulo 4 – Pruebas de Intrusión de Aplicaciones Web. En este capítulo se expone en OWASP como realizar las pruebas de intrusión en aplicaciones Web.

¿Qué es una prueba de intrusión de aplicación web?

Las pruebas de intrusión están enfocadas a la evaluación de la seguridad de una aplicación web, simulando ataques y teniendo en cuenta que las incidencias de seguridad encontradas deben ser presentadas junto con una evaluación del impacto y propuesta de mitigación.

¿Qué es una vulnerabilidad?

“Una vulnerabilidad es una debilidad que se presenta por diferentes causales y puede terminar o generar una amenaza, que puede generar daños y afecta la seguridad de la información.

¿Qué es la metodología de pruebas OWASP?

Es una metodología orientada a la seguridad de aplicaciones web y utilizada para auditorías. Con la aplicación de esta metodología se espera obtener y entender las técnicas descritas en la guía para generar pruebas de intrusión, las cuales se basan en pruebas de caja negra.

7.2.3 Fases de las pruebas. Modo Pasivo: Aplica cuando se realiza una exploración de la aplicación de manera que se comprende el flujo de la aplicación y se familiariza con ella, de esta manera logra conocer todos los puntos de accesos de la aplicación.

Modo Activo: Se inicia con la ejecución de las pruebas de intrusión con las metodologías existentes.

7.2.4 Definición de las pruebas en subcategorías según OWASP V4

- Recopilación de información
- Pruebas de gestión de la configuración y despliegue
- Pruebas de seguridad a la gestión de la identidad
- Pruebas de seguridad al proceso de autenticación
- Pruebas de seguridad al proceso de Autorización
- Pruebas de seguridad al proceso de gestión de sesiones
- Pruebas de seguridad a la validación de entrada
- Pruebas de seguridad al manejo de errores
- Pruebas de seguridad a los mecanismos criptográficos
- Pruebas de seguridad a la lógica de negocio
- Pruebas de seguridad del lado del cliente

7.2.5 Recolección de Información. OWASP define técnicas o procesos para la recolección de la información, teniendo en cuenta que es la primera fase que se debe aplicar en las pruebas de intrusión de manera que se recolecte la mayor información que se pueda sobre el objetivo a atacar.

Tabla 2. OWASP Recopilación de información

Numero de ref.	Nombre de Prueba	Recopilación información
OWASP-IG-001	Spider, robot y Crawlers	Consiste en navegar y capturar recursos.
OWASP-IG-002	Descubrimiento/Reconocimiento mediante motores de búsqueda	Se aplica con la búsqueda de incidencias expuesta públicamente
OWASP-IG-003	Identificación de puntos de entrada de la aplicación	Identificar y catalogar secciones de la aplicación que se consideren deben ser investigadas.
OWASP-IG-004	Pruebas de firma digital de Aplicaciones Web	Determinar la firma de la aplicación, la versión y el tipo de servidor web, permite determinar las vulnerabilidades conocidas y exploit.
OWASP-IG-005	Descubrimiento de Aplicaciones	Identificar las aplicaciones Web que están instaladas en un Servidor Web, Perite revelar otros detalles.

Tabla 2. (Continuación)

Numero de ref	Nombre de Prueba	Recopilación información
OWASP-IG-006	Análisis de Códigos de Errores	En las pruebas de intrusión, la aplicación web releva información de códigos de error, tecnologías y productos usados en la aplicación.

Fuente: <https://goo.gl/8aD5yJ>

7.2.6 Kali Linux. Es una distribución diseñada para revisar temas de la seguridad informática, al ser Linux, es de código abierto y gratuito. Es todo un sistema operativo que contiene muchas herramientas para la auditoría informática. *Kali Linux* es considerada como la evolución de *Backtrack* 6, a diferencia de que esta reescrita en debían por considerarse más segura y eficiente y no en Ubuntu.

7.3 MARCO LEGAL

7.3.1 LEYES EN COLOMBIA DE LA PROTECCION DE DATOS. En Colombia se tiene la ley 1273 de 2009, donde está incluido el tema correspondiente a los delitos informáticos y se crean los siguientes tutelados:

7.3.1.1 Capítulo 1: De la protección de la información y de los datos. Donde define los artículos correspondientes a la protección de la información.

A continuación, se relacionan algunos de los artículos con la descripción correspondiente:

Desde el artículo 269A al artículo 269 H, relaciona todo el tema a la protección de la información, como robo de datos personales o suplantación de estos, el generar daños a la información o el uso de software malicioso, y temas relacionados con ingresos sin autorización y abuso de estos.

A continuación, se describe uno de los artículos más relacionados con el tema de ataques a las WEB.

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 programas para el ingreso sin autorización a sitios web

7.3.1.2 Capítulo2: De los atentados informáticos y otras infracciones. Se exponen dos artículos 269I y 269J, donde se trata el tema de robos por medios informáticos, manipulando los sistemas y la transferencia no autorizada de activos por medio de alguna manipulación informática.

8 PROPUESTA METODOLOGICA

8.1METODOLOGIAS

A continuación, se definen metodologías claves en la aplicación del proyecto de *pentesting* web. Con la Metodología OSSTMM se define como llevar a cabo las pruebas de *Pentesting*.

8.1.1 Metodología OSSTMM. En las pruebas de *Pentesting*, se puede aplicar esta metodología. Para elegir el tipo de test adecuado, hay que conocer primero como están diseñados los modelos para trabajar, se debe evaluar el tipo de negocio y el tiempo disponible.

Existen cuatro fases para la ejecución de esta metodología:

- Fase Regulatoria: Se tiene en cuenta los requerimientos y límites de las pruebas, también se define el alcance y las restricciones.
- Fase de Definiciones: En esta fase se define claramente el alcance de las pruebas.
- Fase de información: Muchas de las pruebas son sobre la información que no está cubierta. Los varios tipos de valores, así como los activos mal ubicados y desentendidos y mal gestionados.
- Fase interactiva de pruebas de controles: Es la fase final de las pruebas de pentesting, se debe confirmar y demostrar lo que fue encontrado en los resultados.
-

Características de la Metodología

La metodología esta subdividida en aspectos de sistemas de información:

- Seguridad de la información
- Seguridad de los procesos
- Seguridad en las tecnologías de internet
- Seguridad en comunicaciones
- Seguridad inalámbrica
- Seguridad Física.

8.1.2 Metodología ISSAF: Es una metodología poco conocida, Ofrece una guía dividida por escenarios de acuerdo con el tipo de servicio como MySQL, Oracle, entre otros.

Fases de la Metodología

- Planificación y preparación: Preparación para la prueba
- Evaluación: Se ejecuta el test de penetración, teniendo en cuenta las siguientes capas:
 - Recolección de información
 - Mapeo de la red de trabajo
 - Identificación de vulnerabilidades
- Penetración
- Obtener Acceso y escalada de privilegios
- Enumeración adicional
- Comprometer usuario remotos y sitios
- Reportes, limpieza y Destrucción de objetos: Esta fase corresponde a la entrega de resultados y entrega de reportes correspondientes.

La división por servicios la convierte en poco funcional por no estar completa y utilizar varios servicios.

8.1.3 OWASP Top 10: De acuerdo al top de OWASP , se describen las 10 vulnerabilidades mas criticas en las aplicaciones Web. El proyecto del top 10, busca generar conciencia de la seguridad en aplicaciones Web, por tal razon el listado de las 10 vulnerabilidades expuesto por OWASP corresponde a los riesgos que estan expuestos las organizaciones.

A continuacion se relaciona el listado de las vulnerabilidades según OWASP 2017:

- A1 - Inyección
- A2 - Perdida de autenticación y gestión de Sesiones
- A3 – Secuencia de comandos en Sitios Cruzados (XSS)
- A4 – Referencia Directa Insegura a Objetos
- A5 – Configuración de Seguridad Incorrecta
- A6 – Exposición de Datos Sensibles
- A7 – Ausencia de Control de Acceso a las funciones
- A8 – Falsificación de Peticiones en sitios (CSRF)
- A9 – Uso de Componentes con vulnerabilidades Conocidas
- A10 – Redirección y reenvíos no válidos.

OWASP, describe de que se trata la vulnerabilidad, como se presenta, expone ejemplos de escenarios del ataque y como se puede prevenir, para cada una de las 10 vulnerabilidades identificadas y definidas.

8.1.3.1 Resumen de los Factores de riesgo Top 10. De igual manera se presenta un resumen expuesto por OWASP donde se observan los factores de riesgo determinados por los estudios y estadísticas generadas. Para cada una de los 10 riesgos definidos se expone: Agentes de amenazas, Prevalencia, Debilidades, Impacto técnico y de negocio. De igual manera se expone otros riesgos a considerar.

Figura 1 Comparativo Top 10 OWASP

OWASP Top 10 2013	±	OWASP Top 10 2017
A1 – Inyección	➔	A1:2017 – Inyección
A2 – Pérdida de Autenticación y Gestión de Sesiones	➔	A2:2017 – Pérdida de Autenticación y Gestión de Sesiones
A3 – Secuencia de Comandos en Sitios Cruzados (XSS)	➔	A3:2017 – Exposición de Datos Sensibles
A4 – Referencia Directa Insegura a Objetos [Unido+A7]	U	A4:2017 – Entidad Externa de XML (XXE) [NUEVO]
A5 – Configuración de Seguridad Incorrecta	➔	A5:2017 – Pérdida de Control de Acceso [Unido]
A6 – Exposición de Datos Sensibles	➔	A6:2017 – Configuración de Seguridad Incorrecta
A7 – Ausencia de Control de Acceso a las Funciones [Unido+A4]	U	A7:2017 – Secuencia de Comandos en Sitios Cruzados (XSS)
A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)	✗	A8:2017 – Deserialización Insegura [NUEVO, Comunidad]
A9 – Uso de Componentes con Vulnerabilidades Conocidas	➔	A9:2017 – Uso de Componentes con Vulnerabilidades Conocidas
A10 – Redirecciones y reenvíos no validados	✗	A10:2017 – Registro y Monitoreo Insuficientes [NUEVO, Comunidad]

Fuente: <https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf>

De acuerdo al cuadro comparativo de la evaluación del Top 10 de OWASP para el 2013 y 2017, se confirma que el ataque de Inyección sigue siendo el primero en la evaluación de los ataques más frecuentes que se presentan en los servidores Web. De igual manera en la evaluación para el 2017 aparecen nuevos ataques producto de nuevas tecnologías y nuevas arquitecturas para las aplicaciones.

9. DESARROLLO DE PRUEBAS

En la monografía se definen los siguientes pasos requeridos para el desarrollo del tema de esta y para el cumplimiento de los objetivos.

9.1 Configuración de Ambientes

Para la ejecución de las pruebas es requerido que se adecue un ambiente para las mismas, por lo anterior se realiza la instalación de *Kali Linux*, ya que cuenta con gran cantidad de herramientas para realizar pruebas de *pentesting*.

Se requiere la instalación de la aplicación *Sugar CRM* seleccionada como herramienta de ejemplo sobre la cual se van a aplicar las pruebas de *pentesting*, para lo anterior se sigue el paso a paso descrito por los fabricantes para la instalación sobre un servidor Web adaptado en una maquina virtual con Windows 8.

9.2 Ejecución de las pruebas

Teniendo instalado y configurado el ambiente con la herramienta *Sugar CRM* en la máquina virtual, se procede a la ejecución de las pruebas de *pentesting* realizando ejemplos de ataques con el uso de algunas herramientas desde *Kali Linux* y con algunos complementos que ofrecen los navegadores, con previa selección de los tipos de pruebas a aplicar según OWASP.

9.3 Consolidación de los resultados

Posterior a la ejecución de las pruebas de *pentesting* aplicadas sobre *SugarCRM*, se genera consolidado de los resultados obtenidos en la práctica para cada una de las pruebas, con la descripción de los detalles correspondientes a la obtención del resultado y la correspondiente tipificación que indica si es o no vulnerable.

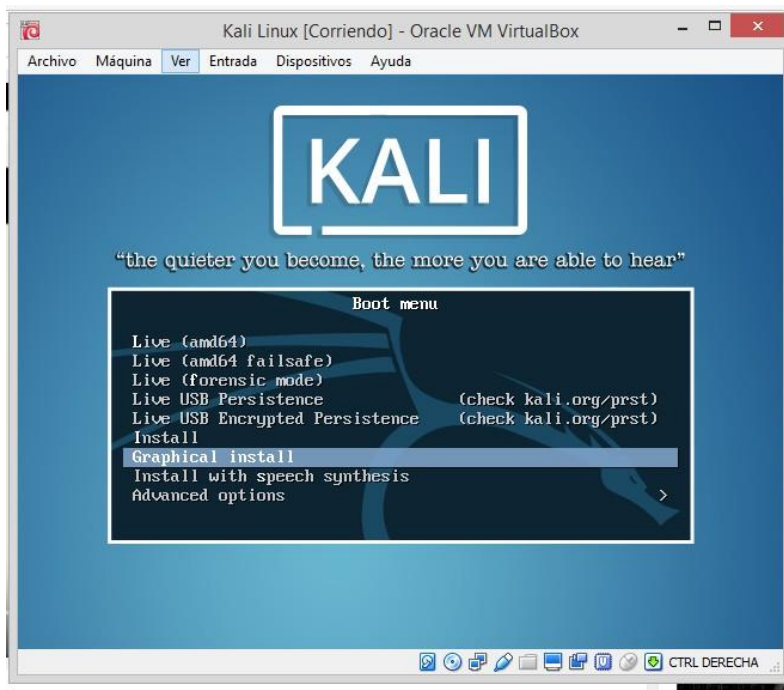
10. ADECUACION DE AMBIENTES

10.1 INSTALACION DEL SOFTWARE REQUERIDO.

A continuación, se describe la instalación del software requerido para cada uno de los ambientes que se requieren para dar inicio a la ejecución de las pruebas de *pentesting* sobre la aplicación de *SugarCRM*, iniciando con la instalación de Kali Linux y posteriormente la de *SugarCRM*.

10.1.1 Instalación de *Kali Linux*: La instalación de Kali Linux se realiza en una maquina virtual utilizando la herramienta Virtual Box para el manejo de maquinas virtuales, todo instalado desde un PC portátil con sistema operativo Windows 10.

Figura 2 Instalación kali Linux

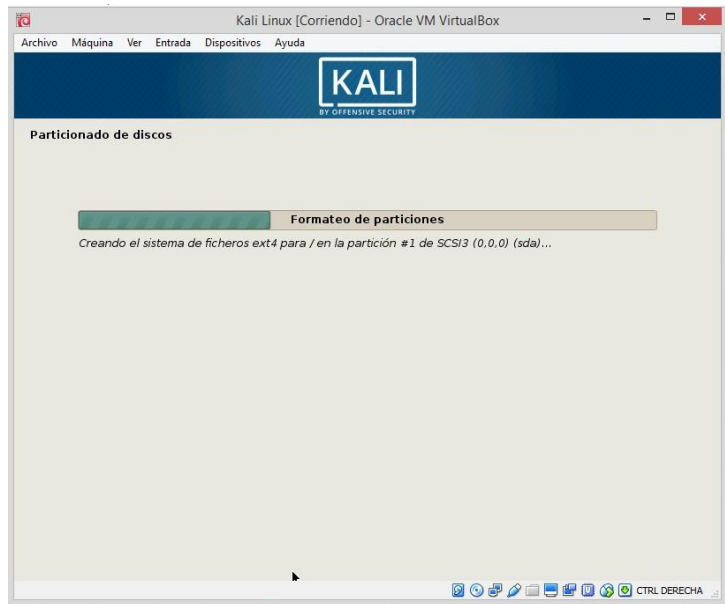


Fuente: El autor

En la figura 2 se observa la primera pantalla para dar inicio a la instalación de Kali Linux con la selección de la opción de Graphical install (entorno grafico), se continua con cada uno de los pasos solicitados en la instalación con la selección del lenguaje: Español, ubicación: Colombia, configuración del teclado: Español, nombre de la maquina virtual: Kali, datos para la creación del Súper usuario y clave. Como otro punto importante se

seleccionan opciones de particiones y partición de discos para dar por terminada la instalación. En la parte de configuración donde se solicita el nombre del dominio se deja vacío la opción es decir no se coloca nombre y no presenta inconveniente ya que no es un campo que sea requerido.

Figura 3. Finaliza la instalación



Fuente: El autor

En la figura 3 se observa una parte de la instalación del software de Kali Linux, una vez terminada se presenta mensaje de confirmación de la instalación exitosa del sistema, para el correspondiente ingreso con los datos del Súper usuario y clave que se crearon durante la instalación.

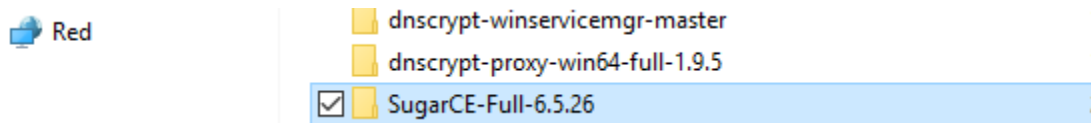
10.1.2 Instalación de *Sugar CRM* en Windows 8

Sugar CRM es una herramienta que permite a las empresas mantener toda la información de sus clientes, almacenándola de una manera más organizada y estructura. Esta herramienta tiene una versión *OpenSource* y también versiones de pago (Enterprise y Profesional).

Las funcionalidades principales del producto *OpenSource* son: Gestión de ventas, Gestión de marketig y Gestión de clientes. A continuación se describen los pasos para la instalación.

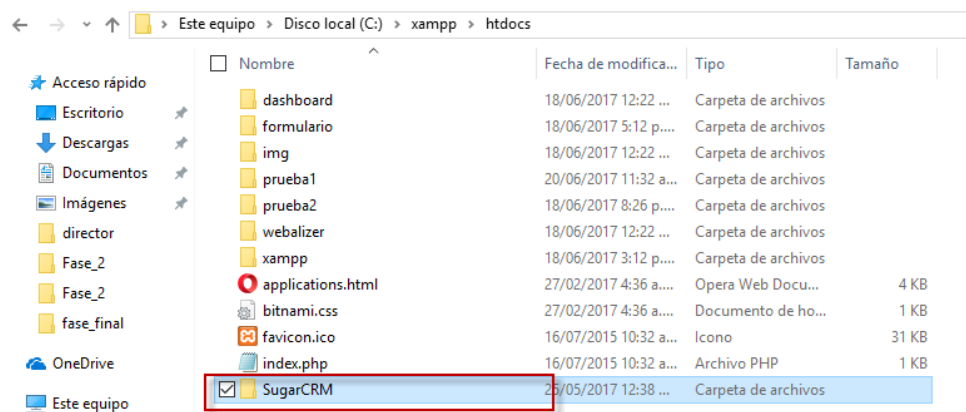
Desde un servidor Web configurado en Windows 8 que se encuentra instalado sobre una máquina virtual, se realiza la descarga de *SugarCRM*. A continuación se especifica la URL donde se realiza la descarga del software para la adecuación del ambiente de pruebas: <https://sourceforge.net/projects/sugarcrm/files/latest/download?source=files>. Luego de realizada la descarga del software, se descomprime y se renombra la carpeta utilizando cualquier nombre, en este caso queda renombrada como *SugarCRM*.

Figura 4 Descarga de sugarcrm



Fuente: El autor

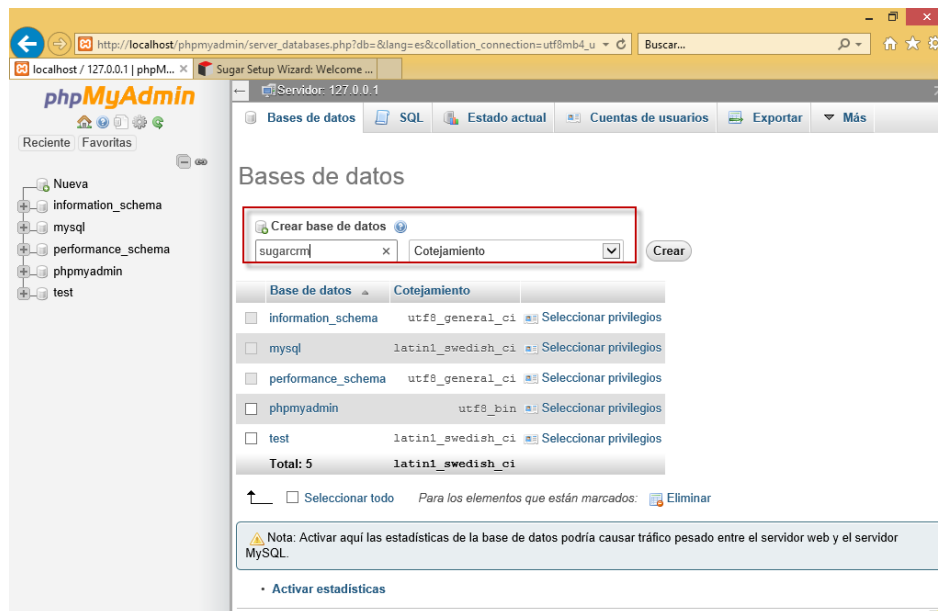
Figura 5 Renombrar el archivo descarga sugar



Fuente: El autor

La carpeta descomprimida y renombrada se debe mover a HTDOCS, en este caso particular, dado que se tiene instalado el servidor web sobre XAMPP. Para la creación de la Base de datos como aparece en la figura 6, se ingresa a phpmyadmin y se crea la base de datos con el nombre de *sugarcrm*, con la opción de cotejamiento.

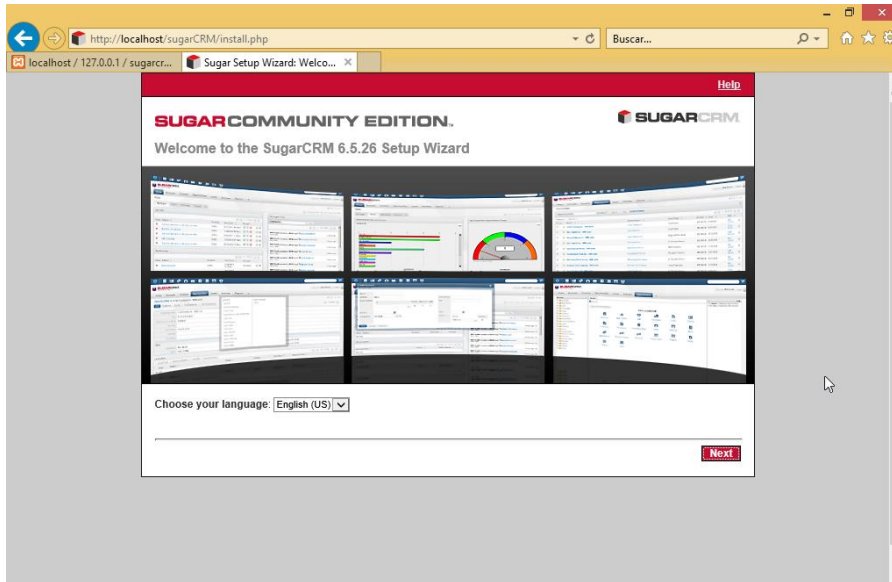
Figura 6 Creación BD sugar



Fuente: El autor

En la figura 7 se observa el ingreso para realizar la instalación en la URL: <http://localhost/SugarCRM/install.php>. El idioma no permite la modificación en este punto de la configuración, de acuerdo a especificación es posible modificarlo posteriormente, en otro punto de la configuración. Para terminar se debe dar click al botón next.

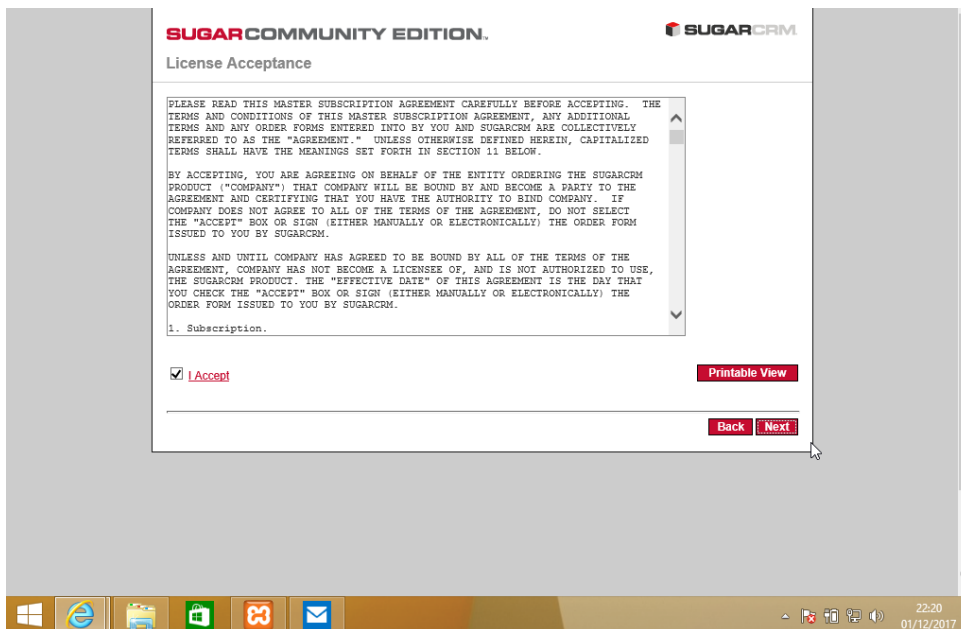
Figura 7 Instalación Sugar CRM



Fuente: El autor

En la figura 8 se visualiza la informacion sobre *SugarCRM* que corresponde a temas de la licencia de la aplicación, con la opcion para seleccionar la aceptacion de terminos y condiciones y continuar con el boton next la configuracion que se valla solicitando.

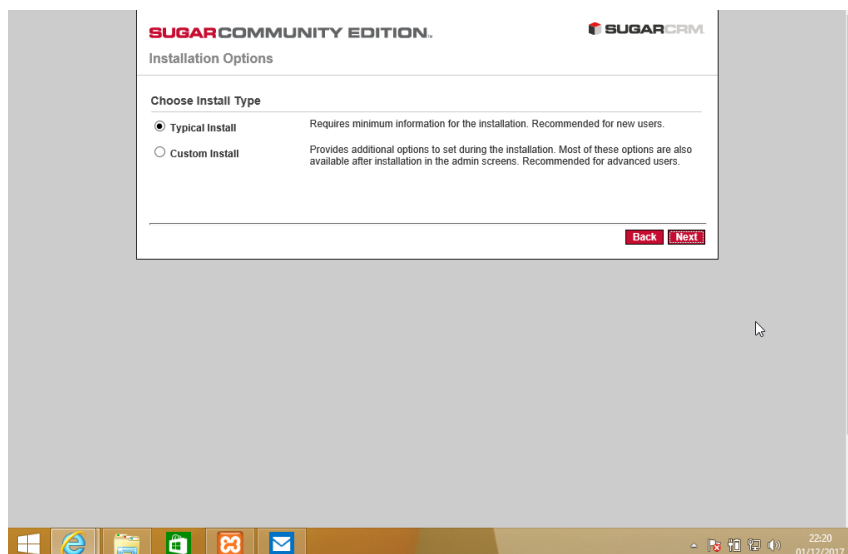
Figura 8 Aceptar términos y condiciones sugar



Fuente: El autor

En la siguiente figura 9 se debe seleccionar la opción de *typical* para que tome la instalación por defecto establecida. La otra opción que corresponde a Custom Install, solicita mas información en la instalación por lo anterior es mas recomendada para usuarios con mas experiencia o conocimiento en la aplicación.

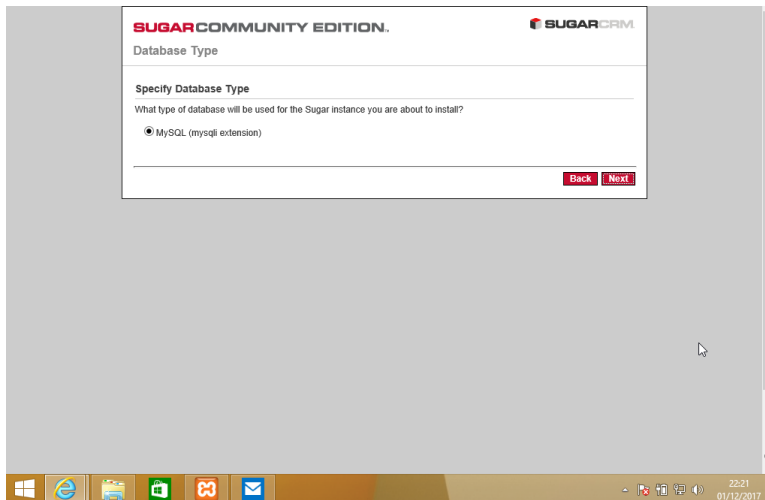
Figura 9 Selección de opción de instalación sugar



Fuente: El autor

La figura 10 que aparece posteriormente es informativa no permite modificar ningún parámetro, por defecto aparece la Base de Datos MySQL, que corresponde a la Base de Datos usada para la aplicación de *SugarCRM.*, por lo anterior no presenta más bases de datos para que sean seleccionadas.

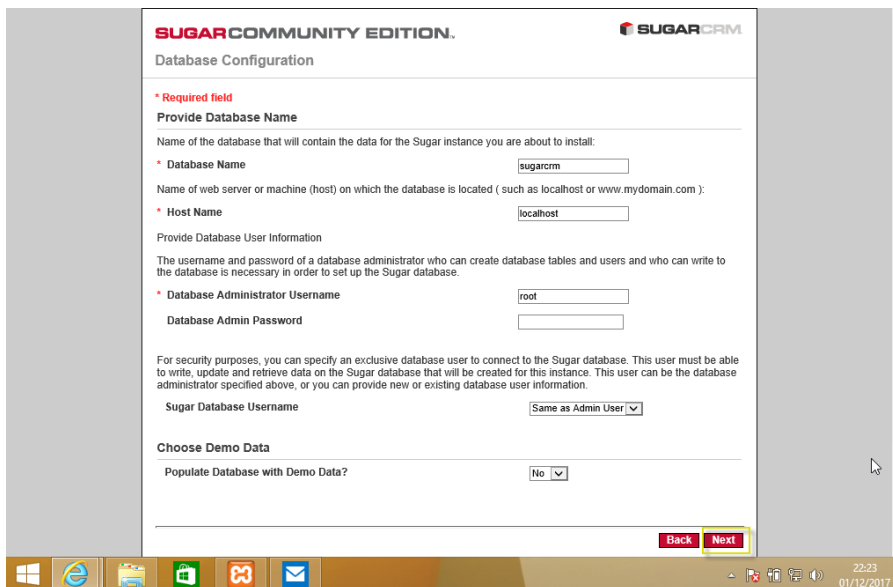
Figura 10 Confirmación instalación sugar



Fuente: El autor

En la figura 11, solicita la información requerida para continuar con la instalación. Se debe diligenciar la información correspondiente al nombre de la base de datos, nombre del host, usuario administrador de la misma y la información de otro usuario con permisos sobre la base de datos.

Figura 11 Configuración SugarCRM

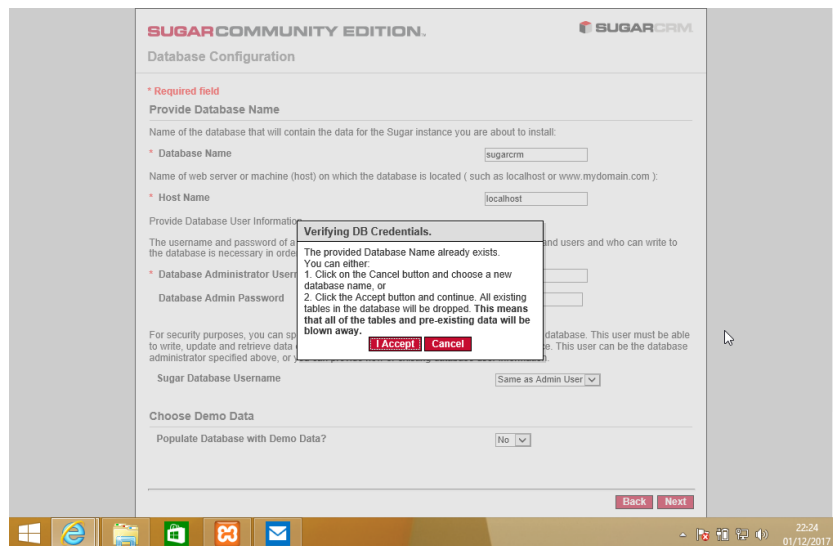


Fuente: El autor

En la figura 12 se confirma la creación exitosa de la Base de datos de *SugarCRM*, realizada con el proceso anterior descrito y los datos requeridos y ingresados. En

la figura aparece mensaje de confirmacion para las credenciales de la base de datos con la opcion de aceptar o cancelar.

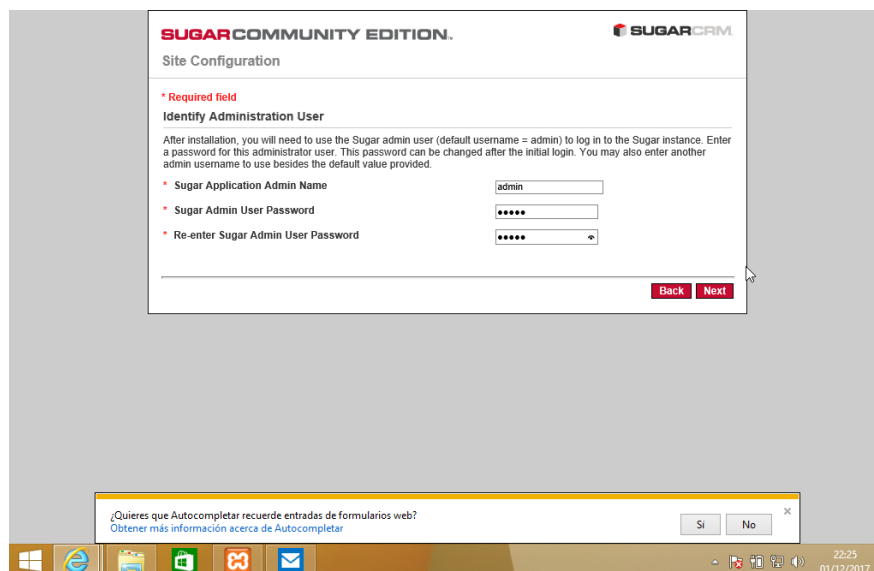
Figura 12 verificacion DB Credentials



Fuente: El autor

En la figura 13 se solicita la creación de un usuario administrador para el ingreso a la aplicación cuando sea solicitado el logueo. Los datos ingresados corresponden al usuario que va tener todos los privilegios sobre la aplicación de *SugarCRM*, de manera que pueda realizar todas las acciones de creación, modificación y lectura.

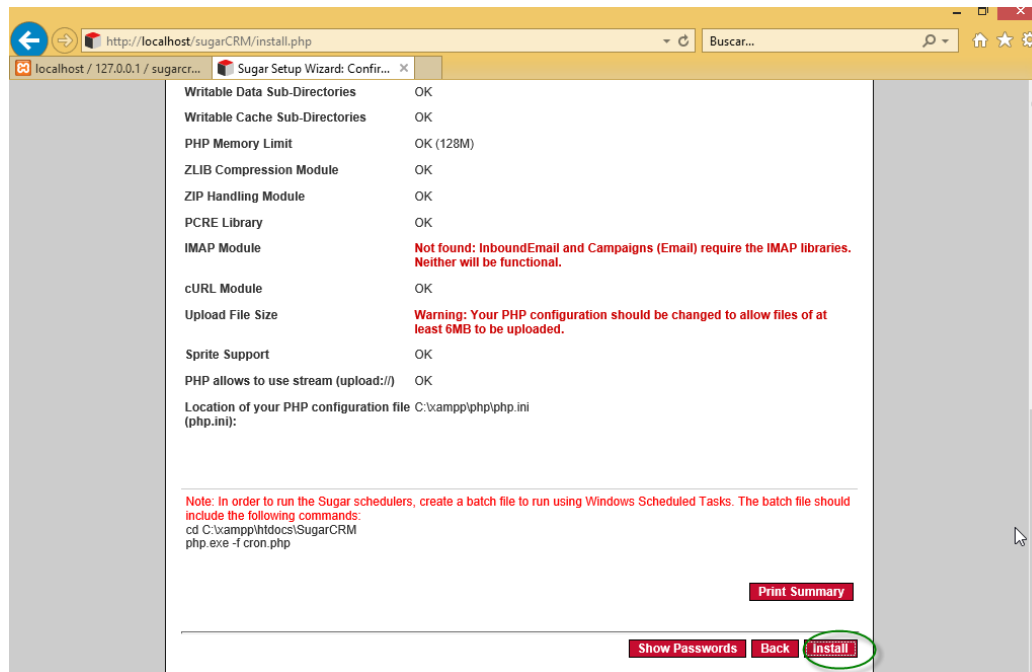
Figura 13 Creación de usuario Administrador



Fuente: El autor

En la figura 14, se confirma la información ingresada en los pasos anteriores para la configuración de *SugarCRM*. Se indica con OK la configuración que se aplicaron de manera correcta y para algunas opciones un mensaje de *warning*, es un checklist para validar la información requerida antes de confirmar la instalación final, también habilita opciones para visualizar *passwords* o regresar.

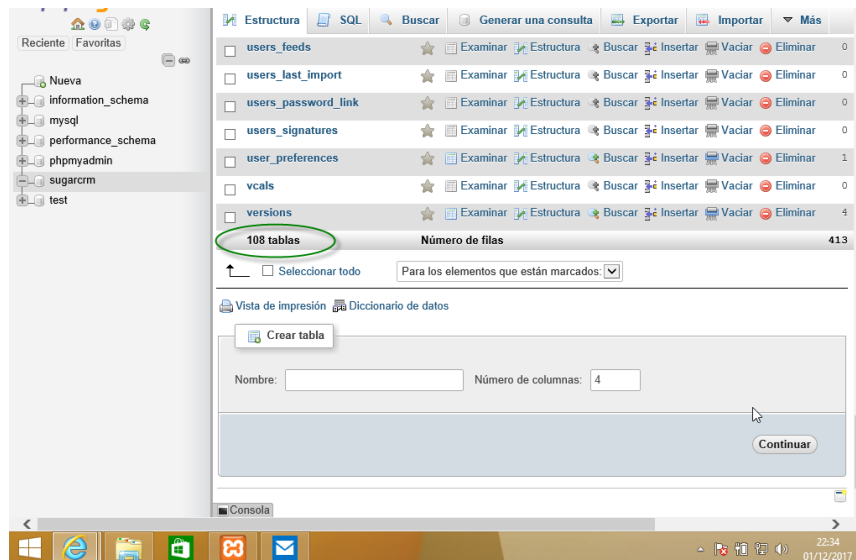
Figura 14 Confirmación de Configuración Sugar



Fuente: El autor

Una vez finalizada la instalación, se valida que en la Base de datos *SugarCRM* estén creadas todas las tablas requeridas para la aplicación *SugarCRM* como se observa en la figura 15. En total se crean 108 tablas, indicando los nombres de cada una de ellas y totalizando la cantidad de las tablas de la aplicación de *SugarCRM*.

Figura 15 Confirmación de la Creación tablas BD

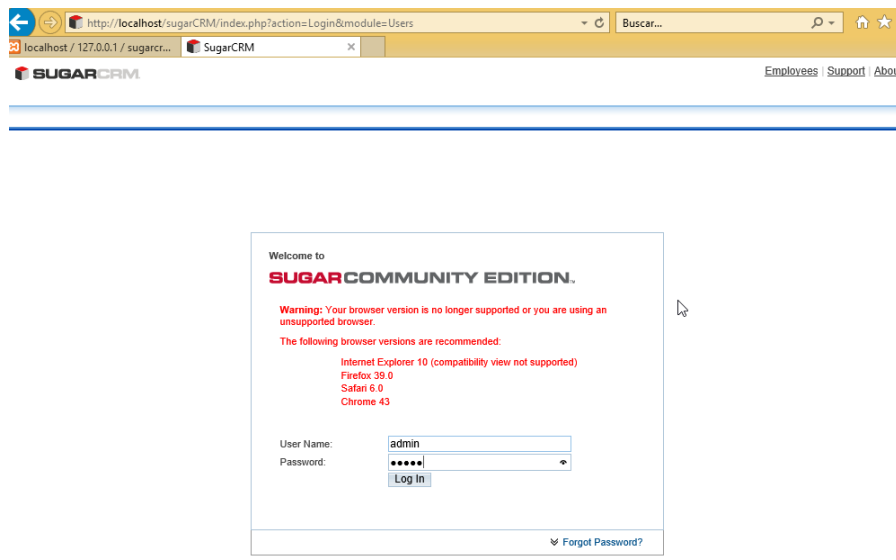


Fuente: El autor

10.1.3 Configuración de *SugarCRM*

Con los pasos anteriores ya queda instalada la aplicación, de manera que ya se pueda realizar el ingreso y dar inicio a los pasos de configuración solicitados de acuerdo a lo que se quiera emplear y ofrezca la aplicación teniendo en cuenta también la necesidad de las empresas y los usuarios.

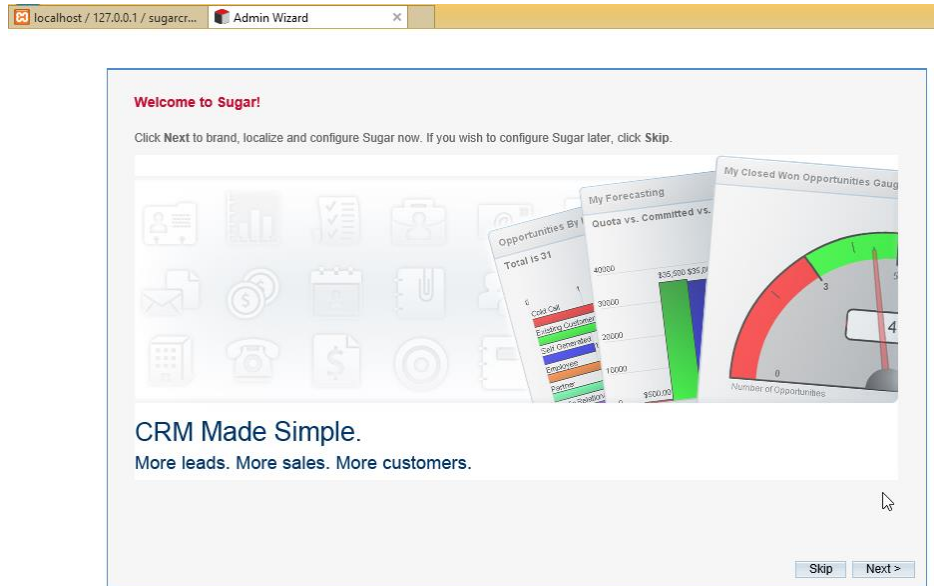
Figura 16 Ingreso a *SugarCRM*



Fuente: El autor

El mensaje de advertencia que aparece corresponde por la versión del navegador utilizado, que no es compatible por la versión. Ingresando los datos de usuario y contraseña configurados anteriormente para el logueo, se realiza el ingreso exitoso a la aplicación de *Sugar CRM* como se visualiza en la figura 17.

Figura 17 Welcome Sugar



Fuente: El autor

En la figura 18 la herramienta permite la elección de un logo de acuerdo con la empresa donde se quiera utilizar para personalizar el uso de esta dentro de determinada empresa, permitiendo el llamado de alguna imagen a través de la opción de examinar para que se cargue en la aplicación.

Figura 18 Cambio de logo sugar

Branding

Provide your organization's name and logo in order to brand your Sugar.

Name:

Select Logo:

Current Logo:

< Back Next >

Fuente: El autor

En la figura 19 que corresponde a la configuración solicita datos para la configuración de la herramienta como fecha y horario estableciendo los formatos que se requieran utilizar para la misma, lenguaje, tipo de moneda y ubicación entre otros.

Figura19 Datos de configuración

System Locale Settings

Specify how you would like data in Sugar to be displayed, based on your geographical location. The settings you provide here will be the default settings. Users will be able to set their own preferences.

Date Format:

Time Format:

Language:

Currency:

Currency Symbol:

ISO 4217 Currency Code:

1000s Separator:

Decimal Symbol:

Name Format:

< Back Next >

Fuente: El autor

En la figura 20 se solicita la información para la configuración de los servidores de correo que se usaran en la herramienta, permite la selección del proveedor de internet como Gmail, Yahoo, Microsoft Exchange y otros. También es solicitada la configuración del puerto designado para los servicios SMTP.

Figura 20 Configuración SMTP Sugar

The screenshot shows the 'SMTP Server Specification' screen in the SugarCRM Admin Wizard. The browser address bar indicates 'localhost / 127.0.0.1 / sugarcrm...'. The page title is 'Admin Wizard'. The main heading is 'SMTP Server Specification'. Below the heading, there is a descriptive text: 'Provide the email account that will be used to send emails, such as the assignment notifications and new user passwords. Users will receive emails from Sugar, as sent from the specified email account.' The 'Choose your Email provider:' section has four tabs: 'Gmail', 'Yahoo! Mail', 'Microsoft Exchange', and 'Other'. The 'Microsoft Exchange' tab is selected. Below the tabs, there are input fields for 'SMTP Server:', 'SMTP Port:' (with a value of '25'), 'Use SMTP Authentication?' (checked), 'SMTP Username:', 'SMTP Password:', and 'Enable SMTP over SSL or TLS?' (set to 'none'). There is also a checkbox for 'Allow users to use this account for outgoing email:'. At the bottom left, there are 'Clear' and 'Send Test Email' buttons. At the bottom right, there are '< Back' and 'Continue' buttons.

Fuente: El autor

En la figura 21 se solicita información para usuario administrador, con los datos requeridos como nombre, apellido, dirección de correo electrónico, teléfono, celular, ciudad, país y código postal. En la configuración que realizamos solo ingresamos los datos obligatorios solicitados en la aplicación de *SugarCRM*.

Figura 21 Datos Administrador

Your Information

Provide information about yourself. The information you provide about yourself will be visible to other Sugar users.
Fields marked with * are required.

First Name:	angela	Last Name: *	Administrator
Email Address: *	angela.lopez06@gmail.com	IM Type:	-none-
Office Phone:		IM Name:	
Mobile:			
Primary Address:			
City:		State:	
Postal Code:		Country:	

< Back Next >

Fuente: El autor

Una vez que se termina con las configuraciones requeridas en *SugarCRM*, en la figura 22 se puede ver el menú de las opciones de configuración de acuerdo a lo que se requiera revisar o realizar nuevamente. En el menú se observa las opciones de iniciar *sugar*, importar data de fuente externa, crear usuarios para el ingreso a la aplicación, visualizar y administrar las configuraciones de la aplicación, configurar la aplicación y ayuda sobre el manejo de la aplicación.

Figura 22 Finalización de configuración

You are ready to use Sugar!

What would you like to do next?

- Start Using Sugar**
Go directly to the application Home page.
- Import Data**
Import data from external sources into the application.
- Create Users**
Create new user accounts for people to use to access the application.
- View and Manage Application Settings**
Manage advanced settings, including default application settings.
- Configure the Application**
Use Studio to create and manage application fields and layouts.
- Visit Sugar University**
Find training materials and classes that will help you get started as a system administrator or end user of the application.

< Back **Finish**

Fuente: El autor

11.EJECUCION DE LAS PRUEBAS DE PENTESTING

Contando con el ambiente adecuado, es decir con la configuración del servidor de la aplicación de *SugarCRM* en una máquina virtual con Windows 8 y utilizando otras máquinas virtuales para la instalación de herramientas que permitan la realización de las pruebas de penetración, se da inicio a la ejecución de las mismas. En la guía de OWASP se presentan varios test de pruebas por cada fase descrita en la guía, a continuación se presentan las pruebas de penetración aplicados sobre *sugarCRM* y basada en la metodología de OWASP.

Tabla 3. Ejecución de pruebas *Pentesting*

Fase	Test de Prueba	Herramienta
Pruebas para gestionar la configuración y la implementación	Prueba de los métodos HTTP (OTG-CONFIG-006)	Complemento <i>RestClient</i> y <i>Curl</i>
Pruebas de Administración de Identidad	Prueba proceso de registro de usuarios (OTG-IDENT-002)	Complemento <i>LiveHTTPHeaders</i>
Pruebas de seguridad al proceso de autenticación	Pruebas para determinar un mecanismo de bloqueo débil (OTG-AUTHN-003)	Hydra
Pruebas de seguridad al proceso de autorización	Pruebas para determinar el escalamiento de privilegios (OTG-AUTHZ003).	WebScara
Pruebas de seguridad al proceso de gestión de sesiones	Prueba para atributos de cookies (OTG-SESS-002)	OWASP-ZAP
Pruebas de validación de entradas	Pruebas de Inyecciones de SQL (OTG-INPVAL-005)	Sqlmap – kali
	Pruebe la saturación del Búfer (OTG-INPVAL-014)	Linux Metasploit
Pruebas para Criptografía débil	Pruebas de Cross Site Scripting almacenados (OTG-INPVAL-002)	Xsssniper – Kali Linux
	Pruebas para el envío de información sensible por canales sin encriptar (OTG-CRYPST-003)	OWASP ZAP

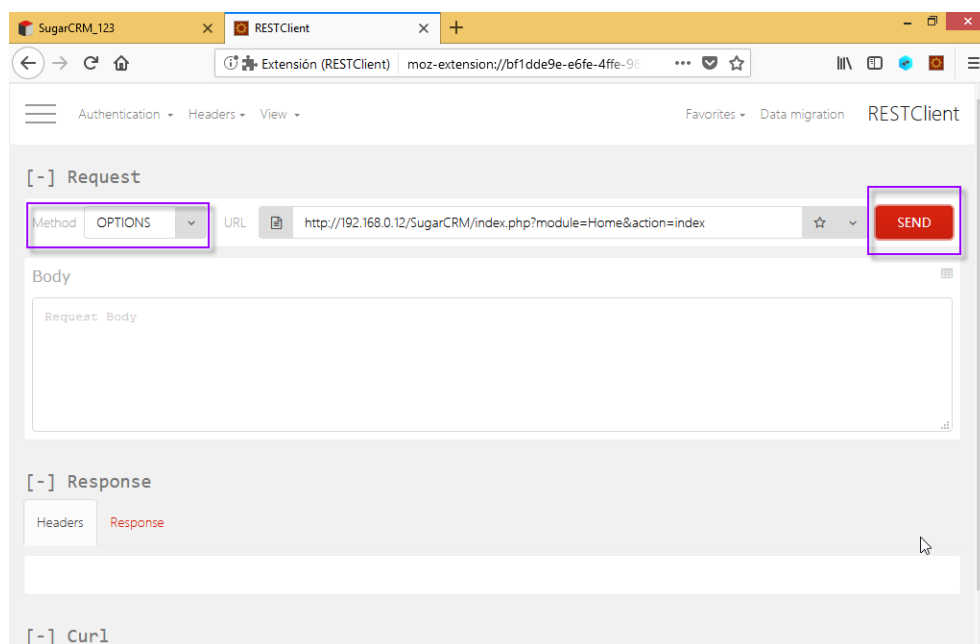
Fuente: El autor

11.1 Pruebas para gestionar la configuración y la implementación

11.1.1 Prueba de los métodos HTTP (OTG-CONFIG-006): El objetivo de esta prueba es evaluar todos los métodos que puede utilizar el protocolo HTTP, para realizar varias acciones en los servidores Web. Es importante la evaluación de esta vulnerabilidad teniendo en cuenta que en caso de presentar configuraciones erradas o no controladas se pueden utilizar con fines malignos.

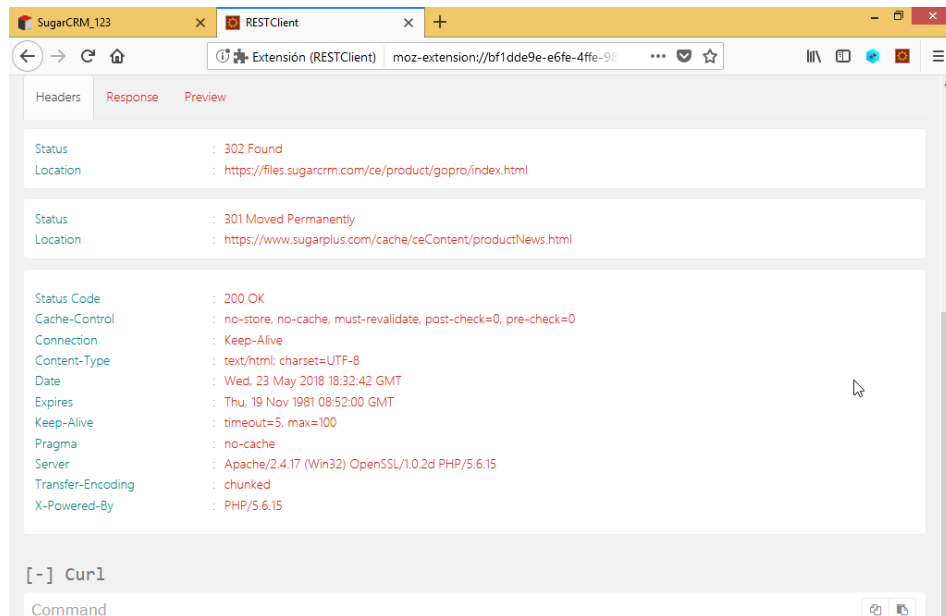
Para la ejecución de esta prueba fue utilizado el complemento de Mozilla restClient utilizado para obtener la información de los métodos HTTP en el servidor de *SugarCRM*. Una vez instalado el complemento en el navegador, se selecciona el método de OPTIONS y se envía la solicitud como esta en la figura 23.

Figura 23. Complemento *RESTClient*



Fuente: El autor

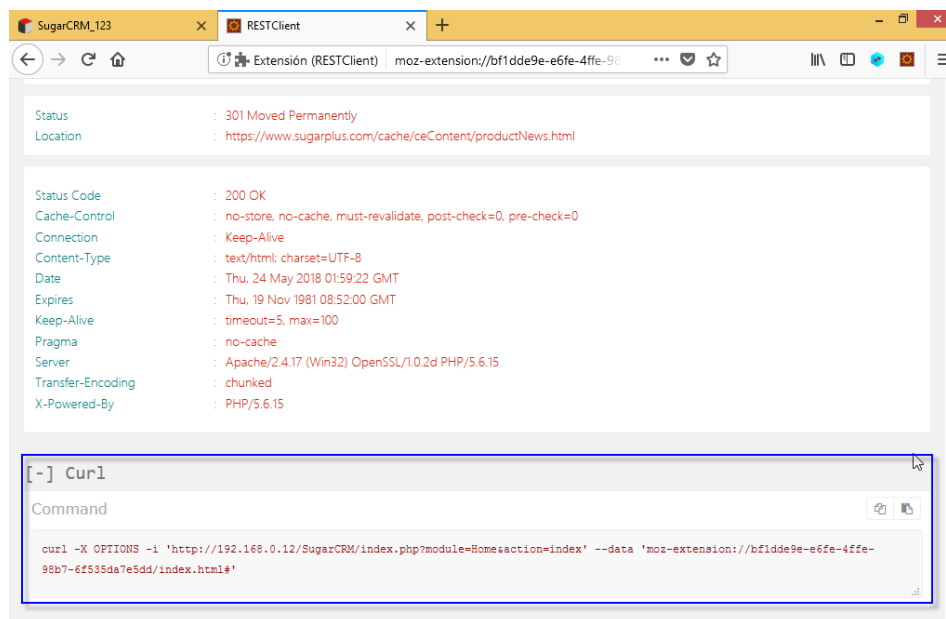
Figura 24 Resultado *Header*



Fuente: El autor

En la figura 24 se observa el resultado del complemento *RESTClient* en la pestaña de *header*. De acuerdo con el resultado observado, no se evidencia ninguna información de los métodos presentes en el servidor *SugarCRM*, normalmente aparecen en un campo que se llama *ALLOW*.

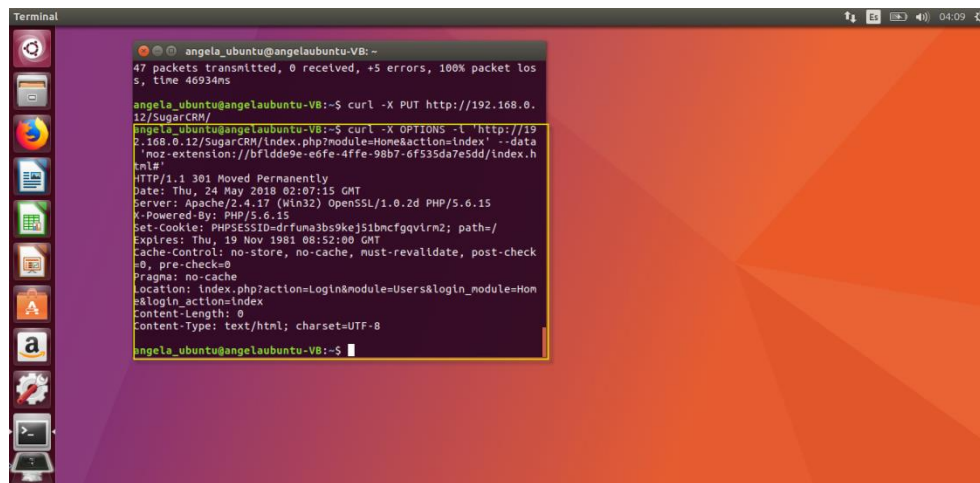
Figura 25. Resultado Curl



Fuente: El autor

En la figura 25 se observa otra información como resultado de la revisión de los métodos, aparece el comando de Curl aplicado al servidor de *SugarCRM* utilizado posteriormente en la prueba que se está ejecutando desde otra máquina virtual para la prueba ejecutada en un sistema operativo Ubuntu.

Figura 26. Ejecución Curl en ubuntu



```
angela_ubunt@angelaubuntu-VB: ~$ curl -X PUT http://192.168.0.12/SugarCRM/
47 packets transmitted, 0 received, +5 errors, 100% packet loss, time 46934ms
angela_ubunt@angelaubuntu-VB: ~$ curl -X OPTIONS -L 'http://192.168.0.12/SugarCRM/index.php?module=Home&action=Index' --data '{"noz-extension://bfldde9e-e6fe-4ffe-98b7-6f535da7e5dd/index.html"}'
HTTP/1.1 301 Moved Permanently
Date: Thu, 24 May 2018 02:07:15 GMT
Server: Apache/2.4.17 (Ubuntu) OpenSSL/1.0.2d PHP/5.6.15
V-Server: PHP/5.6.15
Set-Cookie: PHPSESSID=drfma3bs9kej51bmcfgqvmr2; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: index.php?action=Login&module=Users&login_module=Home&login_action=Index
Content-Length: 0
Content-Type: text/html; charset=UTF-8
angela_ubunt@angelaubuntu-VB: ~$
```

Fuente: El autor

En la figura 26 el resultado de la ejecución del comando Curl desde otra máquina externa con S.O Ubuntu, donde como resultado se observa que no se tiene información sobre los posibles métodos que puede utilizar el servidor *SugarCRM*, de igual manera que en la revisión con el complemento de Mozilla no es visible el campo *ALLOW*.

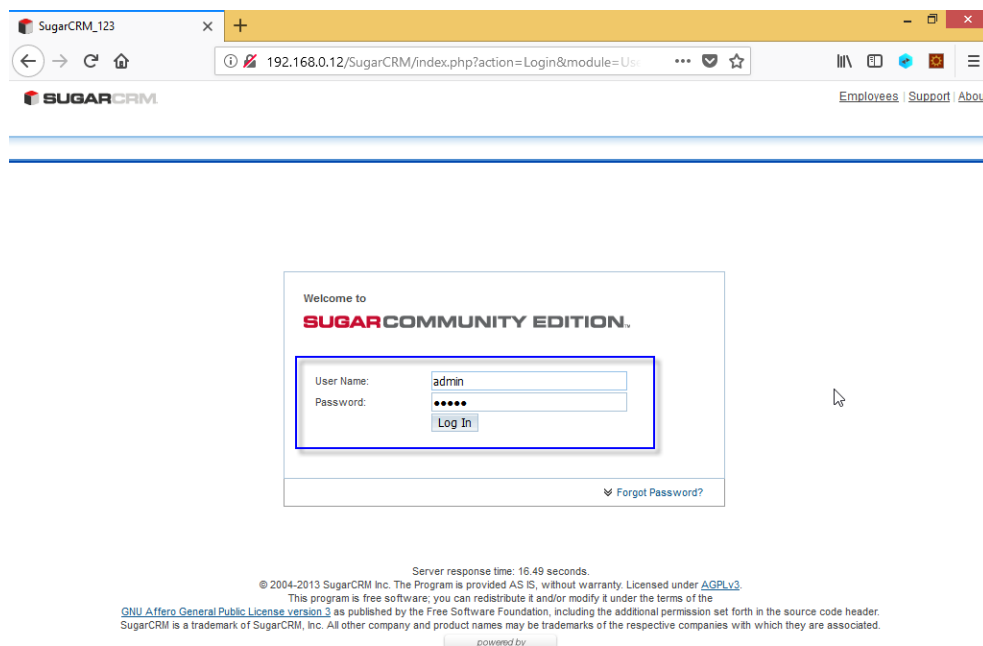
11.1.2 Resultado Prueba de los métodos HTTP (OTG-CONFIG-006) : De acuerdo a los resultados obtenidos en la prueba que se ejecutó con la consulta del método de *OPTIONS* para permitir conocer los recursos y métodos que pueden ser usados por un servidor como *PUT* o *DELETE*, del resultado se puede concluir que no aparece información visible de los métodos usados en el servidor *SugarCRM*, con el complemento *RestClient* y la ejecución del comando *Curl* nunca esta visible el campo *ALLOW* donde normalmente aparece la información de los métodos usados, por lo tanto se asume que el servidor *SugarCRM* tiene el control de los métodos.

11.2 Pruebas de Administración de Identidad

11.2.1 Prueba proceso de registro de usuarios (OTG-IDENT-002): El objetivo de esta prueba es la validación de los requisitos de identidad en el acceso para el servidor de *SugarCRM*, para verificar la seguridad y identificar si se puede conocer, forzar o manipular de manera sencilla.

Para la ejecución de estas pruebas se utiliza el complemento *LiveHTTPHeaders* en el navegador Mozilla, desde donde se ingresa a la interface de *SugarCRM*, para cumplir el objetivo de la misma se van a realizar dos revisiones con el complemento, una con el logueo inicial a la aplicación y otra con la creación de un usuario nuevo.

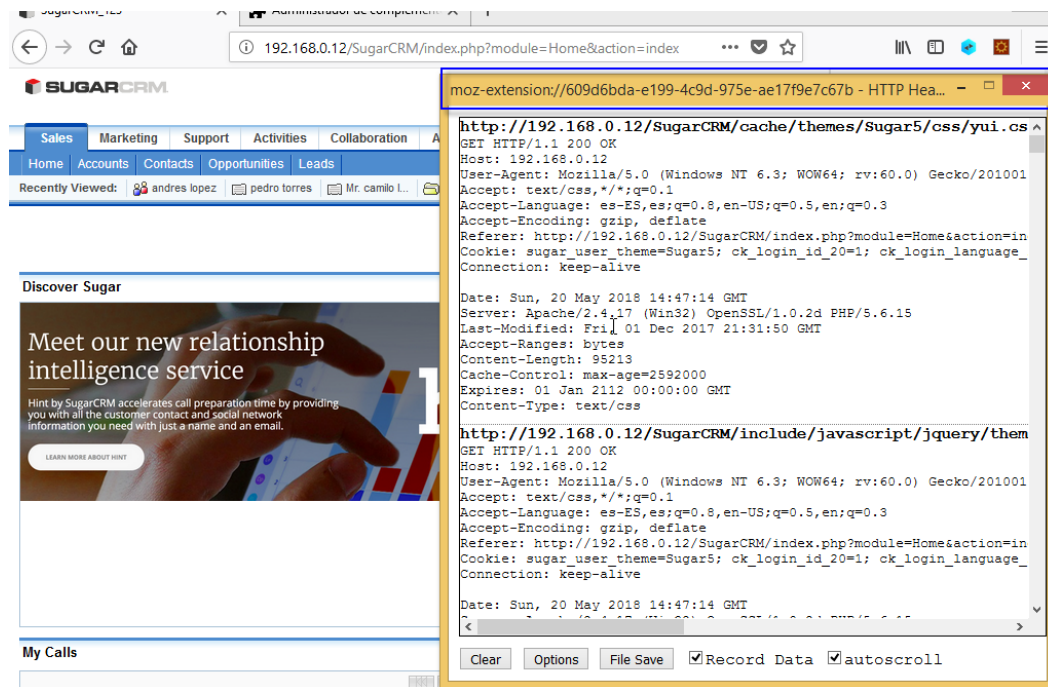
Figura 27. Ingreso *SugarCRM* desde Mozilla



Fuente: El autor

En la figura 27 aparece el ingreso a *SugarCRM* desde Mozilla, en la parte superior del lado derecho se observa el icono del complemento de *LiveHTTPHeaders* instalado. Posteriormente se realiza el logueo a la aplicación de manera que se pueda ir visualizando la información obtenida con el complemento mencionado.

Figura 28. Información del complemento con el logueo

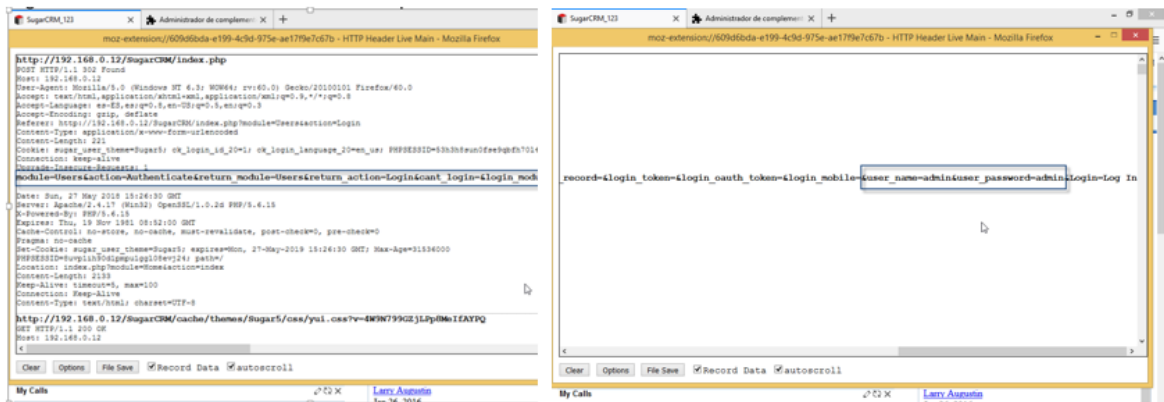


Fuente: El autor

En la figura 28, se observa cómo una vez se genera el logueo a *SugarCRM*, se activa la acción del complemento y registra toda la información del servidor donde se realizó el registro con los datos de usuario y contraseña configurados para el ingreso.

En la figura 29, se selecciona la línea donde se encuentra la información del logueo donde se observan los datos de *username* y *password* de manera clara y sin ninguna encriptación. Como está marcado en la figura 40 los datos de ingreso a *SugarCRM* son admin tanto de usuario como clave.

Figura 29. Información de usuario y password



Fuente: El autor

Figura 30. Opción de Creación usuarios

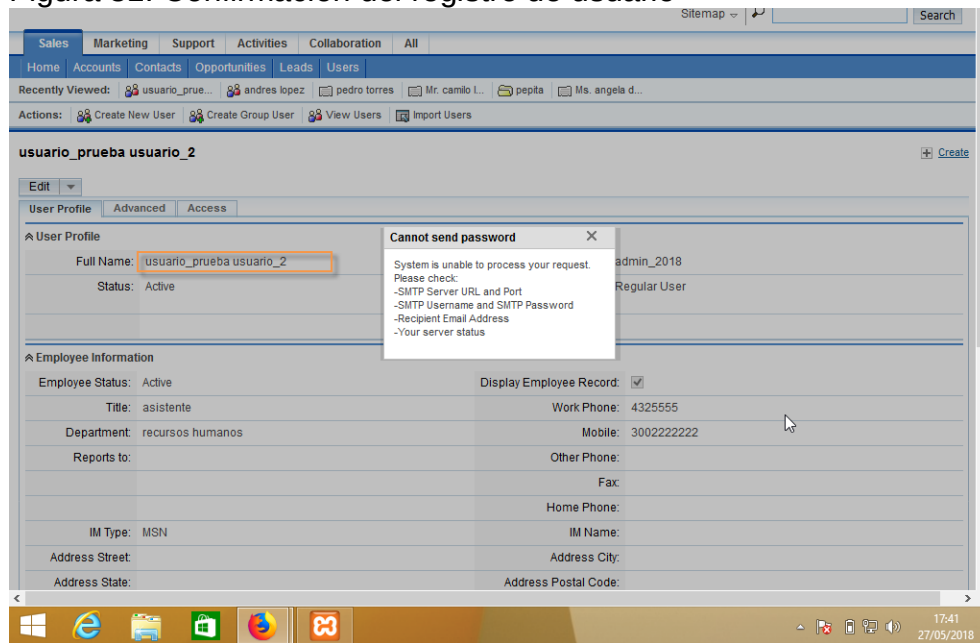
The image shows the 'Create New User' form in SugarCRM. The 'Create' button is highlighted with a green circle. The form is divided into two main sections: 'User Profile' and 'Employee Information'.

- User Profile:**
 - User Name: admin_2018
 - Status: Active
 - User Type: Regular User
 - First Name: usuario_prueba
 - Last Name: usuario_2
- Employee Information:**
 - Employee Status: Active
 - Title: asistente
 - Department: recursos humanos
 - Reports to: (empty)
 - IM Type: MSN
 - Address Street: (empty)
 - Address State: (empty)
 - Address Country: (empty)
 - Display Employee Record: ☒
 - Work Phone: 4325555
 - Mobile: 300222222
 - Other Phone: (empty)
 - Fax: (empty)
 - Home Phone: (empty)
 - IM Name: (empty)
 - Address City: (empty)
 - Address Postal Code: (empty)

Fuente: El autor

Como complemento de las pruebas a continuación, se realiza un registro para la creación de nuevo usuario desde la aplicación de *SugarCRM*, como se observa en la figura 30, a través del módulo de registro se realiza la acción de creación de otro usuario para validar el ingreso nuevamente a la aplicación de *SugarCRM*.

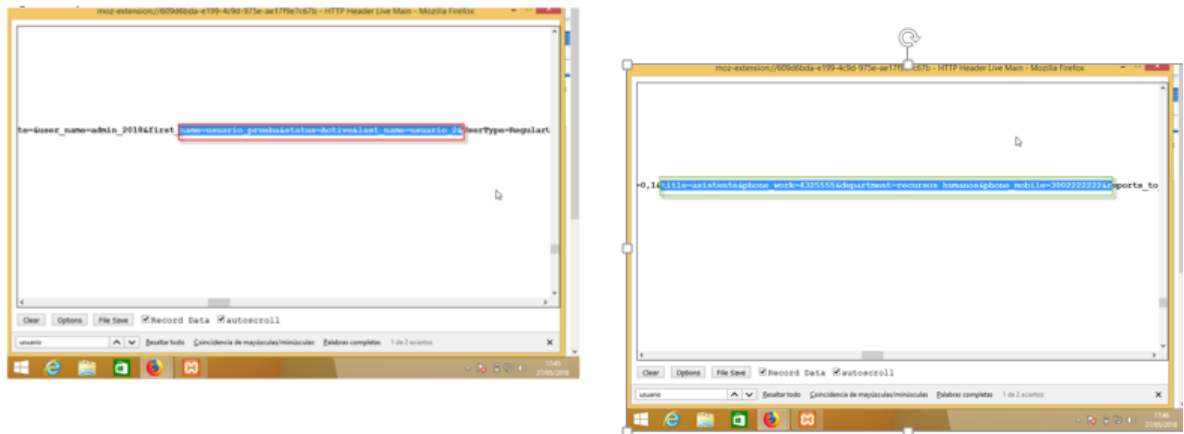
Figura 31. Confirmación del registro de usuario



Fuente: El autor

En la figura 31 se visualiza la ventana de registro del nuevo usuario, el mensaje indica que no se puede enviar la clave por temas de servidor de correo, pero para la prueba no presenta limitación teniendo en cuenta que se está validando únicamente la creación de un nuevo usuario que pueda loguearse en la aplicación.

Figura 32. Información nuevo usuario



Fuente: El autor

En la figura 32, se observan los registros en la cabecera del complemento Live que corresponden a los datos registrados en la creación del nuevo usuario desde *SugarCRM*, datos como usuario, titulo, cargo, teléfono entre otros. La información

se observa clara sin ninguna encriptación y puede ser datos sensibles contando que es información personal.

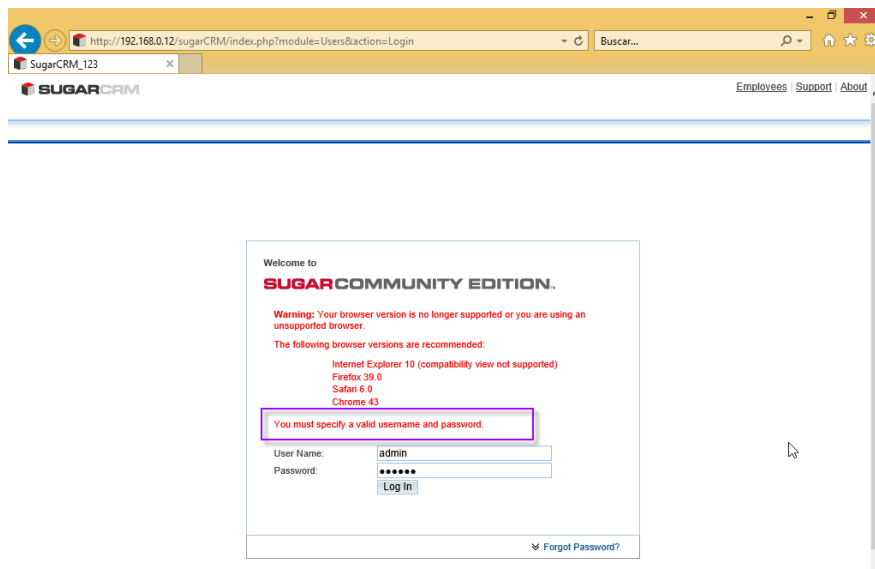
11.2.2 Resultado de la prueba proceso de registro de usuarios (OTG-IDENT-002): De acuerdo a los resultados obtenidos se concluye que la información de logueo y registro de usuarios nuevos se envía de forma clara tal como se ingresa, permitiendo ser capturada a través del complemento *LiveHTTPHeaders* como se visualizó en las pruebas realizadas en el logueo y con la creación de un usuario nuevo, siendo vulnerables para alguna manipulación de usuarios en la herramienta o suplantación de identidad. De igual manera también se puede identificar las variables de registro de información en el logueo inicial y en el formulario de creación de usuarios con los valores enviados sin encriptación, lo que se puede considerar como vulnerabilidad en *SugarCRM* ya que está expuesta a la manipulación o modificación de la información enviada.

11.3 Pruebas de seguridad al proceso de autenticación

11.3.1 Pruebas para determinar un mecanismo de bloqueo débil (OTG-AUTHN-003): El objetivo de esta prueba AUTHN-003 para determinar un mecanismo de bloqueo débil, es validar la vulnerabilidad de *SugarCRM* a un ataque de fuerza bruta para evaluar cómo están los mecanismos de bloqueo de las cuentas y validar si a través de un forzado de contraseñas se logra conocer la información de ingreso o logueo a la aplicación.

Para llevar a cabo la prueba que permita determinar un mecanismo de bloqueo débil (**OTG-AUTHN-003**), se utiliza la herramienta de hydra desde Kali Linux, generando un ataque de fuerza bruta aplicado sobre *SugarCRM*. A continuación, se describe los pasos ejecutados para validar el ataque de fuerza bruta sobre *SugarCRM*.

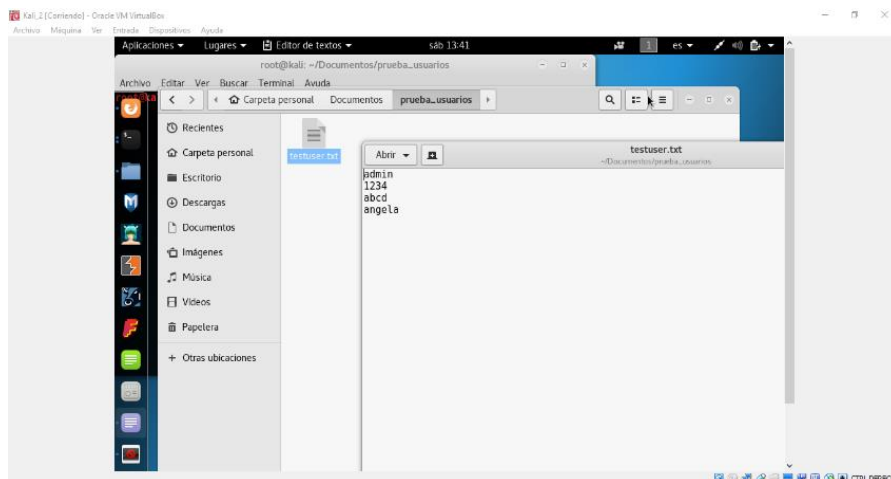
Figura 33. Ingreso a *sugarCRM* con datos errados



Fuente: El autor

En la figura 33 se observa el ingreso al logueo de manera errada con la información de login y password incorrectos. Se obtiene un mensaje de error y en la parte superior de la dirección se obtiene datos de la URL importante para el ataque, como son nombres de directorio y variables del servidor web.

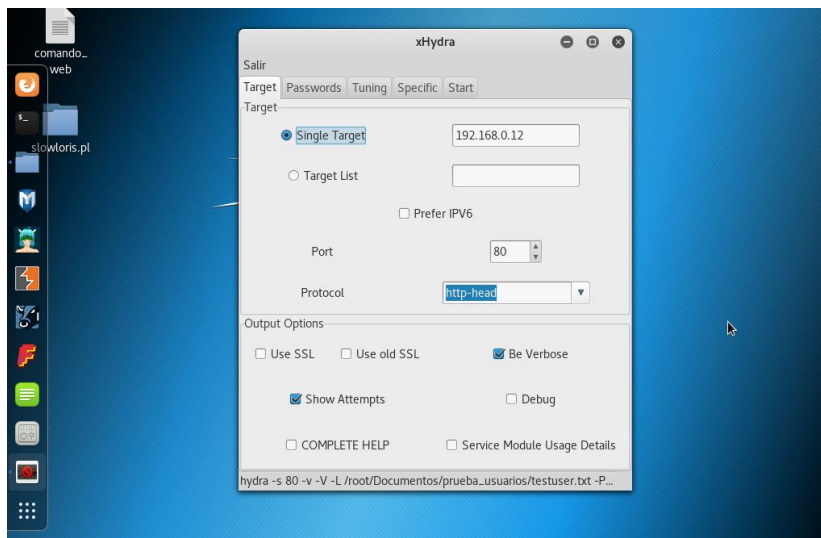
Figura 34. Archivo con lista de usuarios/claves



Fuente: el autor

En la figura 34, se observa la imagen del archivo creado en *Kali Linux* para ser utilizado por la herramienta hydra en el ataque de fuerza bruta que se requiere ejecutar, los datos ingresados son posibles datos de usuarios y claves que se pueden utilizar en la aplicación de *SugarCRM*. Para la prueba a ejecutar utilizamos únicamente cuatros (4) registros.

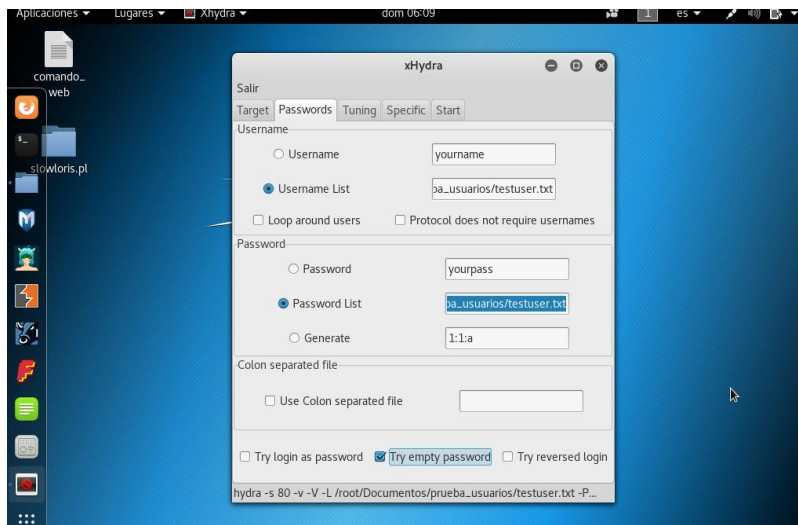
Figura 35. Configuración de Hydra - Target



Fuente : El autor

En la figura 35 se presenta la pantalla inicial de Hydra, en la pestaña target se configura la IP del servidor web donde está instalado *SugarCRM*. De igual manera se incorpora el puerto correspondiente y se selecciona el protocolo a usar, para este caso http-head. En la sección de Output Options se deja seleccionado las opciones de be verbose y Show Attempts.

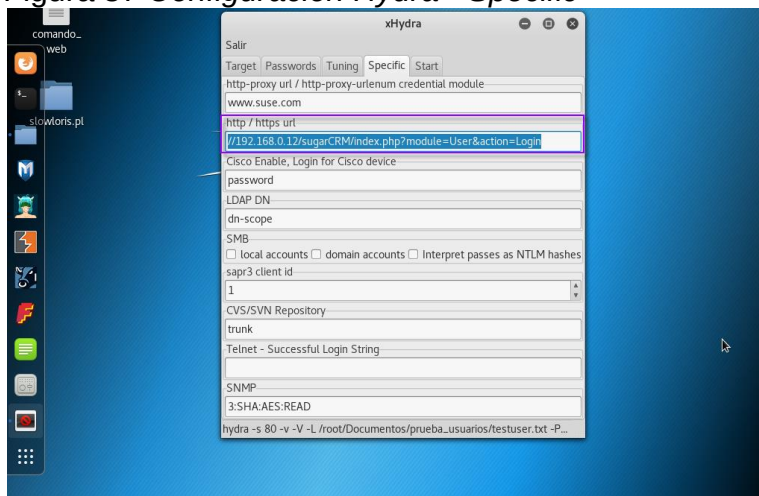
Figura 36. Configuración Hydra -Passwords



Fuente : El autor

En la figura 36 se observa la configuración realizada en la pestaña de *passwords*, donde se elige la opción en *usuarios* y *password* usando un archivo con la lista de datos que se crearon anteriormente, es decir, utilizando el archivo creado con posibles usuarios y claves.

Figura 37 Configuración Hydra - Specific

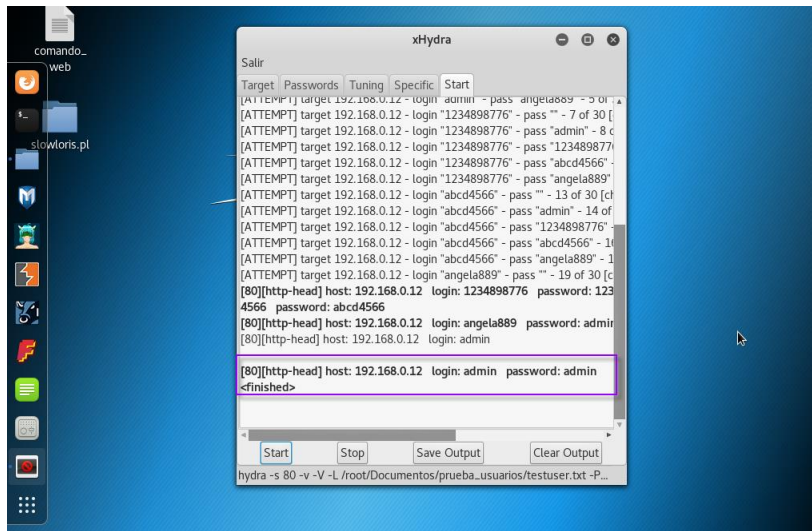


Fuente: El autor

En la figura 37 aparece la configuración o información que se debe ingresar en la pestaña *Specific* para el campo *http / https url*, se coloca la información de la URL que se obtienen en el servidor web cuando se ingresa a *SugarCRM* con datos

errados en el logueo. Los demás campos de esta pestaña no se modifican, no son requeridos para la prueba a validar.

Figura 38. Ataque De fuerza Bruta con Hydra



Fuente: El autor

En la figura 38 se observa el resultado del ataque de fuerza bruta, una vez iniciado el ataque en hydra se muestran los resultados de la ejecución en negrilla, para el ingreso a la aplicación de *sugarCRM*. El usuario de ingreso es admin y la clave es admin, como son generados en la respuesta del ataque.

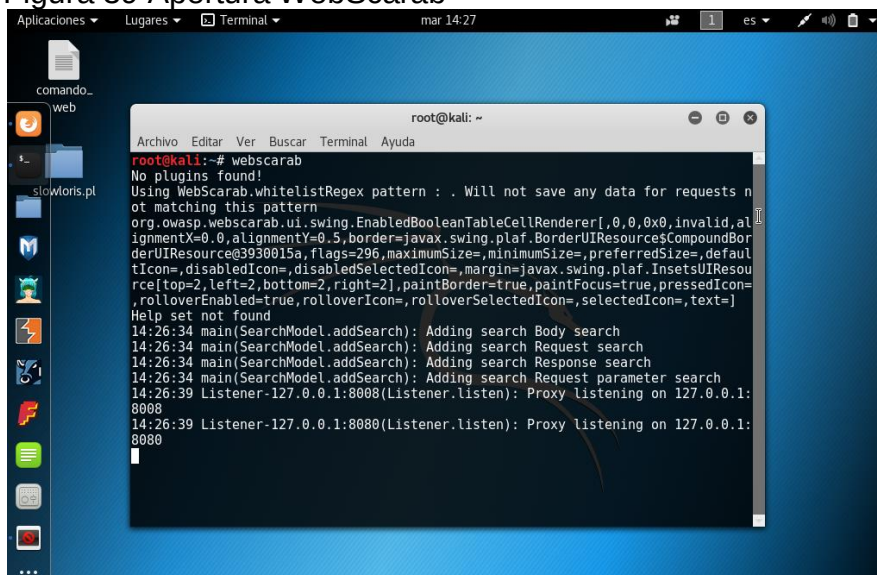
11.3.2 Resultado de las pruebas para determinar un mecanismo de bloqueo débil (OTG-AUTHN-003): De acuerdo al resultado de las pruebas se concluye que *SugarCRM* es vulnerable al ataque de fuerza bruta realizado, utilizando un diccionario o archivo con datos de posibles usuarios y claves, logrando que se conozca los datos de ingreso a la aplicación.

11.4 Pruebas de seguridad al proceso de autorización

11.4.1 Pruebas para determinar el escalamiento de privilegios (OTG-AUTHZ003): El objetivo de esta prueba es validar que la aplicación no permite realizar modificaciones a los privilegios de un usuario o rol dentro de la aplicación. Para esta prueba se utiliza la herramienta de WebScarab desde Kali Linux, con el objetivo de encontrar campos ocultos que permitan alguna modificación de los datos o rol de un usuario.

A continuación en la figura 39, se describe como se utilizó la herramienta para lograr el objetivo de la prueba. Se realiza la apertura de la herramienta WebScarab desde una terminal en Kali Linux, utilizando el comando `webcarab` y se observa la IP y puerto utilizados en la herramienta.

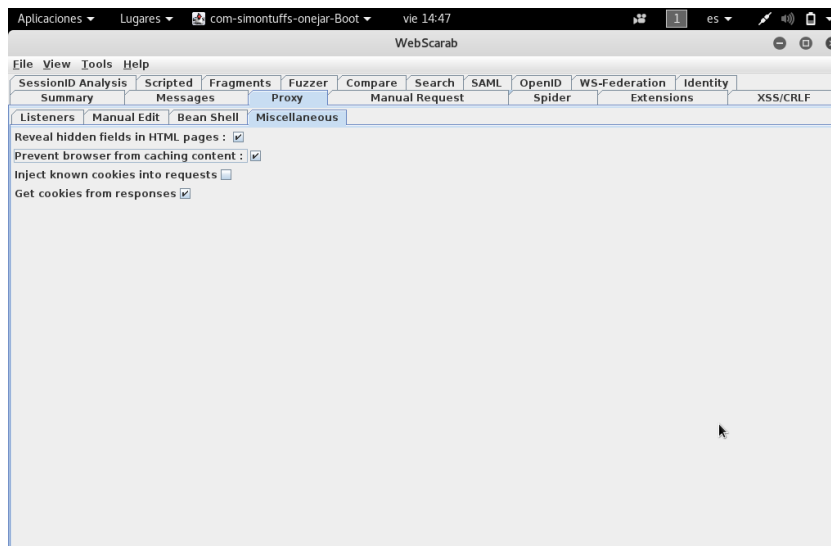
Figura 39 Apertura WebScarab



```
root@kali: ~  
root@kali:~# webcarab  
No plugins found!  
Using WebScarab.whitelistRegex pattern : . Will not save any data for requests n  
ot matching this pattern  
org.owasp.webscarab.ui.swing.EnabledBooleanTableCellRenderer[,0,0,0x0,invalid,al  
ignmentX=0,0,alignmentY=0,5,border=javax.swing.plaf.BorderUIResourcesCompoundBor  
derUIResource@3930015a,flags=296,maximumSize=,minimumSize=,preferredSize=,defaul  
tIcon=,disabledIcon=,disabledSelectedIcon=,margin=javax.swing.plaf.InsetsUIResou  
rce[top=2,left=2,bottom=2,right=2],paintBorder=true,paintFocus=true,pressedIcon=  
,rolloverEnabled=true,rolloverIcon=,rolloverSelectedIcon=,selectedIcon=,text=]  
Help set not found  
14:26:34 main(SearchModel.addSearch): Adding search Body search  
14:26:34 main(SearchModel.addSearch): Adding search Request search  
14:26:34 main(SearchModel.addSearch): Adding search Response search  
14:26:34 main(SearchModel.addSearch): Adding search Request parameter search  
14:26:39 Listener-127.0.0.1:8080(Listener.listen): Proxy listening on 127.0.0.1:  
8080  
14:26:39 Listener-127.0.0.1:8080(Listener.listen): Proxy listening on 127.0.0.1:  
8080
```

Fuente: El autor

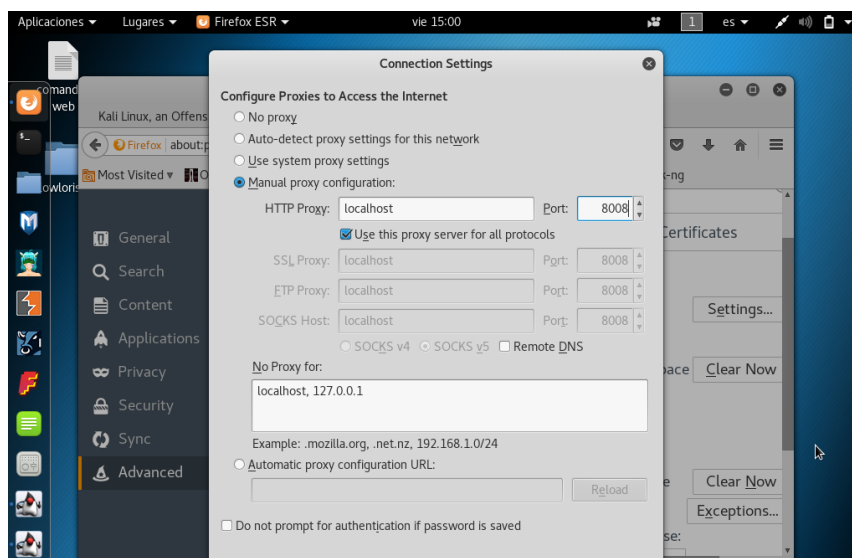
Figura 40 Configuración WebScarab -proxy



Fuente: El autor

En la figura 40 se observa la configuración realizada en la herramienta WebScarab para la prueba. En la pestaña de Proxy, seleccionando la sub pestaña Listeners, se valida la información del puerto en escucha y la configuración proxy, en la siguiente subpestaña Miscellaneous se puede habilitar las opciones como están descritas en la figura 40 de acuerdo a necesidad de la prueba realizada, para la prueba de determinar el escalamiento de privilegios (OTG-AUTHZ003), solo usamos tres de las opciones presentadas en la herramienta.

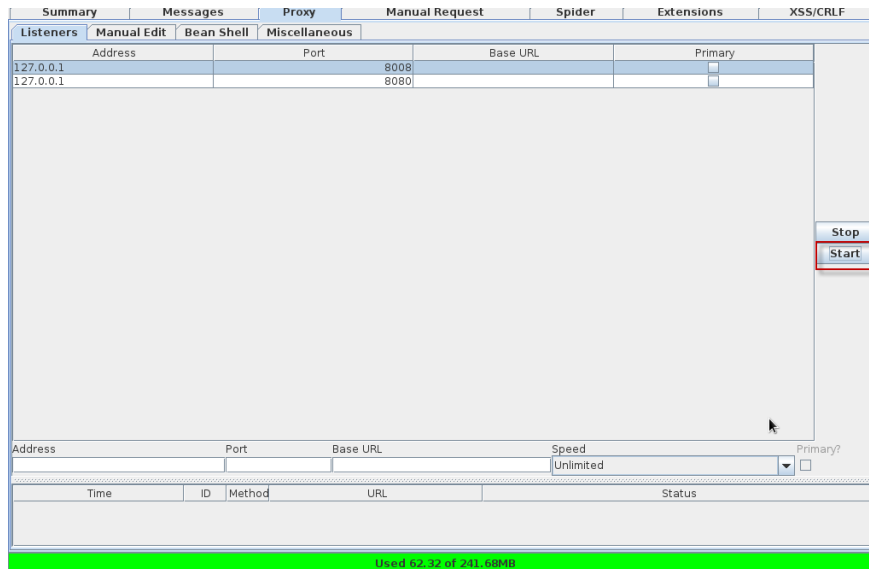
Figura 41 Configuración Navegador como proxy



Fuente: El autor

En la figura 41 se observa la configuración del navegador que se debe usar para el funcionamiento de la herramienta WebScarab como proxy, esta configuración se debe realizar en el navegador usado y con la configuración del puerto 8008, utilizado por la herramienta WebScarab.

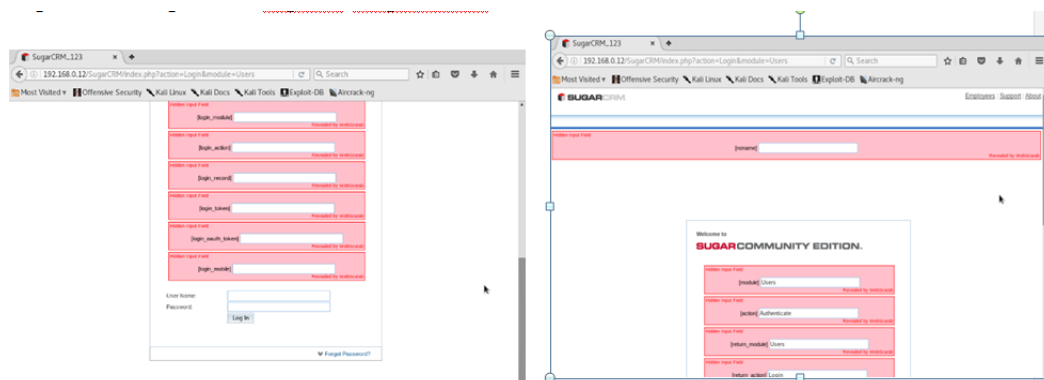
Figura 42 Inicio de WebScarab



Fuente: El autor

En la figura 42, En la pestaña de proxy se configura en el listeners, la IP y puerto de escucha del proxy. Se selecciona la opción de *Start* o inicio de la herramienta. Posterior a lo anterior, se ingresa a la aplicación de *SugarCRM* desde el navegador en kalil inux para validar el escaneo y los resultados que se observan, como se describen a continuación.

Figura 43 Página de logueo *SugarCRM*



Fuente: El autor

En la figura 43, se observa la página principal de logueo de *SugarCRM* y permite observar resaltado en rojo los campos ocultos, algunos de ellos con valores. Las dos figuras corresponden a la misma pantalla tomando la parte superior e inferior de la misma pantalla de manera que se pueda visualizar completamente los campos.

Figura 44 Logueo a la aplicación *SugarCRM*

The screenshot shows the SugarCRM login page with several hidden input fields highlighted in red. The fields are labeled as follows:

- `[search_form]` false
- `[advanced]` false
- `[module]` Users
- `[record]` 768b94b-a577-4d64-a755
- `[isDuplicate]` false
- `[action]`
- `[user_name]`
- `[user_type]`
- `[password_generate]`
- `[old_password]`

Fuente: El autor

En la figura 44, se observa que posterior al logueo en la aplicación *SugarCRM*, se observan varios campos ocultos con valores y otros vacios. De igual manera se visualiza que contienen los nombres para cada uno de los campos ocultos que son mostrados.

Figura 45 Datos de cuenta de SugarCRM

The screenshot shows the SugarCRM user profile page for 'usuario_prueba usuario_2'. The page displays various user details and hidden input fields highlighted in red. The fields are labeled as follows:

- `[fullname]` Active
- `[username]` RegularUser
- `[title]` asistente
- `[work_phone]` 4325555
- `[mobile]` 300222222
- `[department]` recursos humanos
- `[reports_to]`
- `[other_phone]`
- `[fax]`
- `[home_phone]`
- `[im_name]`
- `[address_street]`
- `[address_city]`
- `[address_state]`
- `[address_postal_code]`
- `[address_country]`
- `[description]` Prueba de seguridad sobre sugarCRM
- `[email_address]` wxyz.laboratorio@gmail.com (Primary)
- `[email_client]` External Email Client

Fuente: El autor

En la figura 45, se observa la sección de cuentas de la aplicación *SugarCRM*, donde también se visualizan seleccionados algunos campos ocultos con valores y información de los nombres en cada campo.

11.4.2 Resultado de la Prueba para determinar el escalamiento de privilegios (OTG-AUTHZ003): De la ejecución de la prueba se concluye que existen varios campos ocultos, algunos con información del usuario con el que se ingresó a la aplicación, pero la información de los campos del módulo de logueo y de usuarios no representa algún riesgo ya que no se está guardando ni exponiendo información que permita realizar cambios de información del usuario o rol definido.

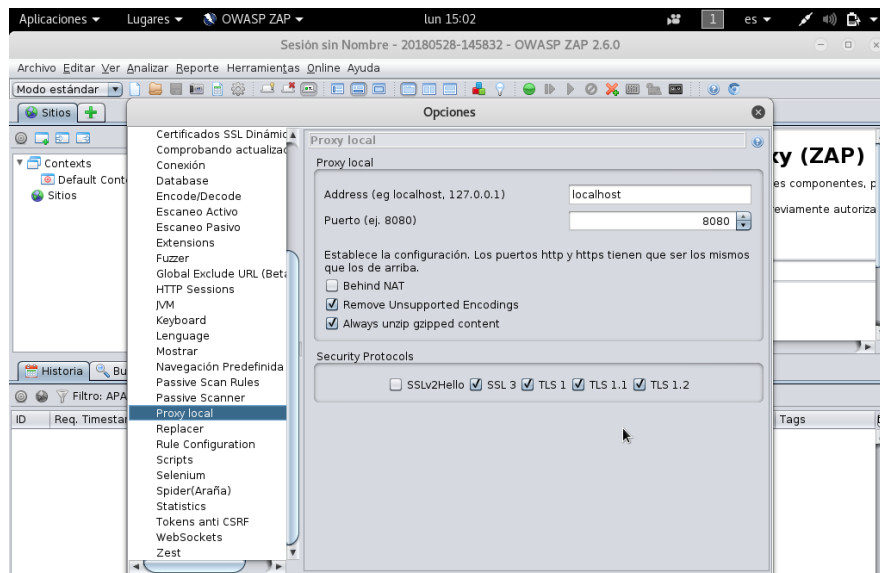
11.5 Pruebas de seguridad al proceso de gestión de sesiones

11.5.1 Prueba para atributos de cookies (OTG-SESS-002): El objetivo de esta prueba es validar que las cookies se estén usando de manera correcta y segura en la aplicación *SugarCRM*. Revisar la configuración de las cookies, permite evaluar los tipos de vulnerabilidades que se pueden presentar como secuestros de sesión, ataques contra Id de sesión o mecanismos de autorización no permitidos, como ejemplo de los tipos de vulnerabilidades para el secuestro de sesión se pueden mencionar los CVE-2018-1000519 que corresponde a una vulnerabilidad de reparación de sesión, ataque explotado por los métodos que permiten configurar las cookies de sesión y la CVE-2018-10813, vulnerabilidad donde los valores codificados son visibles en el código fuente, lo que puede ocasionar que un atacante edite los contenidos de las cookies de sesión y pueda volver a firmarlo.

Por tal razón es de vital importancia tener en cuenta que las cookies pueden ser un vector de ataque clave usado por los delincuentes encontrando la ruta o camino para generar múltiples ataques. Para la prueba es requerido conocer los atributos considerados como cookies, ejemplo de estos son: atributo *Segura*, *HttpOnly* y *Dominio*.

Para la ejecución de la prueba se utiliza la herramienta OWASP-ZAP desde Kali Linux, a continuación se explica la configuración realizada para que funcione como proxy y permita la captura de la información de la aplicación *SugarCRM*. Para utilizar la herramienta OWASP-ZAP como proxy se debe generar una configuración desde la misma y en el navegador como se describe a continuación.

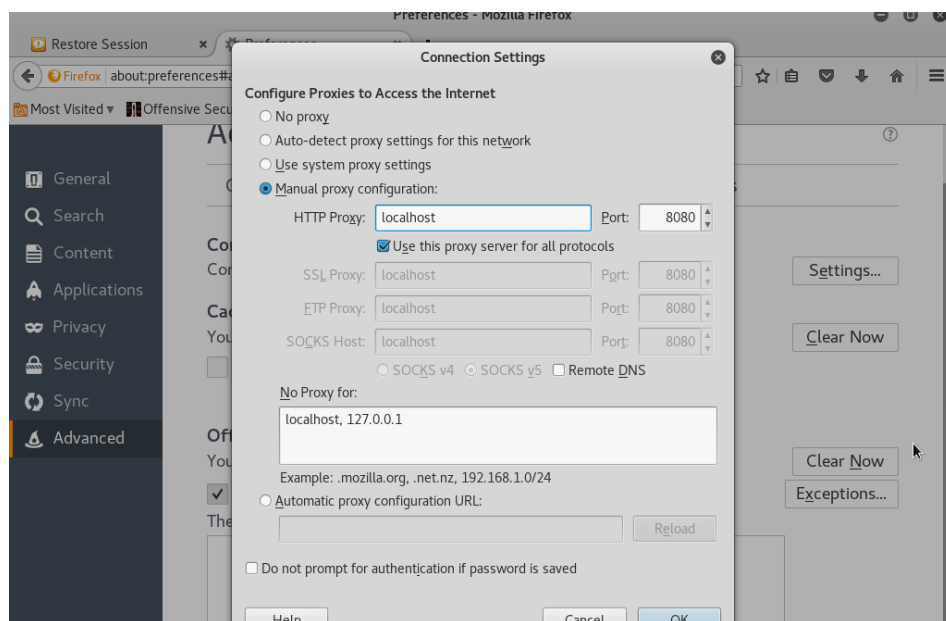
Figura 46 Configuración Proxy de OWASP -ZAP



Fuente: El autor

En la figura 46, se ingresa en el menú de Herramientas, opciones y se elige la opción Proxy local como esta en figura. Se debe colocar la información de Address como localhost y utilizando el puerto 8080, las opciones de protocolos de seguridad se dejan habilitados para SSL, TLS etc.

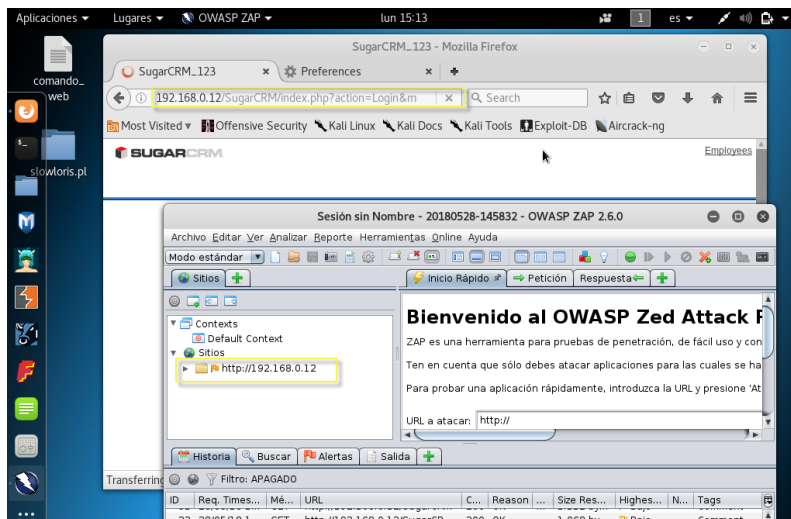
Figura 47 Configuración en el Navegador Mozilla



Fuente: El autor

En la figura 47, se indica la configuración que se debe colocar en el navegador desde la opción de opciones de configuración de manera que se configure como proxy para la prueba a ejecutar, indicando la IP y puerto utilizado. Los datos utilizados tanto en la herramienta OWASP-ZAP como en el navegador usado son la misma, utilizamos la IP de localhost 120.0.0.1 y el puerto 8080.

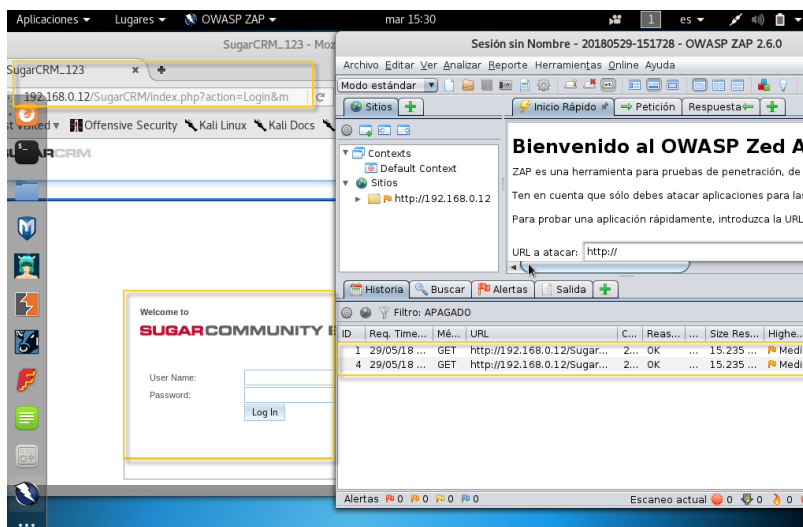
Figura 48 Ingreso a la aplicación *SugarCRM* con interceptación proxy



Fuente: El autor

En la figura 48, se observa el ingreso a la aplicación *SugarCRM* desde navegador configurado con proxy manual y se valida como en empieza a interceptar la información en la herramienta OWASP-ZAP actuando como proxy.

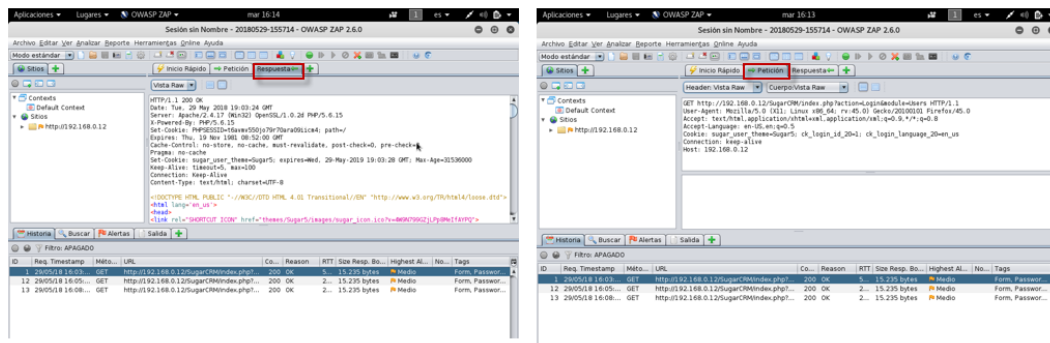
Figura 49 Apertura de la aplicación sin logueo



Fuente: el autor

Para identificar las cookies, primero se realiza la apertura de la aplicación y se valida los registros en la herramienta OWASP-ZAP por cada acción realizada en la aplicación de *SugarCRM*, en la figura 60 aparecen los registros únicamente con la apertura de la aplicación *SugarCRM* sin logueo.

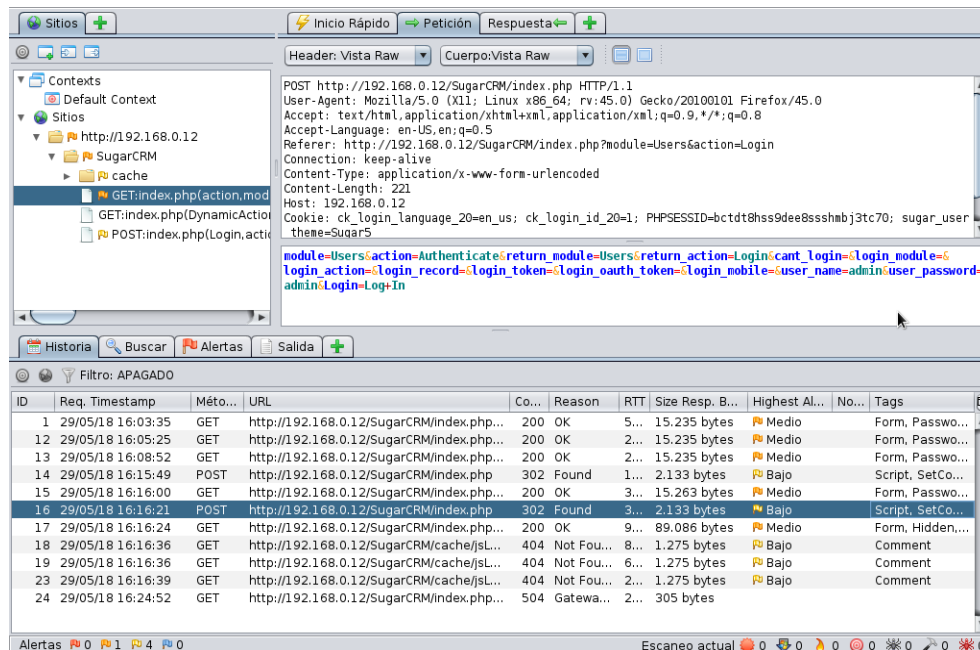
Figura 50 Información tag forma y password



Fuente: El autor

En la figura 50, se identifica la petición realizada y la respuesta donde se puede identificar información en el parámetro de Set cookie como ID de sesión, fecha de expiración, datos del servidor, tipo de conexión entre otros.

Figura 51 Información después de logueo



Fuente: el autor

En la figura 51, validando la información del Tags Script, SetCookies en la información de la petición en la parte inferior se muestra la información del logueo a la aplicación con los datos de usuarios y clave correctos de ingreso a la aplicación y sin encriptación. El usuario y clave de ingreso son admin.

Figura 52 Response de Script, SetCookie

HTTP/1.1 302 Found
 Date: Tue, 29 May 2018 19:16:10 GMT
 Server: Apache/2.4.17 (Win32) OpenSSL/1.0.2d PHP/5.6.15
 X-Powered-By: PHP/5.6.15
 Expires: Thu, 19 Nov 1981 08:52:00 GMT
 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
 Pragma: no-cache
 Set-Cookie: sugar_user_theme=Sugar5; expires=Wed, 29-May-2019 19:16:11 GMT; Max-Age=31536000
 Set-Cookie: PHPSESSID=1aajt549abstin0mdrrcusm6a3; path=/
 Location: index.php?module=Home&action=index
 Content-Length: 2133
 Keep-Alive: timeout=5, max=100
 Connection: Keep-Alive
 Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

ID	Req. Timestamp	Método	URL	Co...	Reason	RTT	Size Resp. B...	Highest Al...	No...	Tags
1	29/05/18 16:03:35	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	5...	15.235 bytes	Medio		Form, Passwo...
12	29/05/18 16:05:25	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	2...	15.235 bytes	Medio		Form, Passwo...
13	29/05/18 16:08:52	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	2...	15.235 bytes	Medio		Form, Passwo...
14	29/05/18 16:15:49	POST	http://192.168.0.12/SugarCRM/index.php...	302	Found	1...	2.133 bytes	Bajo		Script, SetCo...
15	29/05/18 16:16:00	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	3...	15.263 bytes	Medio		Form, Passwo...
16	29/05/18 16:16:21	POST	http://192.168.0.12/SugarCRM/index.php...	302	Found	3...	2.133 bytes	Bajo		Script, SetCo...
17	29/05/18 16:16:24	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	9...	89.086 bytes	Medio		Form, Hidden...
18	29/05/18 16:16:36	GET	http://192.168.0.12/SugarCRM/cache/jsL...	404	Not Fou...	8...	1.275 bytes	Bajo		Comment
19	29/05/18 16:16:36	GET	http://192.168.0.12/SugarCRM/cache/jsL...	404	Not Fou...	6...	1.275 bytes	Bajo		Comment
23	29/05/18 16:16:39	GET	http://192.168.0.12/SugarCRM/cache/jsL...	404	Not Fou...	2...	1.275 bytes	Bajo		Comment
24	29/05/18 16:24:52	GET	http://192.168.0.12/SugarCRM/index.php...	504	Gatewa...	2...	305 bytes	Bajo		

Fuente: El autor

La figura 52 permite validar la información de respuesta correspondiente a la revisión del Tag Script, SetCookie. Con esta información se cumple con los atributos que debe tener las cookies como ubicación, fecha de expiración futura e información del ID de la sesión que se muestra cifrado.

Figura 53 Información TagFormHidden

HTTP/1.1 200 OK
 Date: Tue, 29 May 2018 19:16:14 GMT
 Server: Apache/2.4.17 (Win32) OpenSSL/1.0.2d PHP/5.6.15
 X-Powered-By: PHP/5.6.15
 Expires: Thu, 19 Nov 1981 08:52:00 GMT
 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
 Pragma: no-cache
 Set-Cookie: sugar_login_id=20-1; expires=Mon, 27-Aug-2018 19:16:14 GMT; Max-Age=7776000; path=/
 Set-Cookie: sugar_user_theme=Sugar5; expires=Wed, 29-May-2019 19:16:14 GMT; Max-Age=31536000
 Set-Cookie: PHPSESSID=1aajt549abstin0mdrrcusm6a3; path=/
 Location: index.php?module=Home&action=index
 Content-Length: 2133
 Keep-Alive: timeout=5, max=100
 Connection: Keep-Alive
 Content-Type: text/html; charset=UTF-8

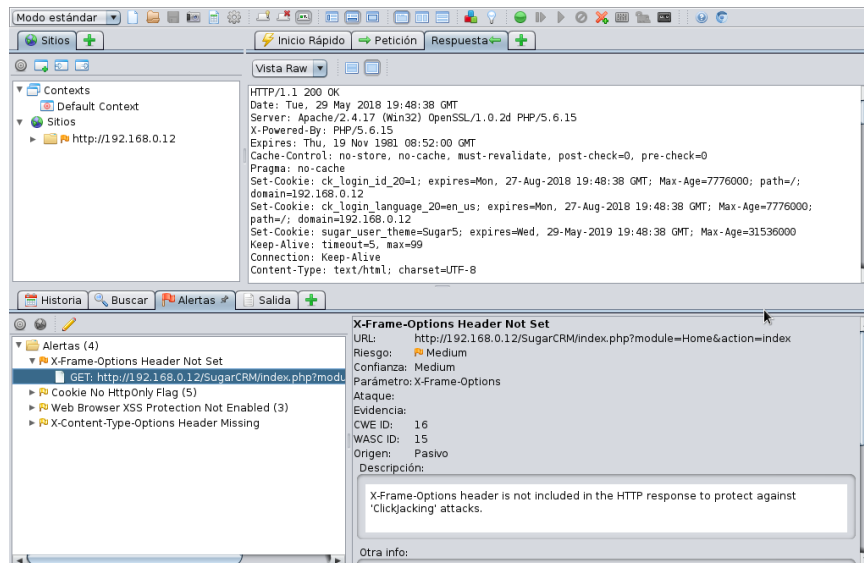
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

ID	Req. Timestamp	Método	URL	Co...	Reason	RTT	Size Resp. B...	Highest Al...	No...	Tags
1	29/05/18 16:03:35	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	5...	15.235 bytes	Medio		Form, Passwo...
12	29/05/18 16:05:25	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	2...	15.235 bytes	Medio		Form, Passwo...
13	29/05/18 16:08:52	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	2...	15.235 bytes	Medio		Form, Passwo...
14	29/05/18 16:15:49	POST	http://192.168.0.12/SugarCRM/index.php...	302	Found	1...	2.133 bytes	Bajo		Script, SetCo...
15	29/05/18 16:16:00	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	3...	15.263 bytes	Medio		Form, Passwo...
16	29/05/18 16:16:21	POST	http://192.168.0.12/SugarCRM/index.php...	302	Found	3...	2.133 bytes	Bajo		Script, SetCo...
17	29/05/18 16:16:24	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	9...	89.086 bytes	Medio		Form, Hidden...
18	29/05/18 16:16:36	GET	http://192.168.0.12/SugarCRM/cache/jsL...	404	Not Fou...	8...	1.275 bytes	Bajo		Comment
19	29/05/18 16:16:36	GET	http://192.168.0.12/SugarCRM/cache/jsL...	404	Not Fou...	6...	1.275 bytes	Bajo		Comment
23	29/05/18 16:16:39	GET	http://192.168.0.12/SugarCRM/cache/jsL...	404	Not Fou...	2...	1.275 bytes	Bajo		Comment
24	29/05/18 16:24:52	GET	http://192.168.0.12/SugarCRM/index.php...	504	Gatewa...	2...	305 bytes	Bajo		

Fuente: El autor

En la figura 53, se observa información para el *tag Form-Hidden* correspondiente a la petición y respuesta, donde aparece la información de *Setcookies* y atributos de los cookies.

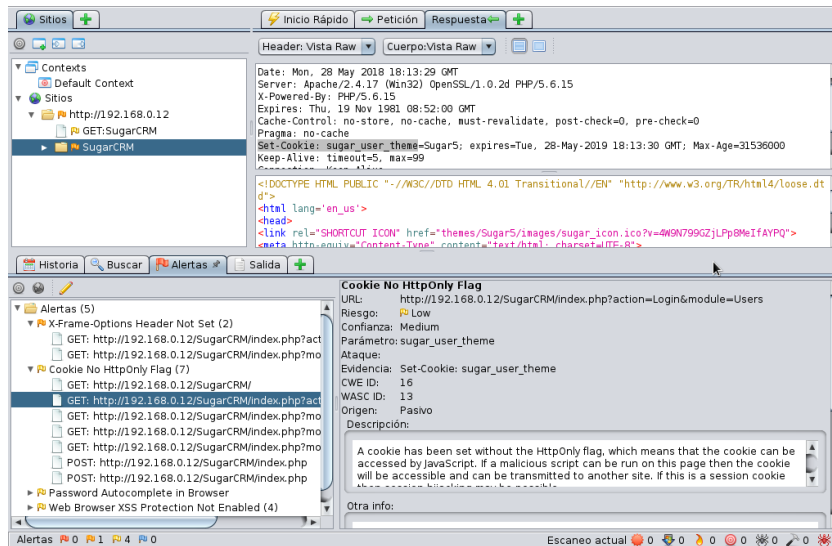
Figura 54 Información de alerta para *X-Frame-OptionHeader*



Fuente: El autor

En la figura 54, aparece como alerta de tipo medio: *X-Frame-OptionHeaderNot Set*, con la descripción correspondiente: *X-FrameOptions* no está incluido en la respuesta HTTP la protección contra "ataques de *clickjacking*". Aclarando que los ataques de *clickjacking* son ataques donde el usuario ingresa a una aplicación web de supuesto, sobre el cual realiza una serie de acciones, este tipo de ataques se pueden presentar a través de una etiqueta `<iframe>` de HTML por donde se coloca una página web dentro de otra.

Figura 55 Alerta Cookie No HttpOnly



Fuente: El autor

En la figura 55, se observa como alerta baja la correspondiente a Cookie No HttpOnly Flag, con la descripción de: Se ha configurado una cookie sin el indicador HttpOnly, lo que ocasiona que se pueda acceder a las cookies mediante JavaScript, ejecutando script malicioso la presencia de la etiqueta <iframe> de HTML, para acceder a las cookies y lo que permite embeber una página web dentro de otra.

11.5.2 Resultado de la prueba para atributos de cookies (OTG-SESS-002): De la prueba ejecutada se concluye que a través de la directiva de http Set-Cookie es donde se almacena la información de las cookies, de igual manera los http de respuesta pueden tener varios Set-Cookies.

Para uno de los atributo como lo es path en el servidor de *SugarCRM*, este se encuentra como “path=/”, lo que es correcto por el carácter de “/”, limitando el uso de la cookies para ingresar a páginas que estén dentro del directorio o ruta.

Para el atributo de expires se cuenta con la información de fecha futura, donde existe información como el id de sesión que no representa vulnerabilidad teniendo en cuenta que está cifrada la información, para el atributo Domain, aparece la IP del servidor donde esta *SugarCRM* sin indicar vulnerabilidad. Con la información anterior se puede concluir que el Servidor de *SugarCRM* cuenta con el control de las cookies requerido.

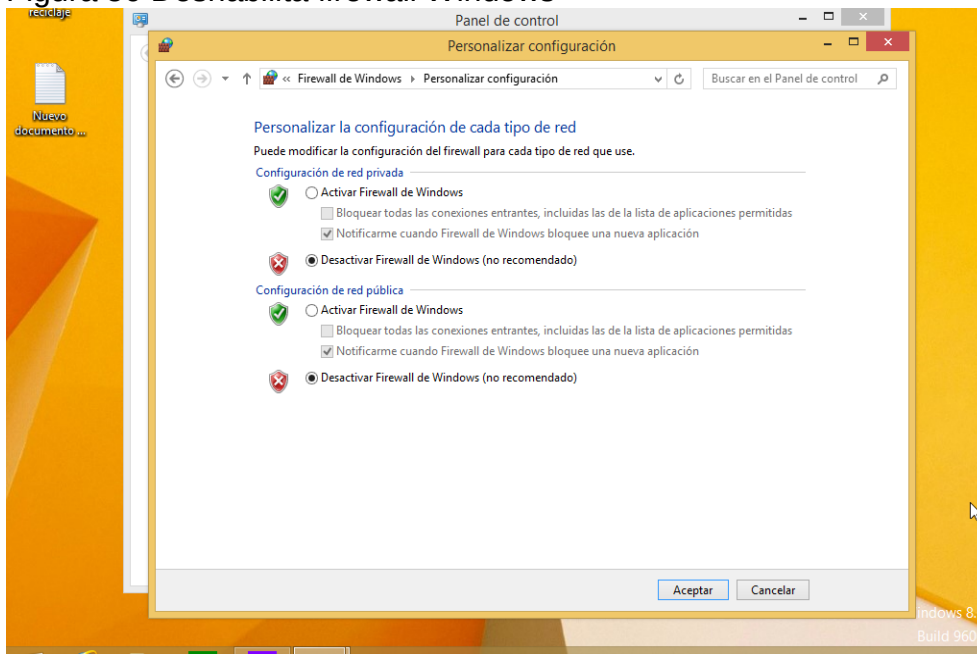
Para las Alertas generadas desde la herramienta de OWASP-ZAP, no se indica vulnerabilidad o alerta para el atributo de HttpOnly, es decir que no estaría expuesto a un ataque de script XSS.

11.6 Pruebas de seguridad a la validación de entrada

11.6.1 Pruebas de Inyecciones de SQL (OTG-INPVAL-005): El objetivo de esta prueba es validar la vulnerabilidad de *SugarCRM* a un ataque de SQL injection, verificando los datos transmitidos desde el cliente a la aplicación web con el objetivo de leer o modificar datos de la base de datos de la aplicación. La prueba es ejecutada desde kali Linux con sqlmap al servidor de *SugarCRM*.

En la figura 56, se deshabilita el firewall de Windows desde la maquina donde está instalado el servidor web de *SugarCRM*, ingresando desde panel de control – firewall de Windows. Este proceso se realiza con el objetivo de realizar la prueba de inyección de SQL sin que se presente alguna restricción.

Figura 56 Deshabilita firewall Windows



Fuente: El autor

Con el comando `sqlmap -u http://192.168.0.12/SugarCRM/ --dbs` descrito en la figura 57 se ejecuta el ataque con sqlmap para obtener toda la información de las bases de datos existentes en el servidor de *SugarCRM* como se observa a continuación.

Figura 57 Ataque SQL Injection

```

root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@kali:~# sql -u http://192.168.0.12/SugarCRM/ --dbs
bash: sql: no se encontró la orden
root@kali:~# sqlmap -u http://192.168.0.12/SugarCRM/ --dbs

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 11:51:38

[11:51:38] [WARNING] you've provided target URL without any GET parameters (e.g.
'http://www.site.com/article.php?id=1') and without providing any POST paramete
rs through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q] y
[11:53:16] [INFO] testing connection to the target URL
sqlmap got a 301 redirect to 'http://192.168.0.12:80/SugarCRM/index.php?action=L
ogin&module=Users'. Do you want to follow? [Y/n] Y
  
```

Fuente: El autor

Figura 58 Ataque Denegado SQL Injection

```

do you want to try URI injections in the target URL itself? [Y/n/q] Y
[12:03:26] [INFO] testing connection to the target URL
sqlmap got a 301 redirect to 'http://192.168.0.12:80/SugarCRM/index.php?action=Login&module=Users'. Do you want
to follow? [Y/n] Y
[12:03:29] [CRITICAL] previous heuristics detected that the target is protected by some kind of WAF/IPS/IDS
[12:03:29] [INFO] testing if the target URL is stable
[12:03:29] [WARNING] URI parameter '#1*' does not appear to be dynamic
[12:03:29] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[12:03:29] [INFO] testing for SQL injection on URI parameter '#1*'
[12:03:29] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[12:03:29] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace'
[12:03:29] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[12:03:29] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[12:03:29] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[12:03:29] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[12:03:30] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[12:03:30] [INFO] testing 'MySQL inline queries'
[12:03:30] [INFO] testing 'PostgreSQL inline queries'
[12:03:30] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[12:03:30] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[12:03:30] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[12:03:30] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[12:03:30] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[12:03:30] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[12:03:30] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[12:03:30] [INFO] testing 'Oracle AND time-based blind'
[12:03:30] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[12:03:30] [WARNING] using unescaped version of the test because of zero knowledge of the back-end DBMS. You can
try to explicitly set it with option '--dbms'
[12:03:30] [WARNING] URI parameter '#1*' does not seem to be injectable
[12:03:30] [CRITICAL] all tested parameters appear to be not injectable. Try to increase '--level'/'--risk' valu
es to perform more tests. Also, you can try to rerun by providing either a valid value for option '--string' (or
'--regexp'). If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could
retry with an option '--tamper' (e.g. '--tamper=space2comment')
[12:03:30] [WARNING] HTTP error codes detected during run:
403 (Forbidden) - 46 times, 404 (Not Found) - 103 times
  
```

Fuente: El autor

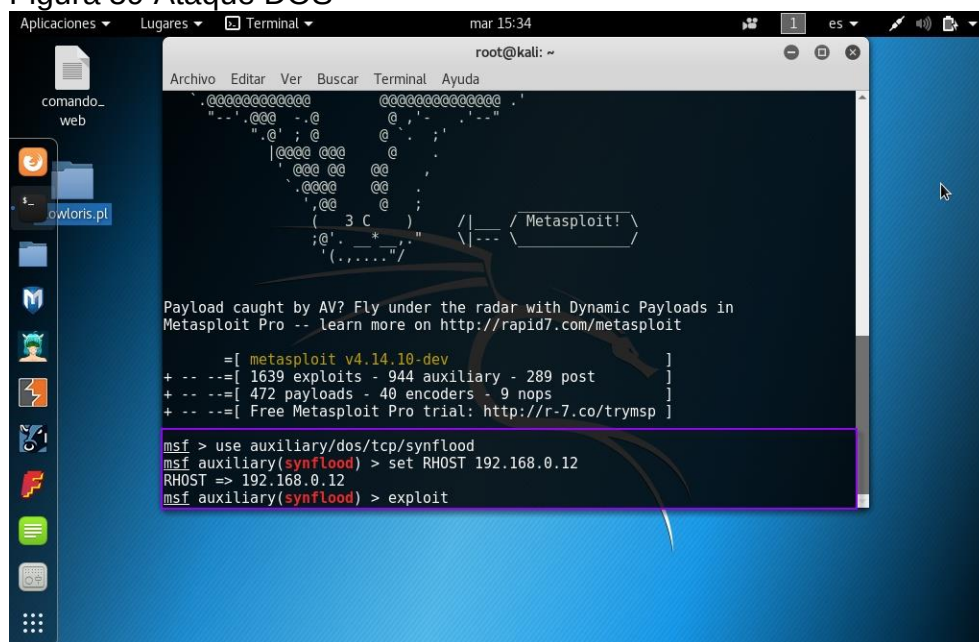
En la figura 58 se observa los mensajes de respuesta despues de la ejecucion del comandodescrito con anterioridad para efectuar la Injection de SQL, que no es

exitosa por que el sitio cuenta con filtros en la entrada de texto, dado por que en la URL no se encontraron variables susceptibles de inyeccion por los controles implementados en el formulario para manejar el ingreso de texto como sentencias sql.

11.6.1.1 Resultado de las pruebas de Inyecciones de SQL (OTG-INPVAL-005): La prueba *pentesting* realizada al servidor *Sugar CRM*, con el ataque de SQL Injection es denegado, en la herramienta usada para el ataque sqlmap se observa los diferentes mensajes descritos en la figura 69, que indican que no fue exitoso el ataque de SQL Injection.

11.6.2 Prueba la saturación del Búfer (OTG-INPVAL-014): El objetivo de esta prueba es validar la vulnerabilidad de *SugarCRM* a ataques de denegación de servicio, enviando varias peticiones TCP de manera que se logre la saturación del servidor y por ende la denegación del servicio. La prueba se realiza desde Kali Linux utilizando la herramienta de metasploit, realizando la apertura por los comandos `msfconsole` – use `auxiliary/dos/tcp/synflood`.

Figura 59 Ataque DOS

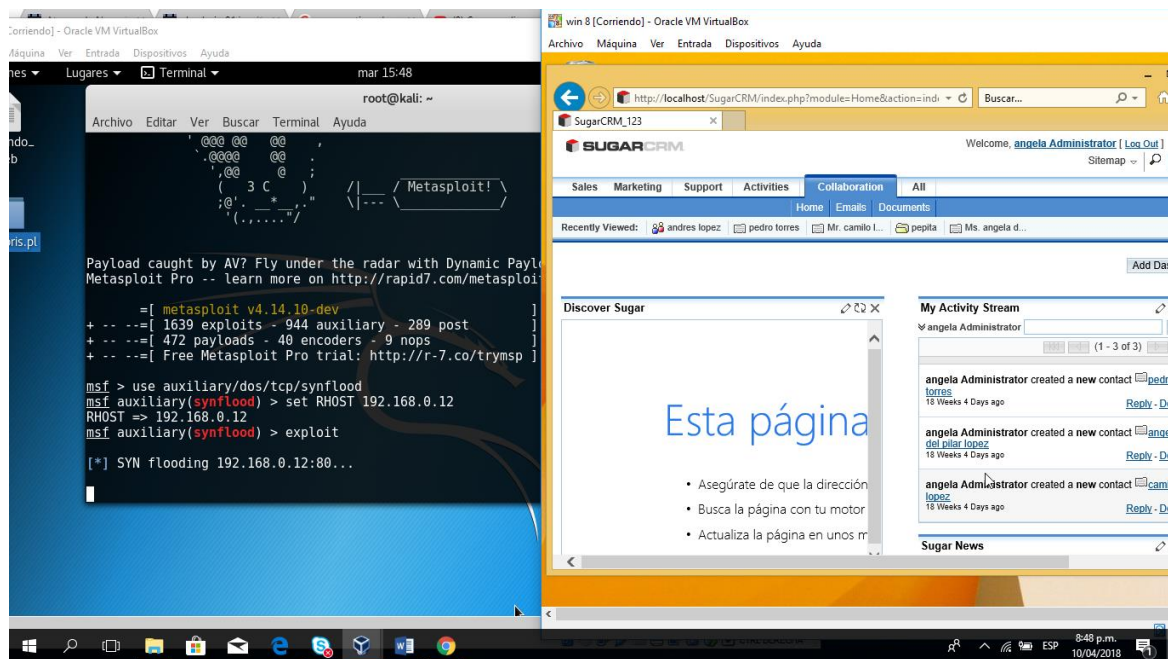


```
root@kali: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
Payload caught by AV? Fly under the radar with Dynamic Payloads in  
Metasploit Pro -- learn more on http://rapid7.com/metasploit  
[ metasploit v4.14.10-dev ]  
+ -- --[ 1639 exploits - 944 auxiliary - 289 post ]  
+ -- --[ 472 payloads - 40 encoders - 9 nops ]  
+ -- --[ Free Metasploit Pro trial: http://r-7.co/trymsp ]  
  
msf > use auxiliary/dos/tcp/synflood  
msf auxiliary(synflood) > set RHOST 192.168.0.12  
RHOST => 192.168.0.12  
msf auxiliary(synflood) > exploit
```

Fuente: El autor

Para ejecutar la prueba de manera exitosa, una vez se tiene el ingreso en la ruta auxiliary(synflood) se usa el comando set con el RHOST de la IP del servidor Web de *SugarCRM*, posterior a la confirmación se finaliza usando el comando exploit para empezar con el ataque. Para observar el ataque, se debe estar actualizando la página constantemente, aproximadamente el tiempo para que sea efectivo el ataque es entre minuto y medio y dos minutos (1:30 y 2:00).

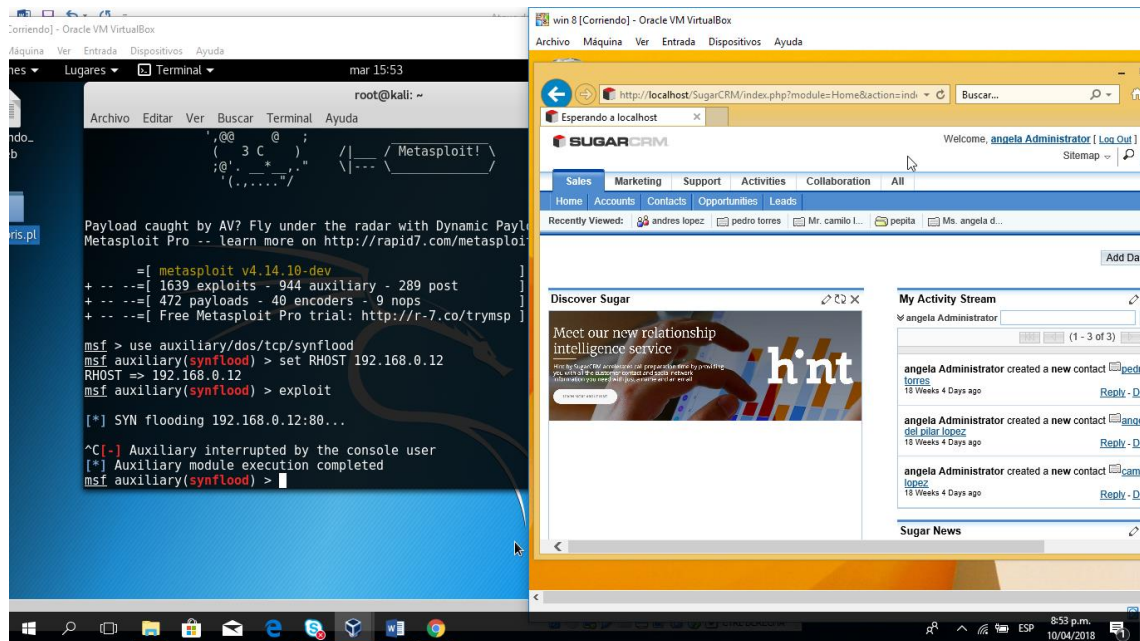
Figura 60 Ataque Dos -Metasploit



Fuente: El autor

En la figura 60 se visualiza el resultado del ataque ejecutado, las imágenes del servidor de *SugarCRM*, indican que no están disponibles y así se genera la actualización de la página no cargan. En la consola de metasploit se visualiza como se inicia el envío de información al servidor de *SugarCRM* hacia el puerto 80.

Figura 61 Luego de detener Ataque Dos



Fuente: El autor

En la figura 61, Se visualiza como se detiene el ataque con el mensaje de confirmación “Auxiliary interrupted by the console user” y en las imágenes se observa cómo se restablecen las imágenes en el servidor de *SugarCRM*.

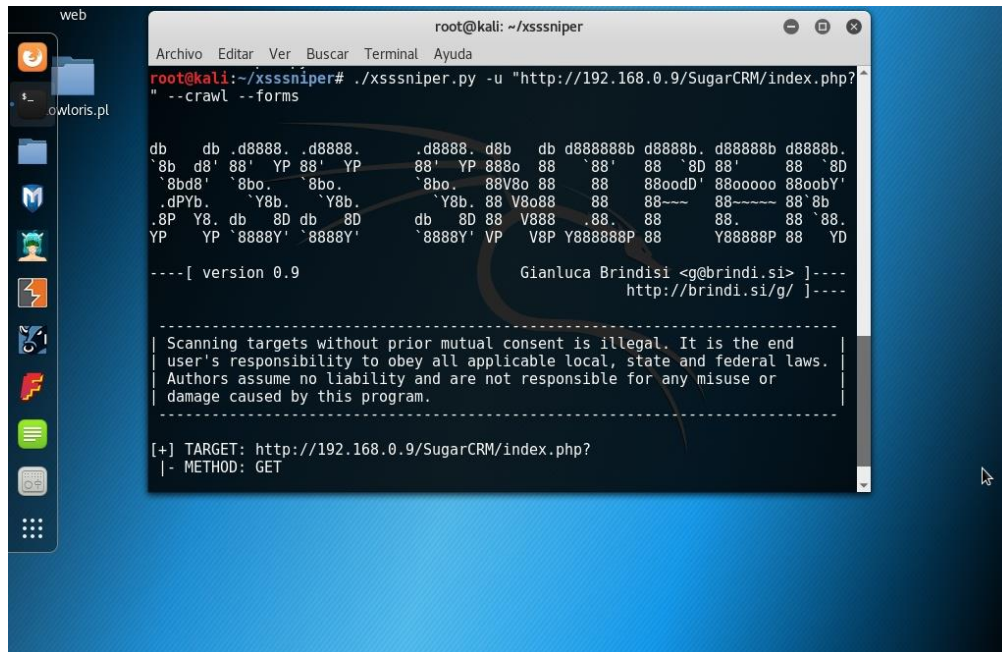
11.6.2.1 Resultado de la prueba de saturación del Búfer (OTG-INPVAL-014):

Con la prueba se ejecuta se observa que el servidor de SugarCRM es vulnerable a el tipo de ataque correspondiente a saturacion del bufer. Con el resultado se concluye que el servidor es saturado por las peticiones que son enviadas desde metasploit, logrando simular una inundacion de las mismas y llevando a la saturacion y denegacion de servicio en sugarCRM

11.6.3 Pruebas de Cross Site Scripting almacenados (OTG-INPVAL-002) :

El objetivo de la prueba es confirmar si el sitio web SugarCRM es vulnerable a los ataques de cross-site Scripting, es decir que permita el almacenamiento de informacion maliciosa que se ejecuta en el navegador posteriormente. La prueba se ejecuta desde Kali Linux con la herramienta *xsssniper*.

Figura 62 Herramienta xsssniper



```
root@kali: ~/xsssniper
Archivo Editar Ver Buscar Terminal Ayuda
root@kali:~/xsssniper# ./xsssniper.py -u "http://192.168.0.9/SugarCRM/index.php?" --crawl --forms

db db .d8888. .d8888. .d8888. d8b db d888888b d8888b. d88888b d8888b.
`8b d8' 88' YP 88' YP 88' YP 888o 88 `88' 88 `8D 88' 88 `8D
`8bd8' `8bo. `8bo. `8bo. 88V8o 88 88 88oodD' 88ooooo 88oobY'
.dPYb. `Y8b. `Y8b. `Y8b. 88 V8o88 88 88~~~ 88~~~~ 88`8b
.8P Y8. db 8D db 8D db 8D 88 V888 .88. 88 88. 88`88.
YP YP `8888Y' `8888Y' `8888Y' VP V8P Y888888P 88 Y88888P 88 YD

----[ version 0.9 Gianluca Brindisi <g@brindi.si> ]----
http://brindi.si/g/ ]----

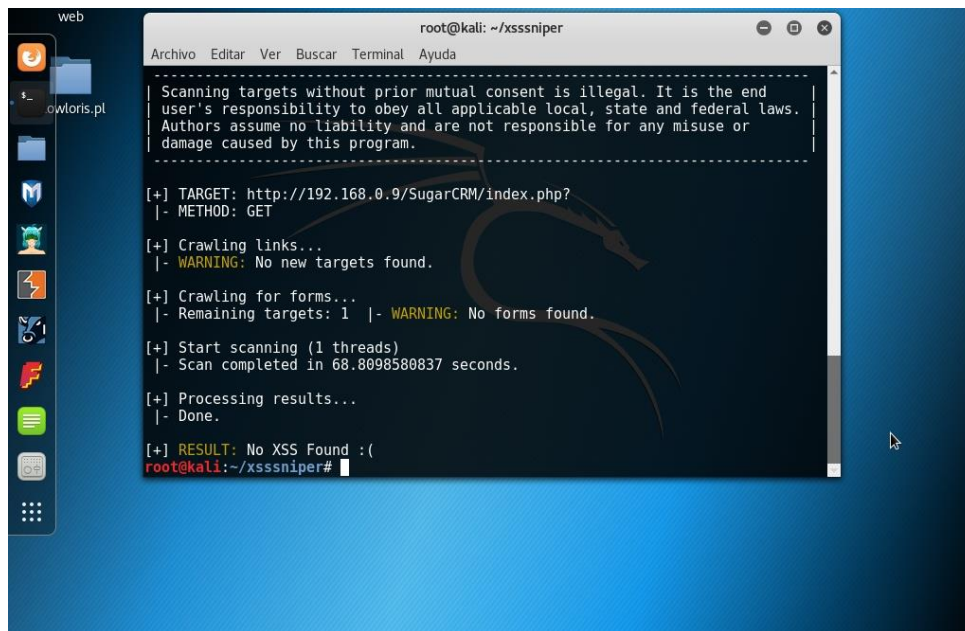
-----
[ Scanning targets without prior mutual consent is illegal. It is the end
user's responsibility to obey all applicable local, state and federal laws.
Authors assume no liability and are not responsible for any misuse or
damage caused by this program.
-----

[+] TARGET: http://192.168.0.9/SugarCRM/index.php?
|- METHOD: GET
```

Fuente: El autor

En la figura 62 se visualiza la herramienta xsssniper utilizada desde Kali linux, realizando la descarga desde el repositorio en github. Para la ejecución del escaneo completo del sitio web de SugarCRM, se usa el siguiente comando: `./xsssniper.py -u "http://192.168.0.9/SugarCRM/index.php?" --crawl --forms`, los comandos utilizados `--crawl` (usado para escaneo específico a un sitio web) y el `--forms` (escaneo de formularios) se usan ambos comandos de manera que se escanee de manera completa el sitio web.

Figura 63 Resultado de Escaneo



```
root@kali: ~/xssniper
Archivo Editar Ver Buscar Terminal Ayuda

-----
Scanning targets without prior mutual consent is illegal. It is the end
user's responsibility to obey all applicable local, state and federal laws.
Authors assume no liability and are not responsible for any misuse or
damage caused by this program.
-----

[+] TARGET: http://192.168.0.9/SugarCRM/index.php?
|- METHOD: GET

[+] Crawling links...
|- WARNING: No new targets found.

[+] Crawling for forms...
|- Remaining targets: 1 |- WARNING: No forms found.

[+] Start scanning (1 threads)
|- Scan completed in 68.8098580837 seconds.

[+] Processing results...
|- Done.

[+] RESULT: No XSS Found :(
root@kali:~/xssniper#
```

Fuente el autor

En la figura 63 se observa el resultado de la revision de la herramienta para el sitio web. Una vez se ejecuta el comando se visualiza que toma el servidor de SugarCRM como objetivo y le metodo GET, de igual manera en el rastreo realizado por los comandos utilizados, la respuesta corresponde a que no se encuentran objetivos. De igual manera se visualiza el tiempo del escaneo y la terminacion del mismo con el mensaje : “No XSS Found”.

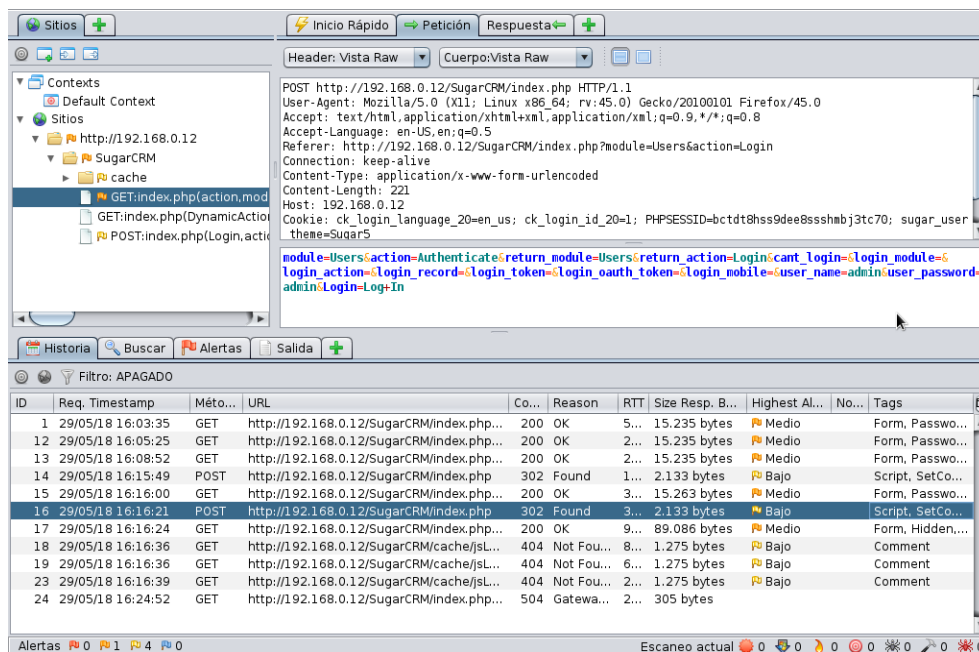
11.6.3.1 Resultado de la prueba de Cross Site Scripting almacenados (OTG-INPVAL-002) : El resultado obtenido de la prueba ejecuta con la herramienta xss sniper al servidor web de *SugarCRM*, confirma que la aplicacion de *SugarCRM* no es vulnerable a este tipo de ataque Cross site scripting almacenados, basando lo anterior en el resultado del comando ejecutado desde la herramienta para rastrear formularios o enlaces, con respuesta final del escaneo de “No XSS Found”. Este ataque no fue efectivo en el servidor de la aplicación de SugarCRM, porque se están usando métodos para filtrar las entradas de datos, evitando la inserción de script maliciosos.

11.7 Pruebas para Criptografía débil

11.7.1 Pruebas para el envío de información sensible por canales sin encriptar (OTG-CRYPST-003): El objetivo de esta prueba es validar los datos sensibles que son transmitidos sin ser cifrados, lo que indicaría una vulnerabilidad en la criptografía. Para la ejecución de la prueba se utiliza la herramienta OWASP-ZAP desde Kali Linux.

Para utilizar la herramienta OWASP-ZAP como proxy se realiza la configuración, desde el menú de herramientas opciones y se elige la opción Proxy local ingresando la información de Address y puerto. De igual manera desde el navegador se realiza la configuración de proxy para la prueba a ejecutar, indicando la IP y puerto utilizado. Los datos utilizados tanto en la herramienta OWASP-ZAP como en el navegador usado son la misma, utilizamos la IP de localhost 120.0.0.1 y el puerto 8080.

Figura 64 Logueo sin cifrado



The screenshot displays the OWASP ZAP web interface. The top pane shows a raw HTTP POST request to `http://192.168.0.12/SugarCRM/index.php` with the following body:

```
module=Users&action=Authenticate&return_module=Users&return_action=Login&cant_login=&login_module=&login_action=&login_record=&login_token=&login_oauth_token=&login_nobile=&user_name=admin&user_password=admin&login=Log-In
```

The bottom pane shows a history table with the following data:

ID	Req. Timestamp	Método	URL	Co...	Reason	RTT	Size Resp. B...	Highest Al...	No...	Tags
1	29/05/18 16:03:35	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	5...	15.235 bytes	Medio		Form, Passwo...
12	29/05/18 16:05:25	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	2...	15.235 bytes	Medio		Form, Passwo...
13	29/05/18 16:08:52	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	2...	15.235 bytes	Medio		Form, Passwo...
14	29/05/18 16:15:49	POST	http://192.168.0.12/SugarCRM/index.php...	302	Found	1...	2.133 bytes	Bajo		Script, SetCo...
15	29/05/18 16:16:00	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	3...	15.263 bytes	Medio		Form, Passwo...
16	29/05/18 16:16:21	POST	http://192.168.0.12/SugarCRM/index.php...	302	Found	3...	2.133 bytes	Bajo		Script, SetCo...
17	29/05/18 16:16:24	GET	http://192.168.0.12/SugarCRM/index.php...	200	OK	9...	89.086 bytes	Medio		Form, Hidden...
18	29/05/18 16:16:36	GET	http://192.168.0.12/SugarCRM/cache/jsl...	404	Not Fou...	8...	1.275 bytes	Bajo		Comment
19	29/05/18 16:16:36	GET	http://192.168.0.12/SugarCRM/cache/jsl...	404	Not Fou...	6...	1.275 bytes	Bajo		Comment
23	29/05/18 16:16:39	GET	http://192.168.0.12/SugarCRM/cache/jsl...	404	Not Fou...	2...	1.275 bytes	Bajo		Comment
24	29/05/18 16:24:52	GET	http://192.168.0.12/SugarCRM/index.php...	504	Gatewa...	2...	305 bytes			

Fuente: el autor

Luego de tener la configuración de la herramienta como proxy, como se detalla en la parte de arriba, se inicia con las pruebas desde el servidor de *SugarCRM*, enviando un logueo a la aplicación para iniciar desde la herramienta de OWASP-ZAP el escaneo que permita validar el resultado de la prueba *pentesting* a realizar Pruebas para el envío de información sensible por canales sin encriptar.

En la figura 64, se muestra la información obtenida desde la herramienta posterior a realizar el logueo en *sugarCRM* con los datos de usuarios y clave correctos de ingreso a la aplicación y sin encriptación. El usuario y clave de ingreso son admin.

11.7.2 Resultado Pruebas para el envío de información sensible por canales sin encriptar (OTG-CRYPST-003): Con la prueba ejecutada se valida que existe una vulnerabilidad importante sobre una aplicación ya instalada que pone en riesgo la confidencialidad e integridad de los datos de usuario, porque viajan los datos sensibles sin ser cifrados. Para evitar este tipo de amenazas los datos se deben proteger en el almacenamiento y también en el transporte.

12. CONSOLIDADO DE RESULTADOS

Finalizadas las pruebas ejecutadas sobre la aplicación sugarCRM se concluye:

Tabla 4.Resultado de las pruebas

Test de Prueba	Herramienta	Vulnerable SI o NO	Descripción del resultado
Prueba de los métodos HTTP (OTG-CONFIG-006)	Complemento <i>RestClient</i> y <i>Curl</i>	NO	Se concluye que no aparece información visible de los métodos del protocolo HTTP usados en el servidor <i>SugarCRM</i> , por lo tanto se concluye que el servidor <i>SugarCRM</i> cuenta con el control de los métodos.
Prueba proceso de registro de usuarios (OTG-IDENT-002)	Complemento <i>LiveHTTPHeads</i>	SI	Se concluye con la ejecución de las pruebas que es vulnerable a la manipulación o modificación de la información enviada, teniendo en cuenta que información sensible e importante viaja de forma clara sin ser cifrada de manera que puede ser interceptada fácilmente.
Pruebas para determinar un mecanismo de bloqueo débil (OTG-AUTHN-003)	Hydra	SI	Se concluye que <i>SugarCRM</i> si es vulnerable a este ataque, con el uso de diccionarios de claves o usuarios.
Pruebas para determinar el escalamiento de privilegios (OTG-AUTHZ003).	WebScara	NO	Se concluye que no es vulnerable, en las pruebas se observan varios campos ocultos que representan amenazas.
Prueba para atributos de cookies (OTG-SESS-002)	OWASP-ZAP	NO	Se concluye en el análisis de la ejecución de las pruebas para cada uno de los atributos que no presentan amenazas para <i>SugarCRM</i> , porque cumple con los parámetros de configuración establecidos para obtener niveles de seguridad en aplicaciones web.

Tabla 5. (Continuación)

Test de Prueba	Herramienta	Vulnerable SI o NO	Descripción del resultado
Pruebas de Inyecciones de SQL (OTG-INPVAL-005)	Sqlmap – kali Linux	NO	Se concluye que no es vulnerable teniendo en Cuenta que el ataque es denegado.
Pruebe la saturación del Búfer (OTG-INPVAL-014)	Metasploit	SI	Se concluye que <i>SugarCRM</i> si es vulnerable a este tipo de ataques.
Pruebas para Cross site scripting almacenados(OTG-INPVAL-002)	XSSSNIPER	NO	Se concluye que <i>SugarCRM</i> no es vulnerable a este tipo de ataques.
Pruebas para el envío de información sensible por canales sin encriptar (OTG-CRYPST-003)	OWASP ZAP	SI	Se concluye que <i>SugarCRM</i> , no está cifrando algunos datos sensibles, de manera que representa una amenaza teniendo en cuenta que fácilmente puede ser conocida y utilizarse con otros fines.

Fuente : el autor

13.CONCLUSIONES

- Para obtener mejores resultados en las pruebas de *pentesting web* según la presente monografía, se establecen los pasos predeterminados como lo son: identificación de la herramienta a atacar, el tipo de pruebas a evaluar y las vulnerabilidades a identificar.
- De acuerdo a los lineamientos como están especificados en la monografía, se siguen todas las pautas de cada una de las fases para realizar *pentesting* como son: reconocimiento, enumeración, análisis, explotación y documentación.
- Se logra obtener resultados óptimos de las pruebas de *pentesting* ejecutados sobre *SugarCRM*, para lo cual se hizo requerido el conocimiento y profundización en el uso de varias herramientas para aplicar pruebas de *pentesting*.
- Con la elaboración de la presente monografía se logra entender y conocer todo lo relacionado con pruebas de *pentesting* y de forma práctica se aplican los conceptos sobre la herramienta de *SugarCRM*.
- Como experiencia a la aplicación de pruebas *pentesting* sobre *SugarCRM*, se logra entender la importancia de aplicar todas las políticas de seguridad a todo nivel en las organizaciones.
- En la presente monografía se expone de manera general las herramientas más utilizadas en el mercado para ejecutar pruebas de *pentesting*, en el desarrollo de la misma se usan algunas de ellas permitiendo que se logre adquirir más conocimiento de la misma.

14.RECOMENDACIONES

Teniendo en cuenta las vulnerabilidades existentes en las aplicaciones web, es importante tomar medidas para la mejorar la seguridad, a continuación, se relaciona algunas de ellas:

- Realizar una adecuada administración de autenticaciones y contraseñas, todos los recursos y páginas deben requerir autenticación.
- Es necesario mantener actualizado el paquete de XAMPP usado en el servidor web para evitar vulnerabilidades en cada una de las herramientas que lo integran.
- Es importante asegurar las variables para que tengan un formato correcto para que se puedan utilizar en la consulta de una manera más segura.
- También se puede aplicar el escapar los caracteres especiales que son usados en las consultas, de manera que sea añadida una barra invertida \ delante de las cadenas de las consultas SQL, para evitar que se dañen o afecten la consulta.
- Utilizar administración de sesiones de manera que sea identificado cada solicitud.
- Manejar políticas de control de acceso para datos y aplicaciones implementando las reglas del negocio de la aplicación, los tipos de datos, criterios para autorización de acceso y los controles asociados para otorgarlos y controlarlos.
- Implementar políticas de roles y privilegios de manera que cada usuario ingrese a las actividades que requiera.
- Debe manejarse registros de logs.
- Proteger los almacenamientos aplicados a los datos sensibles
- Seguridad en las comunicaciones. Asegurar la transmisión de información, se puede usar TLS para proteger la conexión
- Manejar políticas para almacenamiento seguro de la información.
- Utilizar firmas digitales o credenciales cifradas.
- Estar en constante actualización del software.
- Verificación de permisos de acceso y de usuarios, manejar una correcta administración de estos estableciendo los diferentes estados.
- Implementar políticas para el manejo de claves criptográficas.

BIBLIOGRAFIA

CABALLERO Alonso “Pruebas de penetración contra Aplicaciones Web” (Febrero 2006). {En línea}{13 de febrero de 2017} disponible en:
http://www.reydes.com/archivos/slides/eventos/T_PdPAW_Alonso_ReYDeS.pdf.

LEY 1273 DE 2009. (5 de Enero de 2009). Obtenido de Alcaldía de Bogotá:
<http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=34492>

MAULINI Mario “Desarrollo y Seguridad de Aplicaciones Web y Móviles”. (04 diciembre 2010). {En línea}. {13 de febrero de 2017} disponible en:
<http://tecnologiasweb.blogspot.com.co/2010/12/que-es-pen-test-herramientas-de-pen.html>.

OWASP. “Guía de Pruebas de OWASP” (3 de Agosto de 2015). {En línea}. {13 de febrero de 2017} disponible en:
https://www.owasp.org/images/8/80/Gu%C3%ADa_de_pruebas_de_OWASP_ver_3.0.pdf.

OWASP. “Testing for Input Validation.” (8 de Agosto de 2014). {En línea}. {14 de noviembre de 2016} disponible en:
https://www.owasp.org/index.php/Testing_for_Input_Validation.

OWASP “Top 10 -2017”{En línea}. {14 de noviembre de 2016} disponible en:
<https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf>

PREZI “Metodología Issaf”. (25 de marzo 2015). {En línea}. {13 de febrero de 2017}disponible en: <https://prezi.com/5ssonumqypgo/metodologia-issaf/>

REDES ZONE “Herramientas para *pentesting*”. (6 de Enero 2017){En línea}. {13 de febrero de 2017} Disponible en: <https://www.redeszone.net/2017/01/06/las-mejores-herramientas-hacking-este-2017/>.

REPORTE DIGITAL “*Pentesting*: Un mecanismo para encontrar brechas de seguridad en las empresas”.(24 Noviembre 2014). {En línea}. {13 de febrero de 2017} disponible en: <http://reportedigital.com/seguridad/pentesting-mecanismo-encontrar-brechas-seguridad-empresas/>.

SEGURIDAD Y ETICA. “Herramientas de *pentesting*” (11 de abril 2012). {En línea}. {13 de febrero de 2017} Disponible en:
<http://seguridadetica.wordpress.com/2012/04/11/5-heramientas-utiles-en-penetration-testing-para-aplicaciones-web/>.

SEGURIDAD INFORMATICA "*Pentesting* a servicios web: WSDL".(19 Noviembre 2013). {En línea}. {13 de febrero de 2017} disponible en: <https://seguinfo.wordpress.com/2013/11/19/pentesting-a-servicios-web-wsdl/>.

KALI LINUX. "What is Kali Linux?" (2016). {En línea}. {14 de noviembre de 2016} disponible en: <http://docs.kali.org/introduction/what-is-kali-linux>

UNAM "Aspectos Básicos de la Seguridad en Aplicaciones Web".(03 marzo 2010). {En línea}. {13 de febrero de 2017} disponible en: <http://www.seguridad.unam.mx/documento/?id=17>.

ANEXO A - ESCANEOS DE VULNERABILIDADES CON RAPID7

Como parte del proceso de pruebas de *pentesting*, se realiza un escaneo con la herramienta rapid7, utilizada para la detección de amenazas. Se elige esta herramienta porque permite verificar todo el ciclo de vulnerabilidades que implica detección, verificación y descubrimiento. De igual manera genera estadísticas con impacto, riesgo y reportes detallados.

Como resultado del escaneo realizado, se generan dos reportes. A continuación se relaciona las estadísticas del reporte:

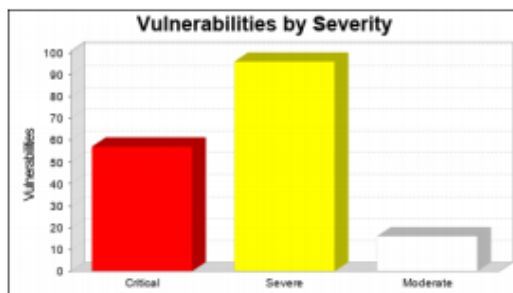
1. Executive Summary

This report represents a security audit performed by InsightVM from Rapid7 LLC. It contains confidential information about the state of your network. Access to this information by unauthorized personnel may allow them to compromise your network.

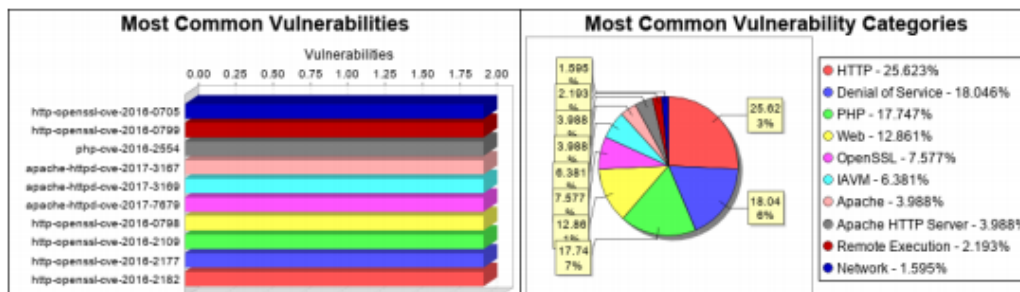
Site Name	Start Time	End Time	Total Time	Status
sugar 23	October 23, 2018 22:41, COT	October 23, 2018 22:44, COT	3 minutes	Success

There is not enough historical data to display overall asset trend.

The audit was performed on one system which was found to be active and was scanned.



There were 169 vulnerabilities found during this scan. Of these, 57 were critical vulnerabilities. Critical vulnerabilities require immediate attention. They are relatively easy for attackers to exploit and may provide them with full control of the affected systems. 96 vulnerabilities were severe. Severe vulnerabilities are often harder to exploit and may not provide the same access to affected systems. There were 16 moderate vulnerabilities discovered. These often provide information to attackers that may assist them in mounting subsequent attacks on your network. These should also be fixed in a timely manner, but are not as urgent as the other vulnerabilities.



There were 2 occurrences of the http-openssl-cve-2016-0705, http-openssl-cve-2016-0799, php-cve-2016-2554, apache-httpd-cve-2017-3167, apache-httpd-cve-2017-3169, apache-httpd-cve-2017-7679, http-openssl-cve-2016-0798, http-openssl-cve-2016-2109, http-openssl-cve-2016-2177 and http-openssl-cve-2016-2182 vulnerabilities, making them the most common vulnerabilities. There were 257 vulnerability instances in the HTTP category, making it the most common vulnerability category.

Vulnerabilidades por severidad

En el resumen resultado del escaneo realizado a la Aplicación de *SugarCRM*, se encuentro un total de 169 vulnerabilidades clasificadas por severidad con los siguientes resultados:

- 57 vulnerabilidades Críticas, catalogadas de esta manera porque son fáciles de explotar por los atacantes, por lo anterior requieren atención inmediata.
- 97 vulnerabilidades severas, que de igual manera requieren atención importante.
- 16 vulnerabilidades moderadas, que pueden ocasionar información a los atacantes para organizar ataques posteriores. De igual manera deben ser solucionadas de manera oportuna.

Vulnerabilidades más comunes

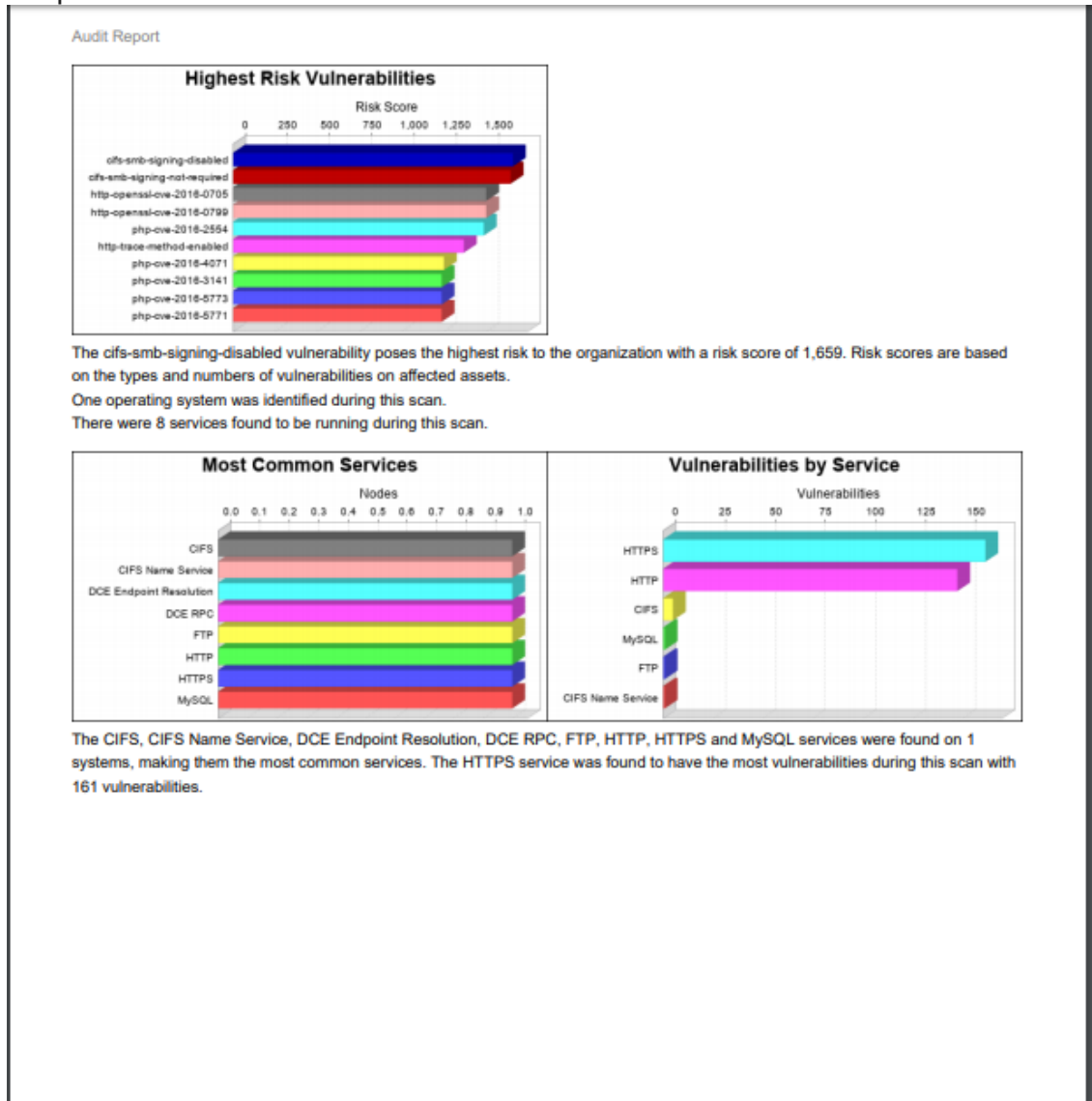
Dentro de las vulnerabilidades más comunes que presenta el reporte están las `http-openssl-cve-2016-XXXX`, que corresponden a un tipo de vulnerabilidades donde los atacantes pueden conocer comunicaciones confidenciales y encontrar claves de cifrado, es importante destacar que estas vulnerabilidades son ocasionadas por el protocolo *OpenSSL*.

En el reporte estadístico las tres (3) vulnerabilidades más comunes como resultado del escaneo son:

- `http-openssl-cve-2016-0705`, que corresponde a una vulnerabilidad de denegación de servicio, ocasionada por DSA (Algoritmo de firma digital) privadas mal construidas. Estos algoritmos son utilizados para la firma y verificación de firma y hacen parte de la generación de claves de 512 y 1024 bits.
- `http-openssl-cve-2016-0799`, corresponde a problemas de memoria, lo que puede ocasionar desbordamientos del buffer.
- `php-cve-2016-2554`, que corresponde a desbordamiento de búfer presentado en algunas versiones de PHP, permitiendo a los atacantes remotos causar una denegación de servicio o bloqueo de la aplicación.

Se debe aclarar que estas vulnerabilidades presentadas en el reporte del escaneo ejecutado desde *rapid7*, no son propiamente de la aplicación de *SugarCRM*, están relacionadas con el protocolo *OpenSSL*, pero generan vulnerabilidad directa en la

aplicación ya que puede ser la entrada por los atacantes para realizar una serie de ataques directos.



Altas Vulnerabilidades de riesgo

En el reporte estadístico, como mayor riesgo del escaneo realizado sobre la aplicación de *SugarCRM*, muestra *Cifs-smb-signing-disabled* con mayor puntuación de riesgo, que corresponde al descubrimiento de vulnerabilidades en la firma SMB deshabilitada. Esta vulnerabilidad indica que el servidor Windows está manejando de forma inadecuada solicitudes en las ejecuciones remotas de código, ocasionando que un atacante pueda inyectar y ejecutar código malicioso. De igual manera es importante aclarar que la firma SMB se puede habilitar de acuerdo al sistema operativo.

2. Discovered Systems

Node	Operating System	Risk	Aliases
192.168.0.7	Microsoft Windows 8.1	65,397	•ANGELA-WINDOWS8

Las imágenes que se adjuntan corresponden a las estadísticas de uno de los reportes generados de la herramienta rapid7 como resultado al escaneo realizado.

RESUMEN ANALÍTICO ESPECIALIZADO – RAE

Resumen Analítico especializado -RAE			
1. Información General			
Tema	Pentesting WEB aplicado a Sugar CRM		
Titulo	Pentesting WEB		
Autor	Ángela del pilar López Molina	Año	2018
Director	Daniel Felipe Palomo		
Fuente Bibliográfica	Aspectos Básicos de la Seguridad en Aplicaciones Web, 03 marzo 2010 disponible en: http://www.seguridad.unam.mx/documento/?id=17. Caballero Alonso “Pruebas de penetración contra Aplicaciones Web”, disponible en: http://www.reydes.com/archivos/slides/eventos/T_PdPAW_Alonso_ReYDeS.pdf. Guía de Pruebas de OWASP, disponible en: https://www.owasp.org/images/8/80/Gu%C3%ADa_de_pruebas_de_OWASP_ver_3.0.pdf. Herramientas de <i>pentesting</i>, Disponible en: http://seguridadetica.wordpress.com/2012/04/11/5-heramientas-utiles-en-penetration-testing-para-aplicaciones-web/. Herramientas para <i>pentesting</i>, Disponible en: https://www.redeszone.net/2017/01/06/las-mejores-herramientas-hacking-este-2017/ Las mejores herramientas online para <i>pentesting</i>, 18 Noviembre 2013 Disponible en: http://www.linuxito.com/seguridad/205-las-mejores-herramientas-online-para-pentesting. Ley 1273 de 2009, Disponible en http://www.deltaasesores.com/articulos/autores-invitados/otros/3576-ley-de-delitos-informaticos-en-colombia. Metodología Issaf, Disponible en: https://prezi.com/5ssonumqypgo/metodologia-issaf/		
Resumen	El tema de la monografía fue elegido por ser una de las actividades principales que se deben llevar a cabo en los temas relacionados con la seguridad de la información, que implica el uso de metodologías para llevar a cabo un correcto proceso de pruebas de penetración. De igual manera es un tema donde se requiere el conocimiento y manejo de herramientas que facilitan el escaneo y monitoreo de vulnerabilidades.		

Palabras Claves	ATAQUES INFORMATICOS, AMENAZAS, VULNERABILIDAD, RIESGO, SITIOS WEB, SPIDERING, SQL INJECTION.
Contenidos	1. TITULO 2. INTRODUCCION 3. DEFINICION DEL PROBLEMA 4. ALCANCE Y DELIMITACION DE PROYECTO 5. JUSTIFICACION 6. OBJETIVOS GENERAL Y ESPECIFICOS 7. MARCO REFERENCIAL 8. PROPUESTA METODOLOGICA 9. DESARROLLO DE PRUEBAS 10. ADECUACION DE AMBIENTES 11. EJECUCION DE LAS PRUEBAS DE PENTESTING 12. CONSOLIDADO DE RESULTADOS 13. PRODUCTO RESULTADO A ENTREGAR 14. CONCLUSIONES 15. RECOMENDACIONES BIBLIOGRAFIA

2. Descripción del problema de investigación
PLANTEAMIENTO DEL PROBLEMA: Las herramientas que escanean Vulnerabilidades en Aplicaciones Web esperan encontrar configuraciones inseguras de servidores y otras vulnerabilidades de seguridad conocidas que pueden generar ataques como por ejemplo: XSS, inyecciones de SQL y accesos a directorios entre los más comunes. Cuando un sitio web tiene vulnerabilidades conocidas como debilidades, se hace posible que los hackers se aprovechen de estas. El problema de la investigación consiste en identificar el concepto de <i>Pentesting</i> de manera que se muestre la importancia de su aplicación en la seguridad de las páginas web. FORMULACION DEL PROBLEMA: En la actualidad las aplicaciones web son muy importantes ya que nos presenta posibilidades ilimitadas y facilitan la vida con aplicaciones como los correos, foros, chats, blogs entre otros que permiten una mejor comunicación e interrelación. Las amenazas en internet se incrementan día a día por lo que tener un sitio web seguro es muy importante para las organizaciones, teniendo en cuenta que pueden ser blanco de muchos ataques desde la web. PREGUNTA DEL PROBLEMA: Con el incremento en el uso de aplicaciones web y todas las funcionalidades que ofrece, es necesario crear aplicaciones con altos niveles de seguridad, confidencialidad e integridad. De igual manera la seguridad de las aplicaciones permite proteger los datos, la pérdida de información que es considerado como uno de los activos más importantes en la actualidad. Teniendo en cuenta las descripciones anteriores, se establece EL siguiente interrogante a tener en cuenta. ¿Cuál sería la manera más eficaz de identificar vulnerabilidades a sitios web que cuentan con la herramienta <i>SugarCRM</i>?

3. Objetivos	
OBJETIVO GENERAL: Dar a conocer el método de <i>pentesting</i> para combatir las vulnerabilidades o brechas de seguridad en las aplicaciones web, logrando a través de las pruebas de <i>pentesting</i> web evaluar los controles y la seguridad de nuestras aplicaciones web.	
OBJETIVOS • Establecer los pasos predeterminados en las pruebas de <i>pentesting</i> web para obtener mejores resultados. • Aplicación de las fases establecidas para realizar <i>pentesting</i> web con las actividades correspondientes en cada una de ellas para lograr los resultados esperados. • Profundizar en el manejo y funcionalidad de las herramientas para pruebas de <i>pentesting</i> web, de manera que se realice uso adecuado de estas con resultados óptimos. • Investigar sobre la conceptualización de los <i>pentesting</i> y su aplicación en las organizaciones de manera que pueda adoptarse como práctica de seguridad. • Conocer la importancia sobre la aplicación de <i>pentesting</i> como política de seguridad en las organizaciones y la finalidad de las pruebas de <i>pentesting</i>. • Presentar una explicación sobre las herramientas en el mercado usadas para llevar a cabo las labores de <i>pentesting</i>.	ESPECIFICOS:
4. Metodología	
TIPO DE INVESTIGACIÓN: La Investigación que se aplica corresponde a la práctica de los conocimientos adquiridos en la especialización en seguridad informática y generando algunas investigaciones sobre cómo aplicar pruebas de <i>pentesting</i>.	
TÉCNICAS DE RECOLECCIÓN DE DATOS: La fuente de datos corresponde a la ejecución de pruebas de <i>pentesting</i> aplicado de manera práctica sobre la aplicación de Sugar CRM.	
TÉCNICAS DE ANÁLISIS DE DATOS: Se hace uso de las pruebas de <i>pentesting</i> planteadas por OWASP, con las cuales se genera la ejecución sobre la aplicación de Sugar CRM, con el resultado obtenido se realiza una breve explicación de porqué se presenta la vulnerabilidad y la manera como se comprueba.	
ETAPAS DEFINIDAS: Para la aplicación de las pruebas de <i>pentesting</i> fue necesaria la realización de las siguientes actividades:	
Configuración	del ambiente de pruebas
Definición	del alcance
Ejecución	de las pruebas de <i>pentesting</i>
Revisión y consolidación de resultados.	

5. Referentes teóricos

Las pruebas ejecutadas se basaron en el planteamiento de pruebas de la cuarta versión de la guía de pruebas de OWASP. Como planteamiento de las pruebas se tiene en cuenta las siguientes fases

1. Recolección y conocimiento del sistema u objetivo antes de la ejecución de las pruebas, para este caso nuestra aplicación de prueba es Sugar CRM, que corresponde a una aplicación gratis que funciona como un CRM.
2. Conocimiento de los aspectos a evaluar del sistema objetivo.
3. Ejecución de pruebas, evaluando la explotación de las posibles vulnerabilidades que tenga la aplicación.
4. análisis y resultado objeto de las pruebas ejecutadas.

6. Referentes Conceptuales

ATAQUES INFORMATICOS: en la actualidad existen varios tipos de ataques informáticos, consiste en tomar el control o causar daño a un sistema o aplicación informática.

AMENAZAS: las amenazas presentadas a los sistemas informáticos se pueden definir como una situación que puede causar daño o pérdida. Cuando las amenazas se materializan se pueden presentar los ataques.

VULNERABILIDAD: se define como una debilidad que puede presentar los sistemas de la información, que pueden ser aprovechadas para causar daño.

RIESGO: se define como la posibilidad de que las amenazas sean materializadas, los análisis de riesgos permite tomar los controles necesarios para la protección del sistema.

SITIOS WEB: corresponde a un sitio virtual, donde se puede tener archivos electrónicos y se tiene un conjunto de páginas web.

SPIDERING: es un programa que inspecciona las páginas del World Wide Web de forma metódica y automatizada. malicioso o investigativo.

SQL INJECTION: ataque generado a una base de datos presentado por la mala filtración de las variables, el ataque inyecta un código creado por el atacante al propio código fuente de la base de datos.

7. Resultados

Son ejecutadas ocho pruebas de *pentesting* de las cuales, en cuatro de ellas se encuentran vulnerabilidades en la aplicación de *Sugar CRM*. De igual manera se indica cual es la herramienta y los pasos para la ejecución de cada una de las pruebas y como se evidencia la vulnerabilidad para las que fueron positivas.

8. Conclusiones

Para obtener mejores resultados en las pruebas de *pentesting web* según la presente monografía, se establecen los pasos predeterminados como lo son: identificación de la herramienta a atacar, el tipo de pruebas a evaluar y las vulnerabilidades a identificar. De acuerdo a los lineamientos como están especificados en la monografía, se siguen todas las pautas de cada una de las fases para realizar *pentesting* como son: reconocimiento, enumeración, análisis, explotación y documentación.

