UACM

Universidad Autónoma de la Ciudad de México

Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES

Implementación de un servidor web para pruebas de seguridad como: autenticación e intercepción de datos en Owncloud así como escaneo de puertos en SurDoc

TRABAJO RECEPCIONAL
PARA OBTENER EL TÍTULO DE LICENCIADA EN
INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES

P R E S E N T A:

Jhoana Magdalena Fernández López

Director del trabajo recepcional

Dr. José Joaquín Lizardi del Angel

Ciudad de México, febrero de 2017.

UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

CA Coordinación Académica

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO
COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS©

DEDICATORIA

Dedicada a:

Mis padres:

Susana López Chimalpopoca y Juan Manuel Fernández Santillán

A mis hermanos:

Susana y Juan Manuel Fernández López

A mis sobrinos:

Susyfer Gómez Fernández

Luis Hiram Gómez Fernández

Juan Jesús Fernández Salgado

Gracias a todos por su apoyo incondicional y comprensión durante este proceso, este logro es gracias a ustedes, los Amo.

Jhoana Magdalena Fernández López.

AGRADECIMIENTOS

Mi agradecimiento dirigido a la Universidad Autónoma de la Ciudad de México por haberme permitido ser parte de ella y darme la oportunidad de estudiar una carrera profesional.

Agradezco también a mi familia por alentarme a seguir mis estudios y por ser una guía en mi vida.

A mi director de trabajo recepcional, **Dr. José Joaquín Lizardi del Ángel**, por su paciencia durante la realización de este trabajo recepcional, por brindarme su apoyo para poder terminar este proceso, que no fue fácil, y al profesor, Joel Yazbek Buendía Gómez, por su colaboración en los inicios del presente trabajo.

.

A mis lectores que también fueron parte importante en la realización de este trabajo recepcional:

M en C. Magali Cortez Vázquez
M. en C. José Alfredo del Oso Acevedo
M. en I. Omar Nieto Crisóstomo
M. en I. Diana Aurora Cruz Hernández

Por tomarse el tiempo de leer el documento y brindarme sus observaciones, correcciones y sugerencias, muchas gracias.

Agradezco a la Universidad Autónoma de la Ciudad de México por su apoyo al otorgarme la beca de impresión y/o empastado de trabajo recepcional.

Jhoana Magdalena Fernández López.

Tabla de Contenido

Indice de figuras y tablas

RESUMEN

Hoy en día las empresas que ofrecen sus servicios para poder almacenar información y consultarla desde cualquier sitio, otorga a los usuarios la facilidad de acceder a sus archivos de manera segura y eficaz ya que tienen la confianza de que la información siempre estará ahí. Actualmente la computación en la nube es la que nos brinda estas características con la finalidad de ofrecer un mejor servicio a los usuarios y/o empresas.

El propósito del presente proyecto es elaborar un diagnóstico de algunos servicios de almacenamiento en la nube con el fin de analizar la problemática de seguridad, sus vulnerabilidades y emitir algunas recomendaciones. Para ello fue necesario, realizar una selección de las distintas nubes comparando sus características y los servicios que ofrecen, luego se llevaron a cabo pruebas de seguridad para conocer las vulnerabilidades y finalmente emitir algunas recomendaciones en base a los resultados obtenidos y literatura consultada.

Para ello se realizó un diagnóstico en algunos servicios de alojamiento y un análisis de vulnerabilidades de seguridad en la información almacenada y enviada sobre un servidor de Owncloud. Debido a que algunos proveedores rechazaron la petición de autorización que se les hizo llegar pidiendo permiso para poder realizar pruebas de seguridad. Además se realizó un escaneo simple de puertos sobre SurDoc (único proveedor que respondió a la solicitud de poder hacer algunas pruebas).

Dicho brevemente, este trabajo contiene un escaneo de puertos en SurDoc y una serie de pruebas de seguridad realizadas a un servidor de nube Owncloud, con el fin de ofrecer al lector un conocimiento más amplio respecto a la seguridad que se emplea en los sistemas de alojamiento en la nube y al momento de elegir alguno de los servicios existentes, se tenga el conocimiento de que las actividades o ataques maliciosos hechos por otros usuarios son posibles pero siempre hay una manera de proteger el contenido que se tiene almacenado.

Palabras clave

- MITM
- Seguridad
- Kali Linux
- Nmap
- Vulnerabilidad
- Pentesting

CAPÍTULO 1
INTRODUCCIÓN

CAPÍTULO 1

Introducción

En este capítulo se describen los antecedentes de las pruebas de seguridad a realizar y también los objetivos a los que se quiere llegar con este proyecto, tomando en cuenta la problemática de seguridad que se tiene en la actualidad con los servicios de alojamiento en la nube.

1.1 Antecedentes

Actualmente el hecho de ofrecer servicios con capacidad de almacenar información y poder acceder a ella desde cualquier sitio, otorga a los usuarios mayor facilidad para acceder a archivos de una manera segura, factible y tener la confianza de que siempre estarán ahí. Es por eso que actualmente la computación en la nube brinda estas características, la cual empezó con proveedores de servicio de internet a gran escala como Google y Amazon los cuales construyeron su propia infraestructura. La finalidad es proporcionar un mejor servicio a los usuarios y empresas. Con el uso de la tecnología de almacenamiento en la nube se tienen ventajas económico-financieras, seguridad, disponibilidad y movilidad.

En cuanto a las desventajas se tiene la dependencia de terceros, es decir, al acceder a una cuenta en la nube si el servicio no está disponible o inestable para los usuarios se detendrá el sistema de tal manera que será imposible acceder, así mismo dependencia de los proveedores de servicios de internet (ISP) y de la velocidad de cable o fibra óptica. Por otro lado, en caso de que los datos no estén cifrados estos pueden ser consultados por terceros ya sea por error o intencionalmente y por último migrar hacia otra plataforma o coordinar varias al mismo tiempo es difícil ya que no todas ofrecen esta opción.

Los ataques a estos servicios de almacenamiento han sido practicados por personas que se dedican a extraer información de cuentas importantes tanto personales como empresariales, cuyo objetivo es pasar desapercibidos, por así decirlo, en un ambiente encubierto por programas y herramientas que hacen de este robo algo más fácil. Con el paso del tiempo estas técnicas han sido perfeccionadas, hecho que se basa en diferentes sistemas de ataque.

De acuerdo con Armando Carvajal (2007) se identifican 4 tipos de ataques [1]:

1. Modificación: También llamados web-defacement buscan comprometer la confidencialidad y la integridad del sistema, por ejemplo cuando un atacante modifica la página web de una organización sin previa autorización.

2. Fabricación: Comprometen la integridad del sistema, por ejemplo al insertar un nuevo usuario en el sistema.
3. Interceptación: Buscan comprometer la confidencialidad del sistema, un ejemplo son los key loggers o spyware y los sniffers.

4. Interrupción: Comprometen la disponibilidad del servicio, un ejemplo serían los ataques de denegación de servicios (DoS).

Cuando alguno de estos ataques se lleva a cabo, la mayoría de los usuarios no se percatan de que están siendo víctimas de un robo de información, por así decirlo, ya que muchas de las herramientas o softwares existentes para realizar este tipo de ataques se ejecutan de manera desapercibida de tal forma que la víctima no pueda hacer nada para evitarlo.

1.2 Planteamiento del problema

Si consideramos que hay tantas opciones que un usuario puede tomar en cuenta para elegir un servicio de nube ya sea de paga o de carácter libre, se tienen que tomar en cuenta aspectos importantes de seguridad como la protección de contenido, ya que existen vulnerabilidades por parte del servicio que se ofrece como son: pérdidas o fuga de datos, uso incorrecto del servicio, vulnerabilidad en archivos compartidos, ataques a dispositivos móviles, ataques en las redes sociales, robo de identidad a los usuarios, entre otros [2].

En el laboratorio de redes B-207, de la UACM se llevan a cabo pruebas para identificar fallos de seguridad y ataques, por lo tanto se decidió analizar la seguridad en los servicios de almacenamiento en la nube.

En la realización de la presente investigación, se plantea la siguiente problemática a resolver ¿de qué manera se pueden detectar vulnerabilidades y hacer recomendaciones hacia el propietario del servicio?

1.3 Justificación

Actualmente la tecnología brinda servicios de almacenamiento de datos "seguros" a gran parte de las personas que tienen acceso a internet, con lo que es posible acceder a ellos de forma relativamente segura desde cualquier terminal que cuente con una conexión a internet. De esta manera se puede prescindir de sistemas de almacenamientos externos (discos duros o memorias USB) para poder llevar a todas partes los archivos. A estos servicios se les llama *Cloud Storage* que significa almacenamiento en la nube, un espacio en internet dedicado a ofrecer servicios para gestionar archivos de manera remota.

Debido a lo anterior y a la problemática de seguridad que a veces se presenta en algunos proveedores de almacenamiento en la nube, en este trabajo recepcional se decide analizar las vulnerabilidades que existen en estos sistemas y realizar una serie de ataques de interceptación con la finalidad de proponer un informe de seguridad.

1.4 Objetivo general

Elaborar un diagnóstico de algunos servicios de almacenamiento en la nube con el fin de analizar la problemática de seguridad, sus vulnerabilidades y emitir algunas recomendaciones.

1.5 Objetivos específicos

1.-Realizar una selección de las distintas nubes comparando sus características y servicios que ofrecen.
2.-Implementar un servidor que ofrezca servicios web para realizar pruebas de seguridad.
3.- Llevar a cabo pruebas de seguridad para conocer las vulnerabilidades y aumentar la seguridad de la información.
4.- Emitir algunas recomendaciones en base a los resultados obtenidos y literatura consultada.

1.6 Metodología

Para el presente trabajo recepcional se llevó a cabo la siguiente metodología:

1.-Revisión de la literatura en general acerca del tema, para tener una base y poder delimitar el tema de estudio.

2.-Búsqueda de servicios de almacenamiento en la nube con el objetivo de realizar pruebas sobre sus servicios, para elaborar un informe de seguridad.

3.- Enviar una petición a los servicios de nube seleccionados, en la cual se solicitó el permiso para realizar pruebas de seguridad con fines académicos.

4.- implementar un servidor web que cuente con seguridad propia (SSL/TLS). Dentro de este se ofrece el servicio de almacenamiento en la nube (Owncloud). Este servidor se ubica en la red interna del laboratorio B-207 y está salvaguardado por un cortafuego (implementado con Pfsense o sistema freebsd).

5.-Desarrollo de pruebas de seguridad tales como: escaneo de vulnerabilidades, intercepción de archivos de texto plano, extracción de información (usuario y contraseña) sobre el servicio de almacenamiento Owncloud.

6.- Por último se brindan algunas recomendaciones en base a los resultados obtenidos y literatura consultada.

1.7 Hipótesis de trabajo

Si realizamos pruebas de vulnerabilidades en servidores locales, obtendremos información de las debilidades del sistema, lo cual permitirá proponer las soluciones adecuadas.

CAPÍTULO 2
MARCO TEÓRICO

CAPÍTULO 2
Marco teórico

En el segundo capítulo se encuentra el marco teórico, que hace mención de distintos conceptos que se deben entender antes de seguir con la implementación de las pruebas de seguridad.

2.1 Almacenamiento en la nube

El almacenamiento en la nube es un modelo de servicio el cual especifica la forma de almacenar, administrar y respaldar datos de un sistema de cómputo, los cuales se encuentran en servidores que están en diferentes lugares, incluso geográficamente, y así puedan ser gestionados de manera remota poniéndolos a disposición de los usuarios a través de una red como lo es el internet; los usuarios pagan por este almacenamiento de datos anualmente y otros más activan cuentas gratuitas por un almacenamiento limitado al espacio que ofrezca cada uno de los proveedores [3].

Existen tres tipos de almacenamiento en la nube [4]:

1.- Público. Este tipo de almacenamiento proporciona un control administrativo mínimo debido a que el acceso es en línea por cualquier persona, el usuario no tiene ninguna visibilidad ni control sobre el lugar donde se alojan los servicios en la nube, proporciona medidas de seguridad y espacios virtuales para cada uno de los usuarios de manera que solo pueden ver lo que les corresponde. Se aloja externamente y se puede acceder mediante internet, normalmente es de bajo costo y requiere poco mantenimiento. Algunos servicios de almacenamiento de nube publica son: Box, Dropbox, Mega, SurDoc, entre otros.

2.- Privado. Este tipo de almacenamiento también se le conoce como nube interna porque su infraestructura se almacena en una plataforma privada cubriendo las necesidades de una persona o empresa. En este modelo el usuario tiene el control administrativo haciendo posible el diseño y la operación del sistema de acuerdo a sus necesidades. Algunos servicios de nube privada son: Cisco, Owncloud, BitTorrent Sync.

3.- Híbrido. Estos sistemas de almacenamiento ofrecen como su nombre lo sugiere una combinación de los almacenamientos de nubes públicas y privadas donde los usuarios pueden personalizar las funciones y aplicaciones que se adapten mejor a sus necesidades, un ejemplo de la configuración de este servicio es almacenar los datos importantes en un sistema de almacenamiento privado y los datos menos importantes se pueden almacenar en uno público, teniendo como inconveniente la compatibilidad entre plataformas.

2.2 Seguridad informática

Actualmente es imposible garantizar la seguridad en un sistema informático, ahora bien para que un sistema se considere seguro debe cumplir con los principios básicos de la seguridad de la información [5].

- Confidencialidad: En este punto se asegura que la información no pueda estar accesible o a disposición de personas o procesos no autorizados, es decir, que la información sea confidencial y que el sistema sea capaz de evitar que personas no autorizadas puedan acceder a la información almacenada en él. Algunas medidas utilizadas para proteger la confidencialidad de los datos son: el control de accesos a los sistemas y el cifrado de la información.

- Integridad: Este principio no debe modificar o corromper la información almacenada ni permitir que alguna persona que no esté autorizada lo haga, este principio de seguridad asegura que la información no sea falsificada, un ejemplo de este principio es que al momento de recibir o recuperar datos estos serán exactamente los mismos que fueron enviados o en su caso almacenados sin que hayan sido modificados, alterados o borrados.

- Disponibilidad: Los sistemas deben mantener la información disponible para los usuarios, es decir, que funcione eficientemente y que este sea capaz de recuperarse rápidamente en caso de fallo. Para el usuario es importante tener sus datos en lugar, momento y forma en que son requeridos.

2.3 Envenenamiento por ARP (Address Resolution Protocol)

Este protocolo trabaja en la capa de enlace de datos, presenta un método para la conversión de las direcciones IP en las redes de área local a las direcciones Ethernet (direcciones MAC), de esta manera los equipos arman una tabla, llamada tabla ARP, esta se almacena en la memoria del equipo y permite establecer la relación entre las direccione IP y las direccione Ethernet de los equipos. Cuando se desee conocer una dirección MAC asociada a una IP, simplemente se localiza la dirección IP en la tabla y se busca dentro del mismo registro de la tabla la dirección MAC asociada.

Ahora bien, cuando un equipo envié información a otro equipo a través de la red local, lo hace por medio de la dirección IP. Lo primero que hace el sistema es verificar si la dirección IP de destino es parte de su mismo segmento de red, si es así, el equipo consulta en su tabla ARP que dirección MAC le corresponde a la IP destino, si la dirección solicitada no se encuentra en ella entonces el protocolo ARP envía un paquete de broadcast (a todos los equipos de la red) con la dirección IP solicitada, los equipos comparan dicha IP con la suya y solo el equipo al que corresponda es el responderá enviando su dirección MAC, hecho esto todos los equipos actualizan su tabla ARP.

Cuando se realiza un envenenamiento por ARP, se pueden infectar uno o varios equipos dentro de la red local, haciendo que uno o varios equipos nos hagan llegar tráfico que está destinado a otro equipo. Se dice que se envenena la tabla ARP de un equipo por que se le envía constantemente respuestas ARP que no son solicitadas, a los equipos que se quiere atacar, con la finalidad de mantener envenenada la memoria caché. Con esto se logra asociar la MAC del atacante con la IP del equipo víctima y también asociada a la Ip del router de la red local, causando que cualquier tráfico destinado a ese equipo se dirija hacia el atacante o bien cuando un usuario envié información a través del router, este no será visto por el router sino por el atacante y viceversa.

2.4 Seguridad en el modelo ISO/OSI

El modelo *ISO/OSI* (*International Standard Organization/Open System Interconnection*) consta de siete capas las cuales tienen una función específica de red, en este punto se tratará la seguridad en cada capa, en la tabla 2.1 se observa el modelo ISO/OSI [7].

Nivel	Nombre
7.-	Aplicación
6.-	Presentación
5.-	Sesión
4.-	Transporte
3.-	Red
2.-	Enlace de datos
1.-	Física

Tabla 2.1 Modelo ISO/OSI.

1.- Capa física

Provee los medios de transporte para los bits que forman la trama de la capa de enlace de datos a través de los medios de red, ya que aquí viajan señales y dependiendo la señal es el tipo de medio, existen tres tipos básicos de medios:

- Cable de cobre, como : coaxial, par trenzado UTP/STP (patrones de pulsos eléctricos)
- Fibra Óptica (patrones de luz)
- Inalámbrico (patrones de transmisiones de radio)

Su función es detectar la señalización, codificar datos y transmitir mediante componentes físicos secuencialmente, es decir, entrega los bits en el mismo orden en que los recibe.

La seguridad en esta capa está dada por los medios de transmisión ya que los alambres de cobre conducen en ocasiones la electricidad de manera no deseada y provocar problemas al personal y el equipo usado.

Por otro lado el revestimiento y el aislamiento de los cables puede llegar a producir emisión de sustancias toxicas además de que puede llegar a ser inflamable cuando se calientan o se queman, debido a esto el personal deberá tomar medidas de seguridad para evitar este tipo de fallos.

2.- Capa de enlace de datos

Es la encargada del intercambio de tramas entre nodos a través de los medios de una red física (conformado por dos o más nodos conectados a un mismo canal de comunicación), toma un medio de transmisión en bruto y lo transforma en una línea que parezca libre de errores de transmisión de la siguiente manera: el emisor divide los datos de entrada en cientos o miles bytes (normalmente) llamados marcos de datos, transmitiéndolos en forma secuencial, y procesando de igual manera los marcos de acuse de recibo que devuelve el receptor.

Su función es la detección de errores de transmisión, así como también el control de flujo en el envió de la información para evitar la saturación, delimitar el inicio y el fin de cada trama para que el receptor pueda reconocer y procesar cada trama de forma individual
Se divide en 2 subcapas:

LLC (Logical Link Control), en español control de enlace lógico, coloca en la trama información que identifica qué protocolo de capa de red se utilizará para la trama, es decir, permite que varios protocolos de la capa 3 como IPv4 e IPv6 utilicen la misma interfaz y los mismos medios de red.

MAC (Medium Acces Control), en español control de acceso al medio, este define los procesos de acceso al medio, el cual proporciona el direccionamiento para determinar cómo se transportaran los datos a través de los medios de transmisión físicos.

La seguridad en esta capa es vulnerable a ataques, poniendo en riesgo la información que se transmite. Un ejemplo podría ser un ataque de ARP Spoofing, el cual utiliza la tabla de direcciones MAC ya que cada dirección MAC tiene asociada una dirección IP y estas relaciones son almacenadas tanto en los switches como en los dispositivos que están conectados a una red. Al realizar peticiones falsas de ARP es posible que se pueda falsificar cualquier dirección MAC en una red, redireccionando a un equipo falso toda la información lo cual permite realizar:

- Sniffing del tráfico de la red interna (incluso de un switch)
- Ataques de denegación de servicio
- Ataques de hombre en el medio (mismo que se explica más adelante)

3.- Capa de red

En este nivel se define el encaminamiento y el envío de paquetes de extremo a extremo, proporcionando las siguientes características:

Una de las primeras funciones de la capa de red es el direccionamiento, es decir, si los dispositivos deben dirigirse a un usuario final este dispositivo debe de tener una dirección única a la cual viajara. Por otro lado los dispositivos no solo deben ser identificados con la dirección si no también con el PDU Unidades de Datos de Protocolo Protocolo de Datos de Usuario, lo hace encapsulando el paquete agregándole un encabezado y etiqueta donde contiene la dirección del equipo al cual será enviado (dirección de destino), y a su vez contiene la dirección de equipo de origen, el paquete podría recorrer muchas rutas diferentes. Luego de esto la capa de red debe proveer los servicios para dirigir, es decir, encaminar el paquete para que tenga una ruta óptima y rápida para llegar al destino final, ya que los equipos no siempre están conectados en la misma red, el paquete podría recorrer diferentes redes para llegar a su destino. Finalmente se verifica la dirección de destino y ver que ha sido enviado correctamente, en caso de que la información sea correcta el paquete es desencapsulado.

La seguridad en esta capa está a cargo del personal, ya que es complicado realizar algún ataque remoto, la seguridad sería violada sólo en caso de que una persona tuviera acceso ya sea al equipo físicamente o al cuarto de telecomunicaciones, provocando fallos en el cableado o interceptando la comunicación entre los equipos, debido a esto se deben de fomentar restricciones al personal respecto al resguardo y acceso a dicha infraestructura.

4.- Capa de transporte

Acepta datos de la capa se sesión, si es necesario los divide en unidades más pequeñas para pasarlos a la capa de red asegurando que todos los pedazos lleguen correctamente al otro extremo. Este nivel garantiza una entrega confiable, segmentando la información ya que evalúa el tamaño de los paquetes con el fin de que éstos contengan el tamaño requerido por las demás capas. Otro aspecto es que los datos no sólo deben entregarse sin errores sino también en la secuencia que proceda. En esta capa se soportan múltiples

conexiones, determina el protocolo o tipo de servicio que garantiza el envío del mensaje y asigna una dirección única de transporte a cada usuario, en esta capa funcionan los protocolos UDP Protocolo de Datagramas de Usuario(no orientado a la conexión, es decir, no tiene control de flujo, ni se sabe si los paquetes fueron recibidos correctamente, tampoco hay confirmación de ello) y TCP (orientado a la conexión, es decir, garantiza ser entregar datos al destino de manera correcta y en el mismo orden que fueron transmitidos).

Los protocolos SSL y TLS son los encargados de la seguridad en esta capa porque son protocolos criptográficos que proporcionan comunicaciones seguras, protegiendo y autenticando la comunicación en una red pública. De tal forma que si no se tiene esta seguridad en los sistemas, serían vulnerables a ataques informáticos como denegación de servicio.

5.- Capa de sesión

Este nivel permite a usuarios de diferentes máquinas establecer una sesión, misma que puede ser usada para efectuar un inicio de sesión en un sistema remotamente o bien para transferir un archivo entre dos máquinas. Mediante esta capa se controla el dialogo, es decir, se establece el orden de los mensajes, quién habla, cuándo y cuánto tiempo, estableciendo el inicio y termino de la sesión al mismo tiempo en caso de fallo tiene la capacidad de recuperar la sesión.

- Uno de los problemas de seguridad que hay en este nivel es el robo de sesiones, es decir, la obtención de información de un usuario para poder identificarse en alguna página web, también se le conoce como credenciales de identificación para una sesión iniciada en páginas como Facebook, Hotmail, aplicaciones web, entre otros.

6.-Capa de Presentación

Es la encargada de manejar las estructuras de datos indefinidas, es decir, se encarga de la sintaxis y la semántica de la información que se transmite y realizar las conversiones de presentación de datos necesarios para la correcta interpretación de los mimos, permitiendo también cifrar los datos y comprimirlos, en pocas palabras se diría que es un traductor.

Sus principales funciones son:

- Formateo de datos: se encarga de la representación de la información traduciendo el formato, si distintos equipos tienen diferentes representaciones internas como: ASCII, EBCDIC, sonidos o imágenes se encarga de que los datos lleguen de manera reconocible.
- Cifrado de datos: es un proceso mediante el cual la información legible s transformada mediante un algoritmo en información legible por así decirlo y puede ser enviada de forma secreta con menos riesgos a que una tercer persona pueda leerlo, es decir, protege la información durante la transmisión.
- Compresión de datos: usa algoritmos para reducir el tamaño de los archivos remplazándolo con un token (patrón de bit mucho más corto, que representa al patrón largo).
- Aplicar a los datos procesos criptográficos
- Definir la estructura de datos a transmitir

La seguridad en este nivel está dada por las claves de cifrado con las cuales la información viaja encriptada y segura por la red sin temer a perder o filtrar información y al llegar a su destino estos mensajes puedan ser descifrados.

7.-Capa de aplicación

En esta capa el usuario normalmente no interactúa directamente con el nivel de aplicación, suele interactuar con programas que a su vez interactúan en este nivel, se basa en las funciones de las capas inferiores para completar los procesos de comunicación proporcionando servicios a los usuarios, suministra las herramientas que el usuario visualiza mediante las aplicaciones y es encargado del acceso general a la red, gestionando los mensajes, la transferencia de archivos y el acceso a las bases de datos, esta capa suministra cada uno de estos servicios a los distintos programas de aplicación con los que cuenta el usuario en su computadora. Entre ellos se encuentra los servicios de correo electrónico SMTP (Protocolo Simple de Transferencia de correo) y aplicaciones de base de datos (cliente/servidor).

Debido a que uno de los protocolos que trabajan en esta capa es el HTTP se recomienda iniciar sesión en páginas web que tengan seguridad empleando en la URL el protocolo HTTPS para que al momento de autenticar en alguna página web los datos personales que se ingresan no queden desprotegidos.

2.5 Pruebas de seguridad

Empresas y personas interesadas en la seguridad del contenido de sus equipos de cómputo, paginas o portales web, y servicios de almacenamiento en la nube, llevan a cabo auditorías internas para detectar vulnerabilidades en sus sistemas y poder corregirlos antes de que un usuario malicioso por así decirlo aproveche esas debilidades e intente robar información y perjudicar de manera permanente a dichas usuarios, algunos de los ataques comúnmente utilizados para adquirir información de manera ilícita se describen a continuación.

Ataque MITM *(Man In The Middle*), que traducido al español sería "hombre en el medio", este tipo de ataque se realiza para adquirir información personal del usuario ya que está presente en medio de la transmisión del origen y el destino, donde el atacante puede observar, modificar, interceptar y capturar todo el tráfico que pasa por él, un ejemplo es el nombre de usuario y contraseña con la que se inicia sesión en cuentas como facebook, correo electrónico, almacenamiento en la nube, páginas personales, entre otros.

La descripción del ataque se puede observar en la figura 2.2. Supongamos que tenemos 3 elementos dentro de una red: victima, servidor y atacante; donde la victima quiere intercambiar información con el servidor, ahora, si el atacante tiene intención de "escuchar" el mensaje que la víctima envía al servidor, sólo tiene que adoptar una apariencia falsa donde se hará pasar por el servidor para así poder recibir los mensajes y poder alterar o no toda la información antes de enviarla al servidor correcto o verdadero [8].

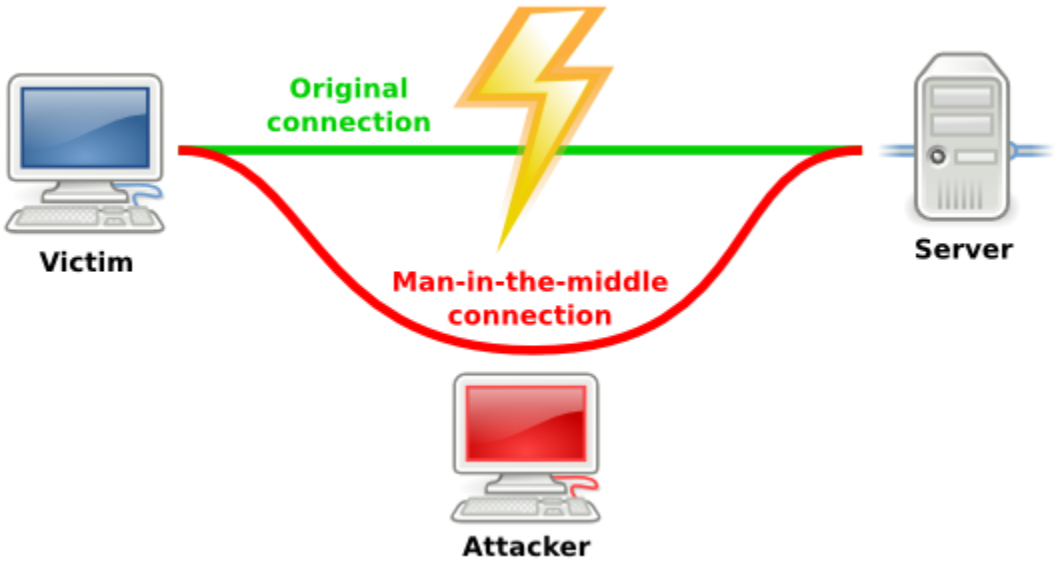


Figura 2.1. Descripción del ataque MITM.

Sniffing network, traducido al español, olfateando la red, consiste en analizar la red y los paquetes que viajan en ella para poder interceptar los datos y paquetes de los usuarios en caso de que accedan a una página en internet e inicien sesión, o bien compartan archivos o imágenes que contengan información privada e importante para el usuario.

Hay varias formas de analizar una red dependiendo el uso y de lo que se quiera obtener, en el caso del programa Wireshark tiene la posibilidad de filtrar paquetes para facilitar la búsqueda de información, es decir, es un Sniffer[1].

Wireshark es un analizador de paquetes de red utilizado para identificar y analizar qué tipo de tráfico está siendo transmitido, el cual tratará de capturar paquetes de red mostrándolos a detalle tanto como sea posible, esto para realizar un análisis para detectar y corregir errores de red y examinar problemas de seguridad. Contiene una amplia gama de filtros que facilitan la búsqueda de protocolos, actualmente soporta aproximadamente hasta 1100. Debido a que wireshark "entiende" la estructura de estos protocolos se puede visualizar cada una de las cabeceras que componen cada paquete analizado [9,10].

[1] El sniffer es un software que permite capturar tramas de la red. Generalmente utilizado con fines maliciosos para capturar textos de emails, chats, datos personales, contraseñas.

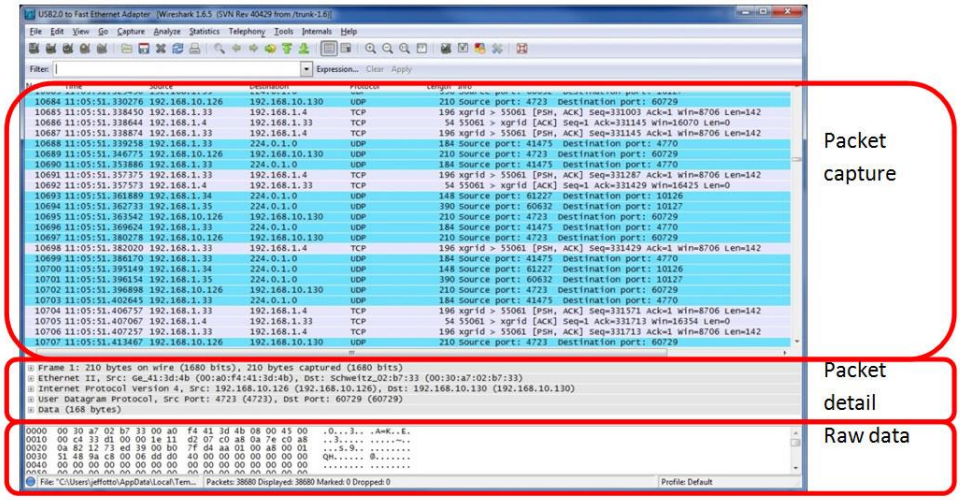


Figura 2.2. Wireshark, analizador de redes.

En la figura 2.3 se puede observar la interfaz de wireshark donde se visualizan los paquetes capturados seguidos del detalle del paquete y finalmente lo que llamaríamos la información en bruto, este último campo nos muestra la información en hexadecimal.

SQL Injection: La inyección SQL brinda la posibilidad de saber y analizar qué es lo que contiene un portal web como lo son sus tablas de contenido, usuarios registrados y contraseñas; es decir, SQL Injection es un ataque que permite a un atacante realizar consultas a una base de datos modificarlas e incluso borrarlas o dejarlas inservibles. Un incorrecto filtrado de la información que se pasa a través de los campos y/o variables que usa un sitio web, es por lo general usado para extraer credenciales y realizar accesos ilegítimos. Un fallo de este tipo puede llegar a permitir ejecución de comandos SQL en el servidor, subida y lectura de archivos, o peor aún, la alteración total de los datos almacenados.

En la figura 2.4, se muestra un ejemplo del ataque SQL injection, se observan dos usuarios uno de confianza y el otro es un usuario malicioso, el primero logra acceder al sistema por medio de credenciales validadas y el segundo realiza una combinación de sentencias SQL logrando burlar el formulario de autentificación, accediendo al sistema para realizar extracción de datos. [11].

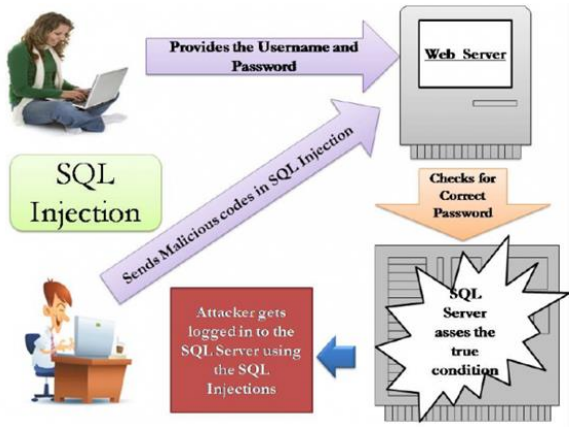


Figura 2.3. Descripción del ataque SQL Injection.

2.6 Elementos de conectividad

Switch

El switch interconecta 2 o más segmentos de red, enviando información de acuerdo con la dirección de control de acceso al medio (MAC), el switch trabaja en la capa dos del modelo OSI, hace posible que los usuarios de una red, envíen información al mismo tiempo sin que se disminuya la velocidad de transmisión, de este modo la información viaja desde el puerto de origen al puerto destino [12].

Router

Es un dispositivo que opera en la capa tres del modelo OSI, que interconecta redes, para enrutar paquetes hacia otra red, utilizando el camino más óptimo para llegar a su destino, su función es saber si el destinatario de un paquete está en la propia red o en una remota, esta decisión la toma basándose en la dirección IP del paquete [12].

Firewall

También conocido como cortafuegos, permite o niega las comunicaciones de una red a otra, dependiendo de las reglas que se le hayan configurado para que el tráfico pueda acceder o salir de nuestra red, en caso contrario el trafico entrante o saliente será bloqueado, normalmente se sitúa entre la red local y la red de internet como un dispositivo de seguridad, evitando que un usuario malicioso pueda acceder a la información confidencial, en resumen, filtra en su totalidad el trafico entrante y saliente que hay entre dos redes [12].

Servidor web

Su principal función es almacenar los archivos de un sitio y enviarlos por internet para que puedan ser consultados por los usuarios, es decir, guarda y transmite. Espera peticiones del cliente, que se encarga de contestarlas de forma adecuada, obteniendo como resultado una página web o información de acuerdo a los comandos solicitados. Integra seguridad para permitir una conexión encriptada entre el servidor y el navegador para que toda la información confidencial viaje segura por la red [12].

Algunos de los servidores web más utilizados son: Apache, Microsoft IIS, Ngnix, Lighttp. Apache es uno de los más usados debido a su estabilidad y confiabilidad y es personalizable, perteneciendo a la capa de aplicación del modelo OSI.

En resumen, cuando se introduce una URL valida en el navegador, la petición de conexión se envía al servidor web (por ejemplo apache) para administrar la petición, de modo que el servidor web apache retorna la página inicial del dominio correspondiente.

2.7 Certificados de seguridad SSL

SSL Secure Socket Layer (capa de conexión segura), brinda seguridad a una página web, lo que permite saber que el sitio es auténtico, real y confiable en caso de tener que ingresar datos personales para hacer compras en línea o iniciar sesión. Este protocolo de seguridad hace que sus datos viajen de manera íntegra y segura mediante conexiones cliente-servidor[2], en nuestro caso van totalmente cifrados con **RSA -2048 bits**, que es un algoritmo de llave pública desarrollado por Ronald Rivest, Adi Shamir y Leonard Adelman en 1977. Los mensajes enviados utilizando este algoritmo se representan mediante números (producto de 2 números primos) elegidos al azar y emplea funciones exponenciales.

Generalmente es el más utilizado ya que la seguridad de este algoritmo[3] radica en que no hay maneras rápidas conocidas de factorizar un número grande en sus factores primos utilizando computadoras tradicionales [13].

[2] Un servidor, es una aplicación que ofrece un servicio a usuarios de internet, el cliente, es el que pide ese servicio. Una aplicación consta de una parte de servidor y otra de cliente y se pueden ejecutar en el mismo o en diferentes sistemas.

[3] Conjunto ordenado y finito de operaciones, que permite hallar la solución a un problema matemático, informático o disciplinas afines.

2.8 Seguridad mediante protocolo TLS

TLS Transport Layer Security, (seguridad de la capa de transporte), este protocolo de seguridad que actualmente sustituye al SSL. El protocolo TLS, establece también conexión segura entre cliente y servidor por medio de un canal cifrado, se ejecuta en la capa de aplicación junto a protocolos como HTTP y HTTPS, ofreciendo seguridad a páginas web.
muestra el protocolo handshake TLS, es decir, permite que ambas partes se autentiquen mutuamente para negociar el cifrado, antes de intercambiar datos y posteriormente establecer la conexión, a continuación se describe el proceso establecido en la intercepción de datos con este protocolo [14]:

Client Hello: el cliente envía un mensaje al servidor, para dar inicio a la conexión cifrada, enviando una cadena de caracteres aleatorios que sirven para generar la clave simétrica[4].
Server Hello: el servidor contesta al cliente escogiendo un cifrado, enviando un ID de sesión correspondiente, como la versión de TLS correspondiente, el tipo de compresión de datos.
Certificate: el servidor ofrece su certificado al ser verificado por el cliente.
Server Key Exchangue: con este mensaje el servidor ofrece un cifrado asimétrico entre el cliente y servidor, la clave pública[5] con la clave del certificado.
Server Hello Done: el servidor da por concluida su fase de negociación asimétrica.
Client Key Exchangue: ya comprobado y validado el certificado, cifra con la clave pública del servidor y se lo envía, después el cliente y el servidor pueden generar la clave secreta usando el cifrado simétrico.
Changue Cipher Spec: con este mensaje el cliente informa que sus mensajes sucesivos estarán cifrados con el cifrado simétrico acordado.
Finished: el cliente da por finalizada su fase de negociación asimétrica, el mensaje garantiza la integridad de la comunicación.
Changue Cipher Spec: el servidor descifra con su clave privada[6], el mensaje enviado por el cliente, de manera que establece un cifrado asimétrico[7].

[4] Solo utiliza una clave para cifrar y descifrar el mensaje, que puede ser un número, una palabra, o una cadena de letras, que tiene que conocer el emisor y receptor previamente.

[5] Una clave pública, se puede entregar a cualquier persona, solo sirve para cifrar datos.

[6] Puede descifrar o hacer las dos cosas, es como una contraseña, solo quien la conozca podrá descifrar el mensaje, es una clave que no se comparte con nadie.

Finished: el servidor finaliza su fase de negociación asimétrica, con este protocolo se logra un canal TLS, que garantiza que todos los datos enviados serán cifrados.

[7] El cifrado asimétrico, hay en juego dos claves, la pública (disponible gratuitamente para cualquier persona que desee enviar un mensaje), la privada, (se mantiene en secreto para que solo una única persona la conozca).

CAPÍTULO 3

DESARROLLO

CAPÍTULO 3
Desarrollo

En el presente capítulo se llevaron a cabo pruebas de seguridad en un servicio de nube local, donde previamente se instaló el servicio de nube Owncloud, se detalló el proceso empleado para la realización de las pruebas y los análisis de vulnerabilidades que se efectuaron en los servicios de nube SurDoc.

3.1 Búsqueda de proveedores de almacenamiento en la nube

Para la elección de los proveedores se realizó una búsqueda de algunas empresas de renombre que ofrecen sus servicios para almacenamiento de datos, en la cual se tomaron en cuenta las características que diferencian unas nubes de otras como lo son capacidad de almacenamiento, seguridad de la información, encriptación de datos, autenticación, costos y disponibilidad. A pesar de que hay numerosos servicios en la nube disponibles solo se seleccionaron 10 de ellos por su capacidad de almacenamiento, son de uso libre y por su máxima transferencia de archivos, es decir, la capacidad máxima permitida para subir un archivo de una sola vez, en la tabla 3.1 se muestran las características tomadas en cuenta para su selección.

Nombre	Capacidad de Almacenamiento	Max. Transferencia de Archivo
SurDoc	100 GB	5 GB
Mega	50 GB	ilimitada
4 Shared	15 GB	2048 MB
Google Drive	15 GB	ilimitada
One Drive	15 GB	10 GB
Box	10 GB	250 MB
Bitrix 24	5 GB	5 GB
Idrive	5 GB	500 MB
Dropbox	2 GB	2 GB
Fiabee	1 GB	250 MB

Tabla 3.1. Descripción de cuentas en la nube.

Cabe destacar que se envió una solicitud por escrito a cada uno de los proveedores, requiriendo una autorización para poder realizar pruebas de seguridad en sus servidores de almacenamiento en la nube con fines académicos, de manera que al hacer las pruebas pertinentes no se tuvieran problemas, pero todos los proveedores a excepción de SurDoc notificaron que no era permitido el uso de sus servicios para pruebas, debido a que se ponía en riesgo la seguridad en las cuentas de los usuarios registrados. Debido a la respuesta negativa de algunos proveedores de nube se creó un servidor que ofreciera servicios web para poder realizar en él las pruebas de seguridad, en particular con los servicios de Owncloud.

3.2 Implementación de un servidor web

Se utilizó una PC como servidor la cual tiene las siguientes características: Lenovo (ThinkCentre), M83 i3, memoria RAM de 8 GB, Disco duro de 1 Tera y sistema operativo Centos 7 minimal. En el Anexo A, se describe el procedimiento completo para la implementación del servidor web [15].

Al terminar la instalación de todos los servicios web se procedió a mover el servidor dentro de una red protegida por un firewall **PfSense**. El objetivo principal de este firewall es proveer un software que brinde la posibilidad de tener una red segura y fácil de configurar a través de un interfaz amigable con el usuario y que se pudiera instalar en cualquier PC. Se trata de una solución muy completa, que esta licenciada bajo BSD lo que significa que es de libre distribución. Este firewall está ubicado en el **laboratorio B-207**, plantel SLT de la Universidad Autónoma de la Ciudad de México, donde se realizaron las pruebas de seguridad.

El servidor se estableció en la red **DMZ** (zona desmilitarizada), es decir que no es una zona segura, que es usada habitualmente para ubicar servidores que es necesario que sean accedidos desde fuera, por ejemplo: servidores de e-mail y servidores web, las políticas de seguridad aplicadas a la DMZ son las siguientes:

Tráfico origen	Destino	Autorizada	Prohibida	Rechazada
De la red externa	hacia la DMZ	✓		
De la red externa	hacia la red interna		✓	
De la red interna	hacia la DMZ	✓		
De la red interna	hacia la red externa	✓		
De la DMZ	hacia la red interna		✓	
De la DMZ	hacia la red externa			✓

Tabla 3.2. Política de seguridad aplicada en la DMZ.

Esta red se configuró por medio de la interface del firewall pfSense para redireccionar la dirección IP del servidor y así acceder a los servicios web externamente por medio de la dirección IP 172.17.120.129 de la red WAN y la IP interna del servidor 192.168.2.4, ahora ya en la DMZ, también se tiene la red interna donde se encuentran los equipos para las pruebas de seguridad como se muestra en la figura 3.1

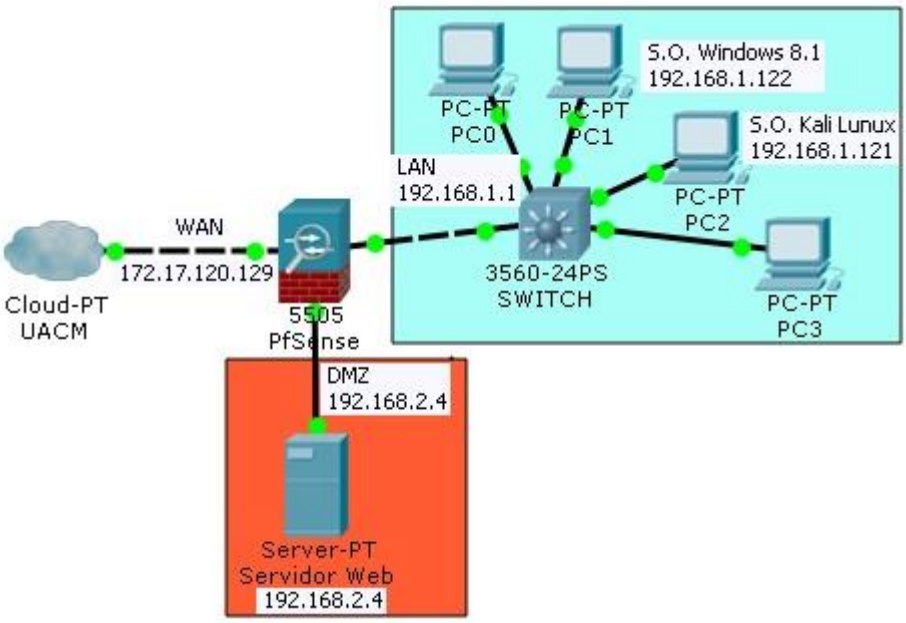


Figura 3.1. Configuración de la red del Laboratorio B-207.

3.3 Instalación de Kali Linux

Para la realización de las pruebas de seguridad, se instaló la distribución Kali Linux 2.0, es una distribución basada en Back Track que es la versión antigua a Kali Linux, este sistema operativo fue instalado en una PC con las siguientes características: DELL (Optiplex 960), windows Vista, intel core 2, memoria RAM de 2 GB y disco duro 40 GB.

Este sistema es apropiado para realizar las pruebas, porque está hecho para pruebas de penetración (pentesting) y auditorias de seguridad con soporte multi-lenguaje para entenderlo mejor, con más de 300 herramientas, se adquiere de manera gratuita y cuenta con un entorno de desarrollo seguro y software de código abierto ya que está disponible para cualquier usuario, en la figura 3.2 se observa el entorno de Kali linux [16,17].

Figura 3.2. Sistema Operativo Kali Linux 2.0.

3.4 Diagnóstico de vulnerabilidades con NMAP

En el primer escaneo se utilizó la herramienta NMAP predefinida en el sistema operativo Kali Linux [18] que se utilizó para detectar vulnerabilidades en los dos servidores de nube SurDoc y Owncloud. Esto se realizó por medio de algunos comandos de uso exclusivo de esta herramienta mismos que se definen en la tabla 3.2.

Comando	Descripción	Ejemplo
nmap IP	Da información sobre los puertos que están abiertos.	nmap 192.168.2.4
nmap –sS IP	Permite no dejar registros en el sistema sobre el escaneo.	nmap –sS 192.168.2.4
nmap dominio	En caso de no tener la IP se realiza por medio del dominio.	nmap Owncloud.com
nmap PN IP	Para ver si una red está protegida por un firewall.	nmap PN 192.168.2.4
nmap –f -script vuln IP	Permite conocer si presenta alguna vulnerabilidad.	nmap –f -scipt vul 192168.2.4
nmap –v IP	Brinda información completa sobre el escaneo.	nmap –v192.168.2.4

Tabla 3.3. Comandos usados para el escaneo de vulnerabilidades.

Las posibles respuestas que nos da un escaneo con NMAP, respecto a los puertos son las siguientes:

- **Abierto (open),** quiere decir que la aplicación en la máquina destino se encuentra esperando conexiones o paquetes en ese puerto.
- **Cerrado (closed)**, el puerto es accesible puesto que podría abrirse en cualquier momento, pero no existe ninguna aplicación escuchándolo.
- **Filtrado (filtered)**, el paquete que se ha enviado ha sido filtrado por un firewall o reglas del router, en este caso nmap no puede determinar si está abierto o no.
- **Sin filtrar (unfiltered)**, quiere decir que el puerto es accesible, pero nmap no es capaz de determinar si está abierto o cerrado.

3.5 Diagnóstico de vulnerabilidades con NESSUS

Nessus es una herramienta de uso libre que provee de usuario y contraseña, fue instalada en el sistema Kali Linux para el segundo escaneo de vulnerabilidades, tiene una interface sencilla y fácil de manejar por lo que hace más fácil el análisis de la información ya que los resultados que muestra están dados por porcentajes y colores dependiendo la vulnerabilidad. Por ejemplo: baja-verde, media-amarillo, alta-roja e información-azul. En cada una de estas etapas muestran el nombre y descripción de la vulnerabilidad encontrada y una posible solución de la misma [19].

A continuación, se muestra la serie de pasos que se siguieron para realizar los escaneos:

1.- Abrir el programa Nessus en Kali Linux
2.- Iniciar el servicio a través de la terminal con el comando ***/etc/init.d/nessus start***
3.- Se procede a iniciar sesión en Nessus con usuario y contraseña, figura 3.3.

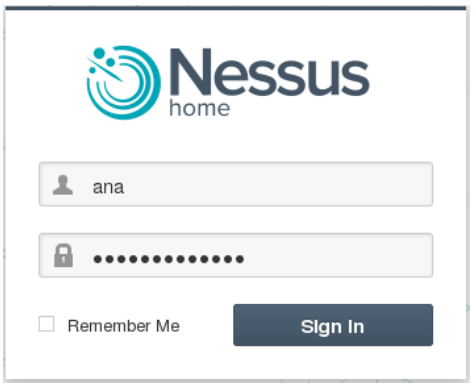


Figura 3.3. Inicio de sesión del programa NESSUS.

4.- Se podrá observar un menú con distintas opciones, del cual se elegirá “Basic Network Scan”
5.- Direccionará a una nueva página donde se selecciona “New Scan”.
6.-Se observará una nueva ventana que cuenta con un formulario, en el cual se debe colocar. Nombre, descripción, ruta para guardar, dirección IP que se va a escanear.
10.- Al terminar de llenar los campos del formulario se selecciona “Save”.
11.- Para poder realizar el escaneo de la red se selecciona “play”
12.- El despliegue de resultados se obtiene seleccionando el escaneo realizado.

3.6 Prueba MITM a Owncloud

La prueba MITM (Man in the Middle) se realizó con dos PC, las cuales están conectadas a una red LAN en el laboratorio B-207 de la Universidad Autónoma de la Ciudad de México, plantel San Lorenzo Tezonco.

Se realizó el ataque de la siguiente manera: la PC1 que opera con Windows 8.1, dispone de varios navegadores para conectarse al servidor web, pero en este caso se utilizó Google Chrome para acceder a dicho servidor. La PC2 con el sistema operativo Kali Linux que va a realizar el ataque MITM. El servidor web que se encuentra en la red DMZ cuenta con el servicio de almacenamiento en la nube de Owncloud es al cual se accederá remotamente para poder extraer información de autenticación.

La PC1 (víctima) se comunicará con el servidor web mediante el router mandando su información personal al momento de llenar el formulario de autenticación para iniciar sesión en Owncloud, esta comunicación será interceptada por la PC2 (atacante), el cual "escuchara" el canal de comunicación por medio del ataque MITM.

Cabe mencionar que la conexión entre la PC1 (víctima), la PC2 (atacante) y el router se realizó a través de la red local. En la figura 3.4 se muestra la conexión utilizada.

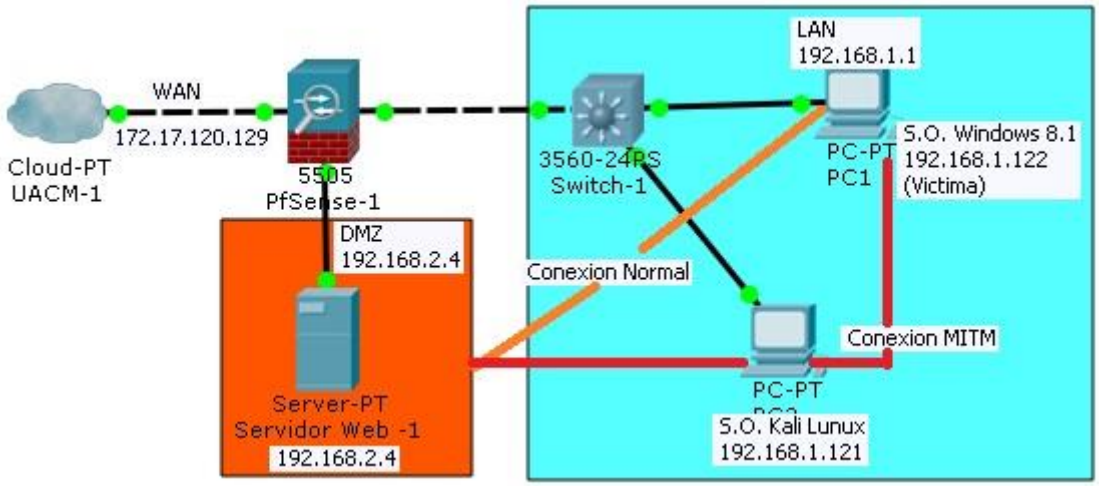


Figura 3.4. Conexión utilizada para prueba MITM.

En este ataque el objetivo es que cuando la PC1 de la red local necesite intercambiar información con el servidor, que puede estar o no en la misma red (en este caso en la red DMZ), esta conexión se verá alterada por la PC2 que es el atacante en Kali Linux que podrá leer, insertar y modificar los paquetes transmitidos entre la PC1 y el servidor, interceptando toda la información entre éstos dos equipos sin que estos se percaten de una conexión alterada.

Para la realización de esta prueba se necesita una serie de instrucciones que utiliza el atacante que se describen y explican a continuación.

Echo: comando para activar el reenvió de paquetes.

Iptables: esta herramienta redirige el tráfico ya sea del puerto 80 (http) o el puerto 443 (https); por ejemplo al puerto 8080 [10].

Envenenamiento ARP (Address Resolution Protocol): en español significa protocolo de resolución de direcciones y trabaja en capa dos del modelo ISO/OSI. Asocian una dirección IP a una dirección física MAC. Su objetivo en esta prueba es el de enviar un paquete ARP falso a la PC1 y al servidor donde la dirección MAC del otro equipo ha sido modificada y reemplazada por la dirección del atacante que en este caso es la PC2 y así cada vez que los equipos intenten comunicarse, los paquetes serán enviados al atacante.

Sslstrip: también juega un papel importante en esta prueba ya que es capaz de descifrar el tráfico y enlaces http o https de forma transparente en una red engañando al servidor, convirtiendo todo el https de una red en http donde la víctima y el atacante se comunican por medio de http, mientras que el atacante y el servidor con https con el certificado del servidor, de modo que el atacante es capaz de ver todo el tráfico en texto plano de la víctima ya que no está cifrado.

Ettercap: es una herramienta que permite analizar el tráfico que pasa por una red colocando la tarjeta en modo promiscuo para ver las conexiones y el tráfico entre dos computadoras, y poder ver qué información intercambian ya que estos datos viajaran sin encriptación [20,8].

3.6.1 Explicación del ataque

En primer lugar se debe conocer la IP de la PC1 (víctima), para esto se utilizó la herramienta NMAP haciendo un escaneo de toda la red local en busca de host activos con el siguiente comando ejecutado en la terminal:

*nmap –T4 –A –v 192.168.1.**

Este desplegó información sobre la IP de la PC1, es decir, 192.168.1.122, por lo tanto se pudo comprobar que pertenece a dicho equipo.
Ahora bien, se re-direccionó el tráfico de la PC2 (atacante) hacia la víctima activándolo de la siguiente manera en la terminal:

echo 1 > /proc/sys/net/ipv4/ip_forward

Una vez activada, se configura iptables para poder filtrar los paquetes y redirigir todo el tráfico del puerto 80, al puerto de escucha que por defecto es 8080. En la terminal se ejecuta el siguiente comando:

iptables –t nat –A PREROUTING –p tcp –destination-port 80 –j REDIRECT –to-port 8080

Teniendo todo el tráfico redirigido hay que realizar el envenenamiento ARP a la PC1 (víctima) utilizando su dirección IP junto con la dirección IP del gateway ejecutando el siguiente comando en la terminal:

arpspoof –i eth1 –t 192.168.1.122(victima) 192.168.1.254(gateway)

En caso de no saber cuál es la IP del gateway se ejecuta el siguiente comando:

netstat –nr

Después de esperar unos segundos se efectuó el envenenamiento y fue necesario abrir una nueva terminal para ejecutar el comando:

sslstrip –l 8080

Para comenzar la conversión de https a http. En otra nueva terminal se ejecuta el comando:

ettercap –T –q –i eth1

Ahora bien, dirigirse al navegador por medio de la PC1 (víctima) accediendo al primer sitio web que es el servidor de nube con dirección https://192.168.2.4/owncloud, mediante el uso de una autenticación de usuario y contraseña.

Por otro lado, en la PC2 (atacante) podremos ver que en la terminal de sslstrip nos informa sobre las peticiones que se están realizando durante la conexión y finalmente en la terminal de ettercap es donde visualizamos la información en texto plano, en este caso se obtuvieron los usuarios y contraseñas de Hotmail para la cuenta de Surdoc y de nuestro servidor Owncloud. En el capítulo 4 se muestran los resultados obtenidos de este ataque.

En la tabla 3.4 se recopilan los pasos para la realización del ataque MITM.

Ataque MITM	*1*	nmap –T4 –A –v 192.168.1.*	Terminal 1
	2	*echo 1 > /proc/sys/net/ipv4/ip_forward*	
	3	*iptables –t nat –A PREROUTING –p tcp –destination-port 80 –j REDIRECT –to-port 8080*	
	4	*netstat –nr (para obtener información sobre el Gateway)*	
	5	*arpspoof –i eth1 –t 192.168.1.122(victima) 192.168.1.254(router)*	
	6	*sslstrip –l 8080*	Terminal 2
	7	*ettercap –T –q –i eth1*	Terminal 3

Tabla 3.4. Comandos para realizar ataque MITM.

3.7 Prueba de seguridad y análisis con Wireshark realizado en el servicio Owncloud

Se realizó la prueba para verificar la seguridad en la transferencia de datos de una terminal al servidor.

Las configuraciones, elementos y herramientas para esta prueba son:

- Configuración espejo en el switch Summit x440-24p del laboratorio B-207
- PC1 (víctima) Windows 8.1
- PC2 (sniffer) Kali Linux 2.0
- Wireshark

Para poder interceptar los datos enviados hacia los servidores de nube, fue necesario hacer una configuración en el switch extreme networks summit x440-24p denominada puerto mirroring traducido al español "puerto espejo" [21]. Lo que esta configuración hace, es básicamente un espejo del puerto para que todo el tráfico que está pasando por un puerto se comunique como si estuviera pasando en otro puerto también.

En esta prueba la PC1 (víctima), está conectada al puerto 5 del switch y la PC2 (sniffer), está conectada al puerto 6 del switch, básicamente la instrucción que se le dará al switch es: selecciona el puerto 6, como el puerto espejo y envía todo el tráfico que entre o salga del puerto 5, al puerto espejo, es decir, todo el tráfico que pase por la PC1 lo vera exactamente igual la PC2. Cabe mencionar que se pueden seleccionar hasta 8 puertos espejo.

Para realizar la configuración se necesita hacer una comunicación serial con el switch, en este caso se hizo mediante una Hyper Terminal[8], que por lo general se encuentra en Windows XP, el proceso de configuración es el siguiente:

1.- Se configura la conexión con los siguientes parámetros:

- ✓ Bits por segundo: 9600
- ✓ Bits de datos: 8
- ✓ Paridad: ninguno
- ✓ Bits de parada:1
- ✓ Control de flujo: ninguno

2.-En la parte inferior de la pantalla indicará que ya está conectado y se puede acceder al switch con un login y password.

3.-Para activar el puerto mirroring, se ejecutan los siguientes comandos

switchb207# enable mirroring to port 6

Habilitar el reflejo del puerto 6 como salida.

switchb207#config mirroring add port 5

Agregar el puerto 5 a la configuración espejo.

[8] La Hyper Terminal, se utiliza para hacer conexiones telnet por medio de los puertos serie, por ejemplo: COM1, con dispositivos externos.

En la figura 3.5 se muestra la configuración empleada en el switch.

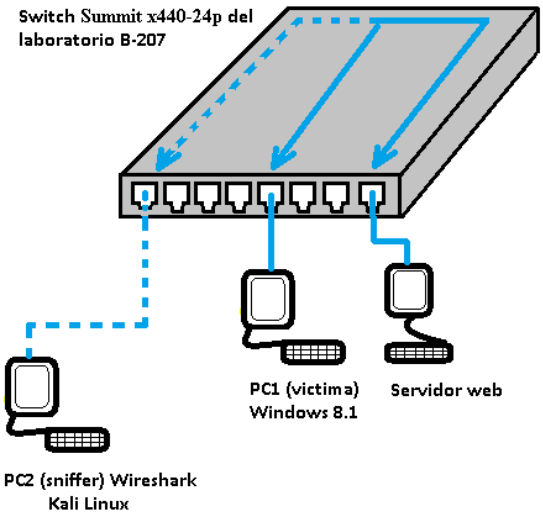


Figura 3.5. Configuración del puerto mirroring.

La herramienta utilizada para interceptar los datos fue **Wireshark,** su objetivo principal es analizar tráfico, así mismo tiene una gama de filtros que facilita los criterios de búsqueda donde actualmente se soportan aproximadamente 1100 protocolos, la interface que maneja es sencilla y fácil de manejar ya que permite desglosar cada uno de los paquetes capturados.

Los datos que viajan durante la conexión realizada son de protocolo *Hypertext Transfer Protocol* (http), traducido al español como protocolo de transferencia de hipertexto, que es un protocolo estándar para la web (cliente-servidor), lo cual significa que hace intercambios de información entre los clientes web y los servidores http.
Para interceptar los datos durante la trasferencia al servidor de nube Owncloud se realizó un análisis con archivos como: imagen JPEG, texto plano, PDF y autenticación ya que la mayoría de estos archivos deberían viajar encriptados hacia su destino pero en esta prueba se comprobó que con un sniffer básico, en este caso wireshark se pueden interceptar los datos [22,10].

En primer lugar se inició la herramienta wireshark desde la PC2 (sniffer) mostrada en la figura 3.6, la interface con la que se va a trabajar en este caso con la eth1 con dirección IP 192.168.1.121 y al seleccionar start queda guardada la dirección como se muestra en la figura 3.7.

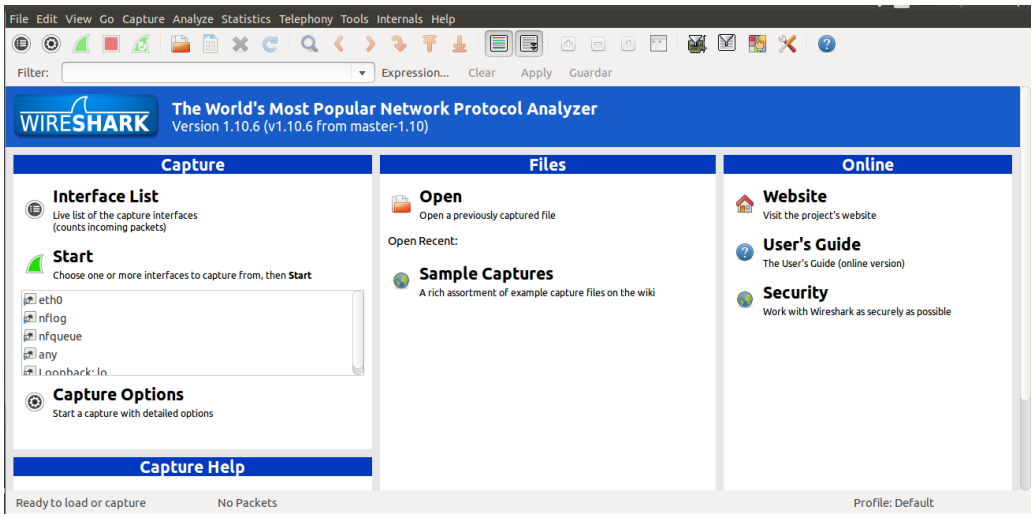


Figura 3.6. Interface inicial de Wireshark.

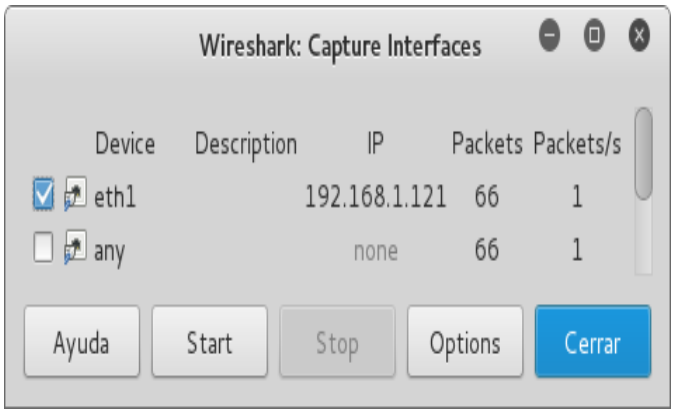


Figura 3.7. Elección de la interface.

A continuación, por medio de la PC1 (víctima) se accedió al servidor de nube Owncloud donde se cargaron dos archivos en texto plano con el siguiente contenido:

- Archivo 1: prueba.txt.
  Contenido: *hola mundo*
- Archivo 2: cuento.txt.
  Contenido: *Al atardecer, llegaron los dueños de la casa. Eran siete enanitos que trabajaban en unas minas. Se quedaron admirados al descubrir a Blancanieves. Ella les contó toda su triste historia y los enanitos la abrazaron y suplicaron a la niña que se quedase con ellos. Blancanieves aceptó y se quedó a vivir con ellos. Eran felices*.[9]

[9] Fragmento del cuento infantil Blancanieves tomado de Cuentos infantiles para leer online www.muchoscuentos.jimdo.com.

Los archivos en la interface de wireshark se interceptan capturando paquetes de protocolos, de acuerdo a su transmisión, que van desde los protocolos ARP, ICMP, TCP, EDP, SSDP, entre otros, hasta llegar a los protocolos http que son los que nos interesan ya que la transferencia de archivos en este caso se hizo cliente-servidor y es el protocolo que se utiliza mediante un cliente web y un servidor http.

En la figura 3.8, se muestra un ejemplo de los paquetes que captura wireshark con algunos protocolos.

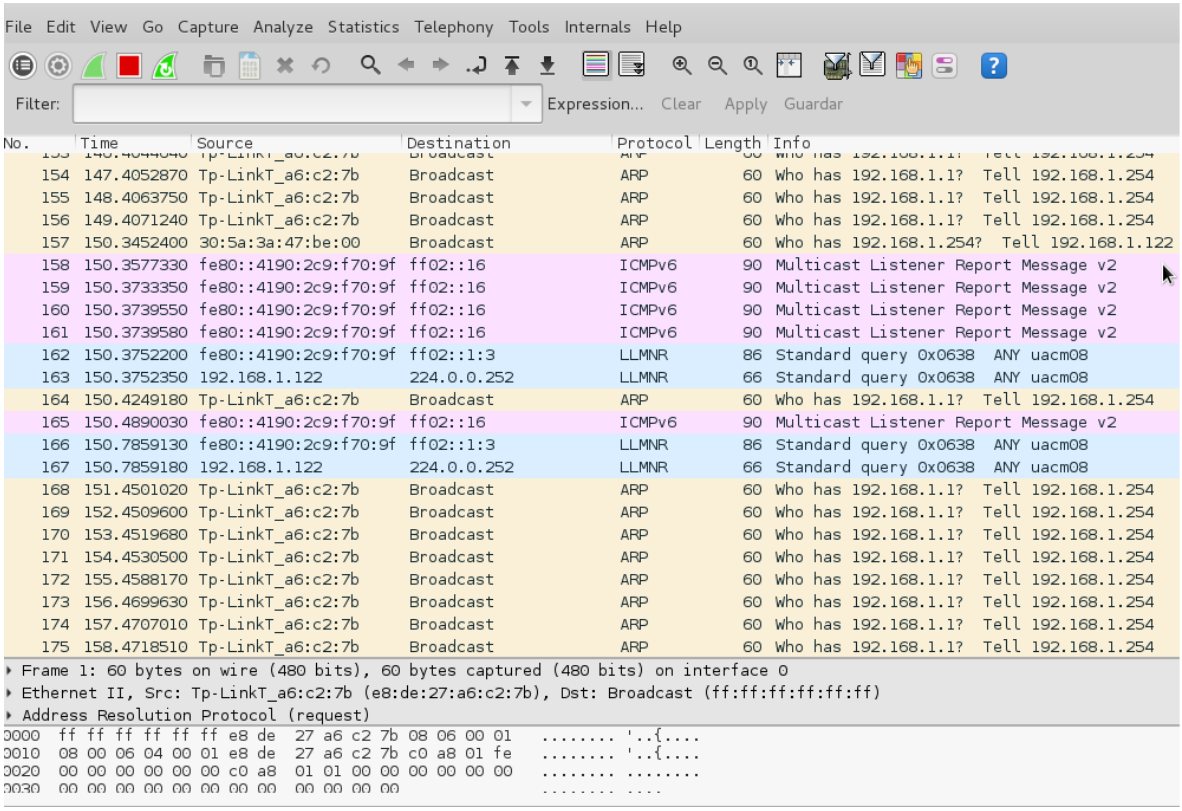


Figura 3.8. Captura de paquetes.

Para poder visualizar los protocolos se selecciona el cuadro rojo para detener la captura. En la parte superior de la interface de wireshark se encuentra un campo que se llama Filter el cual ayuda a filtrar los paquetes dependiendo del protocolo que se quiere analizar más a fondo. En este caso si se quisiera saber sobre los paquetes del cliente-servidor, se detiene la captura de paquetes y se filtran sólo aquellos que son http.

También se ejecutó la prueba con los siguientes archivos:

- **Archivo 3 PDF:** 100 soledad.pdf
  Contenido: texto del libro Gabriel García Márquez. “100 años de Soledad”.
- **Imagen JPEG:** fondo-estrellas.jpg
  Extraída de: www.astromia.com/universo/estrellavisible.htm

Cabe mencionar que al momento de hacer el filtraje se posicionó en el paquete a analizar y con el botón derecho del mouse. Se elige la opción de Follow TCP stream que sirve para visualizar el flujo de transporte de cada paquete y brinda la opción de ver el contenido del paquete en diferentes códigos, logrando así ver el contenido del paquete este caso fue contenido en texto plano. En el capítulo de resultados se muestran las imágenes extraídas de los paquetes y el contenido del mismo.

Se debe agregar que al igual que con el ataque MITM, con wireshark se puede interceptar el inicio de sesión a la página y así obtener información de autenticación del usuario.

3.8 Prueba SQL injection

Es una técnica donde un atacante crea o altera comandos SQL existentes para así poder acceder a la base de datos de un sitio web, exponer datos ocultos, sobrescribirlos o ejecutar comandos en el servidor. Para poder realizar esta prueba se necesita tener una página o portal web que sea susceptible a este ataque [11].

En el servicio de nube Owncloud fue donde se realizó esta prueba, el requisito para hacer que el portal web sea susceptible o vulnerable a este ataque es inyectando códigos SQL para encontrar errores a través del uso de una comilla simple que es interpretada por SQL como el inicio de una instrucción y que se envía por método GET haciéndolo de la siguiente manera, cuando los datos son enviados por este método, los campos de un formulario son incorporados a la URL, si se tuvieran 2 variables, por ejemplo, nombre y edad, éstas son separadas por un &, por ejemplo http://servidor.com/pagina.php?nombre=ana&edad=100, habrá un límite de variables a enviar sin embargo no se recomienda enviar información confidencial como contraseñas por este método.

Se probó poner al final de la URL la comilla simple junto al parámetro que se está enviando a la base de datos y al ejecutarse la consulta se obtiene como resultado un error de sintaxis SQL, lo que indica la no validación de la sentencia y por lo tanto la vulnerabilidad a este ataque.

Algunas sentencias que pueden servir para ver si la página web es vulnerable se muestran en la tabla 3.4.

pagina.cl/search.php?id=NUMERO	**pagina.cl/search.php?id=-1**
pagina.cl/menu.php?act=NUMERO	pagina.cl/menu.php?act='2
pagina.cl/index.php?opc=NUMERO	pagina.cl/index.php?opc=999999999999999999

Tabla 3.5. Ejemplos para ver la vulnerabilidad de una página web.

Para esta prueba se utilizó sqlmap, una herramienta disponible en el sistema operativo Kali Linux, que sirve para inyectar código sql, por así decirlo, para tratar de entrar a la base de datos de algún portal web. Cabe mencionar que este ataque no se hará manual, la herramienta ya ejecuta por si sola el proceso de inyección.

El procedimiento es el siguiente: solicitarle a sqlmap que despliegue qué bases de datos está utilizando la página web a través del comando --dbs e indicando también la url –u. Todo esto se ejecuta en la terminal de la siguiente manera:
sqlmap -u "https://192.168.X.X/pagina web" --dbs

Con esto se tiene la base de datos. Para saber qué tablas tiene, se ejecuta lo siguiente:
sqlmap -u "https://192.168.2.4/owncloud/index.php" -D nombredelabase –tables

Para saber las columnas, se ejecuta:
sqlmap -u "https://192.168.2.4/owncloud/index.php" -D nombredelabase –T user –columns

Y por último para ver la línea donde veremos los datos de autenticación de los usuarios registrados, se ejecuta:

sqlmap -u "https://192.168.2.4/owncloud/index.php" -D nombredelabase –T user –C login,password –dump

Cabe aclarar que las contraseñas desplegadas estarán encriptadas en código MD5 que es una codificación de 128 bits, representada típicamente como un número de 32 dígitos hexadecimal para lo cual ya existen convertidores en línea.

Hecho lo anterior, se prosiguió a evaluar si Owncloud era vulnerable con la siguiente sentencia:
sqlmap -u "https://192.168.2.4/owncloud/index.php" --dbs

A pesar de que la herramienta sqlmap ya realiza el proceso de inyección sql por si solo al ejecutarse la sentencia el proceso se terminaba antes de poder entrar a la base de datos, mostrando un error que hacía referencia a que no se encontraban parámetros para realizar las pruebas en los datos proporcionados, mostrando también que el parámetro GET proporcionado es no inyectable.

Dado este hecho también se intentó probar la vulnerabilidad del portal de Owncloud de la manera manual poniendo en la URL lo siguiente: http://192.168.2.4/owncloud/index.php?catid=1', teniendo como respuesta que el portal web con esta sentencia carga de la misma manera, es decir aun poniendo el carácter de comilla simple el portal no encuentra errores y sigue cargando la página de manera normal pidiendo la autenticación del usuario, esto quiere decir que Owncloud no es vulnerable a ataques de sql inyección ya que por su seguridad no permite realizar este tipo de sentencias para poder entrar a la base de datos, es por esta razón que la prueba no fue concluida.

CAPÍTULO 4
RESULTADOS Y DISCUSIÓN

CAPÍTULO 4

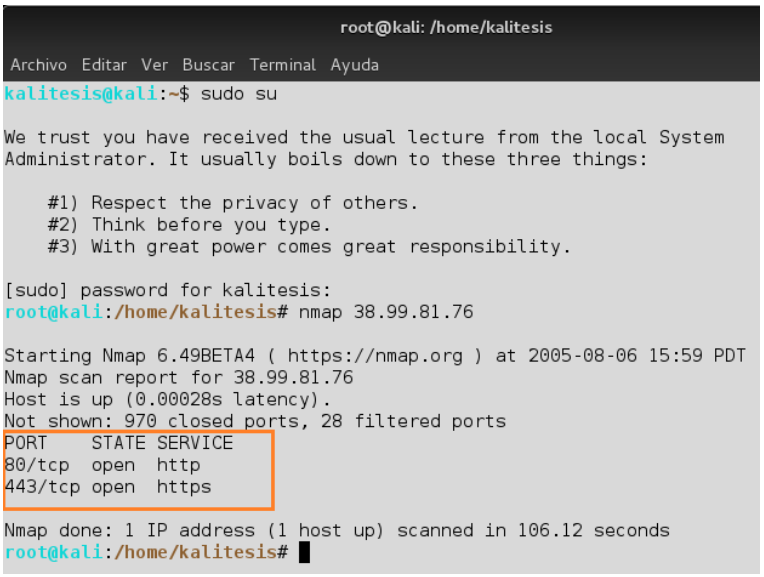
Resultados y discusión

Aquí se redacta los resultados obtenidos en las pruebas empleadas, en cada una de ellas se describe el resultado con una imagen clara, haciendo mención de algunas recomendaciones generales que bien puede tomarse en cuenta al elegir un servicio de almacenamiento o para la protección de los ataques que se realizaron, esto basado en la literatura consultada y en las pruebas hechas.

4.1 Diagnostico de vulnerabilidades con NMAP

En la figura 4.1 incisos (a) y (b), se muestran los resultados obtenidos de los diagnósticos de vulnerabilidades realizados al servidor de nube SurDoc. En el inciso (a), muestra 970 puertos cerrados, lo que significa que pueden ser accesibles en cualquier momento pero no hay una aplicación escuchándolos en este momento; 2 puertos abiertos, el 80 que es para el protocolo http y el 443 que es para el protocolo https, lo que indica que la máquina destino se encuentra esperando paquetes en estos puertos. Además se puede observar en el inciso (b) que este servicio de nube no tiene vulnerabilidades, debido a la seguridad en los servidores de SurDoc el servicio no tiene vulnerabilidad alguna al momento del diagnóstico.

Figura 4.1. Resultados del diagnóstico nmap a Nube SurDoc

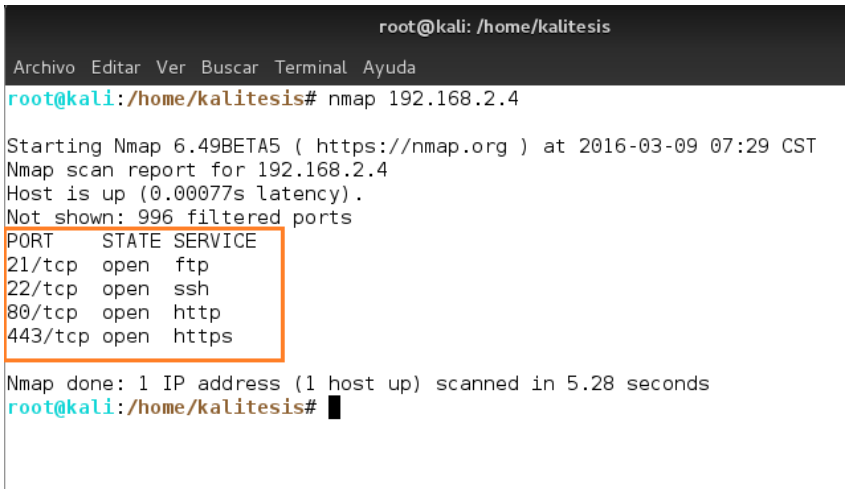


(a). comando *nmap 38.99.81.76*

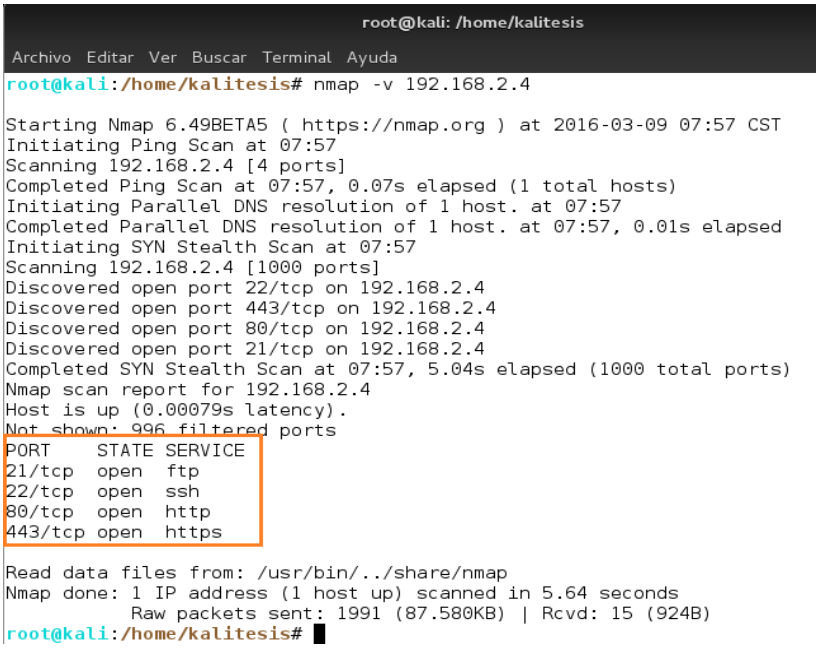
(b). comando nmap –f –script vuln 38.99.81.76

Por otro lado, en las figura 4.2 incisos (a) y (b), se muestran los resultados obtenidos de los diagnósticos realizados al servidor de nube Owncloud. El inciso (a), muestra 996 puertos cerrados, de igual manera no hay una aplicación escuchándololos, aun así son accesibles esos puertos; 4 puertos abiertos, el 80 que es para el protocolo http, el 443 que es para el protocolo https, el 21 para protocolos ftp y por último el 22 para protocolos ssh. Estos puertos son accesibles ya que hay una aplicación escuchando y esperando recibir paquetes en cualquier momento. Además se puede observar en el inciso (b) un despliegue de información más completa con un comando diferente (nmap –v) que muestra información más detallada, en este caso muestra los puertos señalados anteriormente.

Figura 4.2 Resultados de escaneo nmap a Nube Owncloud.



(a). comando *nmap 192.168.2.4*



(b). comando nmap –v 192.168.2.4

Los resultados obtenidos con nmap fueron de gran apoyo para visualizar de una manera más directa el estado de los puertos que tenían las nubes, la información proporcionada es muy amplia y clara respecto a los puertos abiertos, puertos en servicio, si hay presencia de un firewall y si presentan alguna vulnerabilidad en sus puertos. NMAP no es capaz de determinar si un puerto realmente está cerrado, debido a que si no existe una aplicación que este escuchando el puerto aparecerá como filtrado o protegido por un firewall, esto es porque no hay manera de escuchar el puerto por así decirlo y lo que detecta NMAP sería solo la protección que tiene al momento del diagnóstico.

4.2 Diagnostico de vulnerabilidades con NESSUS

A continuación, se muestran los resultados de los diagnostico realizados a los servicios de nube utilizando la herramienta NESSUS.

4.2.1 Resultados del diagnóstico en SurDoc

En la figura 4.5, se pueden observar 28 resultados de información en color azul representa el 87 % y 3 resultados de vulnerabilidad de nivel bajo en color verde que representa el 13 %.

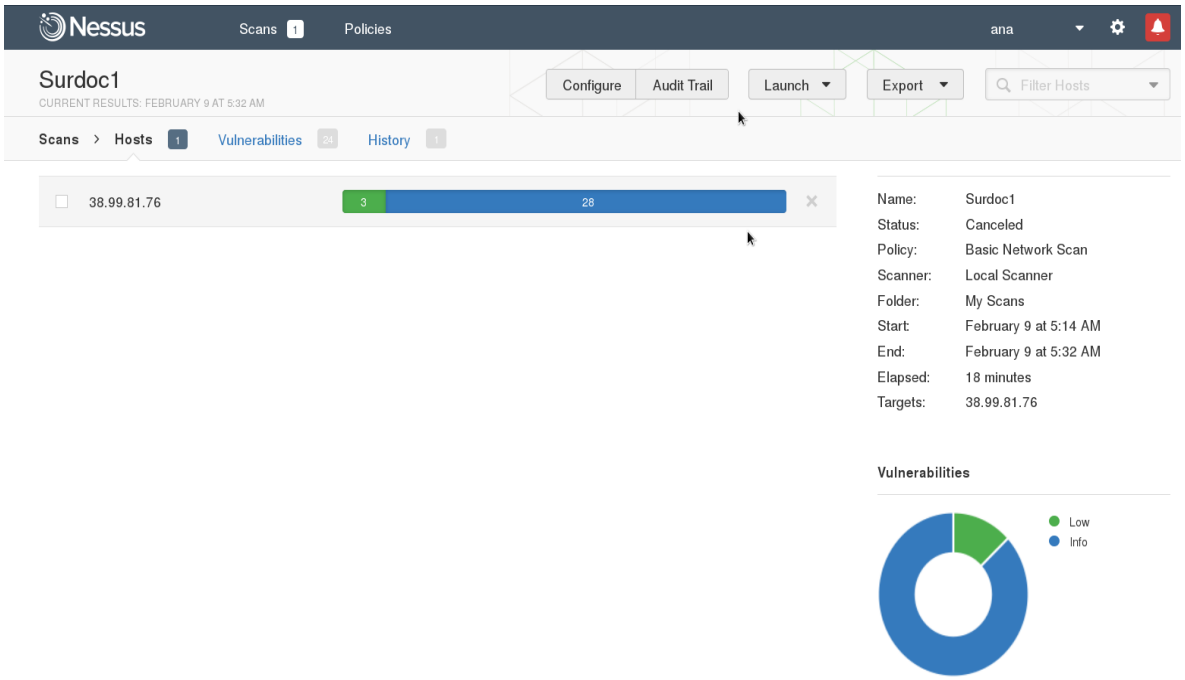


Figura 4.3. Información general del diagnóstico.

En la figura 4.6, se muestra el despliegue de los resultados encontrados, en este caso no muestra puertos sin embargo muestra fallas en la seguridad, que un usuario malicioso podría tomar de referencia para realizar un ataque y poder adquirir información de manera ilícita.

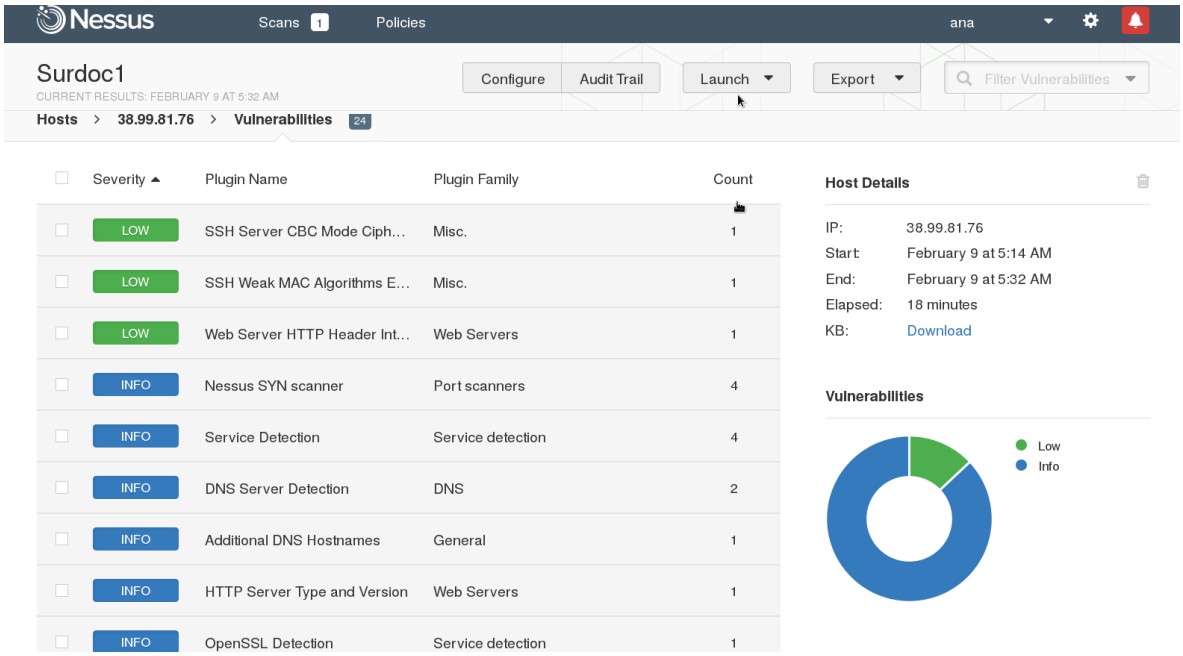


Figura 4.4. Despliegue de vulnerabilidades en SurDoc.

4.2.2 Resultados del diagnóstico en Owncloud

En la figura 4.7, se observan 47 resultados de información en color azul que representan el 79 %, 3 resultados de vulnerabilidad de nivel bajo en color verde que representan el 7 % y 7 resultados de vulnerabilidad de nivel medio en color amarillo que representan el 14 %.

De igual forma en el servicio de nube Owncloud también se tiene un despliegue de información más detallada respecto a las fallas de seguridad en el servicio, mismo que cuenta con una breve recomendación hecha por el mismo NESSUS, en la figura 4.8 se muestran los resultados obtenidos.

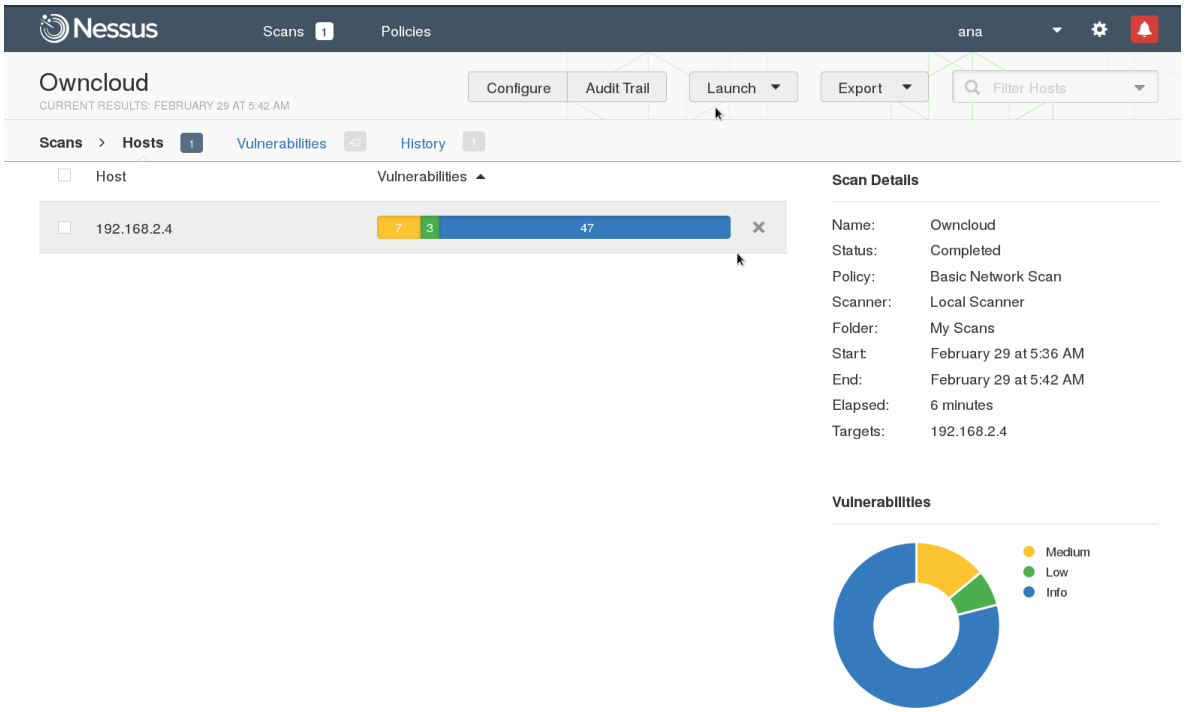


Figura 4.5. Información general del diagnóstico.

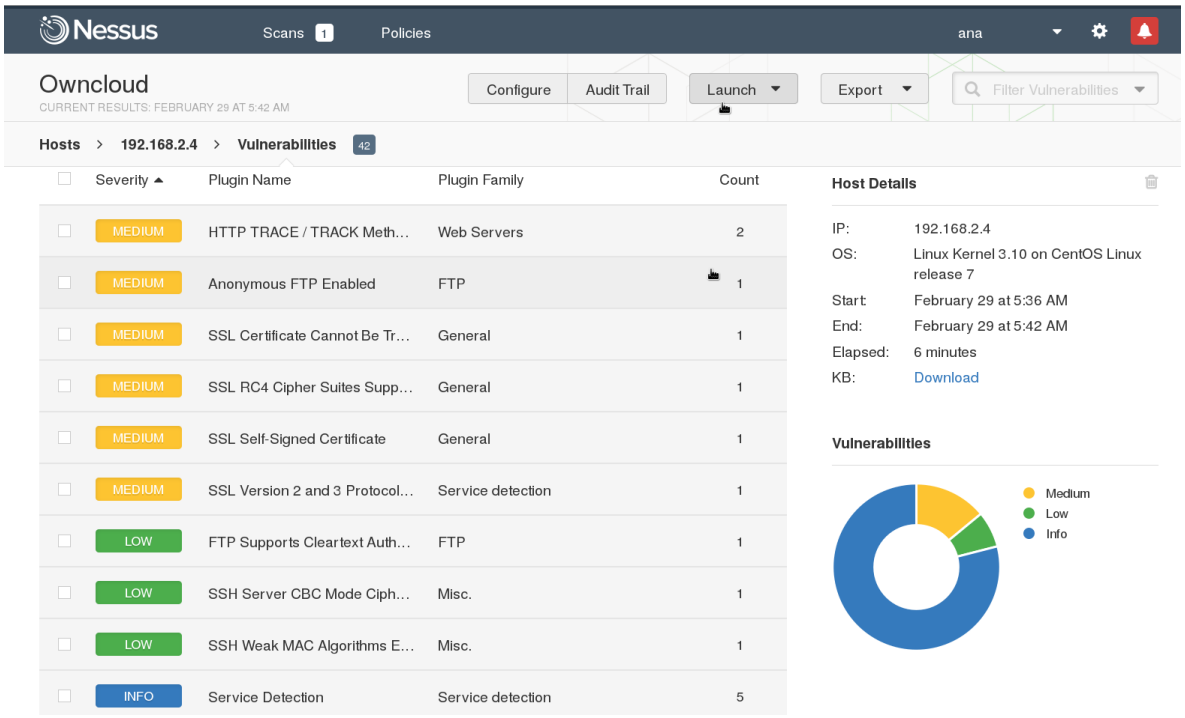


Figura 4.6. Despliegue de vulnerabilidades en Owncloud.

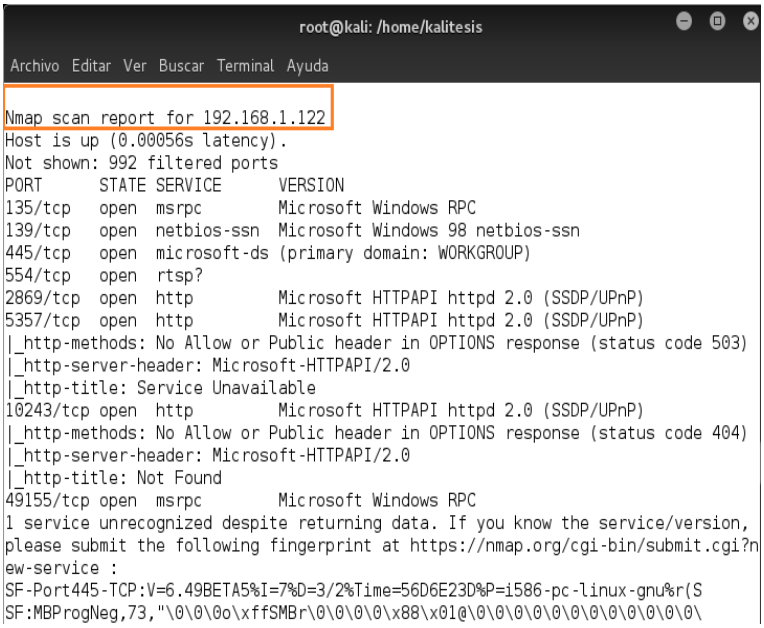
La herramienta de escaneo NESSUS, muestra una prueba más completa respecto de nmap debido a que encontró una serie más amplia de vulnerabilidades y las desplegó dependiendo la gravedad dada en baja, media o alta dando también una serie de recomendaciones para poder corregir el error.

4.3 Prueba MITM a Owncloud

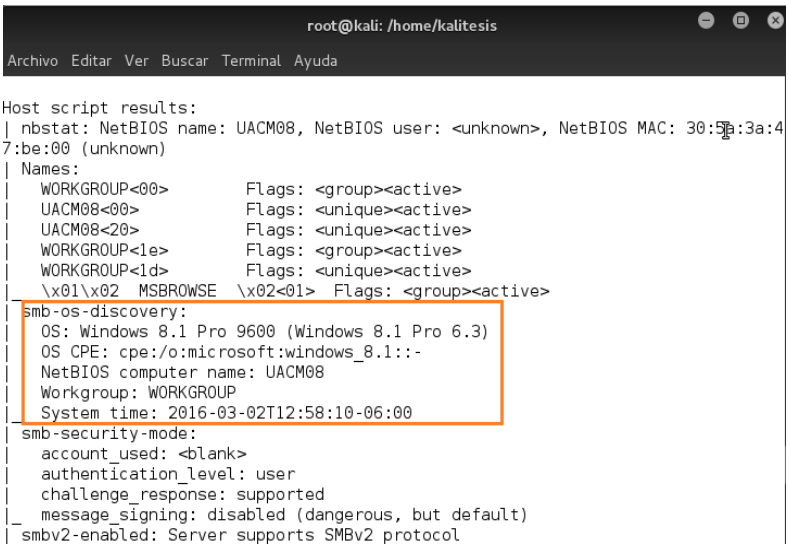
Como ya se había mencionado en el capítulo 3, la finalidad de este ataque es que la maquina atacante se ponga en medio de la comunicación entre el servidor y la maquina víctima de manera que el tráfico que se establezca entre ellas pueda ser visto por el atacante.

La obtención de la dirección IP de la PC1 (víctima) se muestra en la figura 4.7 inciso (a), ya que si el atacante no tuviera la dirección IP exacta de la víctima a la que quiere atacar, con este comando de NMAP se identifica las PC activas en la red. Para el análisis de esta prueba la PC 1 (víctima) fue la IP 192.168.1.122, también muestra algunos puertos en el mismo diagnóstico y a su vez indica que opera con Windows 8.1 como se observa en la misma figura inciso (b).

Figura 4.7. Comando nmap –T4 –A –v 192.168.1.*



(a). Resultado de la IP víctima.



(b). Detalles del S.O obtenidos sobre la víctima.

4.3.1 Envenenamiento ARP

Realización del envenenamiento ARP hacia la PC1 (víctima), en la figura 4.8 observamos él envió de paquetes ARP erróneos, asociando la MAC del atacante a la IP del gateway y también la IP de la víctima, para que todo el tráfico sea dirigido al atacante antes de ser entregado al destino real, que en este caso sería la PC1. En los recuadros azules se observa que la MAC del atacante es asociada a la IP del router para redirigir todo el tráfico que vaya destinado a la víctima.

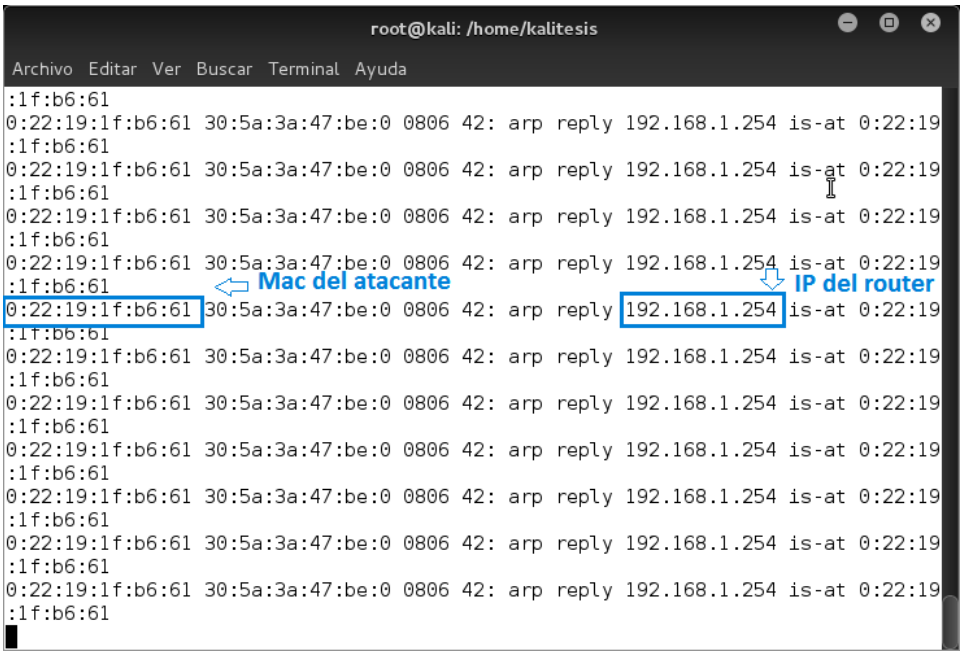


Figura 4.8. Comando arpspoof –i eth1 –t 192.168.1.122 (víctima) 192.168.1.254 (gateway)

4.3.2 Función sslstrip

Peticiones realizadas durante la conexión al servidor de nube Owncloud, ya que la función de sslstrip es hacerse pasar por un navegador seguro, es decir, convierte todo el tráfico HTTPS en tráfico HTTP (sin cifrar), haciendo creer a la víctima que está estableciendo una conexión segura con el servidor web.

En la figura 4.9 se observa que la función sslstrip está corriendo y está lista para "escuchar" en el puerto 8080.

Figura 4.9. Comando sslstrip –l 8080

4.3.3 Función ettercap

En la figura 4.10, se tiene la visualización final en texto plano de la información obtenida, en este caso se obtuvo la información de autenticación de usuario y contraseña de Owncloud, ya que ettercap es un lector de contraseñas de: telnet, ftp, http, entre otras, por tal motivo nos muestra el contenido en texto plano cuando termina el ataque.

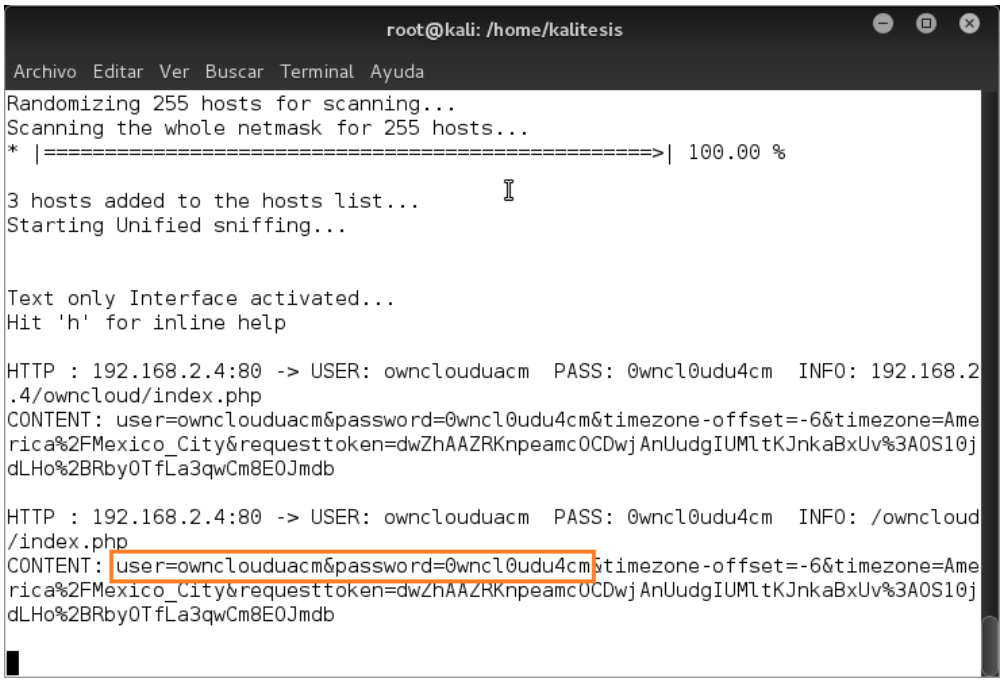


Figura 4.10. Comando y salida de ettercap –T –q –i eth1 para Owncloud.

También se obtuvo usuario y contraseña de Hotmail, que fue el correo creado para la cuenta de nube Owncloud, mostrada en la figura 4.11.

```
root@kali:/home/kalitesis# ettercap -T -q -i eth0

ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team

Listening on:
  eth0 -> 00:14:22:2C:2D:1D
          192.168.1.115/255.255.255.0
          fe80::214:22ff:fe2c:2d1d/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/eth0/use_tempaddr is not set to 0.
Privileges dropped to EUID 65534 EGID 65534...

  33 plugins
  42 protocol dissectors
  57 ports monitored
20388 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Text only Interface activated...
Hit 'h' for inline help

HTTP : 131.253.61.80:80 -> USER: linux_cloud@hotmail.com  PASS: Jm2015@U4CM  INFO: http://login.live.com/login.srf?wa=ws
.1.6195.0&wp=&wreply=https://prodigy.msn.com/es-mx/homepage/Secure/Passpor
CONTENT: loginfmt=linux_cloud%40hotmail.com&passwd=Jm2015%40U4CM&SI=Iniciar+sesi%C3%B3n&login=linux_cloud%40hotmail.com&
En%21JC5LUuZkyg*E15nRB2upEgTv5CJYGE8PhY%215141dNyaiy*cCs6JGGWFC20N3laNzS*02q8dki1F*7%21olMhouAwgFJS757u57RECTHpjKQgjgv17
rTdpwr80ChM7n0m3c7iuiyo%21h8a8SKisl*YeBEeppTTlDPIjK%217X6AIjhwbenWM*f55NkpoEqBgqWGis%21pQ%24%24&PPSX=P&idsbho=1&sso=0&Ne
B17&i4=0&i7=0&i12=1&i13=0&i14=294&i15=903&i17=0&i18=__Login_Strings%7C1%2C__Login_Core%7C1%2C__Login_OTC%7C1%2C
```

Figura 4.11. Comando y salida de ettercap –T –q –i eth1 para Hotmail.

La prueba MITM realizada a Owncloud fue exitosa ya que se situó a la PC2 (atacante) entre las dos partes que mantienen comunicación, en este caso la PC1 (víctima) y el servidor web. Se *envenenó* a la víctima por medio del protocolo ARP con la finalidad de interceptar los mensajes enviados al momento de iniciar sesión a Owncloud para finalmente adquirir la información confidencial del usuario.

4.4 Prueba de seguridad y análisis con Wireshark realizado en el servicio Owncloud

En el capítulo 3 se explica el desarrollo de esta prueba, que básicamente consiste en obtener datos de un usuario en texto plano mediante la herramienta Wireshark, interceptando el tráfico que va dirigido hacia la víctima, configurando un puerto espejo en el switch, que lo que hará es redirigir el tráfico entre la víctima y el servidor exactamente igual hacia al atacante.

En la figura 4.12, se muestra la intercepción de un archivo en texto plano por parte del atacante, un paquete intercambiado entre el servidor web y la víctima, el cual trabaja con el protocolo HTTP y se encuentra encerrado en el recuadro naranja. El atacante usa un sniffer para poder interceptar este archivo o paquete haciendo que el switch haga un espejo del tráfico entre el servidor y la víctima y así el atacante pueda recibir exactamente el mismo contenido. En la figura 4.12, se observa que el archivo interceptado tiene como fuente la dirección IP 192.168.1.122 (víctima) y su destino es la dirección IP 192.168.2.4 (servidor web).

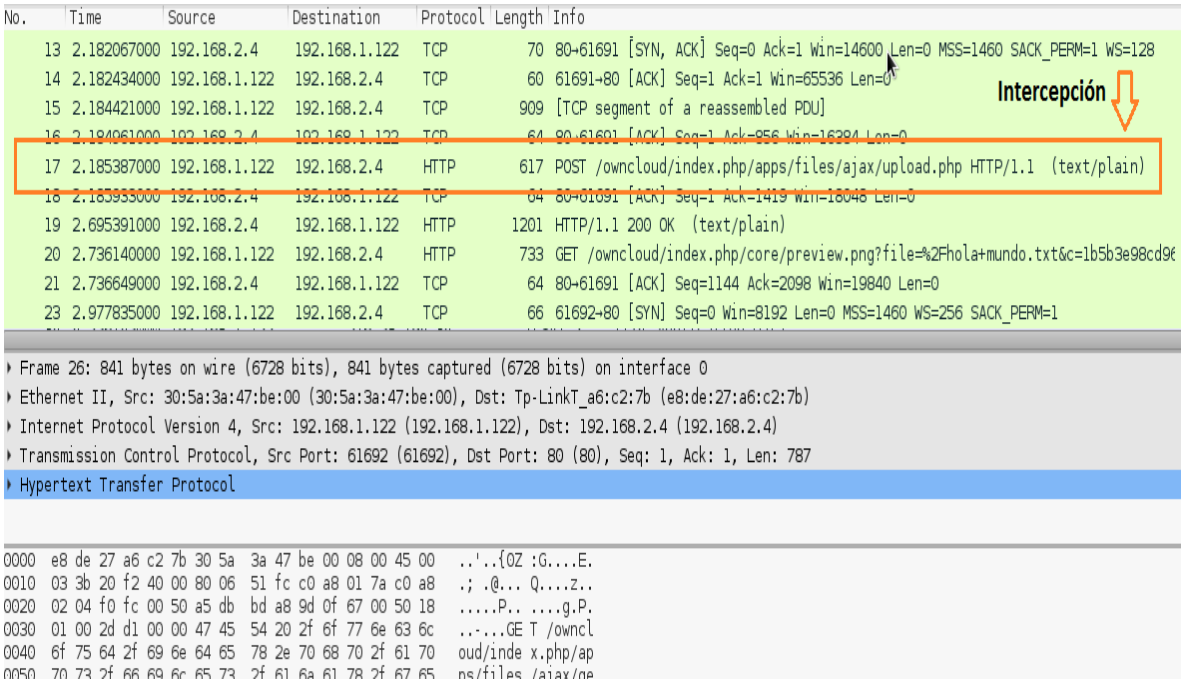


Figura 4.12. Intercepción de archivo txt.

En la figura 4.13,se observa el contenido del paquete, el cual nos despliega el nombre del archivo, la fecha y hora de intercepción y el sistema operativo de la PC que está extrayendo la información, al final de la imagen también se puede observar que el archivo interceptado fue extraído de un servidor apache seguido de su dirección IP.

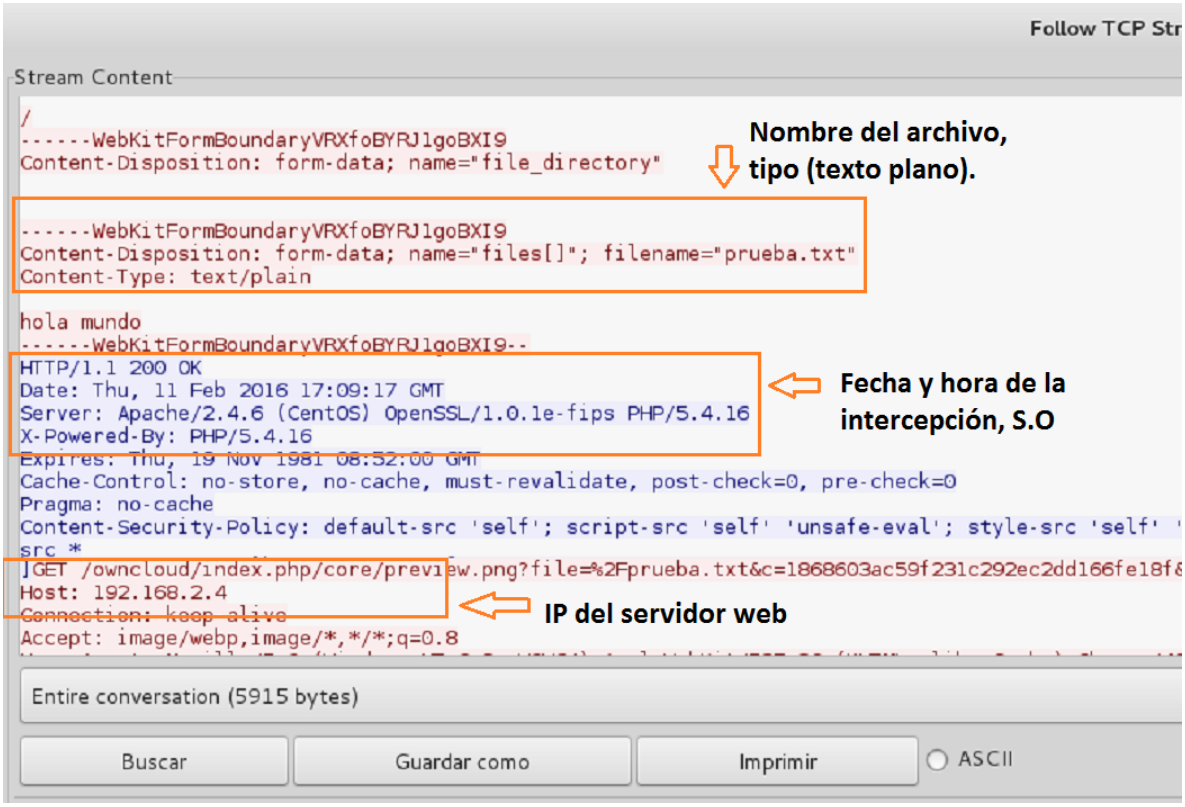


Figura 4.13. Despliegue de contenido de texto plano prueba.txt

En la figura 4.14, se visualiza más de cerca el contenido del paquete enviado al servidor web, donde el atacante pudo interceptar y visualizar el contenido, *"hola mundo "(ver* letras rojas).

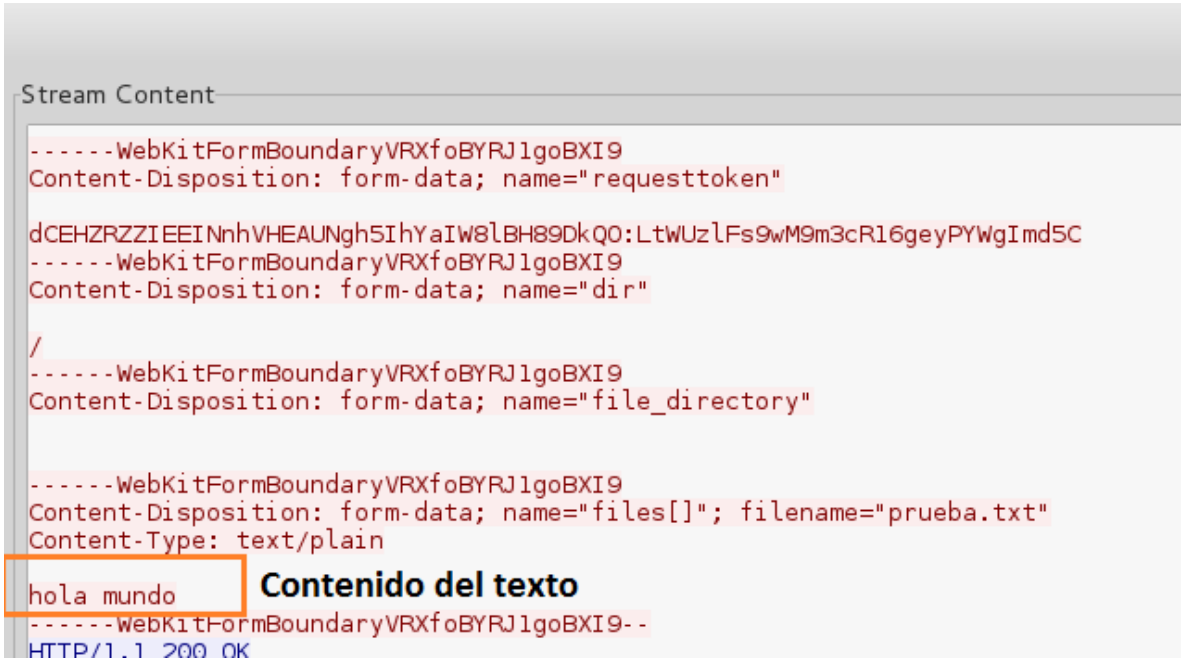


Figura 4.14. Visualización de texto plano prueba.txt

En la figura 4.15, se muestra nuevamente la intercepción de un archivo en texto plano, en este caso el archivo interceptado fue un fragmento de un cuento infantil extraído de internet, la intercepción de este archivo también se observa en el recuadro naranja con IP fuente 192.168.1.22 (víctima) y con dirección IP destino (servidor web).

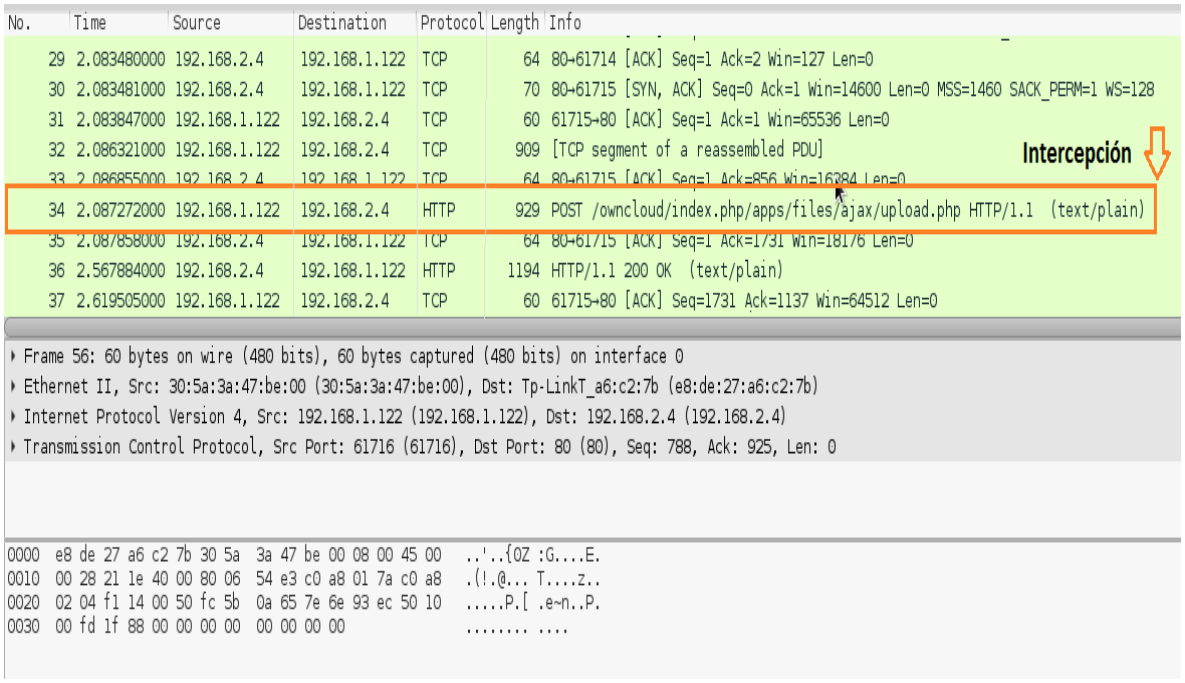


Figura 4.15. Intercepción de archivo txt.

En la figura 4.16 se muestra el contenido del paquete, el cual despliega los mismos datos que en el archivo anterior como son: nombre del archivo, fecha y hora de intercepción y el sistema operativo de la PC que está obteniendo la información.

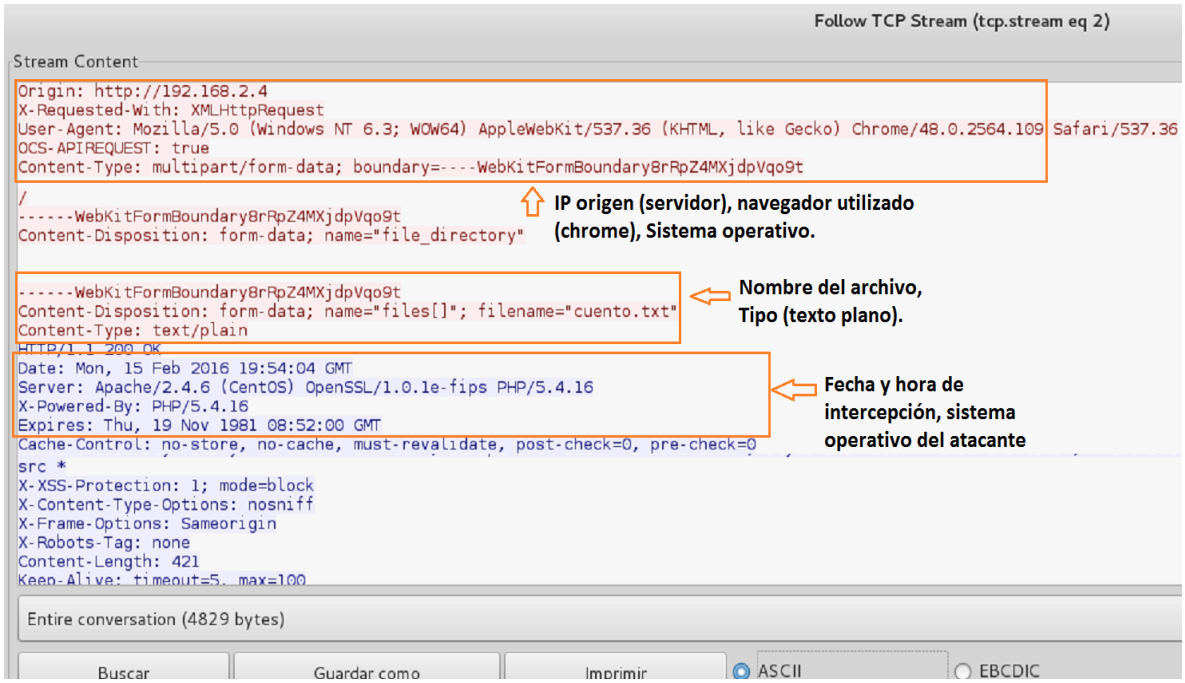


Figura 4.16. Despliegue de contenido de texto plano cuento.txt

En la figura 4.17, se visualiza más de cerca el contenido del paquete enviado al servidor web por la víctima, donde el atacante pudo interceptar y visualizar el contenido, *"Al atardecer, llegaron los dueños de la casa. Eran siete enanitos que trabajaban en unas minas. Se quedaron admirados al descubrir a Blancanieves. Ella les contó toda su triste historia y los enanitos la abrazaron y suplicaron a la niña que se quedase con ellos. Blancanieves aceptó y se quedó a vivir con ellos. Eran felices ".* (Ver letras rojas).

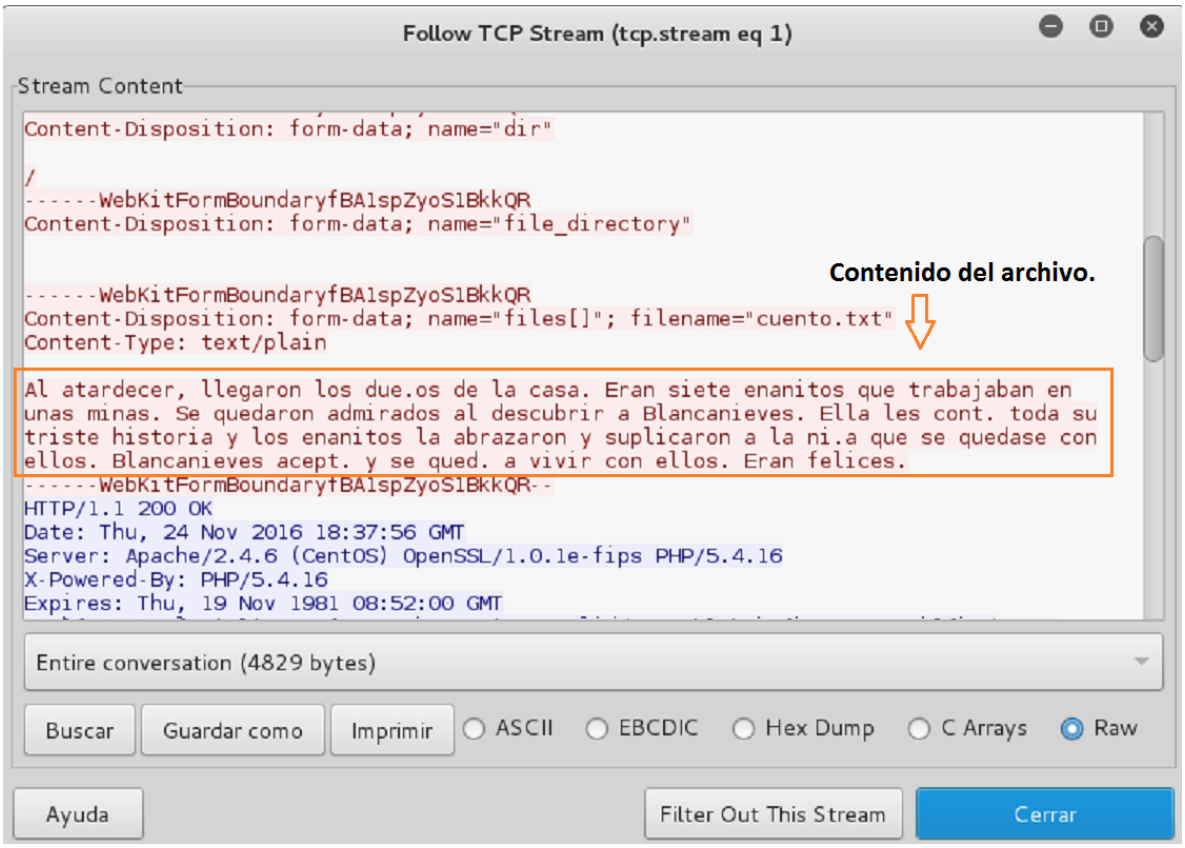


Figura 4.17. Visualización de texto plano cuento.txt

Por otro lado también se interceptó la información de autenticación del usuario al momento de iniciar sesión en Owncloud, en la figura 4.18, se observa la intercepción de esta información (recuadro naranja.)

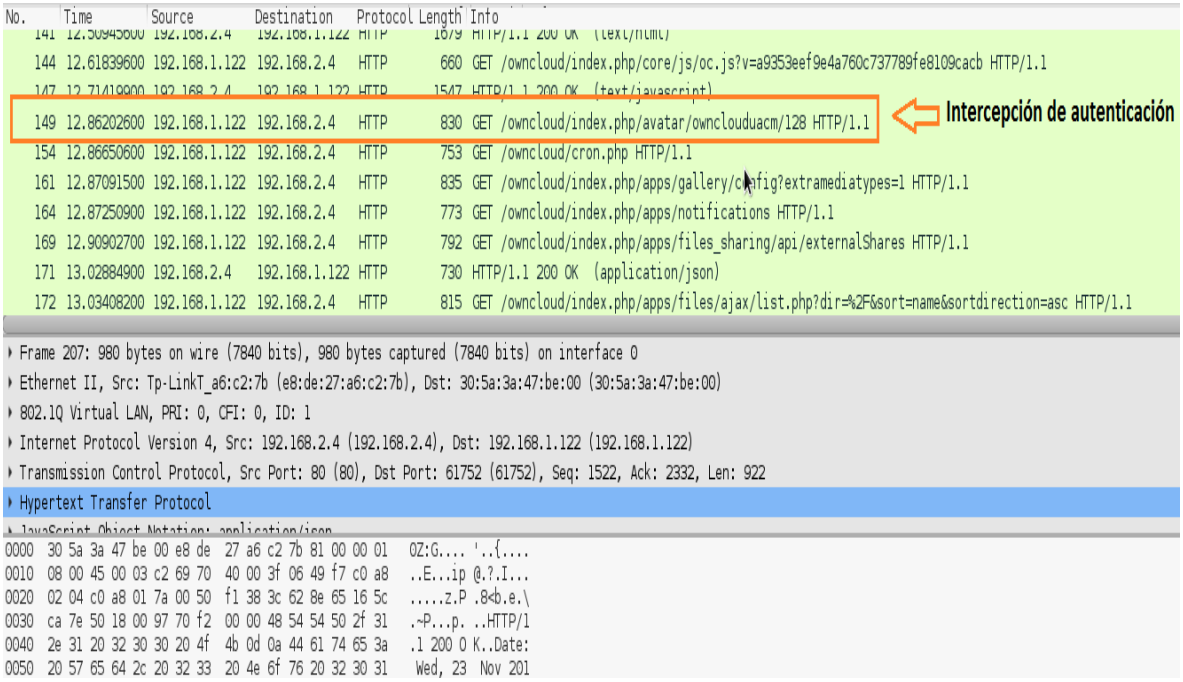


Figura 4.18. Intercepción de datos.

En la figura 4.19, se muestra el despliegue del contenido del paquete interceptado, con la hora y día de la extracción, el sistema operativo, y al final de la misma figura se encuentra también la información detallada del portal web usado en este caso fue el sistema de almacenamiento de Owncloud seguido de su dirección IP.

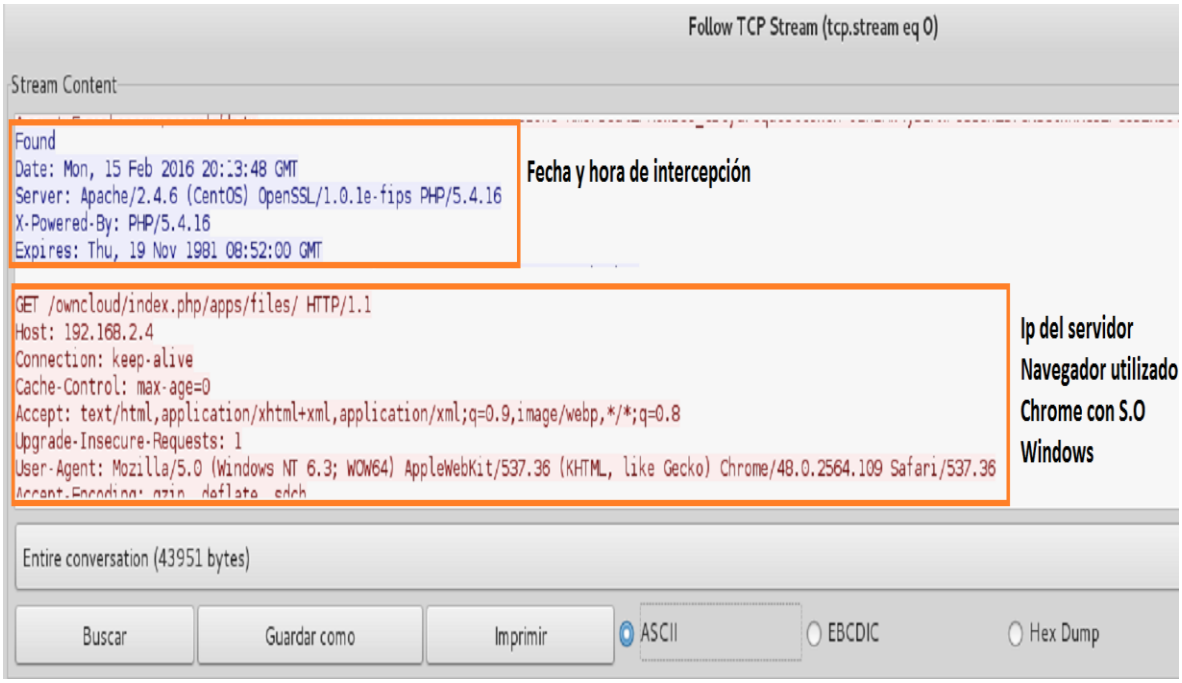


Figura 4.19. Despliegue de contenido del paquete.

En la figura 4.20 se muestra el contenido de la información del usuario un poco más de cerca, extraída al momento de la autenticación con *usuario: owncloudduacm y contraseña: 0wncl0uduacm.*

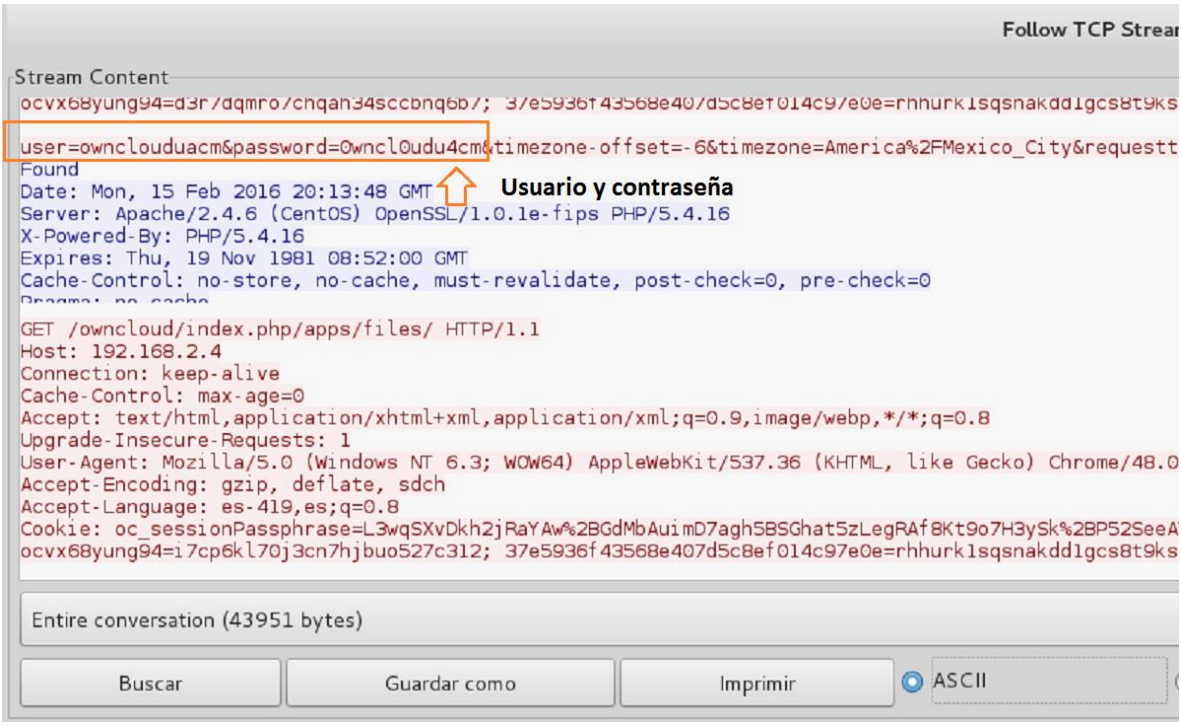


Figura 4.20. Visualización de usuario y contraseña.

Las evidencias mostradas en la prueba de seguridad y análisis con Wireshark (figuras 4.14 y 4.17) que hacen referencia a los resultados de los archivos interceptados, prueba.txt y cuento.txt, estas intervenciones fueron exitosas. Los archivos enviados a Owncloud se filtraron por medio de los protocolos **http**. De cada archivo interceptado se generó un resultado que muestra un paquete de la fuente 192.168.1.122 al destino 192.168.2.4 (figura 4.12).

Lo anterior indico que la intercepción de datos fue lo que se esperaba, ya que se pudo capturar el texto plano enviado al servidor de nube Owncloud.
Además, con esta herramienta se pueden obtener nombres de usuario y contraseña con el que se inicia sesión. Para nuestro análisis de la cuenta de servidor de nube Owncloud, a lo que se realizó el mismo procedimiento para la prueba MITM y se obtuvo lo siguiente:
Usuario: owncloudduacm
Contraseña: 0wncl0udu4cm
Cabe mencionar que estos resultados fueron mostrados en la prueba anterior en la figura 4.10.

4.5 Prueba SQL injection

La prueba consiste en inyectar códigos sql al portal de Owncloud, por medio de la herramienta sqlmap que nos brinda Kali Linux, esto para lograr entrar a la base de datos y así poder obtener información de los usuarios registrados como datos de autenticación de sus cuentas.

En la figura 4.21, se observa el inicio de la ejecución de sqlmap que como ya se mencionó es una herramienta de kali Linux. El proceso para la inyección de código sql al portal web en un inicio no tiene problema pero el proceso queda trunco, mostrando un mensaje de advertencia.

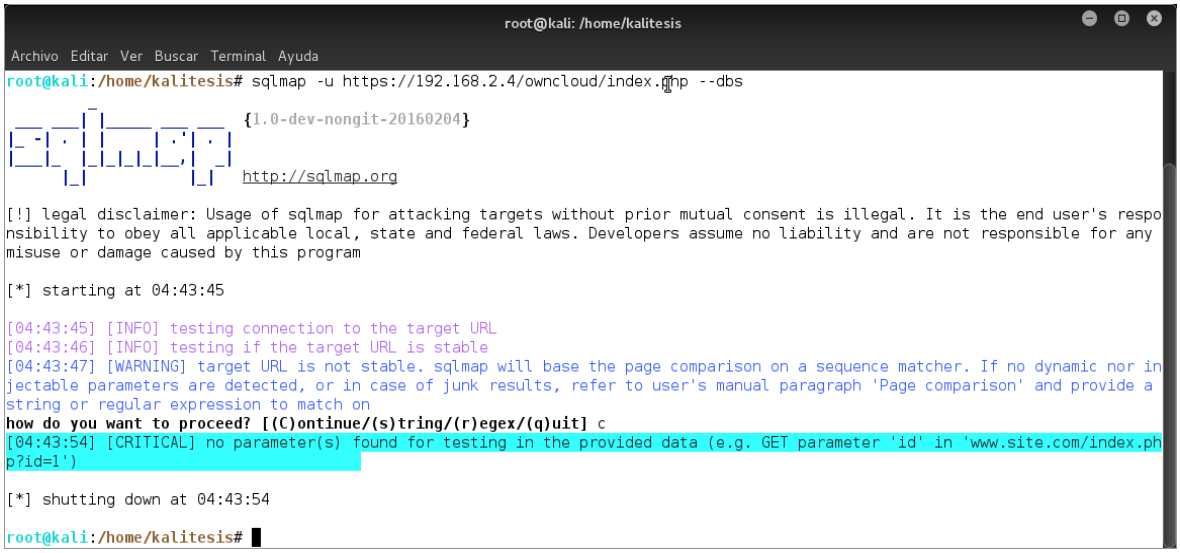


Figura 4.21. Comando y salida de sqlmap –u "https://192.168.2.4/index.php" –dbs

A continuación, en la figura 4.22, se muestra el proceso final que ejecuta la herramienta sqlmap, teniendo como resultado que el portal de Owncloud no es susceptible a inyección de código sql, y es por esta razón que la herramienta no logra terminar el proceso de inyección, para poder ingresar a la base de datos y extraer la información del usuario.

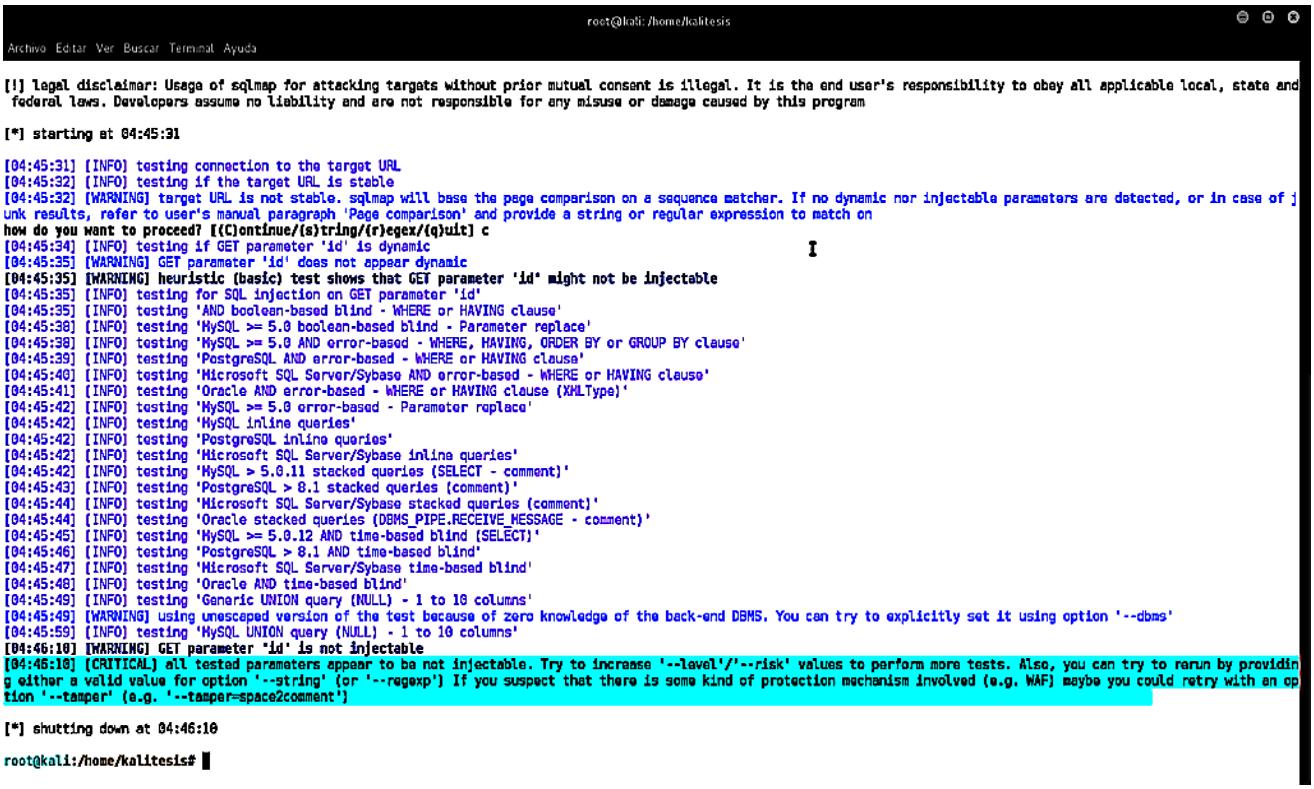


Figura 4.22. Error de ejecución para la inyección SQL.

En el caso particular del servidor de nube Owncloud, no se pudo realizar la inyección SQL, debido a que el portal web no es vulnerable a ningún tipo de código o sentencia SQL, de tal manera que se comprobó que cualquier elemento que se pusiera en la URL de la página no arrojo ningún error de sintaxis de SQL puesto que la página seguía cargando exactamente igual. Esto también se debe a que la versión usada de Owncloud es la 8.2.1, ya que las versiones anteriores a la 3.0.1 y posteriores a la misma liberaron una actualización que elimina una serie de vulnerabilidades críticas, una de ellas cierra agujeros, es decir, no hay errores de sql para el portal de Owncloud.

Por ello, no es posible hacer este tipo de ataque al servidor de nube Owncloud.

4.6 Soluciones y recomendaciones

4.6.1 Solución recomendada para la prueba MITM

Existen diferentes soluciones para la prueba MITM que pueden ser implementadas para no estar en riesgo y ser víctima de este tipo de ataque, esta recomendación está basada en literatura consultada [23]:

- **Usar siempre HTTPS**
  El servidor se verifica a si mismo presentando un certificado digital y se establece un canal cifrado entre el cliente y el servidor a través del cual se envía la información de forma confidencial. Siempre que se visite una página se debe asegurar que la dirección muestre HTTPS en lugar de HTTP, y si no lo hace, escribirlo manualmente. Esto no protege de vulnerabilidades del lado del cliente, pero al menos evita que los ataques menos sofisticados intercepten las comunicaciones.

- **Evitar conexiones WIFI**
  Otra solución sería evitar conectarse a routers wifi abiertos o en su defecto usar una navegación con protocolo HTTPS la cual establece una conexión segura siempre que sea posible.

- **Usar una red VPN** (Virtual Private Network)
  Es una tecnología de red que se utiliza para conectar una o más computadoras a una red privada utilizando Internet. De esta manera la conexión se cifra entre un cliente VPN y un servidor VPN, estableciéndose a través de un túnel de comunicación seguro, aunque también este tipo de comunicación es propenso a ataques MITM, al menos hará más difícil la intercepción.

- **Comprobar correo**
  Los correos electrónicos que se reciben de remitentes desconocidos con links fraudulentos y sospechosos son muy comunes por lo tanto evitar hacer clic en enlaces. Para obtener acceso a sitios seguros, escriba la dirección del sitio en el navegador.

- **Actualizar los navegadores**

  Un factor importante al hacer una consulta segura es descargar la última versión de los navegadores web de alta seguridad, como: Internet Explorer 7 o versión superior, FireFox 3 o versión superior, Google Chrome, Safari u Opera.

- **Verificación de dos pasos**

  Aumentar la seguridad del acceso a las cuentas de usuario, siempre que el mecanismo de verificación de los dos factores que más a delante se ejemplifican sea suficientemente fuerte. Esta es otra línea de defensa contra atacantes, por ejemplo: introducir una contraseña o PIN + huellas dactilares, patrón de iris o reconocimiento de voz.

- **Aplicación Marmita 1.3**

  Esta es una solución con mayor eficacia ya que la alerta ocurre en tiempo real es un software diseñado para detectar ataques de envenenamiento por ARP, donde puede iniciarse en caso de tener duda si el sitio web al que estamos accediendo es seguro [24].

  La aplicación se ejecuta antes de iniciar una conexión a algún sitio web. Al momento de detectarse el ataque y que el usuario intente autentificarse en el servicio de nube Owncloud, la herramienta de inmediato lanza una alerta informando al usuario que está siendo atacado mediante envenenamiento ARP, mostrando también la tabla ARP, analizando los paquetes y cuantificándolos, entre otras cosas.

En las figuras 4.23, 4.24 y 4.25 se muestran los resultados de la aplicación Marmita 1.3 obtenidos al usar esta herramienta y al hacer un ataque MITM al servicio de nube Owncloud.

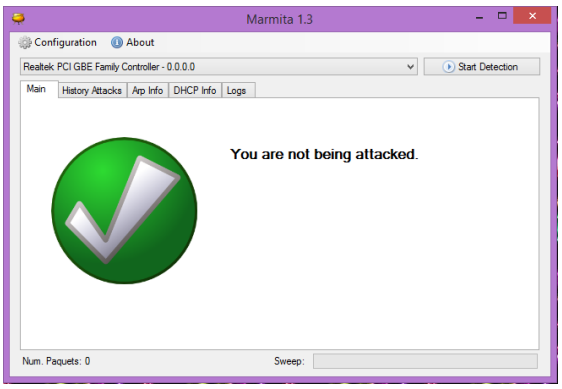


Figura 4.23. Aplicación Marmita.

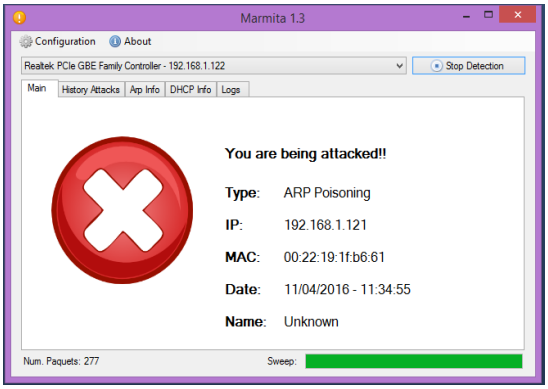


Figura 4.24. Alerta de ataque MITM.

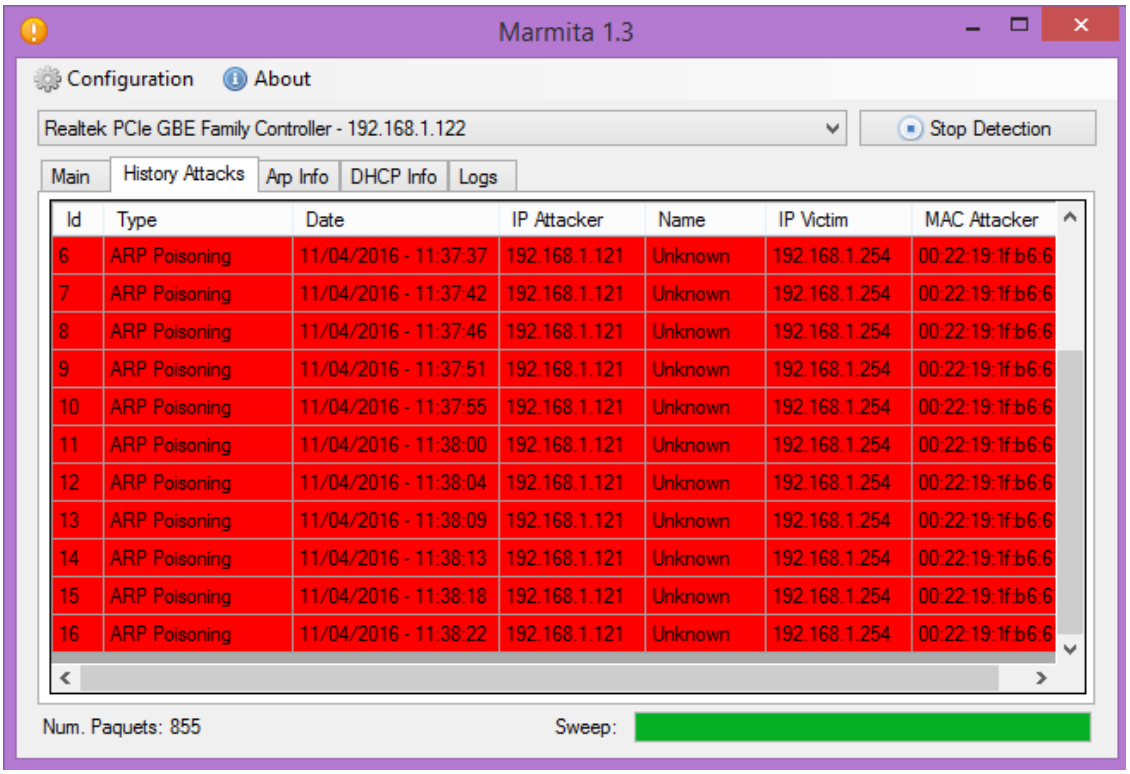


Figura 4.25. Información obtenida sobre el atacante.

4.6.2 Solución recomendada para la prueba de seguridad y análisis con Wireshark

Para tener protección contra ataques de intercepción de datos, es necesario contar con un sistema de cifrado con el que se proteja la comunicación al momento de estar intercambiando información. Esta solución es basada en las pruebas realizadas en el presente trabajo.

En el caso de la prueba hecha al servicio de nube de Owncloud, se tomaron medidas para la protección de los datos almacenados, para ello, se utilizó seguridad SSL ya que es una solución segura en línea porque garantiza a los clientes que el sitio consultado es seguro, autentico, real y confiable, ya sea para una simple visita, hacer compras en línea o iniciar sesión y así prevenir que se filtre información en caso de ser atacados por algún usuario malicioso.

Así mismo, se volvió a realizar la prueba pero ahora rectificando que los datos enviados al servidor se transmitieran encriptados.

Retomando los pasos de esta prueba redactados en el capítulo 3 del presente trabajo, se realizó una intercepción con la herramienta Wireshark a un archivo pdf, en las figuras 4.26 y 4.27 se muestran los resultados obtenidos.

En la figura 4.26, se observa que la intercepción está encriptada, mostrando también el protocolo de seguridad TLS Transport Layer Security, se muestra indicado como Application Data, con dirección IP fuente 192.168.1.122 y dirección IP destino 192.168.2.4, de igual manera en la figura 4.27, se muestra el contenido del paquete, el cual está formado solo por caracteres debido a la encriptación del mismo.

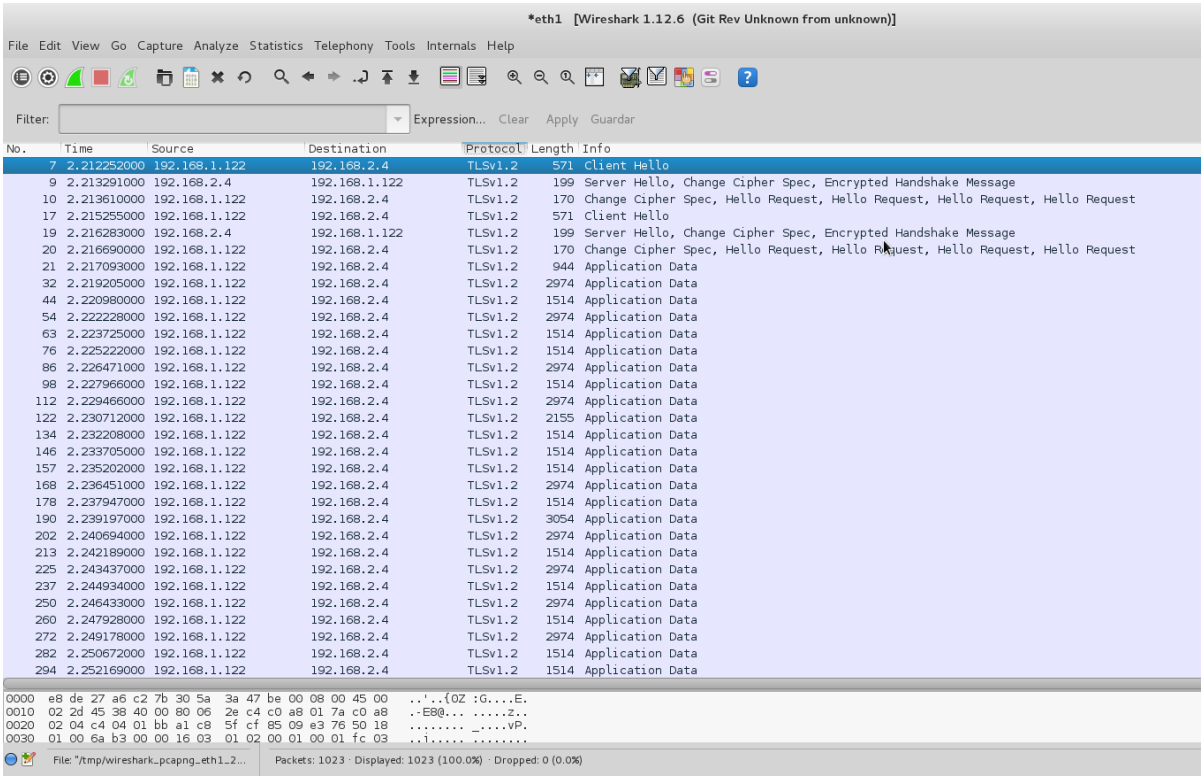


Figura 4.26. Intercepción de archivo pdf encriptado.

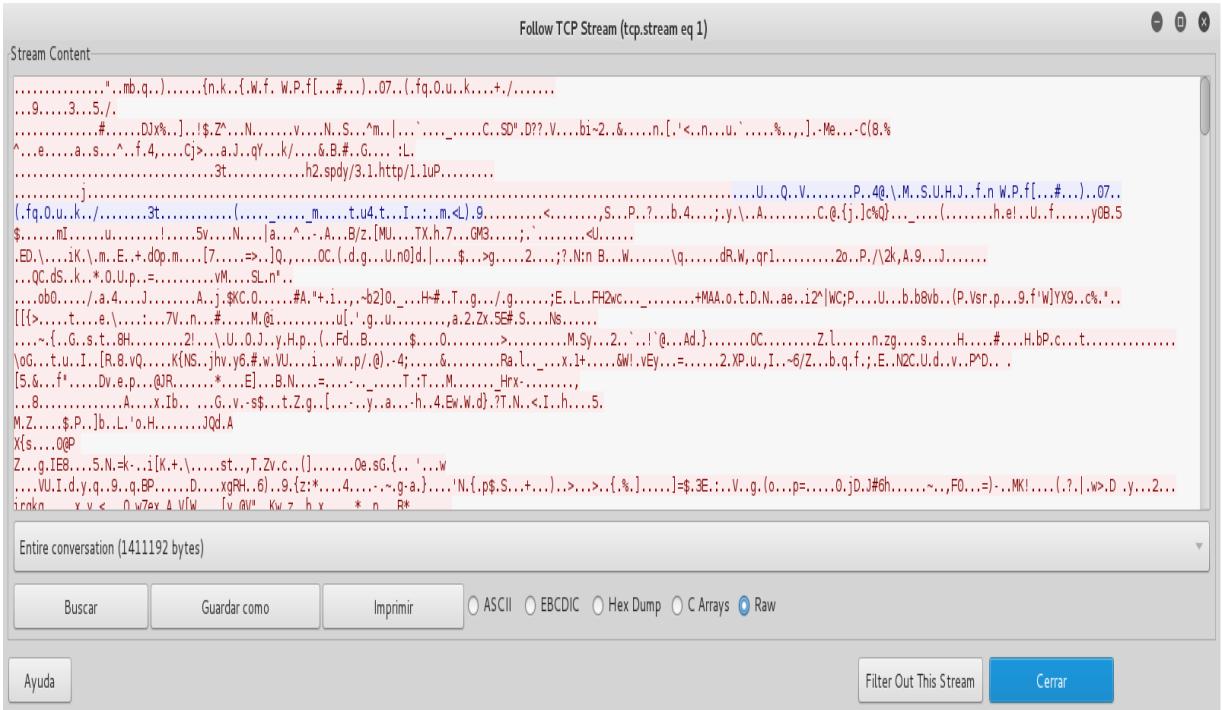
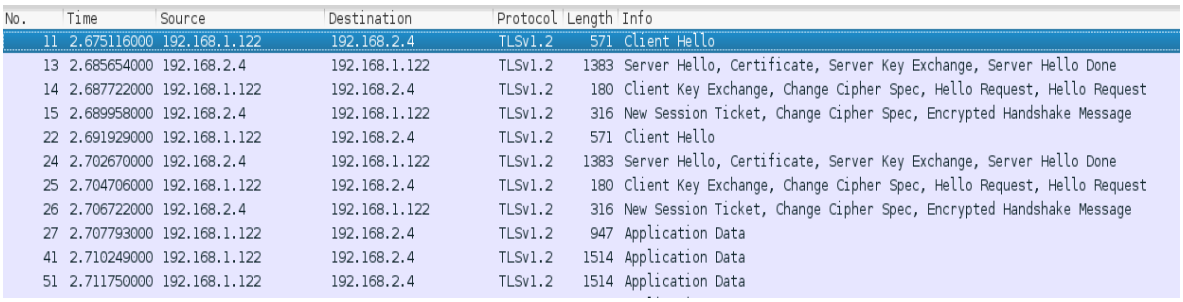


Figura 4.27. Despliegue de datos encriptados.

En la figura 4.28, se muestra la comunicación establecida entre el cliente y el servidor al momento de interceptar el archivo, lo hacen mediante el protocolo handshake para que ambas partes se autentiquen entre sí y puedan intercambiar información.



No.	Time	Source	Destination	Protocol	Length	Info
11	2.675116000	192.168.1.122	192.168.2.4	TLSv1.2	571	Client Hello
13	2.685654000	192.168.2.4	192.168.1.122	TLSv1.2	1383	Server Hello, Certificate, Server Key Exchange, Server Hello Done
14	2.687722000	192.168.1.122	192.168.2.4	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Hello Request, Hello Request
15	2.689958000	192.168.2.4	192.168.1.122	TLSv1.2	316	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
22	2.691929000	192.168.1.122	192.168.2.4	TLSv1.2	571	Client Hello
24	2.702670000	192.168.2.4	192.168.1.122	TLSv1.2	1383	Server Hello, Certificate, Server Key Exchange, Server Hello Done
25	2.704706000	192.168.1.122	192.168.2.4	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Hello Request, Hello Request
26	2.706722000	192.168.2.4	192.168.1.122	TLSv1.2	316	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
27	2.707793000	192.168.1.122	192.168.2.4	TLSv1.2	947	Application Data
41	2.710249000	192.168.1.122	192.168.2.4	TLSv1.2	1514	Application Data
51	2.711750000	192.168.1.122	192.168.2.4	TLSv1.2	1514	Application Data

Figura 4.28. handshake entre el cliente y servidor.

4.6.3 Recomendación para diagnóstico de vulnerabilidades con NMAP

Recomendación para el punto 4.1

- Utilizar puertos que no sean un estándar.
  Al querer tener acceso a información privada de algún usuario el atacante se basa en puertos estándar, es decir, 21 ftp, 22 ssh, 443 https, 80 http, entre otros. Así al utilizar puertos no estándar tendrá mayor dificultad el atacante para encontrar la información [16].

- Abrir solo los puertos necesarios.
  No solo tener abiertos los puertos que se están utilizando si no también otra opción sería cerrar o silenciar los puertos, la diferencia entre estos es que al cerrarlos y al hacer un análisis de vulnerabilidad, el puerto ofrecerá información de que está cerrado y al silenciarlos, no revela información de su estado.
- Actualización de software [16].
  Esta recomendación para algunos usuarios estaría de sobra ya que un software antiguo contendrá errores y fallos de vulnerabilidad donde cualquier script podría derrumbarlo, secuestrarlo o hacer mal uso de él [16].

4.6.4 Recomendación para la seguridad en el portal de Owncloud

En el portal web de Owncloud se encuentran pestañas para configuración personal de la cuenta, en la cual hay una opción llamada administración, en ella se encuentran opciones como avisos de seguridad y privacidad de la cuenta, opciones de correo, administración de los archivo, sugerencias, la versión del servidor, actualizaciones, entre otras cosas.

Para que el servidor se encuentre más seguro a la hora de enviar información confidencial por la red, se puede habilitar una opción la cual cifra el contenido de los archivos desde el momento en que son cargados a él. También se puede hacer una copia de seguridad de toda la instalación de Owncloud, es decir, en caso de pérdida de archivos por una mala administración hecha por el usuario se puede restaurar la instalación de Owncloud con la copia de seguridad, los pasos para generar la copia de seguridad están enlistados en el mismo portal.

Existe una opción la cual indica algunas recomendaciones de seguridad para el usuario entre ellas: Utilizar una conexión segura con HTTPS siempre que se utilice Owncloud para evitar ataques o robo de información, ajuste de rendimiento para el almacenamiento en la memoria caché o vinculaciones a sitios externos para poder acceder a páginas web rápidamente, todas las configuraciones de seguridad que el usuario puede configurar de acuerdo a sus necesidades se encuentran en el manual de administración del servidor, disponible en la siguiente dirección:
https://doc.owncloud.org/server/8.2/admin_manual/configuration_server/oc_server_tuning.html.

CONCLUSIONES

Conclusiones

Como resultado del análisis de seguridad en algunos servicios de alojamiento de archivos y para finalizar el presente trabajo se obtienen las siguientes conclusiones.

Al elaborar una serie de pruebas de seguridad y analizar algunas vulnerabilidades en servicios de nube privados como Owncloud, se demuestra la importancia que se debe tener al elegir una plataforma en la cual se almacenara información de cualquier tipo, como lo es en los servicios de nube, esperando que esta sea protegida debidamente para evitar que usuarios no autorizados tengan acceso a ella.

La extracción de información al momento de iniciar sesión en el servicio de nube Owncloud, indica que no siempre se tiene la certeza de que los archivos almacenados estén seguros, o en su defecto la información personal que se ingresa al iniciar sesión. Por otro lado un escaneo de puertos pone en riesgo la información almacenada en un equipo, ya que de esta forma se podrá saber que puertos son vulnerables para poder extraer información, en caso de no tener acceso algún equipo del cual se desee consultar archivos se podrá manipular el switch para tener una copia idéntica del intercambio de información que se esté dando entre dos equipos. Este tipo de problemas son los más comunes por los cuales la información es filtrada y manipulada por personas malintencionadas.

Finalmente con los resultados obtenidos durante este trabajo, se emiten algunas recomendaciones para poder generar un sistema más seguro y menos vulnerable ante algún atacante o mejor dicho, un usuario malicioso que pretenda extraer información aprovechando las fallas que algunos sistemas presentan en caso de no estar completamente protegidos.

Referencias bibliográficas

[1] Armando Carvajal. (2007). Introducción a las técnicas de ataque e investigación forense, un enfoque pragmático. Colombia: Globalteksecurity.

[2] Carlos tori. (2008). Hacking ético. Buenos Aires, argentina: autor-editor.

[3] Aaron Wheeler, Michael Winburn. (2015). Cloud Storage Security. Amsterdam: Elsevier.

[4] Ester Chicano Tejada. (2015).Gestión de servicios en el sistema informático. Antequera Málaga: IC editorial.

[5] José Salvador Sánchez Garreta, Ricardo Chalmeta Rolaleñ, Oscar Coltell Simon, Pilar Monfort Manero y Cristina Campos Sancho. (2003).Ingeniería de proyectos informáticos: actividades y procedimientos. Universitas.

[6] Héctor Jara & Federico G. Pacheco. (2012).Ethical Hacking: las técnicas de los hackers al servicio de la seguridad. Buenos Aires: red users.

[7] Andrew S. Tanenbaum. (1997). Redes de computadoras. México: Prentice hall.

[8] Rassoul Ghaznavi-zadeh. (2014). Ethical hacking and penetration step by step with Kali Linux. United States: Primedia e-launch LLC.

[9] Borja Merino Febrero. (2011).Análisis de tráfico con wireshark. España: Inteco.

[10] Behrouz A. Forouzan. (2010). TCP/IP Protocol Suite. New York: McGraw-Hill.

[11] Justin Clarke. (2012). SQL Injection attacks and defense. USA: Syngress.

[12] Carlos Eduardo Molina C, [en línea]; 11 de Marzo de 2006, [consulta: 26 de septiembre de 2016], disponible:
http://www.redtauros.com/clases/redes_ii/06_equipos_de_conectividad.pdf

[13] Andrew S. Tanenbaum. (2003). Redes de computadoras. México: Prentice Hall.

[14] María Carmen España Boquera. (2003). Servicios avanzados de telecomunicaciones. España: Díaz de Santos.

[15] Mitchell Anicas, [en línea]; 20 de Noviembre de 2104, [consulta: 28 de enero de 2016], disponible: http://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-centos-7.

[16] Juned Ahmed Ansari. (2015). Web penetration testing with Kali Linux. Birmingham: Packt Publishing.

[17] Abhinav Singh. (2013). Instant Kali Linux. Birmingham: Packt Publising.

[18] Gordon “Fyodor” Lyon. (2008). The official Nmap project guide to network discovery and security scanning. United States: Insecure.Com LLC.

[19] Pablo González, German Sánchez & José Miguel Soriano. (2015). Pentesting con kali 2.0. España: 0xword.

[20] Charles P. Pfleeger & Shari Lawrence Pfleeger. (2013). Security in computing. United States: Prentice Hall.

[21] Extreme networks summit summit24 installation and user manual: port-mirroring commands; port-mirroring [en línea]; 2012, [consulta: 13 de Mayo de 2016], disponible: http://www.manualslib.com/manual/238822/extreme-networks-summit-summit24.html?page=76#manual.

[22] Piyush Verma. (2015). Wireshark network security. Birmingham: Packt Publishing.

[23]Diarioti.com, [en línea]; 07 de Marzo de 2009, [consulta: 27 de Septiembre de 2016], disponible: http://diarioti.com/como-protegerse-de-los-ataques-man-in-the-middle/21709.

[24] Un informático en el lado del mal, [en línea]; 13 de Marzo de 2012, [consulta: 01 de Octubre de 2016], disponible: http://www.elladodelmal.com/2012/03/marmita-13-detector-de-ataques-man-in.html.

[25] Ask Xmodulo, [en línea]; 08 de Noviembre de 2014, [consulta: 28 de Enero de 2016], disponible: http://ask.xmodulo.com/install-phpmyadmin-centos.html.

[26] Digitalocean, [en línea]; 06 de Noviembre de 2014, [consulta: 02 de Febrero de 2016], disponible: https://www.digitalocean.com/community/tutorials/how-to-create-an-ssl-certificate-on-apache-for-centos-7.

[27] IT'zGeek, [en linea]; 02 de Agosto de 2014, [consulta: 05 de Febrero de 2016], disponible: http://www.itzgeek.com/how-tos/linux/centos-how-tos/install-owncloud-7-on-centos-7-rhel-7.html#axzz3EqPPynKx.

Anexo ‘A’. Instalación del Servidor web

Se requirió de la instalación de un servidor con servicios web para la realización de las pruebas de seguridad realizadas en el laboratorio B-207.Los servicios instalados fueron los siguientes: phpMyAdmin, Joomla, Moodle y Owncloud, esto con la finalidad de simular un servidor real, para la presente investigación solo se hizo uso del servicio de nube Owncloud ya que fue donde se realizaron las pruebas.

Los servicios se instalaron en una PC con las siguientes características:
Lenovo (ThinkCentre), modelo: M83 i3, con una memoria RAM de 8GB y un disco duro 1T.

Para empezar la instalación abrir una terminal con el siguiente comando para poder visualizar las configuraciones que respectan ala IP.
#ip add show

Se puede verificar que la tarjeta de Red no está activada por lo cual después de instalar Centos 7 se procede a activar la tarjeta de red, para ello se edita el archivo siguiente [15].

Dirigirse a la siguiente ruta:
#cd /etc/sysconfig/network-scripts/

Para ver las interfaces existentes se ejecuta el comando con:
#ls

Desplegara una lista donde se encuentra la interfaz de nombre "cfg-enp2s0" (puede cambiar respecto al S.O.), en el archivo cfg-enp2s0 se encuentra la descripción de la interface, esta es una nomenclatura distinta, ya que normalmente suelen mostrarse como: eth0, eth1, etc. Sin embargo en esta versión de Centos han cambiado por lo que se encontraran de este tipo: enp3s0, enp4s0, etc.

En el siguiente archivo editar la línea mostrada en la figura A1.
#vi cfg-enp2s0

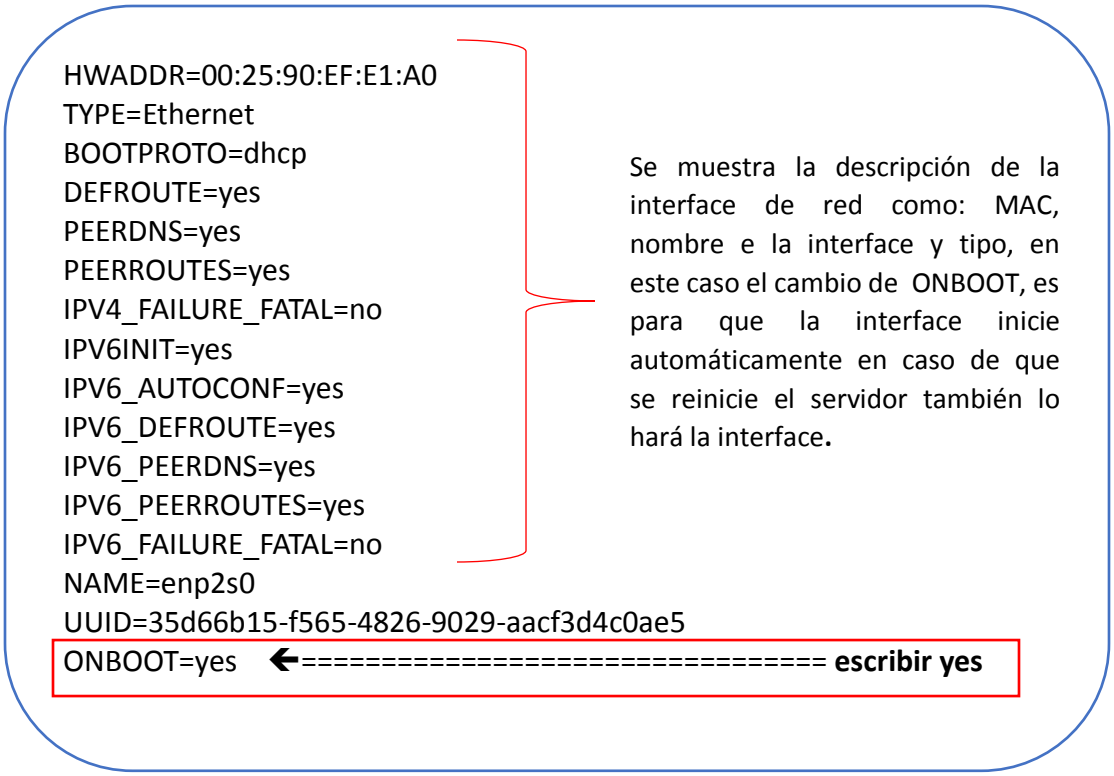


Figura A1. Descripción de la interface e red.

La configuración en modo grafico se basa en NetworkManager, que es el que reinicia las interfaces de red, para seguir con la instalación se reiniciara con el comando siguiente:
#systemctl restart NetworkManager

Si se desea ver cada red, se hae mediante el comando "ifconfig", para ello se procede a instalar net-tools, que es un conjunto de herramientas para configuración de las cuales se puede hacer uso de ifconfig.
#yum install net-tools

Ya que hay conexión a internet. Es necesario actualizar el S.O Centos 7, se ejecuta el siguiente comando para actualizar la lista de los paquetes del sistema.
#yum -y update

A.1 Instalación de httpd (servidor Web apache)

Para la instalación de httpd se ejecuta el siguiente comando [15].
#yum -y install httpd ò #sudo yum -y install httpd

Para que httpd inicie desde el arranque del servidor se habilita el servicio mediante el comando:
#systemctl enable httpd.service

Se desactiva SELINUX en la siguiente ruta.
#vi /etc/selinux/config

En el archivo se modifica la siguiente línea: modificar enforcing por disabled, como se muestra en la figura A2.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing          <========================== (cambiar por disabled)
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Figura A2. Configuración de SELINUX.

Después se reinicia el servidor nuevamente con el siguiente comando:
#shutdown -r now

Para verificar que SELINUX está desactivado y que este es un módulo de seguridad en el cual sus políticas de seguridad se establecen para el control de acceso, en el caso de instalar aplicaciones al sistema es conveniente desactivarlo para no tener complicaciones en la instalación, se ejecuta lo siguiente para verificar su estado:
#sestatus

Es necesario cambiar las reglas del firewall para que inicie por defecto, usando el comando "firewall-cmd" donde agregaremos el servicio http.
#firewall-cmd --add-service=http

Para hacer que esta regla inicie por defecto en nuestro servidor, es decir que inicie al reiniciar el servidor es necesario el comando.
#firewall-cmd --permanent --add-service=http

Reiniciar el Firewall meiante el commando:
#systemctl restart firewalld

Iniciamos el servicio httpd de la siguiente forma:
#sudo systemctl start httpd.service
#firewall-cmd --add-service=http

Colocar en el navegador la ip del servidor en este caso 192.168.1.124 (red LAN) o bien puede ser http://localhost la página de bienvenida se encuentra en la ruta ***/etc/httpd/conf.d***/ en caso de que se desee modificar o eliminar.

A.2 Instalación de mariadb

Es un gestor de base de datos derivado de MySQL, con más funcionalidades donde se puede ver información acerca del servidor como librerías, paqueterías, pluggins, usuarios, entre otros [15].

Para la instalación de mariadb se ejecuta el siguiente comando:
#sudo yum install mariadb-server mariadb

Para iniciar el servicio mariadb se ejecuta lo siguiente:
#sudo systemctl start mariadb

Es necesario configurar mysql para dar permisos de administrador y si lo deseamos cambiar usuario y contraseña, para ello se emplea el siguiente comando.
#sudo mysql_secure_installation

Colocar contraseña de root y dar enter en los siguiente pedimentos (esto regularmente se hace en caso de desear cambiar el usuario y contraseña verificar lo que dice el monitor para poder proseguir).

Para poder activar el Mariadb en el arranque del servidor es necesario realizar.
#sudo systemctl enable mariadb.service

Para crear bases de datos, para ver tablas de las bases.
#mysql -u root –p

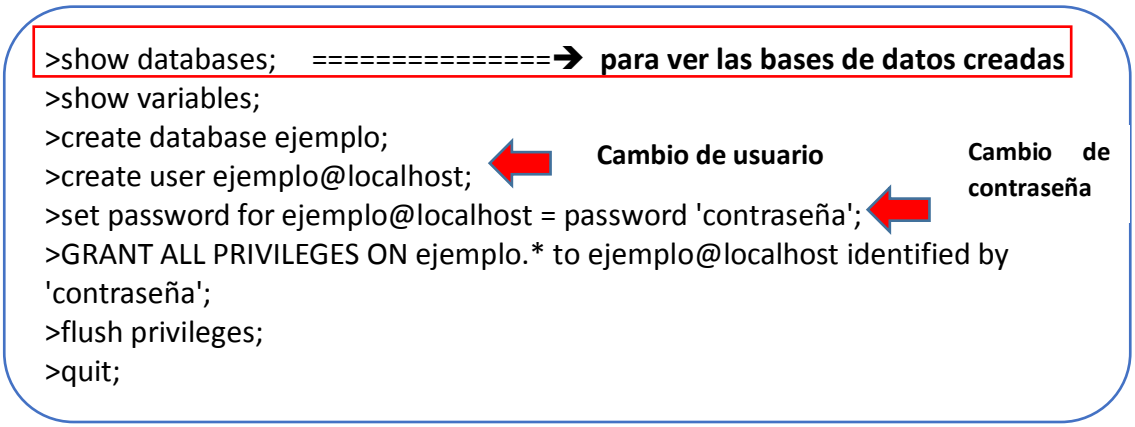


Figura A3. Visualización del contenido de la base de datos.

A.3 Instalación de php5

Para la instalación hay que buscar las versiones de php disponibles con el siguiente comando:

#yum search php

Instalar php-mysql, php es un lenguaje enfocado para el desarrollo web que permite ejecutar scripts y conectarse con base de datos de mysql/mariadb [15].

#sudo yum install php php-mysql ò yum install php php-mysql php-pdo php-gd php-mbstring

Reiniciar el httpd (servidor) para que empiece a trabajar con php.
#sudo systemctl restart httpd.service

Crear y editar el siguiente archivo para verificar el funcionamiento de php.
#vi /var/www/html/info.php
Colocar el siguiente contenido en el archivo de info.php.

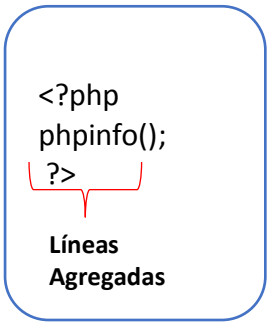


Figura A4. Edición del archivo info.php

Reiniciar nuevamente el servidor.
#systemctl restart httpd

Colocar en el navegador lo siguiente:
192.168.1.124 /info.php o 127.0.0.1/info.php

Se podrá ver la información acerca del servidor como: paqueterías, librerías, plugins, usuarios, etc.
NOTA:
Reiniciar el firewall como ya se ha hecho anteriormente.

#sudo firewall-cmd --permanent --zone=public --add-service=http
#sudo firewall-cmd --permanent --zone=public --add-service=https
#sudo firewall-cmd --reload

A.4 Instalación de EPEL

Más adelante se requerirá instalar phpMyAdmin, sin embargo no está disponible en Centos 7, para obtener los paquetes necesarios se tiene que añadir un repositorio adicional al sistema, un repositorio es como un banco de datos que aloja aplicaciones o paquetes que el sistema necesita, el repositorio EPEL (Extra Packages for Eterprise Linux), este contiene muchos paquetes adicionales entre ellos phpMyAdmin [25].

Para iniciar la instalación ejecutamos lo siguiente:

#yum install wget

#wget -r --no-parent -A 'epel-release-*.rpm'
http://dl.fedoraproject.org/pub/epel/7/x86_64/e/rpm -Uvh
dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-*.rpm

A.5 Instalación de phpmyadmin

PhpMyAdmin es un administrador de base de datos para MyS QL y Mariadb, se maneja por medio de una interface web la cual facilita el manejo de usuarios, cuentas, base de datos, contraseñas, entre otros.

Para ello se utiliza el siguiente comando:
#yum install phpmyadmin

Dirigirse a */etc/phpMyAdmin/* dentro veremos el archivo *config.inc.php* se modificara con el siguiente coamndo.
#vi config.inc.php

Cambiar la siguiente línea:
Originalmente en esta línea aparece la palabra cookie y se cambiara por http. Ya que es el método de autenticación que permite acceder a cualquier usuario valido de la base de datos a través de http, mientras que el cookie es el método de autenticación mediante el cual se da el acceso a cualquier usuario valido de la base de datos con la ayuda de cookies, sin embargo para el caso del servidor web se utilizara autenticación mediante http.

$cfg['Servers'][$i]['auth_type'] = 'http';
// Método de autenticación http, (basado en http o cookie)

En la ruta ***cd /etc/httpd/conf.d/*** se modifica el archivo ***"phpMyAdmin.conf"*** esto para poder permitir que la interfaz gráfica reconozca la ip de servidor.

#vi /etc/httpd/conf.d/phpMyAdmin.conf
De forma predeterminada la configuración de phpMyAdmin solo permite el acceso desde el mismo equipo donde se aloja la aplicación, de manera que hay que ajustar esas reglas para que más adelante se pueda acceder remotamente pidiendo la contraseña de administrador de la base de datos y asi no permita que cualquier usuario se pueda conectar y haga modificaciones, a continuación se edita el archivo como se muestra en la figura A5 [26].

```
<Directory /usr/share/phpMyAdmin/>
  AddDefaultCharset UTF-8

  <IfModule mod_authz_core.c>
    # Apache 2.4
    <RequireAny>
      #Require ip 127.0.0.1 ←======Se Comenta
      #Require ip ::1    ←======Se Comenta
      Require all granted  ←============= Se
agrega
    </RequireAny>
  </IfModule>
  <IfModule !mod_authz_core.c>
    # Apache 2.2
    Order Deny,Allow
    Deny from All
    Allow from 127.0.0.1
    Allow from ::1
  </IfModule>
</Directory>

<Directory /usr/share/phpMyAdmin/setup/>
  <IfModule mod_authz_core.c>
    # Apache 2.4
    <RequireAny>
      #Require ip 127.0.0.1  ←======Se Comenta
      #Require ip ::1      ←======Se Comenta
      Require all granted ←========== Se agrega
    </RequireAny>
  </IfModule>
  <IfModule !mod_authz_core.c>
    # Apache 2.2
    Order Deny,Allow
    Deny from All
    Allow from 127.0.0.1
    Allow from ::1
  </IfModule>
</Directory>
```

Figura A5. Configuración del archivo phpMyAdmin.

Por último reiniciamos el servidor para dar inicio a phpMyAdmin

#sudo systemctl restart httpd.service

Para entrar a phpMyAdmin URL: 192.168.1.124 /phpMyAdmin, como se muestra en la figura A6.

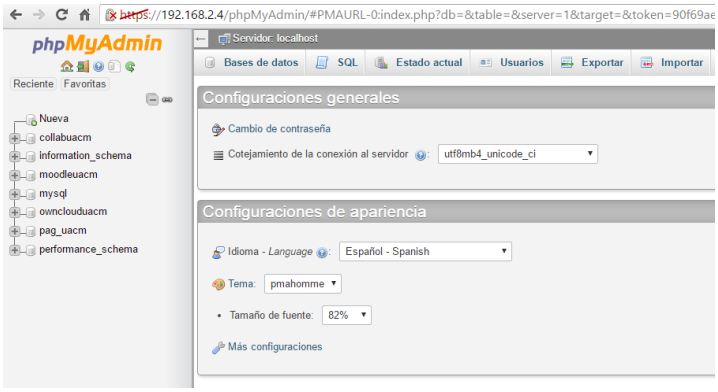


Figura A6. Servicio de phpMyAdmin.

A.6 SSL en httpd

Se instalaron certificados SSL para la seguridad que tendrá el servidor, y al momento de iniciar los servicios web poder entrar con URL segura. Por ejemplo: https
#sudo yum install mod_ssl

Se crea el directorio donde se guardaran los certificados de seguridad.
#mkdir /etc/httpd/ssl

Ahora que tenemos un lugar para colocar los archivos, se crean las claves y certificados SSL con openssl:
#openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/apache.key -out /etc/httpd/ssl/apache.crt

Abrir el archivo de configuración SSL de Apache con cualquier editor de texto mediante privilegios de root de la siguiente manera:
#vi /etc/httpd/conf.d/ssl.conf

Hay que descomentar las líneas como se muestra en la figura A7.

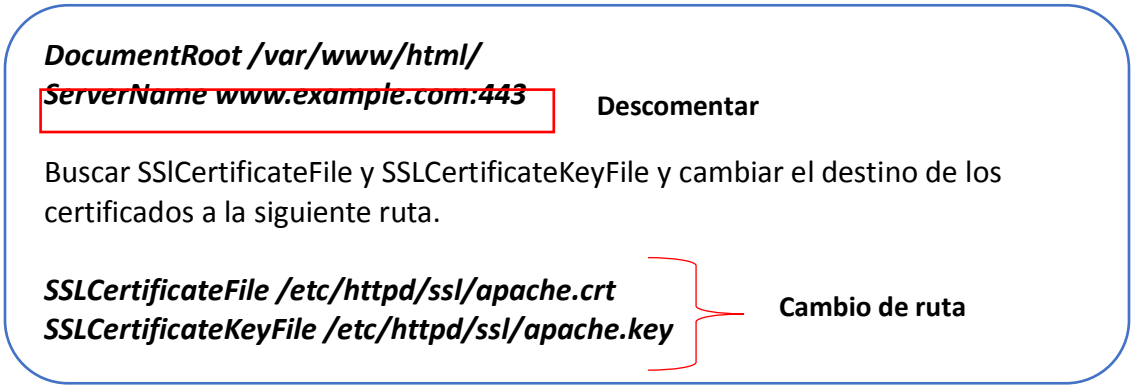


Figura A.7.Configuración de certificados SSL.

Se reinicia el servidor apache para que se activen los cambios realizados con el siguiente comando.
#sudo apachectl restart

Dirigirse al navegador y poner la dirección del servidor, con la diferencia que será por medio de https como se muestra a continuación:

https:// 192.168.1.124

Para encriptar la información se usara los siguientes paquetes que son para generar claves y certificados de seguridad:
#yum install crypto-utils
#genkey your_FQDN

A.7 Instalación de joomla

Es un sistema de gestión de contenidos de código abierto, que permite desarrollar sitios web dinámicos e interactivos a través de un panel de administración.

Descargar el paquete de joomla en el servidor, mediante el siguiente comando.

 # wget http://joomlacode.org/gf/download/frsrelease/19393/158832/Joomla_3.3.0-Stable-Full_Package.zip

Se tiene que descomprimir el archivo y ubicarlo en la carpeta html ya que ahí se encuentran todas las aplicaciones del servidor, para ello ejecutamos el siguiente comando.
 # unzip Joomla_3.3.0-Stable-Full_Package.zip -d /var/www/html/

Hay que dar permisos de administrador a la ubicación para poder configurar el archivo con el siguiente comando.

 # chown -R apache.apache /var/www/html

Copiar el archivo ***htacces.txt*** con el siguiente comando.
 # cp /var/www/html/htaccess.txt /var/www/html/.htaccess

Ir a la dirección ip para terminar la instalación de Joomla en modo gráfico. Es necesario haber creado la base de datos y el usuario para poder terminar la instalación.

Para entrar a la base de datos y crear usuario y contraseña ver la figura A8.
#mysql -u root –p

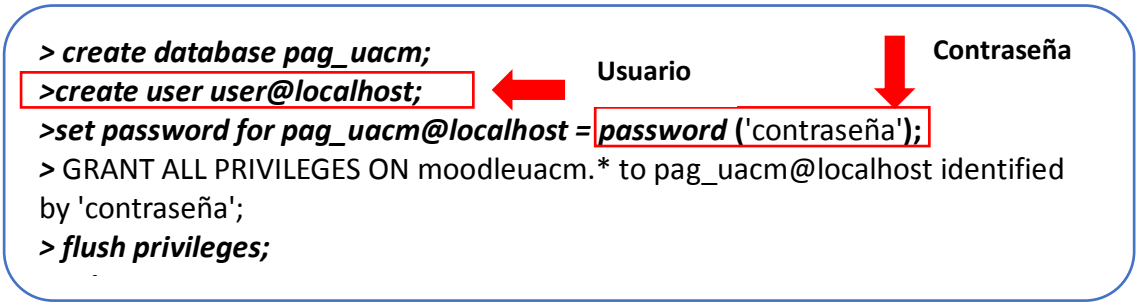


Figura A8. Configuración de base de datos para joomla.

Una vez terminada la configuración se tendrá la página principal de joomla como se muestra en la figura A9.

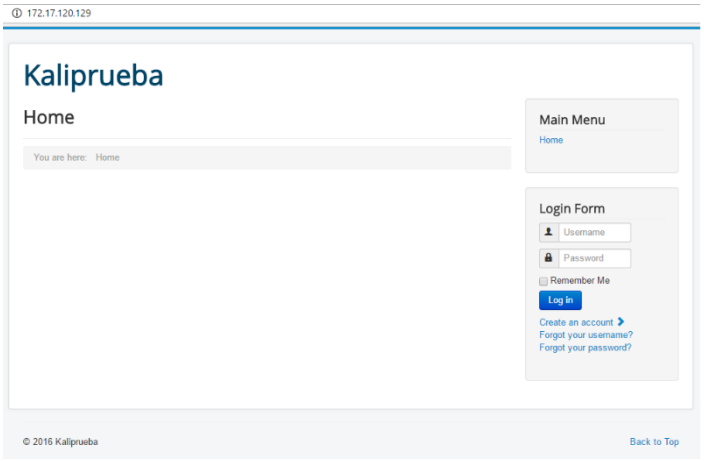


Figura A9. Servicio de jommla.

A.8 Instalación de moodle

Moodle es una aplicación web de tipo ambiente educativo virtual, es un sitio de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. Este tipo de plataformas tecnológicas también se conoce como LCMS (Learning Content Management System) [27].

Es necesario instalar algunas librerías de PHP para poder instalar correctamente Moodle, para ello ejecutamos lo siguiente:

#yum -y install php-common php-cli php-pear php php-pdo php-mysql libXpm php-gd php-xml php-mbstring php-xmlrpc php-intl php-soap

Descargar Moodle como se muestra a continuación:
#wget http://nchc.dl.sourceforge.net/project/moodle/Moodle/stable26/moodle-latest-26.tgz

hay que extraerlo y moverlo a la ruta /var/www/html/Moodle,
#tar xvzf moodle-latest-26.tgz
#mv moodle /var/www/html/moodle

De igual manera que los demás servicios hay que crear una base de datos para moodle.

Ejecutamos el siguiente comando y modificar como se muestra en la figura A10, cabe mencionar que no necesariamente deben ser estos datos.
#mysql -u root –p

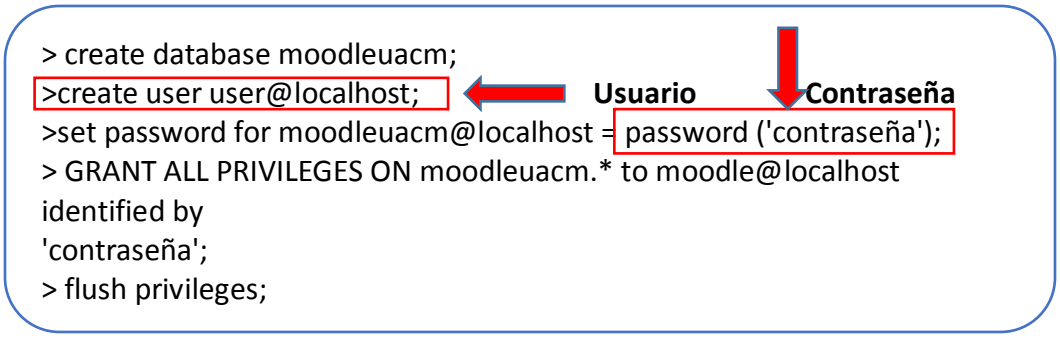


Figura A10. Confoguración de base de datos para Moodle.

Dar permisos de administrador a la carpeta como se muestra a continuación:
#chown -R apache:apache /var/www/html/moodle
#chmod -R 755 /var/www/html/moodle
#chown -R apache:apache /var/www/moodledata
#chmod -R 755 /var/www/moodledata
#setsebool -P httpd_unified 1

Nuevamente reiniciar el firewall.
#firewall-cmd --permanent --zone=public --add-service=http
#firewall-cmd --permanent --zone=public --add-service=https
#firewall-cmd –reload
De igual manera reiniciar el servidor.
#systemctl start httpd.service
 # systemctl start mariadb.service
#systemctl enable httpd.service
#systemctl enable mariadb.service

Acceder con la ip como se muestra en la figura A11.
https:// IP/Moodle

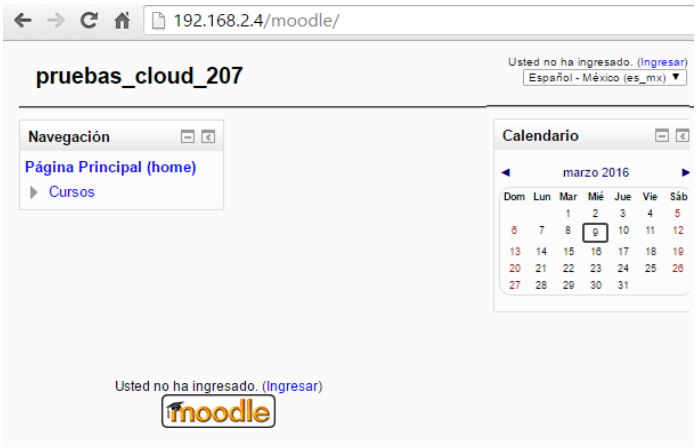


Figura A11. Servicio de moodle.

A.8 Instalación de Owncloud

Es una aplicación libre que permite el almacenamiento en línea y aplicaciones en línea. Owncloud puede ser instalado dentro de un servidor que disponga de una versión reciente de PHP y soporte de Mariadb. Este será el servicio de almacenamiento para la realización de las pruebas de seguridad mediante el cual se va a extraer información y archivos para hacer una revisión de la seguridad empleada al momento de subir archivos en línea.

Para iniciar la instalación hay que Instalar las librerías de PHP5, esto para la conexipn de las bases de dato desde php como se muestra a continuación.

#yum install httpd php php-mysql mariadb-server mariadb sqlite php-dom php-mbstring php-gd php-pdo wget

Verificar que SELINUX este desactivado.

#setsebool -P httpd_unified 1

Cambiar la configuración del Firewall como se ha hecho anteriormente.

#firewall-cmd --permanent --zone=public --add-service=http
#firewall-cmd --permanent --zone=public --add-service=https
#firewall-cmd --reload

Habilitar el servicio httpd en el arranque.

#systemctl start httpd.service
#systemctl enable httpd.service

Habilitamos el servicio de base de datos en el arranche.

#systemctl start mariadb.service
#systemctl enable mariadb.service

Descargar Onwcloud con el siguiente comando.

#wget https://download.owncloud.org/community/owncloud-8.2.1.tar.bz2

Extraer el archivo y enviarlo a la siguiente ruta ***/var/www/html/***

#tar -jxvf owncloud-8.2.1.tar.bz2 -C /var/www/html/
Dar permisos de lectura esto para poder visualizar a Owncloud vía web, para ello ejecutamos lo siguiente.

#chown -R apache.apache /var/www/html/owncloud/

Se crea la base de datos para este servicio de a misma manera que en los servicios anteriores, la configuración empleada para este caso se muestra en la figura A12.
#mysql -u root –p

```
> create database owncloudeuacm;
>create user owncloduacm@localhost;
>set password for owncloduacm@localhost = password ('contraseña');
> GRANT ALL PRIVILEGES ON owncloduacm.* to owncloud@localhost
identified by 'contraseña';
> flush privileges;
>quit;
```

Usuario **Contraseña**

Figura A12. Configuración de base de datos para Owncloud.

Para la configuración del servidor web Apache, se recomienda habilitar .htaccess para conseguir unas características de seguridad mejoradas, con .htaccess por defecto está desactivada en el servidor Apache. Para activarlo, abrir el archivo de host virtual y crea AllowOverride se establecen en All para permitir el acceso, como se muestra en la figura A13. Por ejemplo, crear un archivo de configuración externa en lugar de modificar el archivo principal, para ello se ejecuta el siguiente comando.

#vi /etc/httpd/conf.d/owncloud.conf

```
<IfModule mod_alias.c>
Alias /owncloud /var/www/html/owncloud
</IfModule>
<Directory "/var/www/html/owncloud">
Options Indexes FollowSymLinks
AllowOverride All
Order allow,deny
allow from all
</Directory>
```

Configuración para permitir acceso.

Figura A13. Configuración de Allow Override.

#systemctl restart httpd.service

Dirigirse al navegador web y colocar la URL. http://ip/owncloud, mostrado en la figura A14.

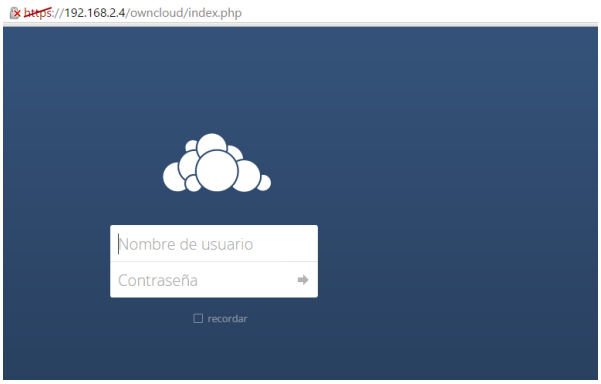


Figura A14. Servicio de Owncloud.